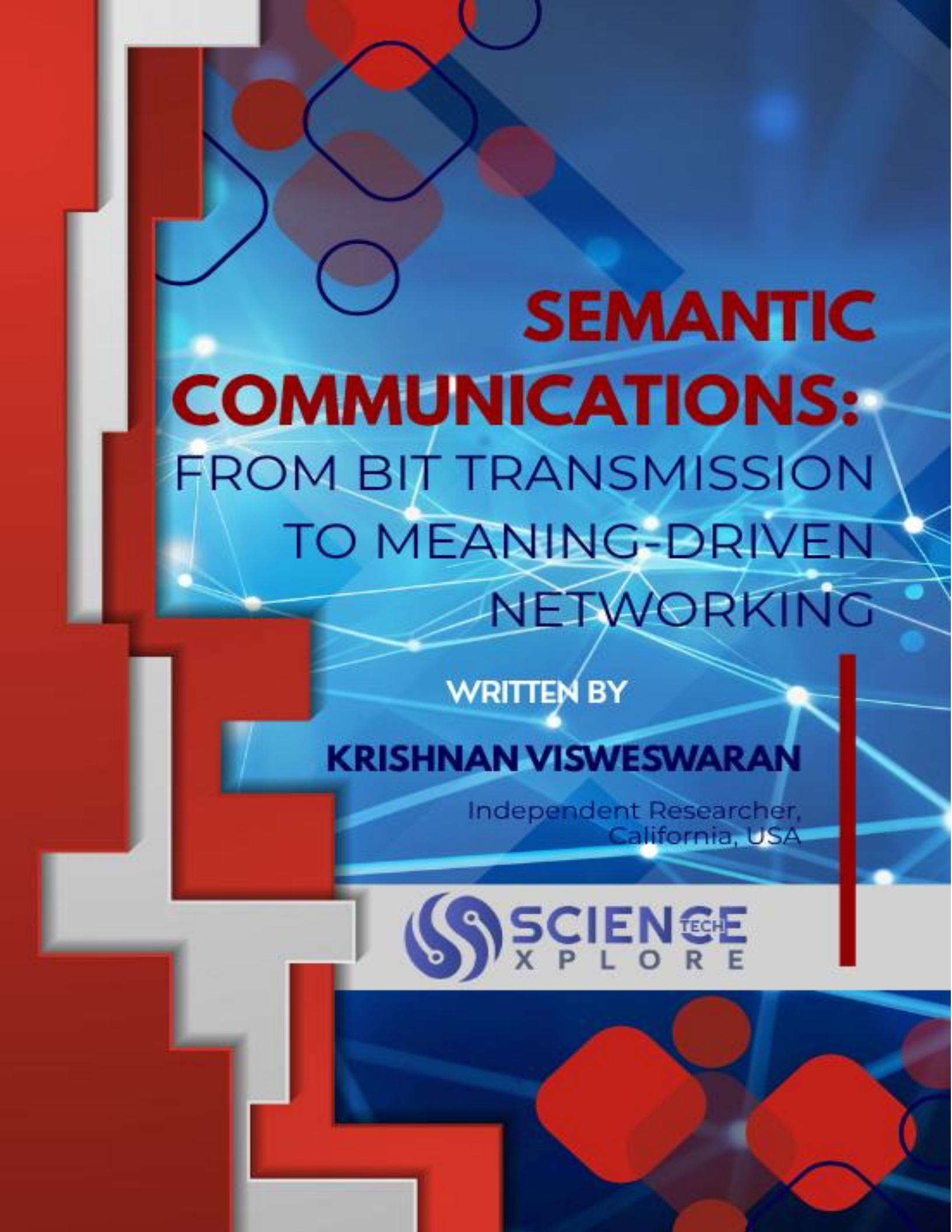




SEMANTIC COMMUNICATIONS: FROM BIT TRANSMISSION TO MEANING-DRIVEN NETWORKING

WRITTEN BY

KRISHNAN VISWESWARAN

Independent Researcher,
California, USA



Semantic Communications: From Bit Transmission to Meaning-Driven Networking

Krishnan Visweswaran

Independent Researcher, California, USA

Published by
ScienceTech Xplore



Semantic Communications: From Bit Transmission to Meaning-Driven Networking

Copyright © 2026 Krishnan Visweswaran

All rights reserved.

First Published 2026 by ScienceTech Xplore

ISBN 978-93-49929-14-2

ScienceTech Xplore

www.sciencetechxplore.org

The right of Krishnan Visweswaran to be identified as the author of this work has been asserted in accordance with the Copyright, Designs, and Patents Act of 1988. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise) without the prior written permission of the publisher.

This publication is designed to provide accurate and authoritative information. It is sold under the express understanding that any decisions or actions you take as a result of reading this book must be based on your judgment and will be at your sole risk. The author will not be held responsible for the consequences of any actions and/or decisions taken as a result of any information given or recommendations made.



978-93-49929-14-2

Printed and Bounded by
ScienceTech Xplore, India

ABOUT THE AUTHOR



Krishnan Visweswaran is a technology professional, researcher, and engineering leader based in California, USA, with extensive experience in wireless communications, software quality engineering, artificial intelligence, embedded systems, and next-generation networking technologies. Throughout his career, he has worked with leading technology organizations including Meta, Amazon, Qualcomm, and Cisco, contributing to complex hardware-software systems, automation frameworks, intelligent platforms, and large-scale system validation.

He pursued graduate studies in Electrical Engineering at California State University, Long Beach, with a focus on communication systems, VLSI, RF and microwave engineering, digital signal processing, computer networks, and embedded systems.

His research interests include 6G wireless systems, AI-native networking, machine learning, edge intelligence, cloud computing, cybersecurity, and intelligent automation. His academic and professional projects span artificial intelligence, radar systems, robotics, VLSI design, embedded systems, wireless technologies, communication networks, and software automation.

His work focuses on bridging artificial intelligence and next-generation communication technologies to develop intelligent, secure, resilient, and scalable systems for the future of global connectivity.

PREFACE

For nearly a century, the theoretical foundation of communication systems has rested on a single, remarkably durable idea: the goal of a communication system is to reproduce, at the receiver, the symbols generated at the transmitter, as faithfully as the channel allows. This framing, formalized by Claude Shannon in his foundational work on information theory, deliberately and productively set aside any question of what those symbols meant. It was an extraordinarily successful abstraction, one that gave rise to nearly every advance in modern telecommunications, from digital modulation to error-correcting codes to the wireless networks that now carry the bulk of human communication. But it was always, by its own design, an incomplete picture of what communication actually is.

As networks are asked to carry not just human conversation but sensor data, machine-to-machine coordination, autonomous decision-making, and increasingly capable AI systems operating at the edge, the limits of a purely bit-faithful model of communication have become harder to ignore. A great deal of what is transmitted across modern networks is redundant, irrelevant to the task at hand, or reconstructible from context without ever needing to cross the channel at all. Two machines coordinating a task, an autonomous vehicle relaying only the features of a scene relevant to a driving decision, a sensor network reporting an anomaly rather than a continuous stream of raw readings — in each of these cases, what matters is not the exact reproduction of bits, but the successful transfer of meaning relevant to a goal. This is the premise from which semantic communications begins: that networks of the future should be designed not merely to move bits accurately, but to convey meaning efficiently.

This book is an attempt to take that premise seriously and follow it through to its practical consequences. It examines how communication systems can be reimaged around semantic relevance rather than symbol fidelity, how AI and machine learning make this reimagining computationally feasible in ways that were not available to earlier generations of researchers, and what new architectures, metrics, and design trade-offs emerge once meaning, rather than bits, becomes the unit of value in a network. The

chapters that follow move from the conceptual foundations of semantic communication theory, through the technical mechanisms that make meaning-aware encoding and transmission possible, to the practical implications for next-generation wireless systems, edge intelligence, and machine-to-machine networks.

Throughout, I have tried to keep the discussion tethered to what is technically achievable and empirically grounded, rather than purely speculative. Semantic communications is still an emerging field, with open questions and unresolved trade-offs, and this book does not claim to settle them. My hope instead is that it offers a coherent framework for understanding where the field stands, why it matters, and where its most promising directions lie — for researchers, network engineers, and technologists who sense, as I do, that the next great shift in communication systems will be defined not by faster transmission of bits, but by smarter transmission of meaning.

ACKNOWLEDGEMENT

This book is the product of not just individual effort, but of the ideas, challenges, and support offered by many people over the course of its development.

I am grateful to the colleagues and collaborators with whom I have discussed the ideas in this book over the years conversations that repeatedly forced me to sharpen vague intuitions about "meaning" in communication into concepts precise enough to be engineered. Their willingness to push back on half-formed ideas made this a far more rigorous book than it would otherwise have been.

I want to thank the reviewers and technical experts who took the time to read early drafts of this manuscript, offering candid feedback that identified gaps, challenged assumptions, and strengthened the overall coherence of the arguments presented here. Their scrutiny is reflected on every page, even where it is not visible.

I am also indebted to the broader research community working in information theory, wireless communications, and AI-driven networking, whose published work laid the theoretical and technical groundwork this book builds upon. Semantic communications as a field exists because of their collective effort, and this book is, in many ways, an attempt to synthesize and extend that work rather than originate it from nothing.

Finally, my sincere thanks to my family and friends, whose patience and steady encouragement carried me through the demanding process of researching and writing this book. Their support made the long hours worthwhile.

Krishnan Visweswaran

CONTENTS

1. About the Author -----	i
2. Preface -----	ii
3. Acknowledgement -----	iv
4. Foundations and Evolution of Semantic Communications -----	1
5. Information-Theoretic Foundations and Mathematical Models of Semantic Communication -----	15
6. Semantic Representation, Knowledge Modeling, and Context-Aware Information Processing -----	37
7. Deep Learning, Transformer Models, and Generative Intelligence for Semantic Communications -----	58
8. Language-Centric Semantic Communications and Large Language Model Integration -----	78
9. Multimodal Semantic Communications for Text, Vision, Speech, and Generative Information -----	95
10. Wireless Semantic Communications and Meaning-Aware Radio Networks ----	110
11. Semantic Networking in 5G, 6G, and AI-Native Communication Infrastructures-- -----	126
12. Edge Intelligence, IoT, and Distributed Semantic Communication Systems --	143

13. Intelligent Applications and Emerging Use Cases of Semantic Communications---	
-----	160
14. Security, Privacy, Trust, and Robustness in Semantic Communication Networks -	
-----	171
15. Semantic Network Optimization, Quality Assessment, and Experimental	
Evaluation -----	187
16. Future Directions toward Fully Meaning-Driven Intelligent Networks -----	202
17. Bibliography -----	215

Chapter 1

Foundations and Evolution of Semantic Communications

1.1 From Conventional Bit-Oriented Communication to Meaning-Centric Information Exchange

1.1.1 Historical Evolution of Digital Communication and Information Transmission

The development of modern communication systems has been closely connected to humanity's ability to represent, transmit, and reconstruct information. Early communication methods relied on physical signals, symbols, written messages, and manually interpreted codes. With the development of electrical and electronic communication, information began to be represented through measurable signals that could travel over long distances. Telegraphy introduced coded symbolic transmission, while telephony enabled the transmission of speech as an electrical waveform. Radio and television subsequently expanded communication from point-to-point interaction toward large-scale wireless and broadcast systems.

The emergence of digital communication represented a fundamental transformation. Instead of transmitting information primarily as continuous physical signals, digital systems represented information using discrete symbols and, eventually, binary bits. Text, images, audio, and video could all be converted into digital representations and processed using common computational infrastructures. This enabled reliable storage, compression, encryption, switching, and transmission across increasingly complex networks.

A major theoretical foundation for this transformation was information theory. The mathematical framework established by Claude Shannon provided a systematic approach for quantifying information, characterizing communication channels, and determining the fundamental limits of reliable information transmission. Communication systems consequently became highly optimized around objectives such as reducing transmission errors, increasing channel capacity, improving spectral efficiency, and achieving reliable reconstruction of transmitted data.

The subsequent evolution of networking introduced packet switching, computer networks, the Internet, wireless communication, mobile systems, cloud computing, and high-speed broadband infrastructures. Each generation increased transmission rates and expanded the range of applications supported by communication networks. More recently, fifth-generation and emerging sixth-generation wireless systems have introduced requirements involving ultra-low latency, massive connectivity, machine-type communication, immersive applications, autonomous systems, and intelligent edge computing.

Despite these advances, the underlying communication principle has largely remained data-oriented: information is converted into bits, transmitted through a communication channel, and reconstructed at the receiver. This architecture has been remarkably successful, but the increasing dominance of artificial intelligence, autonomous systems, and machine-to-machine interaction has exposed situations in which accurate bit reconstruction does not necessarily correspond to useful information exchange.

1.1.2 Fundamental Limitations of Bit-Level Communication Paradigms

Conventional digital communication systems are fundamentally designed around the accurate transmission and reconstruction of symbols or bits. The sender converts information into a digital representation, applies source and channel coding, transmits the resulting signal, and enables the receiver to reconstruct the original data with an acceptably low error probability. This architecture has provided the foundation for reliable digital networks, but its objectives become increasingly restrictive when communication involves intelligent machines, autonomous agents, and task-oriented applications.

One important limitation is that bit-level accuracy does not necessarily represent semantic usefulness. Two messages may differ substantially at the bit level while expressing essentially the same meaning. Conversely, a very small modification to a message may completely change its meaning. Traditional communication metrics such as bit error rate, packet error rate, and symbol error rate cannot directly determine whether the receiver has obtained the intended semantic content.

A second limitation is the transmission of information that may be irrelevant to the receiver's task. Consider an autonomous vehicle receiving visual information from a roadside camera. If the vehicle's immediate objective is to determine whether a pedestrian is entering its path, transmitting every pixel of a high-resolution image may consume substantial communication resources even though only a small portion of the image contributes to the decision. A conventional system attempts to preserve the complete data representation, whereas a semantic system can potentially prioritize task-relevant information.

Another challenge arises from the increasing volume and heterogeneity of modern data. Connected devices continuously generate text, images, video, sensor measurements, audio streams, and machine-generated information. Transmitting all generated data without considering relevance can create significant demands on bandwidth, energy, storage, and computation. These concerns become particularly important for resource-constrained edge and IoT environments.

Bit-oriented communication also treats the communication channel largely independently from the meaning of the transmitted information. In many traditional architectures, the communication system does not understand whether a packet contains critical safety information, redundant sensor measurements, or data that is unlikely to affect the receiver's decision. As a result, communication resources may not be allocated according to semantic importance.

1.1.3 Emergence of Meaning-Oriented and Semantic Communication Systems

The emergence of semantic communication represents a shift in the fundamental objective of information exchange. Traditional systems primarily seek to reproduce transmitted data accurately, whereas semantic communication focuses on conveying information according to its meaning, relevance, and usefulness to the receiving entity. This shift becomes particularly important when communication takes place between intelligent systems capable of interpreting complex information rather than simply reconstructing transmitted symbols.

The concept of semantic communication has roots in earlier discussions of communication beyond purely syntactic information transfer. However, recent advances in artificial intelligence, deep learning, natural language processing, computer vision, multimodal learning, and representation learning have created

practical mechanisms for extracting and reconstructing semantic information. Neural networks can learn compact representations of images, text, speech, sensor observations, and other data modalities. These learned representations provide a foundation for communication systems that can selectively transmit information relevant to a particular meaning or task.

Semantic communication architecture generally introduces intelligence into multiple stages of the communication process. At the transmitter, raw information can be analyzed to identify semantic features or task-relevant representations. A semantic encoder can then transform the original information into a compact representation. This representation is transmitted through a physical or wireless channel, where it may experience noise, interference, fading, congestion, or limited bandwidth. At the receiver, a semantic decoder reconstructs the information or generates an output that preserves the intended meaning.

This architecture changes the definition of successful communication. A received message does not necessarily need to reproduce the source data exactly. Instead, it should preserve the semantic information required by the application. For example, when communicating a traffic scene, a semantic system may prioritize objects, locations, movement, hazards, and relationships rather than transmitting every visual detail. Similarly, in machine-to-machine communication, the system may communicate the information required to make a control decision rather than reproducing the complete sensor stream.

The emergence of foundation models and multimodal artificial intelligence further expands this possibility. Modern models can associate information across language, vision, audio, and other modalities, allowing semantic representations to become increasingly context-aware. Communication systems can consequently move toward representations that capture concepts, relationships, intentions, and task requirements.

Semantic communication should therefore be understood not simply as compression with fewer bits, but as a broader redesign of communication around information meaning and utility. Its development creates a bridge between communication engineering and artificial intelligence, establishing the foundation for intelligent systems in which communication resources are allocated according to what information actually matters.

1.1.4 Transition from Shannon-Centric Communication to Task-Oriented Networking

The transition toward semantic communication does not represent a rejection of Shannon's information theory. Instead, it represents an extension of communication objectives into domains where accurate data reconstruction alone may not adequately describe communication success. Shannon's framework provides fundamental principles for understanding information, entropy, channel capacity, coding, and reliable transmission. These concepts remain essential to the design of modern communication systems. However, intelligent applications introduce an additional layer: whether the transmitted information enables the receiver to accomplish its intended task.

In conventional communication, the communication process can be viewed as a pipeline in which the source generates information, the transmitter encodes it, the channel carries the encoded signal, and the receiver reconstructs the original message. The primary concern is whether the received representation sufficiently matches the transmitted representation. Task-oriented communication introduces another

stage of reasoning: the receiver uses the information to perform an application-specific operation. For example, a connected robotic system may need to determine whether an obstacle exists in its path. The communication objective is not necessarily to reproduce the complete camera image at the remote robot. Instead, the objective may be to provide sufficient information for reliable obstacle detection. Similarly, an industrial monitoring system may not need to transmit every raw sensor measurement if the actual task is anomaly detection. Communication can therefore be optimized according to the downstream objective.

At the first level, the system seeks accurate bit transmission. At the second, it seeks faithful reconstruction of the original data. At the third, it seeks preservation of meaningful information. At the fourth, it evaluates whether the received information supports the intended task. Task-oriented networking extends this principle beyond individual communication links. Network resources including bandwidth, spectrum, computing capacity, energy, routing decisions, and storage can potentially be allocated according to the semantic and task relevance of information. A network could distinguish between information that is critical for immediate decision-making and information that is redundant or less relevant to the current objective.

This creates the foundation for meaning-driven networking, in which communication, computation, and intelligence become increasingly integrated. The network is no longer viewed solely as infrastructure for moving packets; it becomes an intelligent environment capable of understanding information requirements and adapting transmission strategies accordingly. The resulting paradigm can be summarized as a shift from transmit everything accurately toward transmit what is necessary to accomplish the intended objective. This transition provides the conceptual foundation for the subsequent development of semantic encoders, semantic decoders, task-oriented communication, multimodal semantic exchange, and intelligent semantic networking.

1.2 Fundamental Concepts and Principles of Semantic Information Processing

1.2.1 Definition, Characteristics, and Dimensions of Semantic Information

Semantic information refers to information considered in terms of the meaning, significance, and interpretation that it conveys to a receiving entity. In conventional communication systems, information is commonly represented as bits, symbols, packets, images, audio samples, or other data structures. Semantic communication introduces an additional level of abstraction by considering what those representations actually mean and how useful that meaning is to the receiver. Consequently, semantic information is not determined solely by the physical representation of data but by the relationship between the data, its interpretation, and the purpose for which it is used.

A fundamental characteristic of semantic information is meaningfulness. Information becomes semantically significant when it represents concepts, events, objects, relationships, states, or intentions that can be interpreted by a receiver. For example, a sequence of pixels constitutes raw visual data, while the recognition of a pedestrian, vehicle, or traffic signal represents semantic information extracted from that data. The semantic representation therefore provides a higher-level description of what the underlying data communicates.

Another important characteristic is context dependence. The same information may have different meanings depending on the surrounding circumstances, prior knowledge, environment, or application

objective. Semantic information is therefore not always universally defined. Its interpretation can vary according to the receiver, situation, and task.

Semantic information also possesses a task-dependent dimension. Information that is highly important for one application may be irrelevant for another. A detailed temperature measurement may be critical for industrial process control but relatively unimportant for an image-classification task. This means that semantic processing must consider the intended purpose of communication.

Several dimensions can therefore be considered when characterizing semantic information: meaning, context, relevance, importance, intent, granularity, and task utility. Meaning describes what the information represents; context determines the circumstances surrounding its interpretation; relevance measures its relationship to the receiver's objective; importance reflects its contribution to decision-making; and task utility describes how effectively the information supports a particular application.

Semantic information can originate from different modalities, including text, speech, images, video, sensor measurements, and multimodal combinations. Modern artificial intelligence techniques provide mechanisms for converting these heterogeneous data sources into representations that capture higher-level concepts. Thus, semantic information establishes a bridge between raw data and meaningful knowledge. Understanding its characteristics and dimensions is essential for designing communication systems that transmit information according to its significance rather than merely its physical representation.

1.2.2 Relationships among Meaning, Context, Knowledge, and User Intent

Semantic information cannot be fully understood by considering meaning in isolation. Its interpretation is strongly influenced by context, knowledge, and user or application intent. These elements interact to determine how information is understood and which aspects of a message are important to the receiving entity. Consequently, semantic communication requires mechanisms capable of connecting transmitted information with the circumstances and objectives in which that information is interpreted.

Meaning represents the conceptual content conveyed by information. For example, the statement the road is blocked communicates a particular situation involving an obstruction. However, the practical interpretation of this statement depends on context. If the receiver is a pedestrian, the information may suggest taking another route. If the receiver is an autonomous vehicle, it may trigger route replanning. The same semantic content can therefore produce different actions depending on the receiver's circumstances.

Context provides the surrounding information required to interpret a message correctly. Context can include location, time, environmental conditions, previous interactions, system state, and the current communication objective. In intelligent communication systems, contextual information can reduce ambiguity and allow the receiver to interpret incomplete or compressed messages more effectively.

Knowledge provides the background information used to understand semantic content. A receiver with appropriate domain knowledge can infer information that is not explicitly transmitted. For example, an autonomous driving system may understand that a detected red traffic light generally indicates that the vehicle should stop. This prior knowledge allows communication systems to avoid repeatedly

transmitting information that the receiver can already infer. Intent represents the objective or purpose behind the communicated information. Understanding intent is particularly important in task-oriented communication because the receiver does not merely need to understand what was transmitted; it needs to determine what action or task the information is intended to support.

The relationship is dynamic rather than strictly sequential. Context can modify meaning, knowledge can improve contextual interpretation, and intent can determine which aspects of information deserve priority. For example, a camera may observe hundreds of objects, but if the receiver's intent is collision avoidance, only objects that represent potential hazards may require immediate transmission.

This relationship has important implications for semantic communication design. Semantic encoders should consider not only the source data but also the receiver's contextual and task requirements. Semantic decoders, meanwhile, should combine received information with available knowledge and context to reconstruct or infer the intended meaning. Therefore, meaning, context, knowledge, and intent collectively form the semantic foundation of intelligent information exchange. Their integration enables communication systems to move beyond data reconstruction toward purposeful interpretation and decision-making.

1.2.3 Semantic Relevance, Similarity, Fidelity, and Information Importance

Semantic communication requires new ways of determining whether information has been transmitted successfully. Traditional communication systems commonly evaluate performance using measures such as bit error rate, symbol error rate, packet loss, or signal-to-noise ratio. These metrics remain important for characterizing the physical communication channel, but they do not directly indicate whether the receiver has obtained information that is meaningfully equivalent or useful. Semantic relevance, similarity, fidelity, and information importance provide complementary concepts for evaluating communication from a meaning-oriented perspective.

Semantic relevance describes the degree to which transmitted information contributes to the receiver's current objective. Information may be accurate yet irrelevant to the task. For example, in an autonomous driving application, detailed information about distant buildings may have little relevance to immediate collision avoidance, while the position and movement of a nearby pedestrian may be highly relevant.

Semantic similarity concerns the degree to which the meaning of received information corresponds to the meaning of the original information. Two representations may differ substantially at the bit or pixel level while remaining semantically similar. A compressed image with reduced visual detail may still preserve the identity, location, and movement of an object, making it semantically useful despite substantial data-level differences.

Semantic fidelity refers to how accurately the essential meaning of information is preserved during communication. High semantic fidelity does not necessarily require exact reconstruction of the original data. Instead, it requires preservation of the semantic attributes necessary for interpretation or task execution. The required level of fidelity therefore depends on the application.

Information importance represents the relative contribution of particular information to the receiver's decision or objective. Semantic importance can vary across time, context, and applications. In an

emergency-response system, information about the location of a hazard may receive much higher priority than routine environmental measurements. Similarly, in industrial monitoring, an unusual sensor pattern may be more important than continuously transmitted normal readings. These concepts are closely related but should not be treated as identical. Relevance concerns whether information matters; similarity concerns whether two pieces of information have comparable meaning; fidelity concerns how well essential meaning is preserved; and importance concerns the priority or contribution of information to the task.

Together, these concepts support semantic-aware resource allocation. A communication system can prioritize information with high semantic importance, allocate greater resources to highly relevant content, and tolerate greater distortion in information with lower task significance. This represents a fundamental change in communication evaluation: rather than asking only whether the received data resembles the transmitted data, semantic communication asks whether the received information retains the meaning and utility required by the intended application.

1.2.4 Principles of Meaning Preservation during Information Transmission

Meaning preservation is one of the central principles of semantic communication. A communication system should ensure that the essential information contained in a message remains understandable and useful after transmission, even when the received representation differs from the original data. This principle changes the objective of communication from exact data reproduction toward preservation of task-relevant semantic content. The first principle is semantic sufficiency. The transmitted representation should contain enough information for the receiver to recover the semantic content required for its intended task. Not every characteristic of the original data must necessarily be preserved. Information that does not contribute meaningfully to the receiver's objective may be removed or compressed.

The second principle is relevance preservation. Communication resources should prioritize information that is relevant to the current context and task. A semantic encoder should therefore distinguish between critical, useful, redundant, and irrelevant information. This enables selective transmission rather than uniform treatment of all source information. The third principle is context-aware interpretation. Meaning should be preserved relative to the context in which information will be interpreted. A message that appears incomplete in isolation may become fully meaningful when combined with previously available knowledge or contextual information. Semantic communication systems can exploit this property to reduce unnecessary transmissions.

The fourth principle is robustness to channel distortion. Noise, interference, fading, packet loss, and limited bandwidth can alter transmitted representations. A semantic communication system should therefore prioritize representations whose essential meaning remains recoverable under communication impairments. The system should be designed so that degradation in physical signal quality does not automatically result in proportional degradation in semantic usefulness.

The fifth principle is task consistency. Ultimately, the preserved meaning should support the intended receiver task. For machine-to-machine communication, semantic preservation may be evaluated through decision accuracy, classification performance, control effectiveness, detection reliability, or another application-specific measure. The sixth principle is adaptive semantic fidelity. Different situations require

different levels of semantic detail. During routine operation, a coarse semantic representation may be sufficient, whereas safety-critical conditions may require richer information. Semantic communication should therefore adapt its representation and transmission strategy according to changing context, channel conditions, and task requirements.

These principles establish an important distinction between data fidelity and semantic fidelity. A system can tolerate considerable changes in the transmitted data while maintaining high semantic fidelity if the information necessary for the receiver's objective remains intact. Meaning preservation therefore becomes a cross-layer objective involving source representation, semantic encoding, channel transmission, semantic decoding, contextual knowledge, and application-level evaluation. It provides the conceptual foundation for later chapters that examine how semantic representations can be learned, transmitted, reconstructed, and optimized in intelligent communication networks.

1.3 Architectural Foundations of End-to-End Semantic Communication Systems

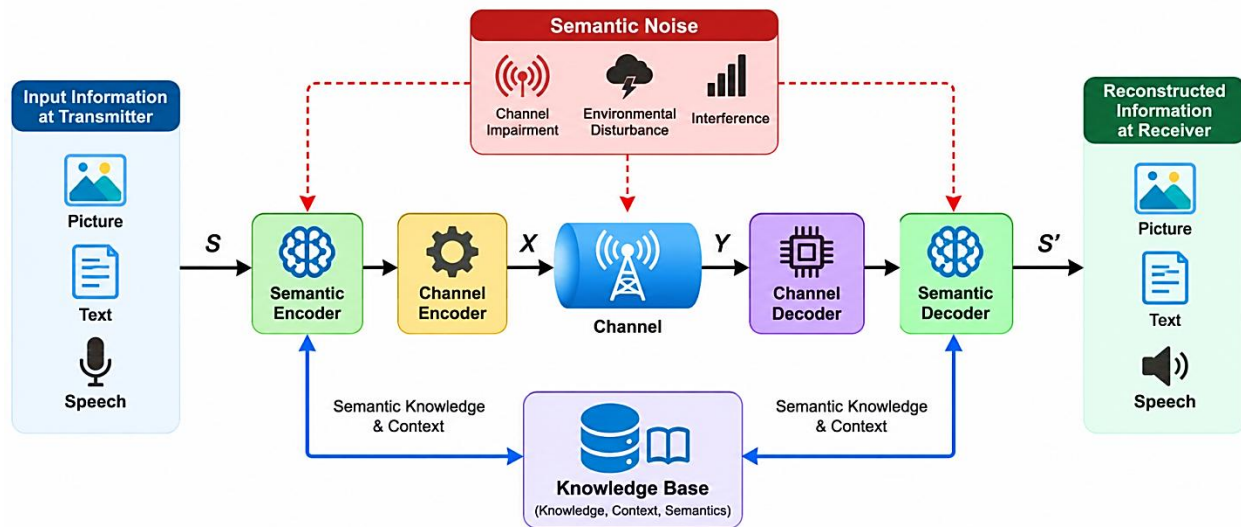


Figure 1: End-to-End Semantic Communication Architecture

The fundamental architecture of an end-to-end semantic communication system, in which information is processed according to its meaning before being transmitted through a communication channel. The system begins with heterogeneous input information at the transmitter, represented in the figure by picture, text, and speech. This source information is denoted by (S) and is provided to the semantic encoder. Unlike a conventional source encoder that primarily focuses on reducing data redundancy, the semantic encoder is intended to extract and represent information that is meaningful and relevant to the communication objective. The resulting semantic representation, denoted by (X), is then passed to the channel encoder for transmission preparation.

Following semantic encoding, the channel encoder converts the semantic representation into a form suitable for transmission through the communication medium. The encoded signal subsequently passes through the physical channel, where it can experience different forms of impairment. The figure represents these effects collectively as semantic noise, including channel impairment, environmental disturbance, and interference. These factors can modify or degrade the transmitted representation before it

reaches the receiver. This stage emphasizes that semantic communication must operate under the same fundamental physical constraints as conventional communication systems while introducing additional mechanisms for preserving meaning under imperfect transmission conditions.

At the receiver, the channel decoder first processes the received signal (Y) and produces a representation that can be interpreted by the semantic decoder. The semantic decoder then reconstructs the information according to the semantic representation received through the channel. The output is represented as (S'), indicating the reconstructed information at the receiver. Importantly, the reconstructed output does not necessarily need to reproduce every element of the original source exactly. Instead, the objective is to preserve the essential meaning and information required by the receiving application. The figure therefore distinguishes the semantic communication process from a purely bit-oriented architecture in which successful communication is primarily associated with accurate reproduction of the transmitted data.

A central feature of the architecture is the semantic knowledge and context shared between the semantic encoder and semantic decoder through the knowledge base. This knowledge base contains information related to knowledge, context, and semantic representations that can support interpretation at both ends of the communication system. By exploiting shared knowledge and contextual information, the transmitter can avoid transmitting information that the receiver can already infer, while the decoder can use available knowledge to reconstruct or interpret incomplete semantic information. The architecture consequently represents communication as an integrated process involving source information, semantic representation, channel transmission, contextual knowledge, and task-relevant reconstruction, providing the foundation for subsequent discussions of semantic encoding, semantic decoding, semantic noise, knowledge bases, and end-to-end semantic communication design.

1.3.1 Semantic Source Modeling and Knowledge Representation

Semantic source modeling constitutes the first major stage in an end-to-end semantic communication system. Its purpose is to represent source information according to the concepts, relationships, attributes, and contextual elements that are meaningful to the intended receiver. In conventional communication, a source is generally modeled as a sequence of symbols, samples, or bits. Semantic communication introduces an additional abstraction in which the source is described in terms of its underlying semantic content. This distinction becomes particularly important for modern sources such as images, speech, text, video, and multimodal sensor observations, where the amount of raw data can be considerably larger than the information required for a particular task.

A semantic source model attempts to capture the structure of information at a level that is useful for interpretation. For example, an image may contain thousands or millions of pixels, but its semantic representation may describe objects, their locations, actions, attributes, and relationships. Similarly, a speech signal can be represented not only as an acoustic waveform but also through words, concepts, speaker intent, and contextual meaning. Such representations allow the communication system to distinguish between raw observations and the information that those observations convey.

Knowledge representation provides an important foundation for this process. A semantic communication system may rely on structured or learned knowledge containing concepts, relationships, contextual information, domain-specific information, and previously acquired representations. This knowledge can

assist both transmitter and receiver in interpreting information consistently. Shared knowledge is particularly useful when some information does not need to be explicitly transmitted because the receiver can infer it from existing knowledge and context.

Semantic source modeling is also inherently context-dependent. The same source information may require different representations depending on the receiver, application, environment, or task. A surveillance system, for instance, may prioritize object identity and movement, whereas an autonomous navigation system may prioritize distance, trajectory, and potential collision risk. Therefore, semantic source models should ideally be adaptable to application requirements rather than treating all information as equally important.

The combination of semantic modeling and knowledge representation establishes the foundation for subsequent semantic encoding. It determines what information is meaningful, what information is relevant, and what information can potentially be inferred. Consequently, effective source modeling enables communication systems to move from representing complete raw data toward representing the concepts and relationships that support meaningful information exchange.

1.3.2 Intelligent Semantic Encoding and Meaning-Aware Compression

Intelligent semantic encoding transforms source information into a compact representation that preserves the semantic content required by the receiving system. This process represents an important departure from conventional compression, where the primary objective is generally to reduce the number of bits required to represent data while maintaining an acceptable level of reconstruction quality. Semantic encoding instead considers the meaning, relevance, context, and intended task associated with the source information.

The process may begin by analyzing raw input such as an image, speech signal, text document, video sequence, or sensor stream. Artificial intelligence models can extract higher-level representations from these inputs and identify features that contribute to their semantic interpretation. Depending on the application, the resulting representation may contain objects, concepts, relationships, events, attributes, instructions, or other task-relevant features. The semantic encoder can then prioritize these features according to their importance to the receiver.

Meaning-aware compression does not necessarily attempt to preserve every detail of the original source. Instead, it seeks to remove information that contributes little to the intended communication objective. For example, when an autonomous vehicle receives information about a road environment, fine-grained visual details that have no influence on navigation may be compressed more aggressively than information describing nearby pedestrians, vehicles, traffic signals, or obstacles. This enables communication resources to be concentrated on information with greater semantic significance.

An important characteristic of intelligent semantic encoding is its ability to adapt to changing conditions. The appropriate semantic representation may depend on channel quality, available bandwidth, computational resources, receiver capabilities, and application requirements. Under favorable communication conditions, a richer semantic representation may be transmitted. Under severe bandwidth or channel constraints, the encoder may prioritize only the most important semantic features.

Semantic encoding can therefore be viewed as a joint interaction between data representation, semantic understanding, compression, and communication objectives. Its effectiveness should not be evaluated solely according to conventional compression ratios. Instead, evaluation should consider whether the encoded representation retains the information necessary for the receiver to understand the source or perform its intended task. This approach establishes a fundamental principle of semantic communication: compression should be guided by meaning rather than by data redundancy alone. By learning which information matters to the receiving application, intelligent semantic encoders can potentially reduce communication overhead while preserving the semantic fidelity and task effectiveness of the transmitted information.

1.3.3 Semantic Channel Modeling and Intelligent Information Transmission

The semantic channel represents the communication environment through which semantic representations travel from the transmitter to the receiver. Although semantic communication introduces meaning-aware processing, the transmitted information remains subject to physical communication constraints such as noise, interference, fading, limited bandwidth, attenuation, and environmental disturbances. Consequently, semantic communication requires an understanding of how physical channel conditions influence the preservation and recovery of semantic information.

In a conventional communication system, channel modeling focuses primarily on characteristics such as signal-to-noise ratio, channel capacity, symbol error probability, fading behavior, and packet reliability. Semantic communication introduces an additional consideration: how channel impairments affect meaning and task performance. A small physical error may have negligible semantic consequences if the essential meaning remains intact, whereas a seemingly minor change in a critical semantic feature may produce a significant impact on the receiver's decision.

Intelligent information transmission therefore seeks to adapt transmission strategies according to both channel conditions and semantic importance. Information with high semantic relevance may receive greater protection or communication resources, while less important information can tolerate greater distortion. This creates the possibility of unequal treatment of information based on its contribution to the receiver's objective.

For example, consider a remote industrial monitoring system in which numerous sensors continuously generate measurements. If the receiver is primarily concerned with detecting abnormal operating conditions, transmitting every measurement with identical reliability may be inefficient. Measurements associated with emerging anomalies may require higher transmission priority, whereas routine and predictable measurements may be transmitted less frequently or represented more compactly.

Semantic channel modeling consequently requires interaction between communication-layer information and application-layer requirements. Channel conditions can influence how much semantic information should be transmitted, while semantic importance can influence how communication resources should be allocated. This creates a more adaptive communication process than the conventional assumption that all transmitted bits should receive approximately equivalent treatment.

The semantic channel should therefore not be interpreted as a replacement for the physical channel. Rather, it represents an additional perspective for understanding how physical transmission conditions interact with semantic information. Intelligent transmission mechanisms can use channel-state information, semantic importance, contextual knowledge, and task requirements to determine appropriate transmission strategies. Ultimately, the objective is not simply to maximize the number of successfully received bits. It is to maximize the amount of useful semantic information delivered under available communication resources, establishing a direct connection between physical channel behavior and application-level communication effectiveness.

1.3.4 Semantic Decoding, Reconstruction, and Knowledge-Based Interpretation

Semantic decoding is the receiver-side process responsible for converting the received representation into meaningful information that can be interpreted or used by the target application. In a conventional communication architecture, decoding primarily attempts to recover the transmitted data from the received signal. In semantic communication, the receiver has an additional responsibility: it must recover the intended semantic content even when the received representation does not exactly reproduce the original source.

The process generally begins with the output of the channel decoder. The received representation may contain distortions caused by noise, interference, fading, or other channel impairments. The semantic decoder processes this representation and attempts to reconstruct the relevant concepts, relationships, attributes, or task information. Depending on the application, the reconstructed output may take the form of an image, text, speech signal, recognized object, inferred event, control instruction, or another meaningful representation.

Knowledge and context can significantly support semantic interpretation. A receiver may possess background knowledge that allows it to infer information that was not explicitly transmitted. For example, if the receiver knows the operating conditions of an industrial process, it may infer the significance of a particular sensor pattern without receiving every associated measurement. Similarly, contextual information such as location, time, previous observations, or task requirements can help resolve ambiguity in incomplete semantic representations.

Semantic reconstruction therefore does not necessarily mean reproducing the source exactly. The receiver may generate a representation that differs from the original data while preserving the concepts required by the application. For instance, a reconstructed image may contain fewer visual details than the original but still accurately identify the objects and relationships relevant to a navigation task. In this situation, semantic reconstruction can be considered successful even when pixel-level similarity is not perfect.

The final stage is knowledge-based interpretation, where reconstructed semantic information is integrated with available knowledge and application context. This stage connects communication with decision-making and task execution. The quality of semantic decoding can therefore be evaluated using application-specific measures such as recognition accuracy, inference reliability, decision accuracy, control performance, or task success. Semantic decoding completes the end-to-end semantic communication process by transforming received representations into usable meaning. Together with semantic source modeling, intelligent encoding, and semantic-aware transmission, it establishes an

architecture in which communication performance is ultimately connected to the receiver's ability to understand and act upon the transmitted information.

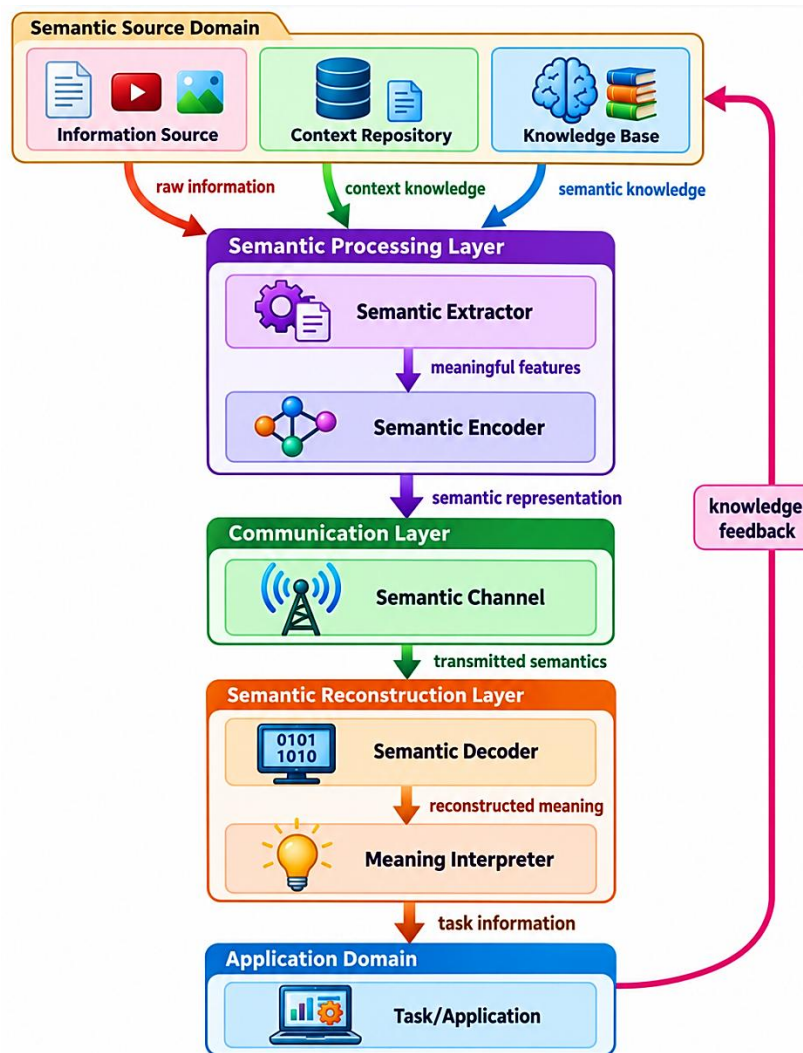


Figure 2: Knowledge- and Context-Aware Architecture for Semantic Information Processing and Task-Oriented Communication

A knowledge- and context-aware architecture for semantic information processing and communication. The process begins within the Semantic Source Domain, which contains three complementary components: the information source, context repository, and knowledge base. The information source represents heterogeneous raw information such as documents, video, images, and other source modalities. The context repository provides contextual information associated with the source and communication environment, while the knowledge base provides semantic knowledge that can support interpretation and representation. The figure therefore emphasizes that meaningful communication does not depend exclusively on raw source data; it can also exploit contextual and previously available knowledge.

The information originating from these components enters the Semantic Processing Layer, where the semantic extractor identifies meaningful features from the available information. These features are subsequently processed by the semantic encoder to generate a compact semantic representation. The

distinction between the semantic extractor and semantic encoder is important: extraction focuses on identifying meaningful information, whereas encoding transforms that information into a representation suitable for communication. This stage demonstrates how semantic communication can reduce dependence on direct transmission of complete raw information by focusing instead on features and representations that carry greater semantic significance.

The resulting semantic representation enters the Communication Layer, represented by the semantic channel in the figure. After transmission, the information reaches the Semantic Reconstruction Layer, where the semantic decoder processes the received representation and the meaning interpreter converts it into reconstructed meaning. The reconstructed meaning is then transformed into task information for the target application. This arrangement illustrates that the ultimate purpose of semantic communication is not necessarily to reproduce the original source exactly, but to provide information that can be interpreted and used effectively by the receiving application. The application domain may involve decision-making, monitoring, recognition, control, or other intelligent tasks.

An important feature of the architecture is the knowledge feedback pathway connecting the application domain back toward the semantic source domain. This feedback mechanism indicates that semantic communication can operate as an adaptive process rather than as a fixed one-way transmission pipeline. Information obtained from application outcomes can influence the contextual or semantic knowledge available to subsequent processing. As a result, the system can potentially adapt its semantic representation according to changing tasks, environmental conditions, and knowledge requirements. The overall architecture therefore illustrates the interaction among raw information, context, knowledge, semantic extraction, semantic encoding, communication, semantic decoding, interpretation, and task execution, providing a useful foundation for understanding how semantic source modeling fits into an end-to-end semantic communication system.

Chapter 2

Information-Theoretic Foundations and Mathematical Models of Semantic Communication

2.1 Classical Information Theory as the Foundation of Modern Communication Systems

2.1.1 Information Entropy, Uncertainty, and Information Representation

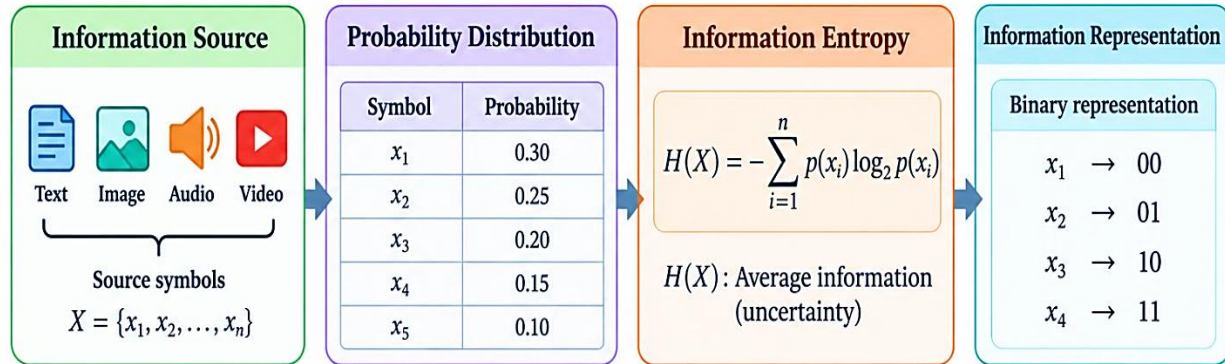


Figure 3: Information Entropy, Uncertainty, and Information Representation

The relationship among an information source, the probability distribution of its possible symbols, information entropy, and the resulting information representation. The process begins with an information source that may generate different types of content, including text, images, audio, and video. The source is represented mathematically as a random variable (X), whose possible outcomes can be expressed as ($X = \{x_1, x_2, \dots, x_n\}$). Each source symbol does not necessarily occur with the same probability. The probability distribution shown in the figure assigns probabilities to the possible outcomes, illustrating that some symbols may occur more frequently than others.

The information content associated with an individual source event can be described using self-information. For a symbol (x_i) with probability ($p(x_i)$), its information content is commonly expressed as

$$I(x_i) = -\log_2(p_i)$$

This relationship shows that less probable events carry greater information because their occurrence is less expected. Conversely, highly probable events provide less new information. The use of the logarithm also allows information from independent events to be combined additively. When the probabilities shown in the figure are considered collectively, the average information produced by the source can be quantified using Shannon entropy.

For a discrete random variable (X), the Shannon entropy is given by

$$H(X) = -\sum_{i=1}^n p_i (\log_2 p_i)$$

Here, $(p(x_i))$ represents the probability of source symbol (x_i) , while $(H(X))$ represents the average uncertainty or average information associated with the source. For the probability distribution illustrated in the figure, where the probabilities are (0.30, 0.25, 0.20, 0.15,) and (0.10), the entropy can be calculated as

$$H(X) = -[0.30\log_2(0.30) + 0.25\log_2(0.25) + 0.20\log_2(0.20) + 0.15\log_2(0.15) + 0.10\log_2(0.10)]$$

$$H(X) \approx 2.23\text{bits/symbol}$$

Information representation, where source symbols are mapped into binary representations. For example, four symbols can be represented using two bits, such as $[x_1 \rightarrow 00, x_2 \rightarrow 01, x_3 \rightarrow 10, x_4 \rightarrow 11]$. This demonstrates the fundamental relationship between probability, information content, entropy, and digital representation. In conventional communication, these principles provide the mathematical foundation for source coding and efficient bit representation. In semantic communication, the same information-theoretic foundation is extended by asking which information is not only statistically informative but also semantically relevant and useful to the receiver's task. Thus, entropy provides an important starting point for understanding how communication systems can quantify information before introducing semantic relevance, semantic fidelity, and task-oriented information measures in later sections.

2.1.2 Mutual Information and Statistical Dependence between Communication Entities

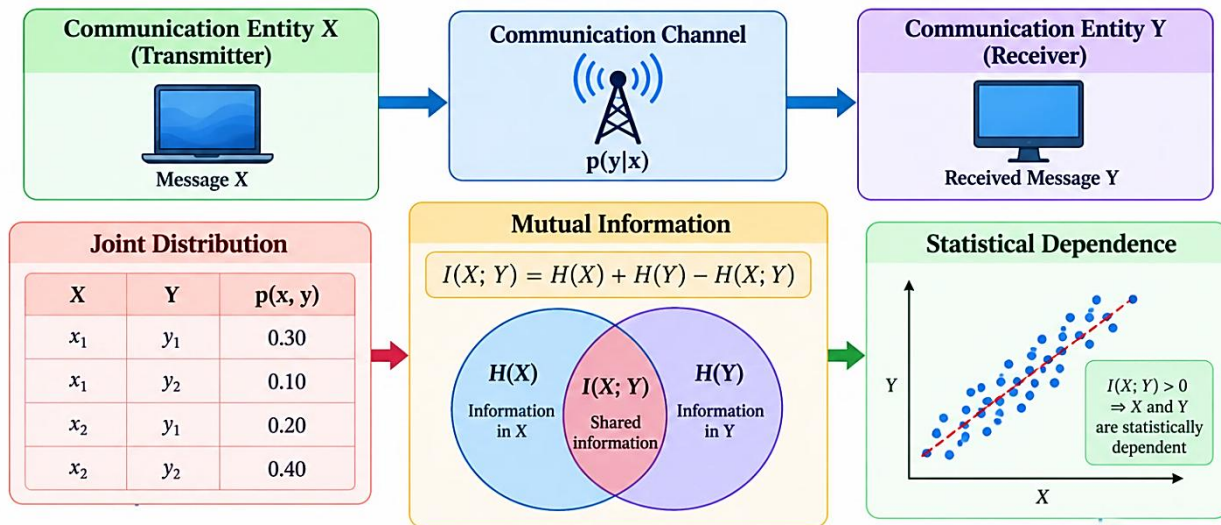


Figure 4: Mutual Information and Statistical Dependence between Communication Entities

The statistical relationship between two communication entities, represented by the transmitter (X) and receiver (Y). The transmitter generates a message (X), which is transmitted through a communication channel characterized by the conditional distribution $(p(y|x))$. The receiver obtains the corresponding received message (Y). Because the communication channel establishes a relationship between the transmitted and received variables, the statistical behavior of (Y) generally depends on (X). This dependence provides the mathematical basis for measuring how much information about the transmitted message is available at the receiver.

The joint distribution shown in the figure describes the probabilities of different combinations of transmitted and received outcomes. For example, the illustrated values assign $p(x_1, y_1)=0.30$, $p(x_1, y_2)=0.10$, $p(x_2, y_1)=0.20$, and $p(x_2, y_2)=0.40$. These probabilities characterize the relationship between the two communication entities. From the joint distribution, the marginal distributions ($p(x)$) and ($p(y)$) can also be obtained by summing over the corresponding variable. The joint entropy is then expressed as

$$H(X, Y) = -\sum_x \sum_y p(x, y) \log_2 p(x, y)$$

The central mathematical concept in the figure is mutual information, which quantifies the amount of information shared between (X) and (Y). It can be expressed as

$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

An equivalent formulation based directly on the joint and marginal probability distributions is

$$I(X; Y) = \sum_x \sum_y p(x, y) \log_2 \left(\frac{p(x, y)}{p(x)p(y)} \right)$$

The Venn-style representation in the figure illustrates ($I(X; Y)$) as the information common to (X) and (Y). When (X) and (Y) are statistically independent, their joint distribution can be written as ($p(x, y)=p(x)p(y)$), resulting in ($I(X; Y)=0$). In contrast, a positive mutual information value indicates that knowledge of one variable provides information about the other.

The statistical-dependence graph shown on the right provides an intuitive interpretation of this relationship. The positively correlated distribution of observations indicates that changes in (X) are associated with changes in (Y), demonstrating statistical dependence. In semantic communication, this concept becomes particularly important because the receiver is expected to recover information that remains related to the source's meaningful content. Mutual information can therefore serve as an important theoretical bridge between classical information theory and semantic communication, although mutual information alone does not fully measure semantic meaning or task effectiveness. Later semantic communication models can build upon this foundation by incorporating semantic relevance, contextual information, and task-specific utility into communication objectives.

2.1.3 Channel Capacity and Fundamental Limits of Reliable Communication

The classical communication-channel model and the fundamental limit imposed by channel capacity on reliable information transmission. The process begins with the transmitter, which generates an input (X) and produces a transmitted signal ($x(t)$). The signal passes through a communication channel represented by the channel model ($h(t)$), where noise ($n(t)$) is introduced before the resulting signal ($y(t)$) reaches the receiver. In a general channel model, the received signal can be represented conceptually as

$$y(t) = h(t) * x(t) + n(t)$$

where $(h(t)*x(t))$ represents the channel response to the transmitted signal and $(n(t))$ represents additive channel noise. This model provides the basic physical foundation for understanding reliable communication in the presence of channel impairments.

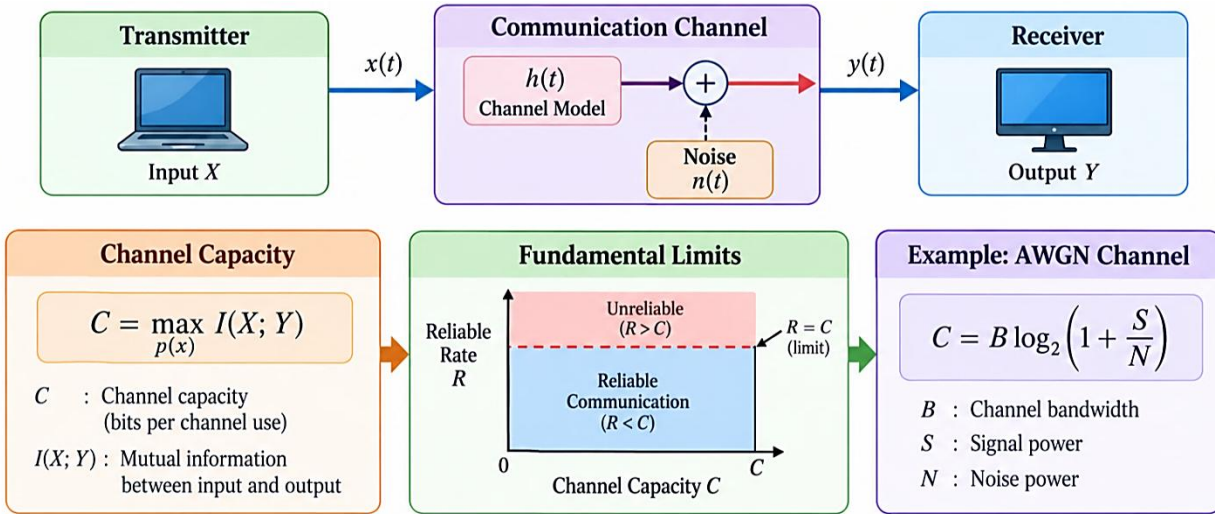


Figure 5: Channel Capacity and Fundamental Limits of Reliable Communication

The central concept shown in the figure is channel capacity, which represents the maximum information rate that can theoretically be achieved while maintaining arbitrarily reliable communication under the assumptions of the channel model. Channel capacity can be expressed in terms of mutual information as

$$C = \max_{p(x)} I(X; Y)$$

where (C) denotes channel capacity, $(p(x))$ represents the input probability distribution, and $(I(X;Y))$ represents the mutual information between the channel input (X) and output (Y) .

$$R < C$$

reliable communication is theoretically achievable under the corresponding channel assumptions. When

$$R > C,$$

reliable communication cannot be achieved asymptotically, regardless of the coding strategy used within the model's assumptions. For an additive white Gaussian noise (AWGN) channel, the figure provides the familiar Shannon capacity expression

$$C = B \log_2(1 + SN)$$

Semantic communication lies in establishing the classical physical-layer baseline against which meaning-oriented communication can be understood. Traditional channel capacity focuses on the reliable transmission of information under physical channel constraints, whereas semantic communication introduces additional objectives concerning meaning, relevance, and task effectiveness. A semantic system may therefore seek to use limited communication resources more intelligently by prioritizing

information that is important to the receiver rather than treating every transmitted bit as equally valuable. Channel capacity remains fundamental, but semantic communication extends the question from how many bits can be reliably transmitted to how much useful meaning or task-relevant information can be conveyed within those communication constraints.

2.1.4 Source Coding, Channel Coding, and Rate-Distortion Principles

Source coding, channel coding, and rate-distortion principles form three closely related foundations of conventional digital communication systems. Together, they address how information can be represented efficiently, transmitted reliably, and reconstructed with an acceptable level of distortion. Source coding primarily focuses on removing redundancy from information so that it can be represented using fewer communication resources. Channel coding, in contrast, introduces controlled redundancy to protect transmitted information against noise, interference, and other channel impairments. Rate-distortion principles establish the fundamental relationship between the amount of information that must be transmitted and the quality of reconstruction that can be achieved.

Source coding is concerned with efficient representation of information generated by a source. Real-world sources often contain substantial redundancy. Natural language, images, audio, and video may include repeated patterns or information that can be represented more compactly. Compression techniques exploit these characteristics to reduce the amount of data that must be stored or transmitted. Lossless source coding aims to preserve the source information exactly, whereas lossy source coding allows controlled information loss when exact reconstruction is unnecessary. This distinction becomes particularly relevant to semantic communication because information that is visually or statistically insignificant may not necessarily be important from a semantic perspective.

Channel coding addresses the opposite requirement. While source coding attempts to remove redundancy, channel coding deliberately introduces redundancy so that the receiver can detect and correct errors caused by the communication channel. Coding strategies allow communication systems to operate reliably even when the transmitted signal encounters noise, fading, interference, or other impairments. The balance between coding overhead, transmission rate, reliability, and available channel resources is therefore a central consideration in communication-system design.

Rate-distortion principles provide a framework for understanding the trade-off between communication rate and reconstruction quality. When communication resources are limited, reducing the amount of transmitted information generally requires accepting some degree of distortion. The acceptable distortion depends strongly on the application. A small degradation may be insignificant for one application but unacceptable for another. For example, reduced visual detail may have little effect on a general monitoring task but could be critical in a safety-sensitive recognition system.

These classical principles provide an important foundation for semantic communication. However, semantic systems introduce a broader notion of distortion in which degradation is evaluated not only according to reconstructed data quality but also according to meaning preservation, semantic relevance, and task effectiveness. This creates a conceptual transition from conventional rate-distortion optimization toward communication strategies that determine how much information must be transmitted to preserve the meaning necessary for a particular application.

2.2 Development of Semantic Information Theory and Meaning-Aware Communication Models

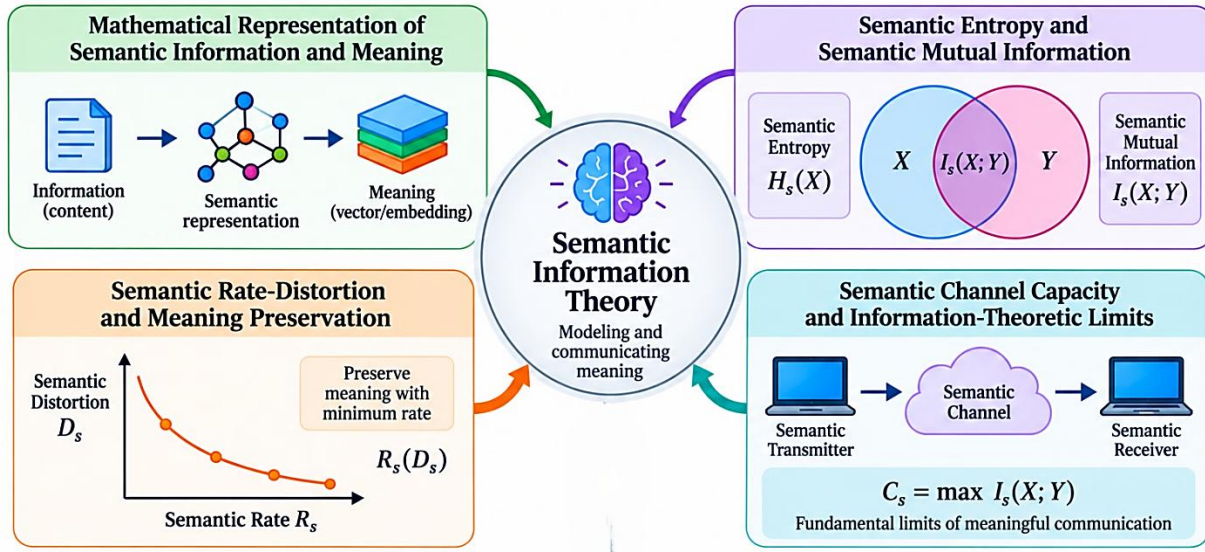


Figure 6: Conceptual Foundations of Semantic Information Theory

A conceptual framework for the development of semantic information theory, showing how classical information-theoretic concepts can be extended toward the representation, quantification, and transmission of meaning. At the center of the figure is semantic information theory, which provides a conceptual connection between information representation and communication of meaning. The upper-left component illustrates the transformation from information content into a semantic representation and subsequently into a meaning-oriented representation such as a vector or embedding. This reflects the fundamental idea that semantic communication requires information to be represented at a level that captures concepts and relationships rather than relying exclusively on raw symbolic or bit-level representations.

The upper-right component introduces semantic entropy and semantic mutual information. Semantic entropy, represented as $(H_s(X))$, is intended to characterize uncertainty associated with semantic information, while semantic mutual information, represented as $(I_s(X;Y))$, describes the shared semantic information between the transmitted and received representations. The Venn-style representation emphasizes that the overlap between (X) and (Y) can be interpreted in terms of shared semantic content. These concepts extend the intuition of classical entropy and mutual information by shifting attention from statistical information alone toward information that has meaning for the communicating entities.

The lower-left component illustrates semantic rate-distortion and meaning preservation. The plotted relationship between semantic distortion (D_s) and semantic rate (R_s) represents the trade-off between the amount of semantic information transmitted and the degree of meaning loss tolerated during communication. The conceptual objective is to preserve the essential meaning using an appropriate or minimum communication rate. This differs from conventional rate-distortion analysis because distortion can be interpreted according to semantic degradation rather than solely according to numerical, signal-level, or perceptual differences. The lower-right component then extends this perspective to semantic

channel capacity and information-theoretic limits, where semantic channel capacity is represented conceptually as

$$C_s = \max I_s(X; Y)$$

This formulation indicates the maximum achievable shared semantic information under the considered semantic channel model. The figure therefore provides a useful conceptual bridge between classical information theory and meaning-aware communication. It also prepares the reader for later sections in which semantic information may be formally modeled, quantified, encoded, transmitted, and reconstructed. Importantly, the terms shown in the figure such as $(H_s(X))$, $(I_s(X; Y))$, $(R_s(D_s))$, and (C_s) should be presented in the subsequent text as proposed or model-dependent semantic extensions, rather than implying that there is one universally accepted set of semantic-information-theoretic definitions. This distinction will make the chapter technically more rigorous.

2.2.1 Mathematical Representation of Semantic Information and Meaning

Mathematical representation of semantic information provides a formal mechanism for describing the meaning contained in raw data and transforming that meaning into representations suitable for computation and communication. In conventional information theory, a source is commonly represented as a random variable (X) with possible outcomes $(x_1, x_2, \dots, x_n)$. Semantic communication introduces an additional representation layer in which the source observation is mapped to a semantic representation that captures concepts, attributes, relationships, or task-relevant information. This representation can be learned using machine learning models, knowledge-based methods, vector embeddings, graphs, or combinations of these approaches.

Let (S) denote the original source information and let (Z_s) denote its semantic representation. A general semantic mapping can be expressed as

$$Z_s = f_{sem}(S, C, K, T)$$

where (C) represents contextual information, (K) represents available knowledge, and (T) represents the intended task. This formulation emphasizes that semantic representation is not necessarily determined by the source alone. The same source can produce different semantic representations depending on the context, knowledge available to the system, and requirements of the application.

Semantic representations can take several forms. A text message may be transformed into a vector embedding representing its conceptual content. An image may be represented through objects, spatial relationships, attributes, and scene-level concepts. Speech may be represented through linguistic content, speaker information, intent, or emotional characteristics when these are relevant to the application. Structured representations such as semantic graphs can explicitly model relationships among entities, while learned latent representations can encode semantic characteristics in a continuous vector space.

A useful distinction is therefore made between syntactic representation and semantic representation. Syntactic representation describes how information is physically or digitally encoded, whereas semantic representation describes what that information means or how it contributes to a task. A semantic

representation should ideally preserve the information necessary for the receiver to interpret or use the source effectively while allowing irrelevant details to be omitted.

The mathematical formulation of semantic information remains application-dependent because there is no single universally accepted representation of meaning. Consequently, semantic variables, embeddings, task representations, and semantic similarity measures must be defined according to the communication scenario. This flexibility is important because semantic communication spans heterogeneous sources and applications.

Thus, mathematical semantic representation establishes the foundation for subsequent concepts such as semantic entropy, semantic mutual information, semantic distortion, and semantic channel capacity. It provides the mechanism through which abstract concepts such as meaning, relevance, context, and intent can be incorporated into communication models and eventually optimized alongside conventional communication resources.

2.2.2 Semantic Entropy and Semantic Mutual Information

Semantic entropy and semantic mutual information extend the intuition of classical information theory toward information represented according to meaning. Classical entropy quantifies the uncertainty associated with a random variable, while mutual information measures the statistical information shared between two variables. In semantic communication, analogous quantities can be introduced to characterize uncertainty and shared information at the semantic level. However, unlike classical Shannon entropy and mutual information, their precise definitions depend on how semantic information, similarity, context, and task relevance are modeled.

Let (Z_s) represent a semantic variable associated with source information. A semantic entropy can be conceptually represented as

where (H_s) describes uncertainty associated with the semantic states of the source. The interpretation of this quantity depends on the underlying semantic representation. If semantic states are discrete and probabilistically defined, entropy-like formulations can be constructed using their probability distribution. For continuous semantic embeddings, alternative measures may be required because conventional discrete entropy cannot be directly applied without appropriate modeling or quantization.

Semantic mutual information is intended to characterize the amount of semantic information shared between a transmitted semantic representation (Z_s) and a received representation $(\{\hat{Z}\}_s)$. It can be represented conceptually as

$$I_s(Z_s; \hat{Z}_s)$$

where the subscript (s) emphasizes that the quantity is defined with respect to a semantic representation or semantic information measure. A simple conceptual extension may take the form

$$I_s(Z_s; \hat{Z}_s) = H_s(Z_s) - H_s(Z_s | \hat{Z}_s)$$

This expression emphasizes the reduction in semantic uncertainty achieved through the received information.

The key distinction from classical mutual information is that semantic similarity or semantic utility may not be equivalent to statistical dependence. Two representations can differ substantially at the bit level while conveying essentially the same concept. Conversely, two statistically related representations may not preserve the specific meaning required for a particular application. Semantic mutual information therefore requires an underlying definition of semantic equivalence, relevance, or utility.

For example, consider an image containing a pedestrian crossing a road. A conventional information measure may evaluate the statistical relationship between transmitted and received image representations. A semantic measure may instead focus on whether the receiver retains information about the pedestrian, location, movement, and potential hazard. If those critical concepts remain available despite changes in visual details, semantic information preservation may remain high.

Therefore, semantic entropy and semantic mutual information should be introduced as model-dependent extensions rather than universally standardized replacements for Shannon measures. Their usefulness depends on the chosen semantic representation, probability model, similarity criterion, contextual knowledge, and application task. These concepts nevertheless provide an important theoretical bridge between classical information theory and meaning-oriented communication by enabling uncertainty and information sharing to be considered at a semantic level.

2.2.3 Semantic Rate-Distortion and Meaning Preservation

Rate-distortion theory provides a fundamental framework for studying the trade-off between the amount of information transmitted and the quality of the reconstructed information. In conventional communication systems, distortion is commonly defined according to differences between an original source and its reconstruction. Depending on the application, distortion may involve numerical error, signal degradation, image quality, perceptual similarity, or another source-specific criterion. Semantic communication extends this idea by considering whether the meaning of the source has been preserved despite compression, transmission errors, or deliberate information reduction.

Let (D_s) denote semantic distortion and (R_s) denote the semantic transmission rate. A conceptual semantic rate-distortion relationship can be represented as

$$R_s(D_s)$$

which describes the minimum semantic communication rate required to achieve a specified level of semantic distortion under a particular source and communication model. Unlike conventional rate-distortion theory, the central challenge is defining (D_s) in a meaningful and application-relevant manner.

Semantic distortion can reflect differences in concepts, relationships, attributes, intent, or task-relevant information between the original and reconstructed representations. For example, an image reconstructed with lower resolution may have substantial pixel-level distortion but very little semantic distortion if the receiver can still correctly identify the objects required for its task. Conversely, a small modification that

changes the identity of a critical object may result in substantial semantic distortion despite relatively small numerical differences.

The semantic rate-distortion perspective therefore encourages communication systems to allocate resources according to meaning preservation rather than uniform data fidelity. A semantic encoder may discard details that have low relevance while preserving information that strongly influences the receiver's objective. This can potentially reduce communication requirements while maintaining acceptable task performance. Semantic distortion can be defined using different approaches depending on the application. Embedding-space distances can measure differences between semantic representations, while classification accuracy, task loss, decision error, or application-specific utility can provide indirect measures of semantic preservation. In multimodal systems, semantic consistency between modalities may also be incorporated.

The resulting trade-off is not necessarily a single universal curve. Different semantic representations, tasks, receiver knowledge, and distortion measures can produce different rate-distortion relationships. Therefore, semantic rate-distortion theory should be presented as a general modeling framework rather than as one universally established mathematical formulation. The importance of this concept lies in changing the central optimization question. Conventional rate-distortion theory asks how much data must be transmitted to achieve a specified reconstruction quality. Semantic rate-distortion analysis instead asks how much information must be communicated to preserve the meaning necessary for the receiver's objective. This provides a direct theoretical connection between information compression, semantic fidelity, communication efficiency, and task-oriented communication.

2.2.4 Semantic Channel Capacity and Information-Theoretic Limits

Channel capacity represents one of the most important concepts in classical information theory because it establishes the maximum rate at which information can theoretically be transmitted reliably through a communication channel under specified assumptions. Semantic communication extends this perspective by asking whether communication limits can also be described in terms of the amount of meaningful or task-relevant information that can be conveyed. This creates the concept of semantic channel capacity, although its exact mathematical definition remains dependent on the semantic representation and performance criterion being used.

In a conventional discrete communication model, channel capacity is expressed as the maximum mutual information between the channel input and output:

$$C = \max_{\mathbf{x}} I(\mathbf{X}; \mathbf{Y})$$

For a semantic communication model, a corresponding conceptual formulation can be written as

$$C_s = \max_{\mathbf{Z}_s} I(\mathbf{Z}_s; \hat{\mathbf{Z}}_s)$$

where $(\mathbf{Z}_s)$ denotes the transmitted semantic representation and $(\{\hat{\mathbf{Z}}\}_s)$ denotes the received or reconstructed semantic representation. Here, (C_s) represents a model-dependent measure of the maximum achievable shared semantic information under the assumptions of the semantic channel.

The interpretation of semantic capacity differs from conventional channel capacity because successful communication is not necessarily determined by exact reconstruction of the transmitted representation. A receiver may successfully accomplish its task even when substantial low-level information has been lost. Consequently, semantic capacity may depend on semantic fidelity, task success, contextual knowledge, receiver capabilities, and the semantic representation itself.

For example, consider a communication system supporting autonomous navigation. A conventional capacity analysis may focus on the number of bits that can be transmitted reliably under a given bandwidth and signal-to-noise ratio. A semantic analysis may additionally consider whether the transmitted information is sufficient for identifying obstacles, estimating their positions, and making safe navigation decisions. Two systems operating at the same physical transmission rate may therefore provide different levels of task-relevant communication performance if they use different semantic representations or resource-allocation strategies.

Semantic channel capacity should consequently be treated carefully. Unlike Shannon channel capacity, there is no single universally accepted semantic-capacity equation applicable to all semantic communication systems. Different research models may define semantic information, semantic distortion, and task utility differently. Some approaches may formulate semantic capacity through shared semantic representations, while others may connect it directly to task performance or utility.

The value of semantic channel-capacity analysis is therefore not to replace classical capacity theory but to extend the communication objective. Physical channel capacity continues to establish fundamental constraints on available communication resources, while semantic models investigate how effectively those resources can be used to convey meaning. This provides a theoretical foundation for meaning-aware resource allocation, task-oriented transmission, semantic coding, and intelligent networking, where the ultimate objective becomes maximizing useful information rather than maximizing transmitted bits alone.

2.3 Semantic Distortion, Fidelity, and Task-Oriented Communication Quality

A conceptual framework for evaluating semantic communication quality beyond conventional bit-level transmission accuracy. The upper-left portion distinguishes bit-level distortion from semantic distortion. Bit-level distortion is associated with errors in binary symbols or transmitted data, whereas semantic distortion concerns changes in the underlying meaning of the information. This distinction is fundamental because the same number of bit or symbol errors can produce very different semantic consequences. An error affecting an irrelevant detail may have little influence on the receiver's interpretation, while an error affecting a critical object, instruction, or relationship may significantly alter the intended meaning.

The upper-right portion of the figure focuses on semantic fidelity and meaning preservation. Semantic fidelity concerns how successfully the intended meaning of the original information is retained after communication. The objective is therefore not necessarily exact reproduction of every element of the source, but preservation of the semantic content required by the receiving system. This perspective is particularly important for lossy semantic transmission, where information may intentionally be removed or transformed. A communication system can tolerate certain data-level changes as long as the concepts, relationships, or information required by the receiver remain sufficiently preserved.

The lower-left component introduces task-oriented semantic quality and application-level accuracy. It illustrates a progression from communication output to task processing and ultimately to task performance. This emphasizes that communication quality should be evaluated according to what the receiving system can accomplish with the transmitted information. For example, in an autonomous system, successful communication may be reflected in accurate object detection, reliable navigation, or correct decision-making rather than simply high reconstruction similarity. The lower-right component extends this idea through context-aware evaluation, incorporating factors such as environment, user, domain, and situation. Semantic quality can therefore vary according to the circumstances in which information is interpreted and used.

Taken together, the figure establishes a multidimensional view of semantic communication quality: data-level accuracy alone is insufficient; semantic fidelity, meaning preservation, task success, and contextual relevance must also be considered. This framework provides a useful transition from classical communication metrics toward application-aware evaluation. It also supports the later development of semantic distortion measures and task-specific quality metrics, where communication performance can be assessed according to whether the transmitted information enables the intended receiver to understand, infer, decide, or act successfully.

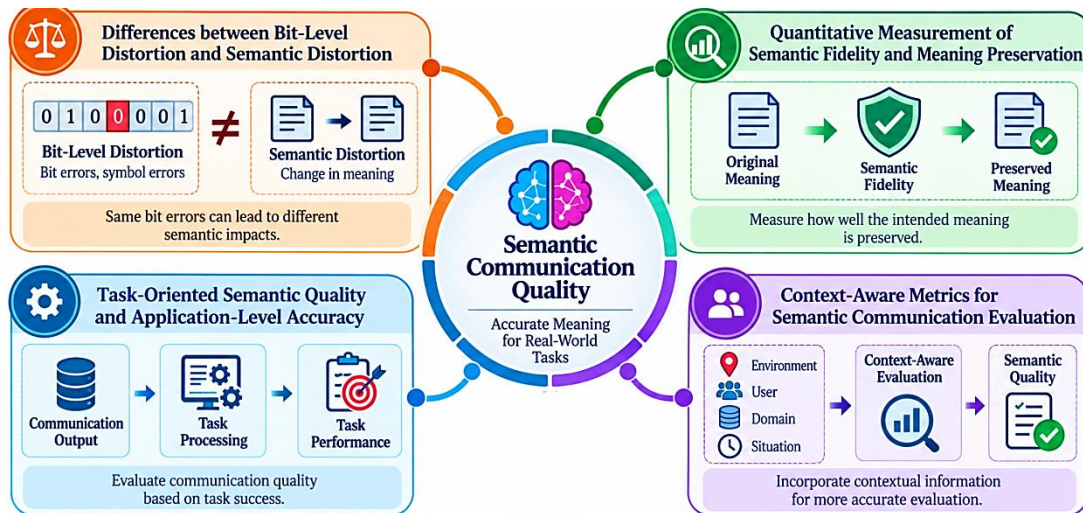


Figure 7: Semantic Communication Quality: Distortion, Fidelity, and Task-Oriented Evaluation

2.3.1 Differences between Bit-Level Distortion and Semantic Distortion

Bit-level distortion and semantic distortion represent two fundamentally different perspectives for evaluating the degradation of information during communication. In conventional digital communication systems, distortion or error is generally assessed by examining differences between transmitted and received bits, symbols, samples, or reconstructed signals. Metrics such as bit error rate, symbol error rate, packet error rate, and signal-level distortion provide valuable information about the reliability of the physical communication process. However, these measures do not necessarily indicate whether the receiver has preserved the intended meaning of the transmitted information.

Semantic distortion focuses instead on the difference between the intended meaning and the meaning recovered by the receiver. A communication system may experience several bit errors while retaining the

essential semantic content of a message. For example, a small alteration in the background of an image may produce measurable pixel-level distortion but have almost no effect on object recognition. Conversely, a single error affecting a critical word, object identity, numerical value, or control instruction may produce a substantial change in meaning. Therefore, bit-level distortion and semantic distortion do not necessarily have a direct one-to-one relationship.

This distinction can be represented conceptually by defining a source semantic representation (Z_s) and a reconstructed semantic representation ($\{\hat{Z}\}_s$). A general semantic distortion measure can be written as

$$D_s = d_s(Z_s, \hat{Z}_s)$$

where ($d_s(\cdot)$) represents a semantic distance or task-relevant discrepancy. The exact definition of (d_s) depends on the application and may involve differences in concepts, embeddings, relationships, intent, or task-relevant attributes.

The distinction becomes particularly important in multimodal and intelligent communication systems. For text, semantic distortion may involve changes in the intended statement or meaning. For images, it may involve incorrect object identification or spatial relationships. For speech, it may involve changes in recognized words or speaker intent. For machine-to-machine communication, semantic distortion may instead be reflected in incorrect system states or decisions.

Thus, semantic distortion should not be interpreted simply as a more advanced version of bit error. It represents a different evaluation level. Bit-level measures characterize how accurately data has been transmitted, whereas semantic distortion characterizes how much meaningful information has changed. This distinction provides the foundation for semantic communication systems in which communication resources can be allocated according to the importance of information and the consequences of its degradation.

2.3.2 Quantitative Measurement of Semantic Fidelity and Meaning Preservation

Semantic fidelity describes the extent to which the intended meaning of transmitted information remains preserved after communication. Quantifying semantic fidelity is considerably more challenging than measuring conventional transmission accuracy because meaning is dependent on representation, context, knowledge, and application requirements. A suitable semantic fidelity measure should therefore evaluate whether the information that matters to the receiver remains sufficiently consistent after transmission.

One approach is to compare the semantic representations extracted from the original and reconstructed information. Let (Z_s) represent the semantic representation of the source and ($\{\hat{Z}\}_s$) represent the corresponding reconstructed representation. A similarity-based fidelity measure can conceptually be expressed as

$$F_s = \text{sim}(Z_s, \hat{Z}_s)$$

where ($\text{sim}(\cdot)$) denotes an appropriate semantic similarity function. Depending on the application, this function may operate on learned embeddings, structured semantic representations, concepts, relationships, or other meaning-oriented representations.

Semantic fidelity can also be evaluated through the preservation of specific attributes. For example, a visual communication system may evaluate whether object identities, positions, and relationships remain correct. A natural-language system may evaluate whether the key concepts and intent of a statement are preserved. In speech communication, fidelity may involve preservation of linguistic content or task-relevant speech characteristics. These application-specific measures can provide a more meaningful assessment than generic signal reconstruction metrics.

Another useful perspective is to relate semantic fidelity to semantic distortion. If (D_s) represents semantic distortion, a normalized fidelity measure can conceptually be expressed as

$$F_s = 1 - D_s$$

where D_s is a normalized semantic distortion value. This relationship is useful for conceptual analysis, although the exact normalization and interpretation must be defined for the specific application.

Meaning preservation should also consider the relative importance of different semantic elements. Critical information may deserve greater weighting than peripheral information. For example, in a road-scene communication system, the location and movement of a pedestrian may be considerably more important than background scenery. A weighted semantic fidelity measure can therefore provide a more realistic evaluation of communication quality.

Importantly, there is no single universal semantic-fidelity metric suitable for all semantic communication systems. The appropriate measure depends on the semantic representation, modality, context, receiver capabilities, and intended task. Consequently, semantic fidelity should be evaluated using a combination of representation-level, perceptual, and application-specific measures where appropriate.

The central objective is to determine whether the meaning that matters has survived communication. This shifts evaluation away from asking whether the reconstructed data is identical to the original and toward determining whether the receiver retains sufficient semantic information for reliable interpretation and subsequent task execution.

2.3.3 Task-Oriented Semantic Quality and Application-Level Accuracy

Task-oriented semantic quality evaluates communication according to the ability of the receiving system to perform its intended application successfully. This represents an important extension beyond source reconstruction and semantic similarity. In many intelligent communication scenarios, the ultimate purpose of transmitting information is not to reproduce the source but to enable a specific task such as classification, detection, prediction, control, decision-making, navigation, or anomaly identification.

Consider a remote camera communicating with an autonomous vehicle. The camera may observe a complex environment containing numerous visual details, but the vehicle may only require information about nearby obstacles and their trajectories. A communication system that transmits fewer details but enables accurate obstacle detection may be more effective for the task than a system that reconstructs visually similar images but occasionally misses critical objects. This example illustrates why application-level performance can provide a more meaningful measure of communication quality.

Let (T) denote the intended task and let (A_T) represent the resulting task accuracy. Semantic communication quality can then be evaluated through an application-specific function such as

$$QT = f(AT)$$

where $(f(.))$ maps task performance to a communication-quality measure. For classification tasks, accuracy, precision, recall, or F1-score may be appropriate. For regression tasks, suitable prediction-error measures can be used. For control applications, task success, control stability, or cumulative performance may provide more meaningful indicators.

Task-oriented evaluation also allows communication systems to distinguish between important and unimportant information. Information that substantially influences the task can receive greater transmission resources, stronger protection, or higher semantic priority. Less important information can be compressed more aggressively or omitted. This creates an explicit connection between semantic encoding and application requirements. An important consequence is that communication quality becomes task-dependent. The same transmitted information can have different values for different applications. A detailed weather image may be highly useful for meteorological analysis but relatively unimportant for a simple object-detection task. Similarly, a sensor measurement may be essential for industrial fault detection but unnecessary for another application.

Task-oriented semantic quality therefore provides a bridge between communication engineering and intelligent application systems. It recognizes that successful communication ultimately depends on whether the receiving system can use the information effectively. However, task accuracy should not completely replace conventional communication metrics. Physical-layer reliability, semantic fidelity, latency, energy consumption, and computational cost can all influence the overall system. A comprehensive evaluation should consequently consider communication quality and task performance jointly. This perspective transforms semantic communication from a purely transmission-oriented problem into an end-to-end optimization problem in which communication resources are selected according to their contribution to the intended application.

2.3.4 Context-Aware Metrics for Semantic Communication Evaluation

Semantic communication quality is strongly influenced by context. The same information may have different meanings, importance, or utility depending on the environment, user, domain, system state, and current task. Consequently, semantic communication evaluation should not rely exclusively on fixed metrics that assume identical conditions across all communication scenarios. Context-aware evaluation incorporates the circumstances surrounding information exchange so that semantic quality can be assessed more realistically.

Several forms of context may influence semantic communication. Environmental context can include location, time, weather, surrounding objects, or physical conditions. User context can include the receiver's information requirements and preferences. Domain context can include application-specific knowledge, while situational context can describe the current task, system state, or operational condition. These factors can influence which semantic information is important and how much distortion can be tolerated.

A context-aware semantic quality measure can be represented conceptually as

$$Q_s = f(Z_s, \hat{Z}_s, C, T)$$

where (Z_s) and $(\{\hat{Z}_s\})$ represent the original and reconstructed semantic information, (C) represents context, and (T) represents the intended task. This formulation emphasizes that semantic quality should be evaluated relative to both the information exchanged and the circumstances in which it is used.

Context can also influence semantic distortion thresholds. For example, a minor positional error may be acceptable in a general monitoring application but unacceptable in an autonomous driving scenario involving a nearby obstacle. Similarly, approximate information may be sufficient during routine system operation but insufficient during an emergency. A context-aware system can therefore dynamically modify semantic priorities and evaluation criteria according to changing conditions.

Context-aware metrics can combine several dimensions, including semantic similarity, meaning preservation, task accuracy, latency, reliability, and information importance. Rather than treating these dimensions independently, an evaluation framework may aggregate them according to application requirements. For example, safety-critical systems may place greater emphasis on task correctness and critical-event detection, while entertainment applications may emphasize perceptual or experiential quality. This approach is particularly important for multimodal and intelligent communication systems, where the meaning of one modality can depend on information available from another. Text, image, speech, and sensor information may need to be evaluated jointly rather than independently. Context can help determine which modalities are relevant and how their information should be combined.

Ultimately, context-aware evaluation recognizes that semantic quality is not an absolute property of a transmitted message. It is a relationship among information, receiver, context, and task. This perspective enables more realistic assessment of semantic communication systems and supports adaptive resource allocation, dynamic semantic encoding, and task-aware transmission. It also provides a foundation for evaluating semantic communication in practical environments where communication objectives and operating conditions continuously change.

2.4 Mathematical Optimization Frameworks for Semantic Communication Systems

2.4.1 Semantic Utility and Meaning-Aware Objective Functions

Semantic communication requires objective functions that capture more than the accuracy of transmitted bits or reconstructed signals. In conventional communication systems, optimization commonly focuses on metrics such as throughput, error probability, spectral efficiency, or reconstruction distortion. In semantic communication, the optimization objective should additionally reflect how useful the transmitted information is to the receiver and how effectively it supports the intended task. This leads to the concept of semantic utility, which represents the value obtained from successfully communicating meaning-relevant information. Semantic utility can depend on several factors, including semantic fidelity, relevance, task accuracy, information importance, and contextual consistency. For example, an autonomous vehicle may assign greater utility to information describing a nearby pedestrian than to information describing distant buildings. Similarly, an industrial monitoring system may assign greater utility to an abnormal sensor event than to routine measurements. The objective function should therefore

distinguish between information that contributes directly to the application and information that provides little practical value.

A general semantic utility function can be represented as

$$U_s = f(F_s, R_s, A_T, C)$$

where (F_s) represents semantic fidelity, (R_s) represents semantic relevance, (A_T) represents task performance, and (C) represents contextual information. The exact form of this function depends on the application and the semantic representation being used. It should therefore not be interpreted as a universal formulation.

Meaning-aware objective functions can also combine communication efficiency with semantic performance. For example, an optimization objective may conceptually maximize

$$\max U_s - \lambda R$$

where (U_s) represents semantic utility, (R) represents communication resource consumption or transmission rate, and (λ) controls the relative importance assigned to communication efficiency.

Semantic utility can also be dynamic. As the environment, user requirements, or task changes, the value of particular information can change. A communication system should therefore be capable of adapting its objective according to current circumstances. This makes semantic utility an important foundation for intelligent resource allocation and task-oriented communication. Ultimately, meaning-aware objective functions provide the mathematical link between semantic representations and system-level optimization. They allow communication systems to evaluate information according to its usefulness, relevance, and contribution to the intended objective, establishing the basis for joint optimization of communication, computation, and semantic quality.

2.4.2 Joint Optimization of Communication, Computation, and Semantic Quality

Semantic communication systems increasingly combine communication resources with computational intelligence. Semantic extraction, representation learning, encoding, decoding, inference, and task execution can require substantial computational resources, particularly when artificial intelligence models are deployed at edge devices or distributed nodes. Consequently, optimizing communication performance independently of computation may lead to inefficient system operation. A more comprehensive approach jointly considers communication resources, computational resources, and semantic quality.

Communication resources include bandwidth, transmission power, spectrum, and channel access. Computational resources include processor capacity, memory, accelerator availability, and inference time. Semantic quality describes the effectiveness with which the transmitted information preserves meaning and supports the intended task. These dimensions are interdependent. For example, a more sophisticated semantic encoder may improve semantic fidelity while requiring additional computation. Conversely, performing more processing locally may reduce the amount of data that must be transmitted but increase device energy consumption and latency.

A conceptual joint optimization problem can be expressed as

$$\text{max}_x U_s(x)$$

subject to

$$R(x) \leq R_{\max}, C_{\text{comp}}(x) \leq C_{\max},$$

where $(\{x\})$ represents system decisions, (U_s) denotes semantic utility, (R) represents communication resource consumption, and (C_{comp}) represents computational resource consumption.

The optimization variables may include semantic representation size, model selection, transmission rate, power allocation, computation placement, compression level, and decoding complexity. For example, a semantic encoder could operate at different levels of representation complexity depending on available bandwidth and processor capacity. Under a congested channel, additional computation might be used to extract a more compact semantic representation. Under severe computational constraints, the system might use a simpler model and transmit richer information.

This interaction is especially important in edge and distributed semantic communication. A device may choose whether to process information locally, transmit raw data to an edge server, or transmit a compact semantic representation. Each choice produces different communication, computation, latency, and energy costs. The optimal decision therefore depends on the complete system environment rather than on a single communication metric.

Joint optimization also encourages cross-layer design. Instead of independently optimizing the physical channel, semantic model, and application, the system can consider their interactions. A change in channel conditions may trigger adaptation of semantic compression, while a change in computational availability may alter encoding or decoding strategies.

The central objective is therefore to achieve an appropriate balance between semantic quality and resource consumption. Such joint optimization provides a foundation for intelligent communication systems in which communication and computation cooperate to deliver the required meaning efficiently.

2.4.3 Latency, Reliability, Energy, and Resource Allocation Constraints

Practical semantic communication systems operate under multiple constraints that directly influence their ability to deliver meaningful information. Among the most important are latency, reliability, energy consumption, and communication-resource availability. Unlike theoretical models that may focus primarily on semantic fidelity or transmission rate, real-world systems must balance these requirements simultaneously. This is particularly important in autonomous vehicles, industrial control, healthcare monitoring, robotics, and other applications where delayed or unreliable information can reduce task effectiveness.

Latency represents the time required to acquire, process, transmit, decode, and interpret semantic information. A semantic representation that arrives after the relevant event has passed may have little practical value even if its semantic fidelity is high. Consequently, semantic systems should consider end-

to-end latency rather than only transmission delay. Local semantic processing may reduce communication volume but increase computational delay, creating a trade-off that must be considered during optimization.

Reliability is another critical constraint. Conventional reliability is often associated with packet or bit error probabilities, while semantic communication additionally considers whether important meaning has been preserved. A system may tolerate some physical-layer errors if they do not affect task performance, while information associated with safety-critical decisions may require substantially stronger protection.

Energy consumption is particularly important for mobile and IoT devices. Semantic extraction, artificial intelligence inference, encoding, transmission, and decoding all consume energy. Reducing transmitted data does not automatically guarantee lower total energy consumption if sophisticated semantic models require substantial computation. Energy-aware optimization must therefore consider the complete processing and communication pipeline.

Resource allocation can be formulated conceptually by constraining available resources:

$$R_{\text{comm}} \leq R_{\text{max}}, E \leq E_{\text{max}}, L \leq L_{\text{max}}$$

Where (R_{comm}) denotes communication-resource consumption, (E) denotes energy consumption, and (L) denotes end-to-end latency.

These constraints can be combined with semantic-quality requirements. For example, a system may seek to maximize semantic utility while ensuring that latency remains below an application-specific threshold and that energy consumption remains within the device budget. The allocation of bandwidth, transmission power, computation, and semantic representation size can then be adapted dynamically.

The importance of these constraints is that semantic quality cannot be optimized independently of system feasibility. A highly accurate semantic representation may be unsuitable if it consumes excessive energy or produces unacceptable latency. Conversely, an extremely compressed representation may satisfy resource constraints but fail to preserve task-critical meaning.

Therefore, practical semantic communication requires coordinated management of latency, reliability, energy, and communication resources. These constraints establish the boundary within which semantic optimization must operate and provide the foundation for multi-objective optimization in meaning-driven networks.

2.4.4 Multi-Objective Optimization for Meaning-Driven Networking

Meaning-driven networking involves simultaneously optimizing several objectives rather than maximizing a single communication metric. Semantic utility, task accuracy, communication efficiency, latency, reliability, energy consumption, computational cost, and resource utilization may all influence the overall performance of an intelligent communication system. Because these objectives can conflict with

one another, semantic networking requires multi-objective optimization frameworks capable of identifying appropriate trade-offs.

For example, increasing semantic fidelity may require transmitting a richer representation, which can increase bandwidth consumption and latency. Increasing reliability may require additional redundancy or transmission power, which can increase energy consumption. More sophisticated semantic models may improve task accuracy but require greater computational resources. Consequently, there may be no single solution that simultaneously provides the best performance across every dimension.

A conceptual multi-objective formulation can be expressed as

$$\max[Us(x), AT(x), -R(x), -L(x), -E(x)],$$

where (U_s) represents semantic utility, (AT) represents task performance, (R) represents communication-resource consumption, (L) represents latency, and (E) represents energy consumption. The vector formulation emphasizes that multiple objectives are considered simultaneously.

One common conceptual approach is to combine these objectives into a weighted function:

$$J = w_1Us + w_2AT - w_3R - w_4L - w_5E$$

where ($w_1, \dots, w_5$) represent application-dependent weighting factors. The weights should not be treated as universal constants. They may depend on the communication scenario, application priorities, system state, and user or network requirements. Another important concept is the Pareto-optimal solution. A solution is Pareto-optimal when improving one objective would require sacrificing at least one other objective. For example, one configuration might provide excellent semantic fidelity with higher energy consumption, while another might reduce energy consumption at the cost of slightly lower semantic quality. Neither configuration is necessarily universally superior; the preferred solution depends on application priorities.

Multi-objective optimization is particularly relevant to meaning-driven networking because network decisions can incorporate semantic importance. Routing, scheduling, spectrum allocation, transmission power, computing placement, and semantic compression can all be coordinated according to the value of information to the target application. The ultimate objective is therefore not simply to maximize throughput or minimize communication cost. Instead, the network should seek a balanced operating point in which available resources are used to deliver the most valuable meaning under application-specific constraints. This perspective transforms networking from a packet-centric resource-allocation problem into a meaning-aware optimization problem and provides a mathematical foundation for intelligent, adaptive, and task-oriented semantic networks.

A mathematical optimization framework for semantic communication in which source information is transformed into semantic representations and optimized according to communication requirements and application objectives. The process begins with source information, which provides the underlying data and source relationships. This information enters the semantic information model, where semantic

entropy characterizes uncertainty associated with the semantic source and semantic mutual information represents useful information shared between communicating entities. Semantic rate-distortion is then used to characterize the relationship between semantic transmission rate and the acceptable level of semantic distortion. Together, these components provide the mathematical information required for subsequent optimization.

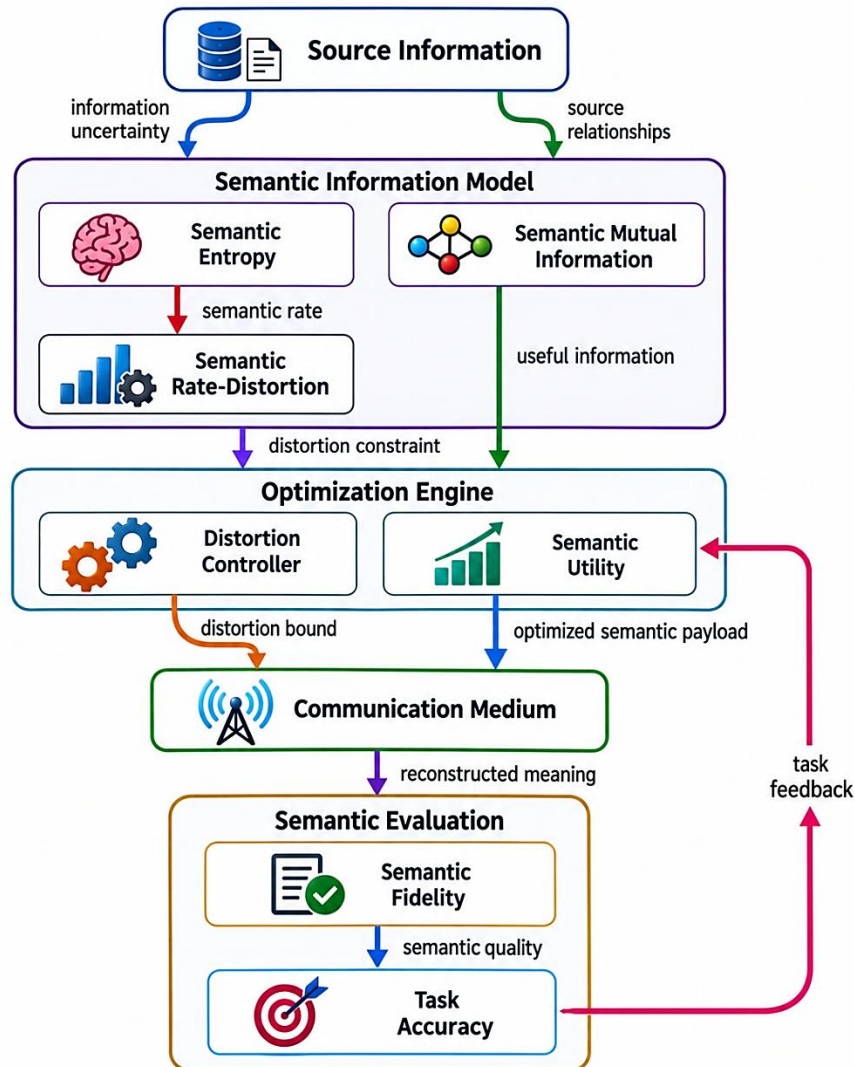


Figure 8: Optimization Framework for Meaning-Aware Semantic Communication

The optimization engine forms the central decision-making component of the framework. It contains a distortion controller and a semantic utility mechanism. The distortion controller establishes an acceptable distortion bound, while semantic utility represents the value or usefulness of the information to the intended receiver or application. These elements influence the generation of an optimized semantic payload for transmission. The optimization process therefore seeks to balance semantic quality against communication constraints rather than simply maximizing the quantity of transmitted data. The figure illustrates this interaction through the flow of semantic rate and distortion constraints from the semantic information model toward the optimization engine.

The optimized semantic payload is then delivered through the communication medium, after which the receiver obtains reconstructed meaning. The reconstructed information enters the semantic evaluation stage, where semantic fidelity is assessed before task accuracy is determined. This arrangement emphasizes that communication quality cannot be evaluated exclusively by examining the transmitted signal. The effectiveness of the communication process ultimately depends on whether the reconstructed information preserves sufficient meaning and enables the intended application to perform its task successfully.

Chapter 3

Semantic Representation, Knowledge Modeling, and Context-Aware Information Processing

3.1 Advanced Methods for Representing Semantic Information

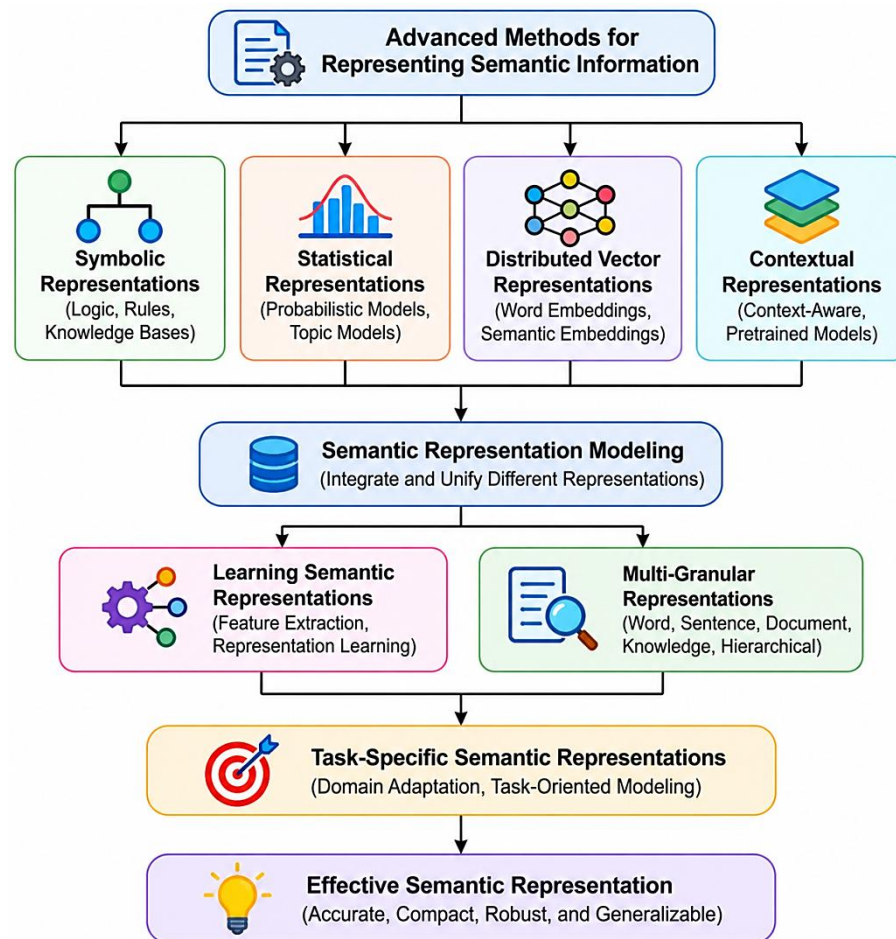


Figure 9: Advanced Methods for Semantic Information Representation

A comprehensive framework for representing semantic information using multiple complementary approaches. At the top, the framework identifies four major representation paradigms: symbolic representations, statistical representations, distributed vector representations, and contextual representations. Symbolic representations describe information using explicit logic, rules, and knowledge bases, making relationships and reasoning structures relatively interpretable. Statistical representations characterize information through probabilistic models and topic-based approaches, allowing uncertainty and patterns within information sources to be captured. Distributed vector representations encode semantic information within continuous vector spaces, including word and semantic embeddings, while contextual representations incorporate surrounding information and contextual relationships through context-aware or pretrained models.

The outputs of these representation approaches are integrated within the Semantic Representation Modeling stage. This stage emphasizes that semantic information may require combining different forms of representation rather than relying on a single representation mechanism. A semantic representation can incorporate structured knowledge, statistical relationships, learned embeddings, and contextual information according to the requirements of the communication system. Such integration is particularly relevant when information originates from heterogeneous sources or when the receiver must interpret meaning using both learned representations and available contextual knowledge.

The framework then identifies two important directions: learning semantic representations and multi-granular semantic representations. Learning-based methods can extract useful features and construct representations directly from data, while multi-granular representations allow semantic information to be modeled at different levels, such as words, sentences, documents, knowledge structures, or hierarchical concepts. These capabilities allow a semantic communication system to select an appropriate level of representation depending on the application. For instance, a communication task may require detailed information at the word level in one situation but only high-level conceptual information in another. Finally, the framework leads toward task-specific semantic representations and an effective semantic representation characterized by accuracy, compactness, robustness, and generalizability. Task-specific modeling allows semantic representations to be adapted to particular domains and application objectives rather than treating all information as equally important. The final representation should preserve the information required for the intended task while avoiding unnecessary redundancy.

3.1.1 Symbolic, Statistical, and Knowledge-Based Semantic Representations

Symbolic, statistical, and knowledge-based approaches provide important foundations for representing semantic information in intelligent communication systems. Unlike raw data representations, these approaches attempt to describe information through concepts, relationships, rules, probabilities, and domain knowledge. They are particularly useful when the communication system needs not only to transmit information but also to interpret, reason about, or make decisions using that information.

Symbolic representations describe semantic information using explicit structures such as symbols, logical expressions, rules, ontologies, and knowledge graphs. Concepts can be represented as entities and relationships, while logical rules can describe how particular conditions lead to conclusions. For example, an intelligent transportation system may represent a pedestrian as an entity and connect that entity with attributes such as location, direction, and movement. The explicit structure of symbolic representations provides interpretability and can support reasoning when the underlying knowledge is clearly defined.

Statistical representations model semantic information through probability distributions and statistical relationships. They are useful when information is uncertain, incomplete, or variable. Statistical models can identify patterns in language, visual information, sensor observations, and other data sources. Rather than representing a concept through an explicit rule, a statistical approach may estimate the probability that an observation corresponds to a particular semantic state. This makes statistical representations useful for handling ambiguity and uncertainty in real-world communication environments.

Knowledge-based representations incorporate previously available domain knowledge into semantic processing. Knowledge bases, ontologies, knowledge graphs, and structured databases can provide background information that helps transmitters and receivers interpret information consistently. A

receiver may therefore infer certain information without requiring every associated detail to be transmitted. This can reduce communication overhead while improving semantic interpretation.

These representation approaches are not necessarily independent. A semantic communication system can combine symbolic knowledge with statistical models and learned representations. Such integration can provide both structured interpretability and flexibility in handling uncertain or complex information. The appropriate representation depends on the source modality, communication objective, available knowledge, computational resources, and target application.

Consequently, symbolic, statistical, and knowledge-based representations establish an important foundation for semantic communication. They enable systems to move beyond representing information as sequences of bits toward representing concepts, relationships, uncertainty, and domain knowledge. Their integration with modern representation-learning methods provides a pathway toward more flexible semantic systems capable of interpreting and transmitting information according to its meaning and application relevance.

3.1.2 Distributed Vector Representations and Semantic Embeddings

Distributed vector representations provide a powerful approach for representing semantic information in machine-readable form. Instead of explicitly describing concepts through symbolic rules or predefined relationships, distributed representations encode information as numerical vectors in a multidimensional space. These vectors, commonly referred to as embeddings, can capture statistical and semantic relationships between words, objects, sentences, images, and other forms of information. Their importance has increased substantially with the development of deep learning and modern artificial intelligence systems.

The central idea is that semantically related information can be represented by vectors with related positions or structures in an embedding space. For example, words with similar meanings may be mapped to nearby regions of a vector space. More complex models can generate representations for complete sentences, documents, images, speech signals, or multimodal observations. These representations provide a compact computational interface through which semantic information can be processed, compared, transmitted, and reconstructed.

A semantic embedding can be represented conceptually as

$$z = f\theta(x)$$

where (x) represents the original information, $(f\theta)$ represents a learned representation model, and $(\{Z^l\})$ represents the resulting semantic embedding. The parameters (θ) are learned from data so that the resulting representation captures useful characteristics of the source.

Similarity between semantic representations can also be evaluated using measures such as cosine similarity:

$$\text{sim}(z_1 z_2) = \frac{z_1 z_2}{\|z_1\| \|z_2\|}$$

Such measures can help determine whether transmitted and reconstructed representations remain semantically consistent.

Distributed representations are particularly valuable for semantic communication because they can provide compact representations of complex information. Instead of transmitting an entire high-dimensional source, a communication system may transmit a learned representation containing information relevant to the receiver's objective. This can potentially reduce communication requirements, especially for high-volume sources such as images, video, and multimodal data.

However, embeddings also introduce challenges. Their meaning is often distributed across many dimensions and may not be directly interpretable. The quality of an embedding depends on the training data, model architecture, domain, and task. A representation that works effectively for one application may not preserve the information required by another. Embeddings can also be sensitive to contextual changes and distribution shifts.

Therefore, distributed vector representations should be viewed as flexible semantic representations rather than universal definitions of meaning. Their integration with contextual information, domain knowledge, and task requirements is essential for reliable semantic communication. When appropriately designed, semantic embeddings provide an effective bridge between raw multimodal information and intelligent meaning-aware transmission.

3.1.3 Contextual Representations for Meaning-Aware Information Processing

Contextual representations address an important limitation of static semantic representations: the meaning of information often depends on the circumstances in which it appears. A word, image, signal, or event can have different interpretations depending on surrounding information, previous observations, environmental conditions, user requirements, and the current task. Contextual semantic processing therefore attempts to represent information together with the contextual factors required for accurate interpretation.

In language processing, for example, the meaning of a word may change according to the sentence in which it occurs. Similarly, an image of an object can have different significance depending on its location, surrounding objects, time, or application. In an autonomous system, detecting a vehicle may have limited significance in isolation, but detecting a rapidly approaching vehicle in the system's planned trajectory can represent critical information. Context therefore changes not only interpretation but also semantic importance.

A contextual representation can be expressed conceptually as

$$z_c = f_\theta(x, c)$$

where (x) represents the source information, (c) represents relevant context, and (z_c) represents the resulting contextual semantic representation. Context may include environmental conditions, temporal

information, spatial information, previous communication, user requirements, domain knowledge, or task objectives.

Contextual representations are particularly important for semantic communication because they can reduce unnecessary transmission. If the receiver already possesses relevant contextual information, the transmitter may only need to communicate information that changes or updates the receiver's current understanding. This creates opportunities for context-aware compression and adaptive semantic transmission. For instance, a sensor system operating in a stable environment may transmit only significant changes rather than repeatedly communicating information that is already known.

Modern pretrained and foundation models can generate contextual representations by incorporating surrounding information into learned representations. These models can process relationships across text, images, speech, and other modalities, allowing semantic communication systems to exploit richer contextual dependencies. However, the use of contextual models also introduces computational requirements and potential challenges related to model consistency between transmitter and receiver.

An effective contextual representation should therefore satisfy several requirements. It should preserve task-relevant meaning, incorporate appropriate contextual information, remain robust when conditions change, and avoid introducing irrelevant dependencies. Context must also be interpreted consistently by the communicating entities; otherwise, differences in background knowledge can lead to semantic mismatches.

Contextual representation consequently provides an important foundation for meaning-aware information processing. It shifts semantic representation from a static description of information toward a dynamic representation that considers what the information means under particular circumstances. This capability is essential for intelligent communication systems operating in changing environments, where relevance and meaning may continuously evolve according to the current state, user, and application objective.

3.1.4 Hierarchical and Task-Specific Semantic Representations

Hierarchical and task-specific semantic representations provide mechanisms for adapting semantic information to different levels of abstraction and application requirements. Real-world information often contains multiple levels of meaning. A visual scene, for example, may contain pixels, edges, objects, object relationships, activities, and an overall scene interpretation. Similarly, textual information can be represented at the word, sentence, paragraph, document, and conceptual levels. Hierarchical representations organize these different levels so that a communication system can select the appropriate semantic granularity.

A hierarchical semantic representation can conceptually be expressed as a sequence of levels,

$$Z(1) \rightarrow Z(2) \rightarrow \dots \rightarrow Z(L)$$

where each level (Z^l) represents information at a particular semantic abstraction. Lower levels may contain detailed information, while higher levels capture increasingly abstract concepts and relationships.

The appropriate level depends on the communication objective. A detailed representation may be necessary for one task, whereas a high-level representation may be sufficient for another.

Task-specific semantic representations further adapt this hierarchy to the requirements of a particular application. Instead of attempting to preserve every aspect of source information, the communication system identifies the semantic information that contributes most strongly to the target task. For an object-detection application, object identity and location may be prioritized. For navigation, position, trajectory, and obstacles may be more important. For industrial monitoring, abnormal states and their temporal evolution may receive greater semantic priority.

Task-specific representations can therefore be viewed as a function of source information and task requirements:

$$ZT = f\theta(X, T, C)$$

where (X) represents source information, (T) represents the target task, and (C) represents contextual information. The resulting representation is designed to preserve information that contributes to task performance. An important advantage of hierarchical and task-specific representations is adaptive communication efficiency. A system can transmit detailed semantic information when the task requires precision and use a more compact representation when only high-level information is necessary. This can reduce bandwidth, computation, and energy requirements while maintaining the required level of semantic quality.

However, task-specific representations may reduce generality. A representation optimized for one task may not contain sufficient information for an unexpected task. This creates a trade-off between specialization and generalization. Future semantic communication systems may therefore combine task-specific representations with flexible shared representations that can support multiple applications. Overall, hierarchical and task-specific semantic representations enable communication systems to control what level of meaning is transmitted and why it is transmitted. They provide a direct connection between semantic representation and application objectives, supporting adaptive, efficient, and meaning-aware communication across diverse intelligent networking environments.

3.2 Knowledge Graphs, Ontologies, and Structured Semantic Knowledge

3.2.1 Knowledge Graph Construction and Semantic Relationship Modeling

Knowledge graphs provide a structured mechanism for representing information as interconnected entities, attributes, and relationships. In semantic communication systems, they can provide explicit representations of concepts that complement learned vector and contextual representations. Rather than treating information as an unstructured collection of data elements, a knowledge graph organizes information into a network in which entities are connected through meaningful relationships. This structure can help communication systems understand not only individual concepts but also how those concepts are related.

A typical knowledge graph consists of entities or nodes and relationships or edges. For example, an autonomous transportation system may represent a vehicle, pedestrian, road, and traffic signal as entities,

while relationships such as located-on, approaching, near, or controlled-by connect those entities. A simple semantic relationship can be represented as a triple:

(subject, relation, object).

For example, *(vehicle, approaching, intersection)* captures a meaningful relationship that can be processed independently of the raw sensor data from which it was derived.

Knowledge graph construction generally involves several stages, including entity identification, entity linking, relationship extraction, attribute assignment, and consistency checking. Information may originate from structured databases, documents, sensor systems, multimodal observations, or learned models. The resulting knowledge graph can then be updated as new information becomes available. In dynamic communication environments, this ability to evolve is particularly important because the semantic state of an environment can change continuously.

Semantic relationship modeling is especially valuable when the meaning of information depends on relationships among multiple entities. A single object label may provide limited information, whereas relationships can describe interactions, spatial dependencies, temporal events, and causal associations. These relationships can support reasoning and allow a receiver to infer information that does not need to be explicitly transmitted.

Knowledge graphs can also contribute to communication efficiency. Instead of repeatedly transmitting complete descriptions, a system may communicate only new or changed relationships when the receiver already possesses a corresponding knowledge structure. This creates opportunities for semantic updates and knowledge-aware transmission. However, knowledge graphs require appropriate domain definitions and mechanisms for handling incomplete, uncertain, or conflicting information. Their effectiveness also depends on maintaining sufficiently consistent knowledge between communicating entities. Overall, knowledge graph construction provides a structured foundation for representing meaning through entities and relationships. It complements learned semantic embeddings by providing explicit relational structure and can support semantic encoding, contextual reasoning, information inference, and task-oriented communication.

3.2.2 Entity, Attribute, and Relationship Representation for Communication

Entity, attribute, and relationship representation provides a fundamental structure for communicating semantic information in knowledge-based systems. An entity represents an identifiable concept or object, an attribute describes a characteristic of that entity, and a relationship specifies how one entity is connected to another. Together, these elements allow communication systems to represent information in a form that is more meaningful and interpretable than raw data alone.

For example, consider an intelligent transportation scenario containing a vehicle and a pedestrian. The vehicle and pedestrian can be represented as entities, while attributes may include position, speed, direction, and object type. Relationships can describe concepts such as near, approaching, following, or crossing. A semantic representation can therefore capture not only that a pedestrian exists but also how that pedestrian is related to the surrounding environment.

A simple entity representation can be described as

$$E = \{a_1, a_2, \dots, a_m\},$$

where (E) represents an entity and (a_i) denotes one of its attributes. Relationships can then connect entities through structured semantic associations. This provides a foundation for representing complex situations as collections of interconnected semantic elements.

From a communication perspective, this structure enables selective transmission. Instead of transmitting all attributes associated with every entity, the system can identify which elements are relevant to the receiver's task. For example, an autonomous vehicle may require the position and movement of a nearby pedestrian but may not require attributes describing distant objects. Similarly, an industrial monitoring system may prioritize the operating state and abnormal relationships of critical equipment. Entity and relationship representations can also support incremental communication. If the receiver already possesses a valid representation of an entity, the transmitter may communicate only attributes or relationships that have changed. This reduces redundant transmission and can be particularly useful in dynamic environments where only a small portion of the semantic state changes over time.

Another advantage is interoperability. Explicit entity and relationship representations can provide a common semantic structure between heterogeneous systems. Different sensors or applications can contribute information about the same entities while maintaining a shared conceptual representation. Nevertheless, semantic representation must address ambiguity, uncertainty, temporal changes, and differences in domain interpretation. An entity may have multiple possible identities, attributes may be uncertain, and relationships may change rapidly. Effective semantic communication must therefore provide mechanisms for updating and validating these representations. Entity, attribute, and relationship modeling consequently provides a practical bridge between structured knowledge and communication. By identifying what exists, what properties it possesses, and how it relates to other entities, semantic systems can transmit information according to its relevance and support more efficient interpretation and task execution.

3.2.3 Ontologies, Semantic Schemas, and Domain Knowledge Integration

Ontologies and semantic schemas provide formal structures for defining concepts and relationships within a particular domain. While knowledge graphs represent interconnected instances of information, ontologies establish the conceptual framework that specifies what those entities and relationships mean. In semantic communication, this distinction is important because communicating systems require sufficiently compatible interpretations of the information being exchanged.

An ontology can define concepts such as objects, events, properties, and relationships, together with constraints describing how these concepts are connected. For example, an intelligent transportation ontology may define concepts such as vehicle, pedestrian, road, intersection, traffic signal, and hazard. It can also specify relationships such as located-in, approaching, and has-speed. Such structures provide a common semantic vocabulary that allows different systems to interpret transmitted information consistently.

Semantic schemas serve a related purpose by defining how information is organized and represented. A schema can specify the expected fields, relationships, data types, and semantic categories associated with a particular information structure. In communication systems, this can simplify the exchange of structured semantic information because the receiver already understands the expected representation.

Domain knowledge integration involves combining information from multiple sources into a coherent semantic framework. Knowledge may originate from databases, expert systems, sensor observations, documents, pretrained models, or knowledge graphs. Integrating these sources can improve semantic interpretation because the receiver can combine newly transmitted information with existing domain knowledge. This integration is particularly useful when communication resources are limited. If both transmitter and receiver share a common ontology or knowledge structure, the transmitter does not necessarily need to transmit information that can be inferred from the shared model. Communication can instead focus on new observations, exceptions, updates, or task-critical changes.

However, semantic interoperability is a major challenge. Different systems may use different vocabularies, ontologies, levels of granularity, or interpretations. Domain knowledge can also become outdated, incomplete, or inconsistent. Semantic communication systems therefore require mechanisms for knowledge alignment, ontology mapping, version management, and uncertainty handling. The integration of ontologies with learned representations can provide additional flexibility. Explicit ontological structures offer interpretability and consistency, while learned embeddings can capture complex statistical relationships that may be difficult to specify manually. Combining these approaches can support richer semantic processing. Thus, ontologies and semantic schemas establish a shared conceptual foundation for meaning-aware communication. They enable communicating entities to interpret concepts consistently, integrate newly received information with existing knowledge, and reduce unnecessary transmission by exploiting shared semantic structures.

3.2.4 Knowledge Graph-Based Semantic Communication Architectures

Knowledge graph-based semantic communication architectures integrate structured semantic knowledge directly into the communication process. Instead of treating communication as an isolated transmission of source data, these architectures use knowledge graphs to identify relevant concepts, relationships, and contextual information before transmission and to support interpretation after reception. This creates a communication framework in which the transmitter and receiver can exploit shared or coordinated semantic knowledge.

A typical architecture begins with raw or multimodal source information. Semantic processing identifies entities, attributes, events, and relationships and maps them into a knowledge graph. The transmitter can then determine which portions of the graph are relevant to the current communication task. Rather than transmitting the complete source representation, it can communicate selected semantic information, updates, or graph changes. At the receiver, the transmitted information is integrated with an existing knowledge graph to reconstruct the relevant semantic state.

The process can be represented conceptually as

$$G_s = fKG(X, C, K),$$

where (X) denotes source information, (C) represents context, (K) represents existing knowledge, and (G_s) denotes the resulting semantic knowledge graph. The transmitted information may then correspond to a relevant subset or update of (G_s), depending on the task and communication constraints.

One major advantage of this architecture is knowledge-assisted inference. If the receiver already possesses relationships and domain knowledge, it may infer information that does not need to be explicitly transmitted. This can reduce communication overhead while maintaining task-relevant semantic information. For example, if a receiver already knows the structure of a road network, the transmitter may only need to communicate a newly detected hazard and its relationship to a known road segment. Knowledge graphs can also support dynamic semantic communication. As environmental conditions change, the system can communicate graph updates rather than repeatedly transmitting complete descriptions. This is particularly useful for smart cities, industrial IoT, autonomous systems, and distributed sensing environments.

However, knowledge graph-based communication introduces challenges involving graph synchronization, knowledge consistency, scalability, privacy, uncertainty, and computational complexity. Transmitter and receiver may possess different knowledge states, and incorrect or outdated knowledge can result in incorrect semantic inference. The architecture must therefore account for knowledge freshness and confidence. Overall, knowledge graph-based semantic communication provides a structured approach for connecting semantic representation, contextual knowledge, communication, inference, and application tasks. It represents an important step toward meaning-driven networking in which communication resources are used not simply to move data but to maintain and update the knowledge required for intelligent distributed decision-making.

3.3 Context-Aware and Intent-Aware Semantic Information Modeling

A context-aware and intent-aware framework for semantic information modeling in which semantic interpretation is influenced by multiple sources of contextual information. At the top of the framework, four contextual dimensions are identified: user context, task context, environmental context, and network context. User context includes information such as user profiles, behavioral and interaction history, preferences, and user intent. Task context describes task requirements, objectives, application domains, and service goals. Environmental context captures location, physical surroundings, surrounding entities, and social conditions, while network context represents channel conditions, network status, available computing resources, latency, and bandwidth. Together, these contextual sources provide information needed to determine what information is meaningful and relevant under a particular situation.

The contextual information flows into the Context-Aware and Intent-Aware Semantic Information Modeling stage. This central component combines the different context dimensions to construct a semantic representation that is appropriate for the current communication scenario. The framework subsequently identifies several complementary processing functions, including context modeling for adaptive semantic communication, user-intent and task-oriented information representation, environmental and network-context integration, and dynamic semantic adaptation to changing conditions. This arrangement emphasizes that semantic information should not be modeled independently of its surrounding circumstances. Instead, semantic representations can be adjusted according to what the user

wants, what the system needs to accomplish, what is happening in the environment, and what communication resources are currently available.

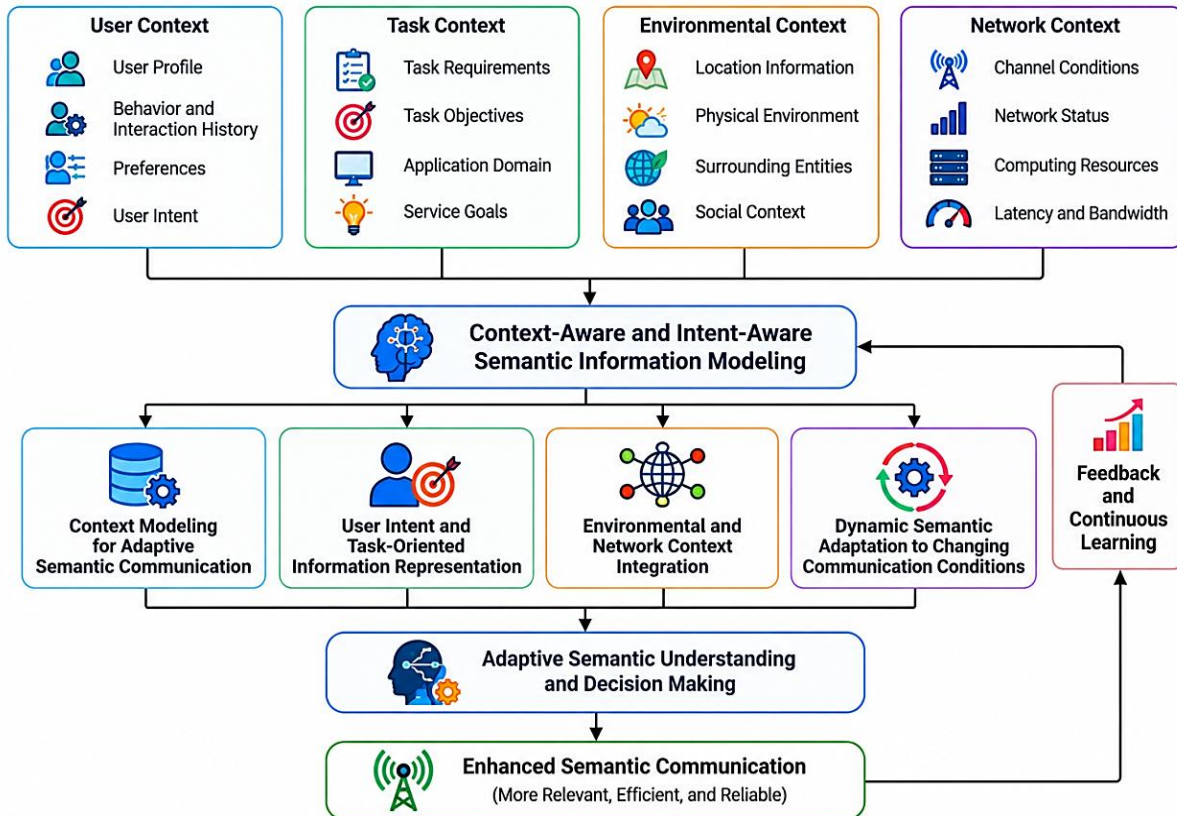


Figure 10: Context-Aware and Intent-Aware Semantic Information Modeling Framework

The outputs of these context-aware processes are combined in adaptive semantic understanding and decision-making. This stage represents the transition from semantic interpretation toward practical system behavior. The objective is to generate communication information that is sufficiently relevant, efficient, and reliable for the intended application. For example, the same source information may require different semantic representations when the user's objective, task requirements, environmental situation, or network conditions change. The framework therefore supports dynamic selection and adaptation of semantic information rather than relying on a fixed representation.

3.3.1 Context Modeling for Adaptive Semantic Communication

Context modeling is a fundamental component of adaptive semantic communication because the meaning and importance of information can change according to the circumstances in which it is generated, transmitted, and interpreted. In conventional communication systems, information is often processed without explicitly considering the broader context surrounding a message. Semantic communication instead attempts to understand information together with relevant environmental, user, task, temporal, spatial, and system conditions. This allows communication resources to be directed toward information that is meaningful under the current situation.

Context can include several dimensions. User context may describe preferences, interaction history, behavioral patterns, and current requirements. Task context identifies the objective that the receiving system is expected to accomplish. Environmental context describes physical surroundings, location, nearby entities, and changing conditions. Network context includes communication quality, available bandwidth, latency, congestion, and computing resources. These dimensions can interact with one another and collectively influence the semantic interpretation of source information.

A context model provides a structured mechanism for collecting, organizing, and interpreting these contextual elements. The model can combine information obtained from sensors, user interactions, databases, communication networks, knowledge bases, and previous system states. By maintaining contextual information, the semantic communication system can determine which source information is relevant and which information can be omitted because it is already known or can be inferred.

Context modeling is particularly valuable in dynamic environments. Consider an intelligent transportation system in which the same visual information is available under different operating conditions. During normal traffic, a system may prioritize general traffic-flow information. During an emergency, information about obstacles, pedestrians, emergency vehicles, or road closures may become significantly more important. Context modeling allows the semantic communication process to recognize these changes and modify the information representation accordingly.

Effective context modeling should also account for context freshness and consistency. Outdated contextual information can lead to incorrect semantic interpretation, while conflicting contextual sources can create ambiguity. The system must therefore identify which contextual information is reliable and relevant to the current communication situation. Context modeling transforms semantic communication from a static transmission process into an adaptive information-exchange process. It enables the system to determine what information matters, how it should be represented, and how it should be prioritized under specific circumstances. This capability provides the foundation for user-aware, task-aware, environment-aware, and network-aware semantic communication systems.

3.3.2 User Intent and Task-Oriented Information Representation

User intent represents the underlying objective or purpose that motivates an information request, interaction, or communication process. In semantic communication, identifying intent is important because the receiver may not need the complete source information; it may only require the information necessary to satisfy the user's objective or accomplish a particular task. Intent-aware communication therefore shifts attention from transmitting everything associated with a request toward transmitting the information that directly supports the intended outcome.

Intent can be explicit or implicit. An explicit intent may be directly stated by the user, such as requesting navigation to a particular destination. An implicit intent may need to be inferred from interaction history, behavioral information, environmental conditions, or task requirements. Intelligent semantic systems can combine these sources to construct a more complete understanding of what the user or application actually requires.

Task-oriented representation builds on this understanding by identifying the information necessary for successful task execution. Different tasks may require different semantic representations of the same source information. For example, an image of a road may contain vehicles, buildings, pedestrians, signs, and environmental details. A navigation task may prioritize road geometry, obstacles, and vehicle movement, while a traffic-monitoring task may emphasize vehicle density and traffic patterns. The underlying source is identical, but its useful semantic representation changes according to the task.

User intent can therefore influence semantic extraction, prioritization, compression, and transmission. Information that directly contributes to the intended task can receive higher semantic priority, while information with limited relevance can be represented more compactly or omitted. This can improve communication efficiency while preserving the information that matters most to the receiver.

Intent-aware communication also requires continuous interpretation. User objectives can change during an interaction, and application requirements may evolve as new information becomes available. A system should therefore avoid treating intent as a permanently fixed property. Instead, intent can be updated according to new instructions, observed behavior, task progress, and contextual changes.

An important challenge is ensuring that the inferred intent is correct. Incorrect intent interpretation can cause the communication system to prioritize inappropriate information, potentially reducing semantic quality or task performance. Intent models must therefore account for ambiguity, uncertainty, and changes in user requirements. Overall, user intent and task-oriented information representation provide a direct connection between semantic understanding and practical objectives. They allow communication systems to determine not merely what information is available, but what information is actually needed to accomplish the intended task. This principle is central to meaning-driven communication because it enables semantic resources to be aligned with the goals of users and intelligent applications.

3.3.3 Environmental and Network Context Integration

Environmental and network context integration combines information about the physical or operational environment with information about the current state of the communication infrastructure. This integration is important because semantic communication takes place within dynamic environments where both the meaning of information and the availability of communication resources can change continuously. A semantic representation that is appropriate under one set of environmental and network conditions may become inefficient or inadequate when those conditions change.

Environmental context can include geographical location, time, weather, physical surroundings, nearby objects, user activity, and other situational information. For example, an autonomous vehicle may need to distinguish between routine traffic conditions and a sudden hazard. Similarly, an industrial monitoring system may need to interpret sensor observations differently depending on machine operating states and environmental conditions. Environmental information can therefore influence which source information is relevant and how it should be represented.

Network context describes the communication conditions under which semantic information is transmitted. Important factors include available bandwidth, channel quality, latency, congestion, connectivity, computing resources, and network load. These factors influence how much semantic

information can be transmitted and how quickly it can reach the receiver. When network resources are limited, a semantic system may prioritize only the most important information or use more compact representations.

Integrating environmental and network context allows the semantic communication system to make coordinated decisions. For example, if an autonomous system detects an emergency situation while network congestion is high, the system may prioritize safety-critical semantic information and reduce the transmission of less important content. Conversely, when network resources are abundant, the system may transmit richer semantic representations to improve the receiver's understanding.

This integration also supports cross-layer semantic adaptation. Traditional communication architectures often separate application information from network management. A context-aware semantic architecture can allow application requirements to influence network decisions while allowing network conditions to influence semantic processing. Communication resources can therefore be allocated according to both the importance of information and the conditions under which it must be transmitted.

A major challenge is the heterogeneity and dynamism of contextual information. Environmental data may arrive from sensors, while network information may be obtained from communication infrastructure. These sources can have different time scales, levels of accuracy, and update frequencies. Effective integration therefore requires mechanisms for synchronization, consistency, relevance assessment, and context management.

Environmental and network context integration ultimately enables semantic communication to become situationally aware. By jointly considering what is happening in the environment and what communication resources are currently available, the system can determine what information should be transmitted, at what level of detail, and with what priority. This provides an important foundation for adaptive semantic networking in real-world intelligent systems.

3.3.4 Dynamic Semantic Adaptation to Changing Communication Conditions

Dynamic semantic adaptation enables a communication system to modify its semantic representation and transmission strategy as communication conditions, application requirements, or contextual circumstances change. This capability is essential because real-world communication environments are rarely static. Channel quality, available bandwidth, network congestion, computational resources, user requirements, and task priorities can all change over time. A semantic communication system that uses a fixed representation may therefore become inefficient or fail to preserve the information required by the receiver.

One important adaptation mechanism involves changing the level of semantic detail. When communication resources are abundant, the system may transmit a richer representation containing additional concepts, attributes, and relationships. When bandwidth is limited or latency requirements become more stringent, the system can prioritize only the most important semantic information. This allows communication quality to degrade gracefully rather than simply experiencing a proportional loss of transmitted data.

Semantic adaptation can also respond to changes in channel conditions. If interference increases or connectivity deteriorates, the system may increase the protection of critical semantic information, reduce representation complexity, or transmit only task-essential content. When communication conditions improve, additional semantic information can be transmitted to enhance the receiver's understanding. Task requirements can also trigger semantic adaptation. A system may initially operate in a routine monitoring mode and later transition to an emergency response mode. During the emergency, information that was previously considered low priority may become critical. Dynamic semantic adaptation enables the communication system to recognize this change and modify its semantic priorities accordingly.

Feedback plays an important role in this process. Information about semantic fidelity, task performance, network conditions, and receiver behavior can be used to determine whether the current communication strategy remains appropriate. If task performance decreases, the system may increase semantic detail or modify the information being transmitted. This creates a continuous adaptation loop between communication and application-level outcomes.

Dynamic adaptation must nevertheless be carefully controlled. Frequent changes in semantic representation can increase computational complexity and may create inconsistencies between transmitter and receiver. Both communicating entities may need compatible semantic models, contextual information, or knowledge structures to interpret changing representations correctly. The overall objective is therefore not to maximize semantic detail at all times but to maintain an appropriate balance between semantic quality, communication efficiency, latency, reliability, and computational cost. Dynamic semantic adaptation transforms semantic communication into a responsive system capable of adjusting its behavior according to changing conditions. It is particularly important for autonomous systems, IoT networks, mobile devices, edge intelligence, and other environments in which both communication conditions and application objectives evolve continuously.

3.4 Semantic Representation Compression and Intelligent Information Selection

A semantic communication framework in which raw multimodal information is transformed into compact semantic representations before transmission. The left side of the figure represents the available semantic information sources, including image, text, speech, and video. Instead of directly transmitting all raw data, the system extracts meaningful semantic features and constructs a semantic representation. This approach allows communication to focus on information that is relevant to the intended users or applications while reducing redundant information that may have limited value for the communication task.

The upper-right portion of the figure shows the semantic representation compression and intelligent information selection process. Semantic features are first extracted from the source information and represented using a structured semantic representation. The representation is then compressed to reduce its communication requirements. Following compression, important information is selected according to its relevance to the intended communication objective. This sequence demonstrates an important distinction between conventional compression and semantic compression: the objective is not simply to reduce the size of the data, but to retain the semantic information that is most valuable for subsequent interpretation or task execution.

The base station acts as an intermediate communication entity connecting the semantic information source with multiple users. The figure distinguishes between uplink transmission of compressed semantics and downlink transmission of selected semantic information. The semantic representations associated with different users, such as (G_1) , (G_i) , and (G_U) , can be reconstructed according to individual requirements. This illustrates the possibility of personalized semantic communication in which different receivers do not necessarily need identical information. Instead, the communication system can select and deliver information according to the specific needs of each user or application.

The lower portion of the figure emphasizes task-oriented reconstruction. At each receiving user, the transmitted semantic representation is reconstructed into the information required for the corresponding application, illustrated through recovered text representations. The key principle is that reconstruction does not necessarily require reproducing the complete original source. Instead, the system aims to recover the information necessary for the user's task. This architecture therefore connects semantic extraction, compression, intelligent information selection, multi-user transmission, and task-oriented reconstruction into a unified framework. It provides a strong conceptual basis for discussing semantic compression techniques, importance-aware information selection, adaptive semantic rates, and personalized semantic communication in the following subsections.

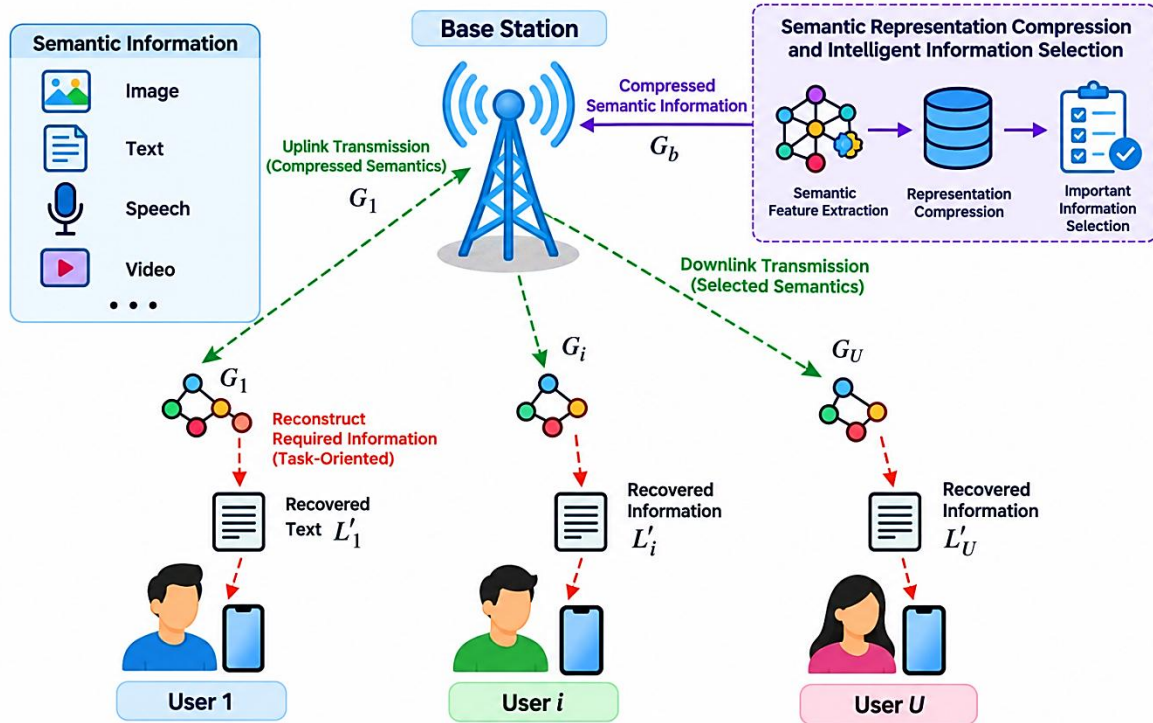


Figure 11: Semantic Representation Compression and Intelligent Information Selection Framework

3.4.1 Semantic Redundancy Identification and Information Reduction

Semantic redundancy identification is an important step in reducing the amount of information that must be transmitted in a semantic communication system. Conventional data often contains repeated,

predictable, or application-irrelevant information. At the semantic level, redundancy can occur when multiple pieces of information convey similar concepts, when contextual knowledge already available at the receiver makes certain information unnecessary, or when several features contribute little additional value to the intended task. Identifying such redundancy allows communication systems to remove unnecessary information while preserving the semantic content required by the receiver.

Semantic redundancy can occur at different levels of representation. In text, repeated concepts or contextual information may be unnecessary when the receiver already possesses the relevant background knowledge. In images and video, background regions or repeated visual features may contribute little to the target task. In sensor networks, measurements that remain unchanged over time may not need to be transmitted repeatedly. Knowledge graphs can similarly contain relationships that can be inferred from existing knowledge. Therefore, redundancy identification should consider both the source representation and the knowledge available at the receiving side.

The reduction process should distinguish between redundant information and important information. Removing information simply because it appears statistically repetitive can be harmful if that information is critical for a particular task. For example, two pieces of information may appear similar but provide different implications for a safety-critical decision. Semantic redundancy identification should therefore consider contextual relevance, task requirements, semantic relationships, and information importance. A semantic communication system can use learned representations, similarity analysis, knowledge bases, contextual models, and task feedback to identify information that can safely be reduced. Shared knowledge between transmitter and receiver is particularly valuable because information that can already be inferred at the receiver does not necessarily need to be transmitted again.

Semantic information reduction therefore represents more than conventional compression. Its objective is to reduce communication requirements while maintaining the meaning necessary for interpretation and task execution. The effectiveness of reduction should consequently be evaluated according to semantic fidelity and application performance rather than only the reduction in data size. Semantic redundancy identification provides the foundation for efficient semantic communication. By recognizing repeated, predictable, inferable, and low-value information, systems can concentrate communication resources on information that contributes most strongly to the receiver's understanding and task objectives.

3.4.2 Importance-Aware Semantic Feature Selection

Importance-aware semantic feature selection determines which semantic features should receive priority during communication. A source may contain a large number of features, but not all of them contribute equally to the receiver's interpretation or application task. Selecting features according to their importance allows semantic communication systems to allocate limited bandwidth, energy, computation, and transmission opportunities toward information with the greatest practical value.

Semantic features can represent different aspects of information depending on the modality and application. In an image, features may correspond to objects, locations, shapes, movements, or relationships. In text, they may represent keywords, concepts, entities, intent, or relationships. In sensor data, relevant features may include abnormal measurements, trends, or changes in system state. The importance of each feature depends strongly on the intended task and contextual conditions.

Feature importance can be determined using several complementary mechanisms. Learned neural models can estimate the contribution of features to a target prediction or decision. Attention mechanisms can identify information that receives greater relevance within a model. Knowledge-based systems can prioritize features associated with predefined domain rules. Context-aware mechanisms can modify feature priorities according to environmental conditions, user requirements, or network status. These approaches can also be combined to produce more robust selection decisions.

An important principle is that semantic importance is task-dependent. A feature that is critical for one application may be irrelevant to another. For example, precise object location may be essential for autonomous navigation but less important for a general image-classification task. Similarly, a user's intent can determine which portions of a message should be prioritized. Importance-aware selection should therefore be aligned with the intended application rather than relying on a universal ranking of features.

Feature selection can also be dynamic. As the task or environment changes, previously low-priority features may become important. An intelligent transportation system, for example, may prioritize ordinary traffic information during routine operation but give immediate priority to pedestrians or emergency vehicles when conditions change. The selected features can then be passed to subsequent semantic compression and transmission stages. By reducing the representation before communication, the system can potentially lower bandwidth and energy requirements while maintaining task-relevant semantic quality. Thus, importance-aware semantic feature selection provides a mechanism for deciding what should be communicated and what can be safely deprioritized. It transforms semantic communication from indiscriminate information transmission into an application-oriented process in which communication resources are concentrated on features that provide the greatest semantic and task value.

3.4.3 Semantic Representation Compression and Efficient Transmission

Semantic representation compression aims to reduce the communication requirements of semantic information while preserving the information necessary for effective interpretation and task execution. Conventional compression generally attempts to reduce the size of a data representation by exploiting statistical or structural redundancy. Semantic compression extends this principle by considering the meaning and usefulness of information. The objective is therefore not necessarily to reproduce the original data exactly, but to create a compact representation that retains the semantic content required by the receiver.

Semantic compression can operate after semantic feature extraction and importance-aware feature selection. Once the system has identified the concepts, attributes, relationships, or learned features that are relevant to the target task, these representations can be transformed into a more compact form. Compression may involve dimensionality reduction, quantization, representation pruning, latent-space encoding, or other learned mechanisms. The appropriate method depends on the semantic representation and application requirements.

A useful conceptual distinction is between source reconstruction and semantic reconstruction. In source reconstruction, the objective is to recover the original data with an acceptable level of distortion. In semantic reconstruction, the receiver aims to recover the meaning or task-relevant information

represented by the source. This distinction allows a semantic system to tolerate certain low-level changes when they do not significantly affect the intended interpretation.

Efficient transmission also requires coordination between the semantic representation and the communication channel. The compressed semantic representation should be adapted to available bandwidth, channel quality, latency requirements, and energy constraints. When network resources are limited, the system may use more aggressive compression or transmit only the highest-priority semantic features. When resources are available, a richer representation can be transmitted to improve semantic fidelity.

Receiver-side knowledge can further improve efficiency. If the transmitter and receiver share semantic models, contextual information, or knowledge structures, the transmitter may communicate only the information necessary to update the receiver's existing semantic state. This reduces repeated transmission and enables more efficient exchange of meaning. However, excessive compression can remove information that is important for the task. The compression process must therefore balance representation compactness against semantic fidelity and application performance. The appropriate operating point is application-dependent and may change with context. Semantic representation compression consequently provides a central mechanism for reducing communication overhead while maintaining useful meaning. When combined with intelligent feature selection and adaptive transmission, it enables communication systems to use available network resources more efficiently without treating every source detail as equally necessary.

3.4.4 Robustness and Generalization of Learned Semantic Representations

Robustness and generalization are essential requirements for learned semantic representations because semantic communication systems operate across diverse environments, users, data sources, and communication conditions. A representation that performs well on the data used during training may fail when the input distribution changes, when the communication channel introduces significant impairments, or when the system encounters previously unseen situations. Robust semantic representations should therefore preserve important meaning despite variations in data and operating conditions.

Robustness refers to the ability of a semantic representation to remain useful when information is affected by noise, distortion, missing features, channel errors, changes in modality quality, or other disturbances. For example, a semantic representation of an image should ideally preserve the identity of task-critical objects even when the transmitted representation is compressed or partially corrupted. Similarly, a speech-based semantic representation should remain useful when the audio contains background noise or transmission degradation.

Generalization concerns the ability of a learned representation to perform effectively beyond the specific examples and conditions encountered during training. Semantic communication systems may encounter new users, environments, objects, languages, domains, or task conditions. A representation that is overly specialized to a particular training distribution may lose semantic accuracy under these changes. Training strategies, domain adaptation, transfer learning, and diverse datasets can help improve generalization.

Robustness also depends on the relationship between the semantic model and the communication channel. A representation designed without considering channel impairments may be highly sensitive to small transmission errors. Channel-aware semantic encoding can instead prioritize features that remain meaningful after communication. Redundant or robust semantic representations can also provide additional protection for critical information without requiring complete duplication of the source.

Another important consideration is semantic consistency between transmitter and receiver. Both sides need sufficiently compatible models, representations, and contextual knowledge to interpret transmitted semantics correctly. Differences in model versions or domain knowledge can produce semantic mismatches even when the communication channel itself is reliable. Robustness should ultimately be evaluated at multiple levels. Representation-level similarity can indicate whether semantic features remain stable, while task-level performance can determine whether the preserved representation continues to support the intended application. Stress testing under changing channel and environmental conditions can further reveal weaknesses that are not visible under ideal conditions. Therefore, robust and generalizable semantic representations are essential for practical deployment. They allow semantic communication systems to operate reliably beyond controlled experimental settings and across changing conditions. Their development supports the broader objective of creating semantic communication systems that are compact, accurate, adaptive, transferable, and resilient, while preserving the meaning required by diverse real-world applications.

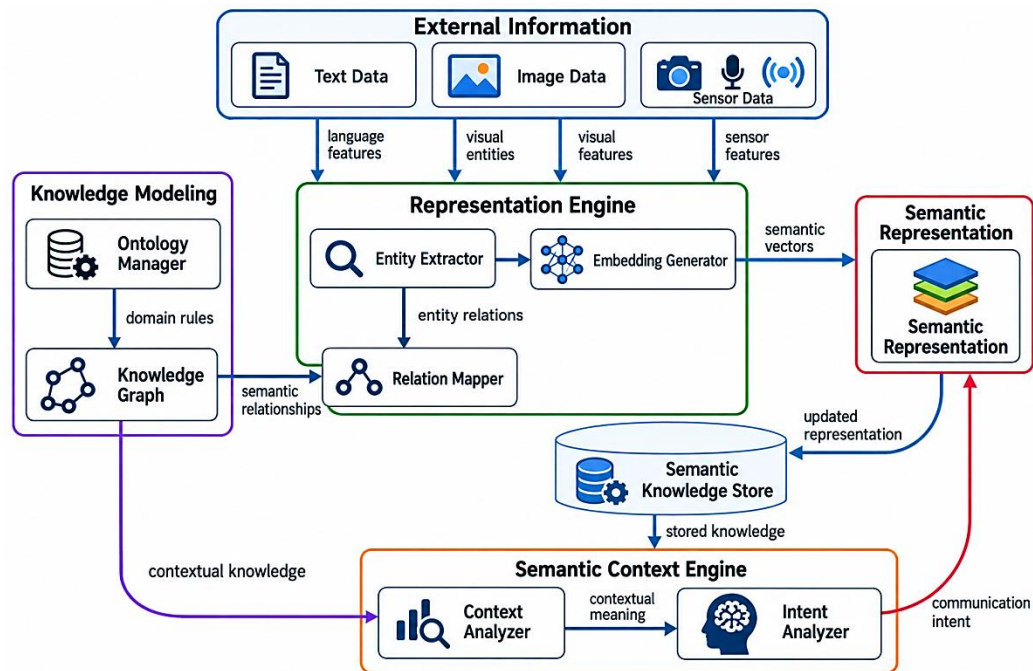


Figure 12: Integrated Semantic Representation and Context-Aware Processing Architecture

An integrated architecture for transforming heterogeneous external information into structured and context-aware semantic representations. The process begins with external information, which includes text data, image data, and sensor data. These different modalities provide complementary information about the environment and the underlying communication scenario. Text contributes language features, images provide visual entities and features, and sensors provide observations about physical conditions.

The multimodal inputs are directed toward the representation engine, where they can be processed into forms that are suitable for semantic interpretation and communication.

The representation engine contains several interconnected components, including an entity extractor, embedding generator, and relation mapper. The entity extractor identifies meaningful entities from the incoming information, while the embedding generator transforms relevant information into semantic vectors. The relation mapper represents relationships among extracted entities, allowing the system to capture structured semantic associations rather than isolated concepts. In parallel, the knowledge-modeling component uses an ontology manager and knowledge graph to provide domain rules and semantic relationships. These knowledge structures support the interpretation of newly extracted information and help establish consistency between incoming information and existing domain knowledge.

The resulting semantic information is organized within a semantic representation and semantic knowledge store. The semantic knowledge store provides persistent or reusable knowledge that can support subsequent semantic processing. The architecture also shows that updated semantic representations can be fed back into the semantic representation component, allowing the knowledge state to evolve as new information becomes available. This is particularly relevant to dynamic environments where entities, relationships, and observations may change over time. The architecture therefore goes beyond one-time semantic extraction and supports continuous semantic knowledge updating.

Chapter 4

Deep Learning, Transformer Models, and Generative Intelligence for Semantic Communications

4.1 Deep Learning Architectures for End-to-End Semantic Communication

4.1.1 Neural Semantic Encoder and Decoder Architectures

End-to-end neural architecture for semantic communication in which source information is transformed into a meaning-oriented representation before transmission. The source can contain multiple modalities, including text, images, speech, and structured information. These inputs are first represented as input features and provided to the semantic encoder. Unlike a conventional communication architecture that primarily focuses on transforming source information into bits, the semantic encoder is designed to extract features that capture information relevant to the intended meaning and application.

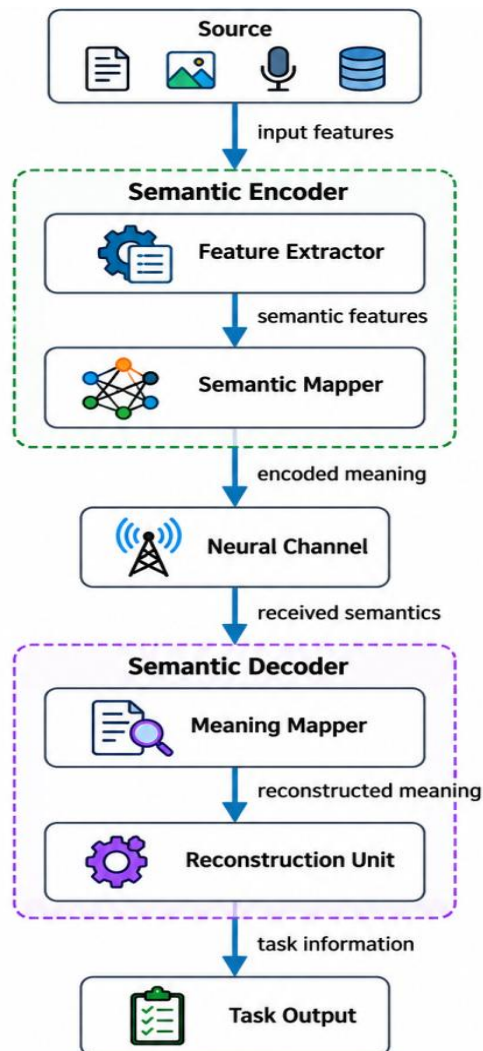


Figure 13: End-to-End Neural Semantic Encoder–Decoder Architecture

The semantic encoder consists of two conceptual stages: a feature extractor and a semantic mapper. The feature extractor identifies meaningful characteristics from the incoming source, while the semantic mapper transforms these features into a semantic representation suitable for communication. The resulting encoded meaning is then transmitted through the neural channel. The neural channel represents the communication process through which the semantic representation is affected by channel conditions before reaching the receiving side. This arrangement highlights the integration of representation learning and communication within an end-to-end neural framework.

At the receiver, the transmitted information is processed by the semantic decoder, which contains a meaning mapper and reconstruction unit. The meaning mapper interprets the received semantic representation and transforms it into a form that can support reconstruction. The reconstruction unit then generates the reconstructed meaning or task-relevant information. Importantly, the architecture does not necessarily require exact reproduction of the original source. Instead, the decoder is designed to recover the information needed by the target application, reflecting the task-oriented nature of semantic communication.

4.1.2 End-to-End Joint Source-Channel Semantic Learning

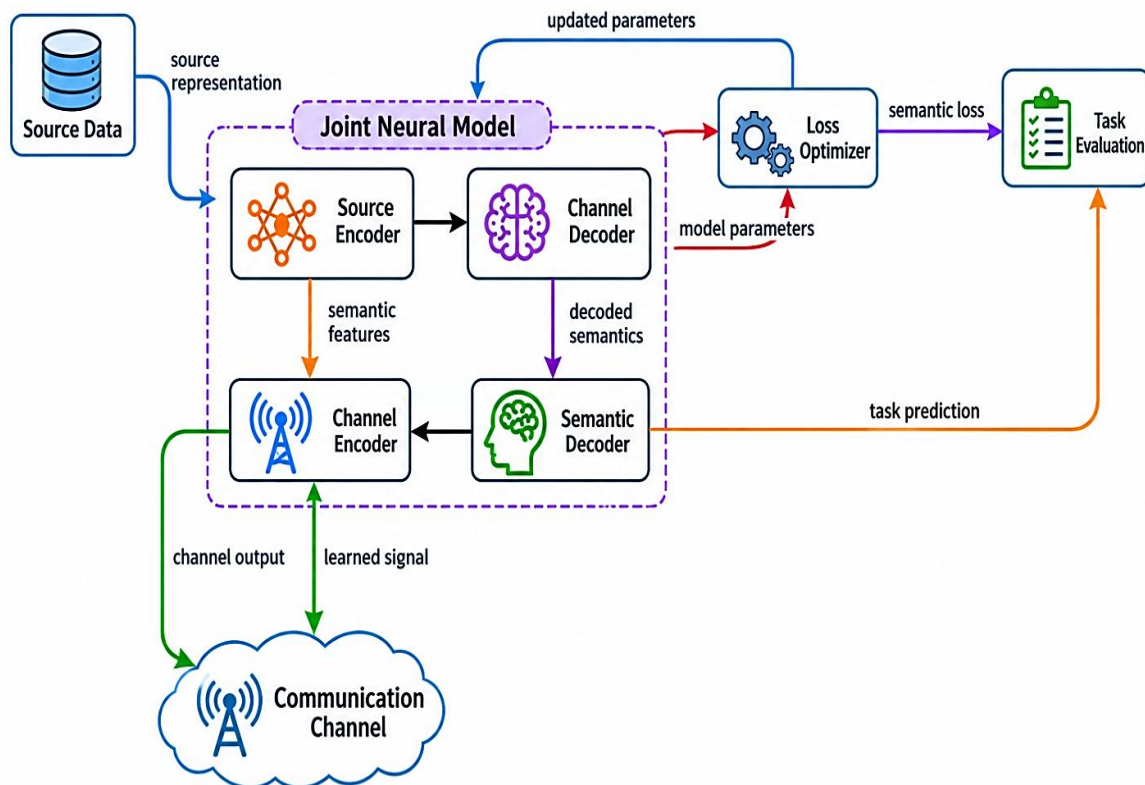


Figure 14: End-to-End Joint Source-Channel Semantic Learning Framework

End-to-end joint source-channel semantic learning framework in which source representation, semantic processing, channel transmission, and reconstruction are optimized as interconnected components of a single neural learning system. The process begins with source data, which is converted into a source representation and provided to the joint neural model. Within this model, the source encoder extracts

semantic features from the input, while the channel encoder transforms these features into a learned signal suitable for transmission through the communication channel. This integration allows the semantic representation and its communication format to be learned together rather than designing source and channel processing independently.

The communication channel introduces the conditions through which the learned signal must pass before reaching the receiver-side processing components. The channel decoder processes the channel output, after which the semantic decoder generates decoded semantic information. This reconstructed semantic information is then used to produce a task prediction. The architecture therefore connects physical communication with semantic interpretation and application-level processing. The important feature is that the neural model is trained with awareness of the entire transmission path, allowing the learned representation to become more suitable for the characteristics of the communication channel.

The right side of the figure shows the optimization and task-evaluation mechanism. The semantic decoder produces information that is evaluated according to the intended task, while a semantic loss is supplied to the loss optimizer. The optimizer uses the resulting loss information to adjust the model parameters. This creates an end-to-end learning process in which the source encoder, channel processing components, and semantic decoder can progressively adapt to improve the desired communication objective. Rather than optimizing only reconstruction accuracy or channel reliability independently, the framework can incorporate semantic and task-level requirements into the learning process.

4.1.3 Autoencoder-Based Semantic Communication Systems

Autoencoder-based semantic communication architecture in which source information is transformed into a compact latent representation before being transmitted through a communication channel. The input can contain different modalities, including images, text, speech, video, and other source information. The encoder first performs preprocessing and extracts learned features from the source. These features are subsequently processed by the semantic feature-learning stage, where the system attempts to capture the information that is most relevant to the underlying meaning rather than preserving every low-level characteristic of the original input.

The central component of the encoder is the semantic bottleneck, which performs meaning-aware compression. The bottleneck forces the system to represent the source using a more compact latent representation. This is particularly important for semantic communication because the objective is to reduce transmission requirements while retaining information that is useful for the receiver. The compressed semantics are then passed to the transmission stage, where the latent representation is communicated through a wireless or wired channel. The channel output represents the received latent information that becomes available to the decoder.

At the receiver, the semantic decoder processes the received latent representation through its reconstruction layers and output layer. The decoder attempts to recover the semantic information required for the intended application rather than necessarily reproducing every detail of the original source. The figure therefore emphasizes the distinction between conventional data reconstruction and semantic reconstruction. The quality of the reconstructed information can be evaluated using a semantic loss, which

provides feedback concerning how effectively the transmitted representation preserves the desired information.

The feedback path between the semantic loss and the semantic bottleneck illustrates how the compression level can be adjusted during learning. If excessive compression causes important semantic information to be lost, the system can modify the representation strategy to improve reconstruction quality. Conversely, when greater compression can be achieved without significantly affecting the required meaning, the representation can become more communication-efficient.

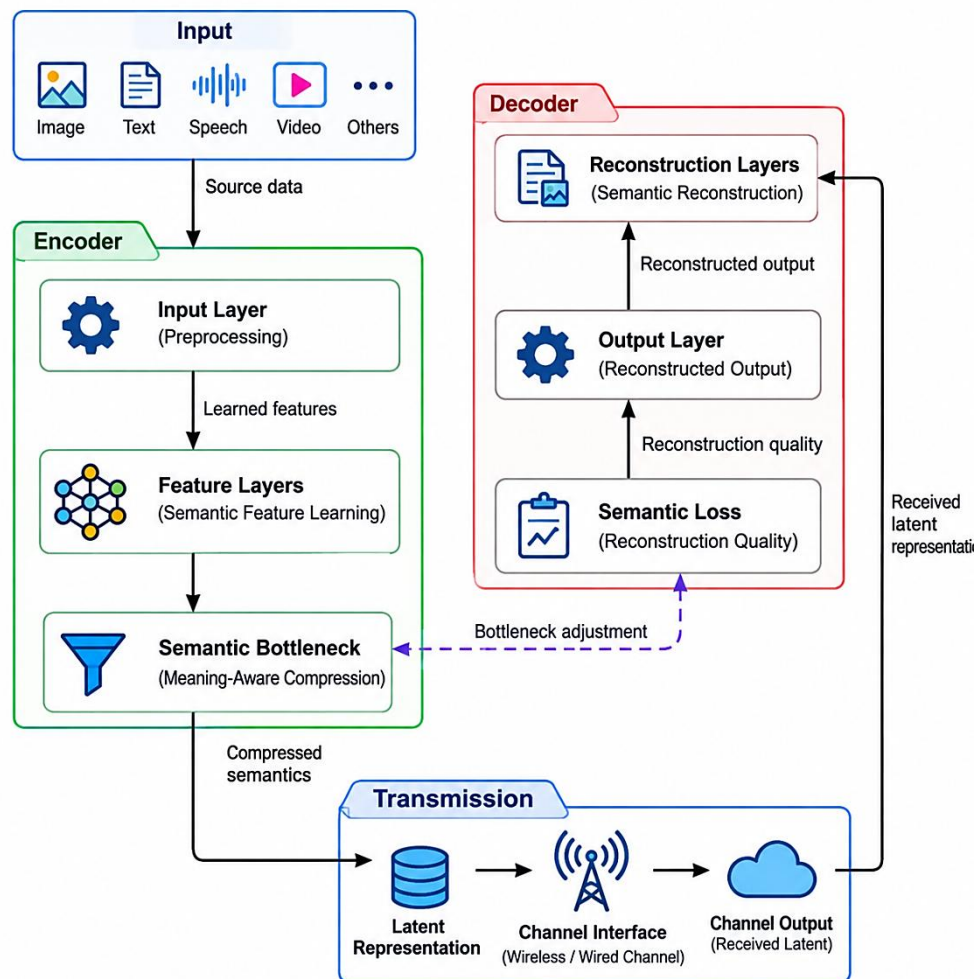


Figure 15: Autoencoder-Based Semantic Communication Architecture

4.1.4 Adaptive Neural Models for Dynamic Communication Environments

An adaptive neural semantic communication framework designed to operate under dynamically changing communication conditions. The central dynamic environment contains three major factors: traffic load, task demand, and channel state. These factors can change over time and directly influence how semantic information should be processed and transmitted. For example, increasing traffic load may require more efficient semantic representations, while changing task requirements may alter which information should receive priority. Similarly, variations in channel state can affect the reliability and efficiency of semantic

transmission. The framework therefore treats communication conditions as dynamic inputs to the semantic processing system.

The neural communication model on the right contains a semantic encoder, an adaptive model, and a semantic decoder. The semantic encoder initially extracts semantic features from the source information. These features are then processed by the adaptive model, which modifies the representation according to the current environmental and communication conditions. The resulting adapted representation is subsequently provided to the semantic decoder for reconstruction and interpretation. This arrangement demonstrates that the semantic representation does not necessarily remain fixed; instead, it can change according to the current requirements of the communication environment and application.

The left side of the figure introduces a decision controller and performance monitor. The performance monitor observes task performance and provides performance readings to the decision controller. At the same time, the decision controller receives information associated with the dynamic environment and determines an appropriate adaptation policy. This policy influences the neural communication model, allowing the system to respond to changes in traffic variation, task requirements, and channel conditions. The architecture therefore creates a connection between network conditions, semantic processing, and application-level performance rather than optimizing each component independently.

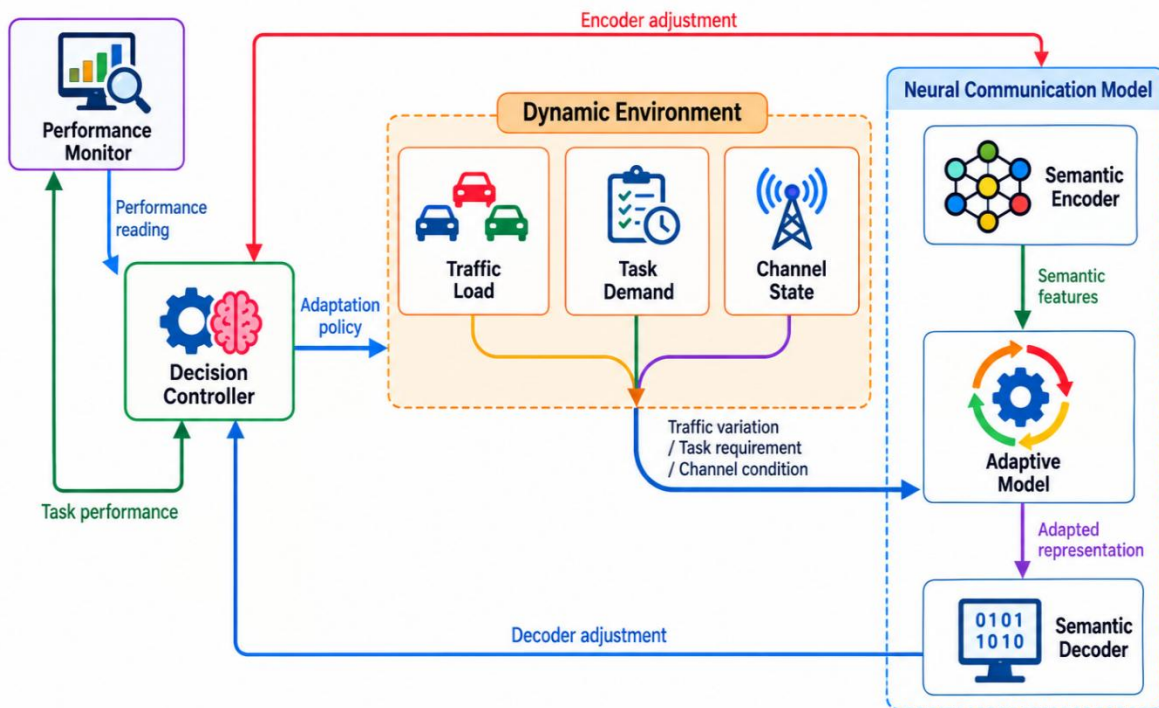


Figure 16: Adaptive Neural Semantic Communication Model for Dynamic Environments

The feedback mechanisms are a particularly important feature of the architecture. Encoder adjustment and decoder adjustment allow the neural communication model to adapt its processing according to observed system behavior, while task-performance feedback enables the decision controller to evaluate whether the

current configuration remains effective. This creates a continuous adaptation cycle in which environmental conditions influence semantic processing, semantic processing affects task performance, and task performance feeds back into future adaptation decisions.

4.2 Transformer Architectures and Attention Mechanisms for Meaning Extraction

4.2.1 Transformer-Based Semantic Representation and Feature Extraction

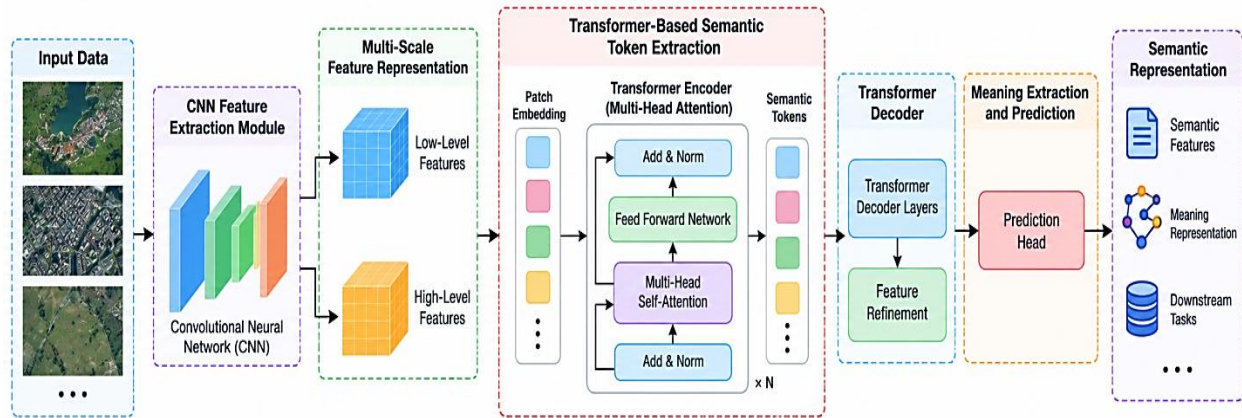


Figure 17: Transformer-Based Semantic Representation and Feature Extraction Framework

Transformer-based framework for extracting and representing semantic information from complex input data. The process begins with input data, represented in the figure through multiple visual samples. These inputs are first processed by a CNN feature extraction module, which extracts spatial and structural characteristics from the original data. The CNN stage provides an initial representation that can capture local patterns and features before the information is passed to the Transformer-based processing stages. This combination is useful when semantic information depends on both local characteristics and broader relationships within the input.

Following CNN processing, the extracted information is organized into a multi-scale feature representation containing low-level and high-level features. Low-level features can capture relatively detailed characteristics, while high-level features provide more abstract information about the input. The resulting representations are then transformed through patch embedding, which converts the feature information into a sequence of representations suitable for Transformer processing. This stage establishes the input tokens required by the Transformer encoder and provides a structured representation of the extracted features.

The Transformer-based semantic token extraction stage contains a Transformer encoder based on multi-head self-attention. The encoder uses self-attention to examine relationships among different embedded features and determine how information from different parts of the input contributes to the overall representation. The repeated Transformer encoder layers progressively transform the embedded features into semantic tokens. These tokens provide a more compact representation of the information that is

relevant for subsequent semantic interpretation and processing. The resulting semantic tokens are passed to the Transformer decoder, where decoder layers and feature refinement further process the representation. The refined features are then supplied to the meaning extraction and prediction stage, which generates task-related predictions. Finally, the framework produces a semantic representation containing semantic features, meaning representations, and information suitable for downstream tasks.

4.2.2 Self-Attention and Cross-Attention for Contextual Meaning Modeling

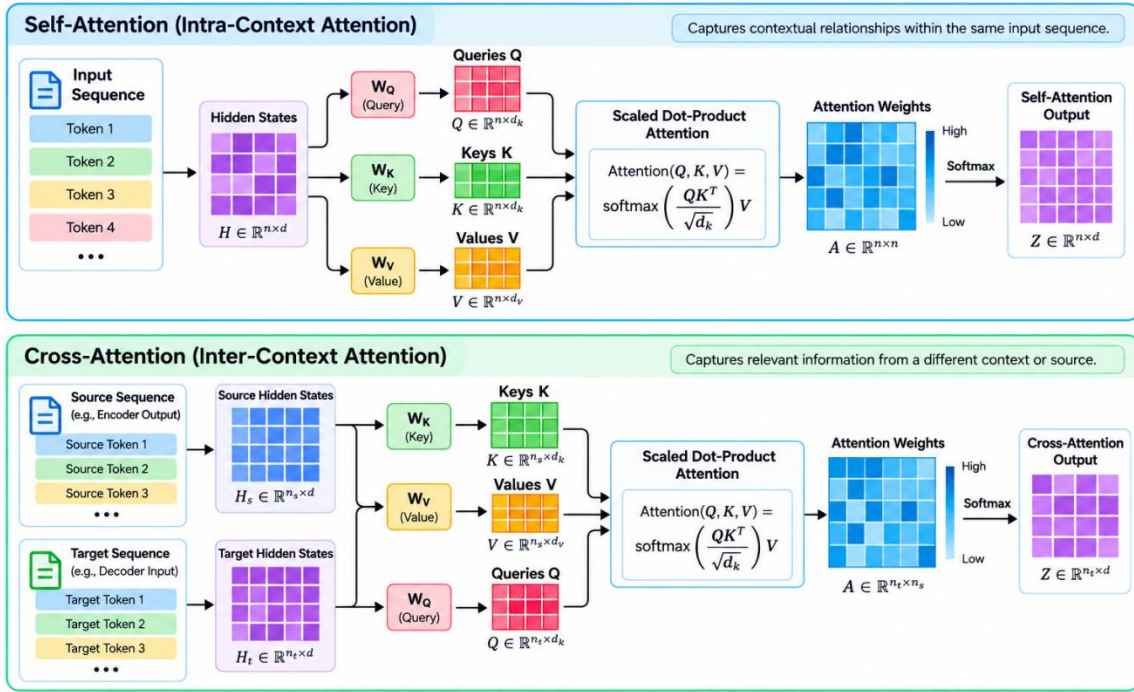


Figure 18: Self-Attention and Cross-Attention for Contextual Meaning Modeling

Two fundamental attention mechanisms used in Transformer architectures: self-attention and cross-attention. The upper part of the figure represents self-attention, also referred to as intra-context attention, where the model examines relationships among elements within the same input sequence. The input sequence is first represented through hidden states, from which three representations are generated: queries, keys, and values. These representations allow the model to determine which elements of the sequence are most relevant to one another when constructing contextualized semantic representations.

In the self-attention mechanism, the query, key, and value representations are generated using corresponding learned transformations. The figure shows the scaled dot-product attention operation, where the compatibility between queries and keys determines the attention weights. These weights are normalized through a softmax operation and used to combine the value representations. The resulting self-attention output therefore contains contextual information influenced by relationships among the tokens within the same sequence. This is important for semantic communication because the meaning of an individual element can depend strongly on other elements occurring within the same source representation.

The lower portion of cross-attention, or inter-context attention, in which information from one representation is selectively accessed by another. The figure distinguishes a source sequence, such as encoder output, from a target sequence, such as decoder input. The target hidden states provide the queries, while the source hidden states provide the keys and values. Consequently, the target representation can identify and selectively use the most relevant information from the source representation. This mechanism is particularly useful when semantic information must be transferred between different representations or processing stages.

Cross-attention is highly relevant to meaning-aware and multimodal processing because it allows one information stream to attend to another. For example, a textual representation can attend to visual features, or a decoder representation can attend to encoded source information. The attention weights provide a mechanism for determining which source elements contribute most strongly to the current semantic interpretation. Overall, the figure establishes the distinction between self-attention for contextual relationships within the same representation and cross-attention for relationships between different representations. This distinction provides an important foundation for understanding Transformer-based semantic fusion, multimodal communication, contextual meaning extraction, and encoder–decoder architectures discussed in subsequent sections.

4.2.3 Multi-Head Attention for Long-Range Semantic Dependencies

The internal structure of multi-head attention and the scaled dot-product attention mechanism used within each individual attention head. The right side of the figure presents the operation of one attention head, beginning with the input query, key, and value representations. These representations are compared through a matrix multiplication between the queries and the transposed keys. The resulting similarity scores are scaled according to the dimensionality of the key representation, and an optional mask can be applied before the scores are normalized using the softmax operation. The resulting attention weights determine how strongly different input elements contribute to the final attention output.

The scaled dot-product mechanism shown in the figure can be expressed as

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T d_k)V$$

where (Q), (K), and (V) denote the query, key, and value matrices, respectively, and (d_k) represents the dimensionality of the key vectors. The scaling factor helps control the magnitude of the similarity scores before the softmax operation. The attention output therefore represents a weighted combination of the value vectors, with the weights determined by the relationships between queries and keys. The multi-head attention. Instead of performing attention only once, the input representation is transformed into multiple sets of queries, keys, and values through separate learned linear projections. Each attention head independently performs scaled dot-product attention. Because the heads operate in parallel, different heads can learn to emphasize different relationships within the input. In semantic communication, this is useful because meaning may depend simultaneously on multiple types of relationships, such as local associations, long-range dependencies, contextual relationships, entity interactions, or task-relevant features.

After the individual attention heads generate their outputs, the results are concatenated and passed through a final linear projection. This produces the combined output representation shown at the top of the figure. The multi-head structure therefore allows the Transformer to construct a richer semantic representation than relying on a single attention pattern. By learning multiple relationship patterns simultaneously, the mechanism can connect information that is separated by many tokens or feature positions. This capability is particularly important for semantic communication, where the meaning of information may depend on relationships between distant elements rather than only neighboring features. The figure consequently provides a clear foundation for discussing long-range semantic dependencies, contextual representation learning, and attention-based semantic feature extraction.

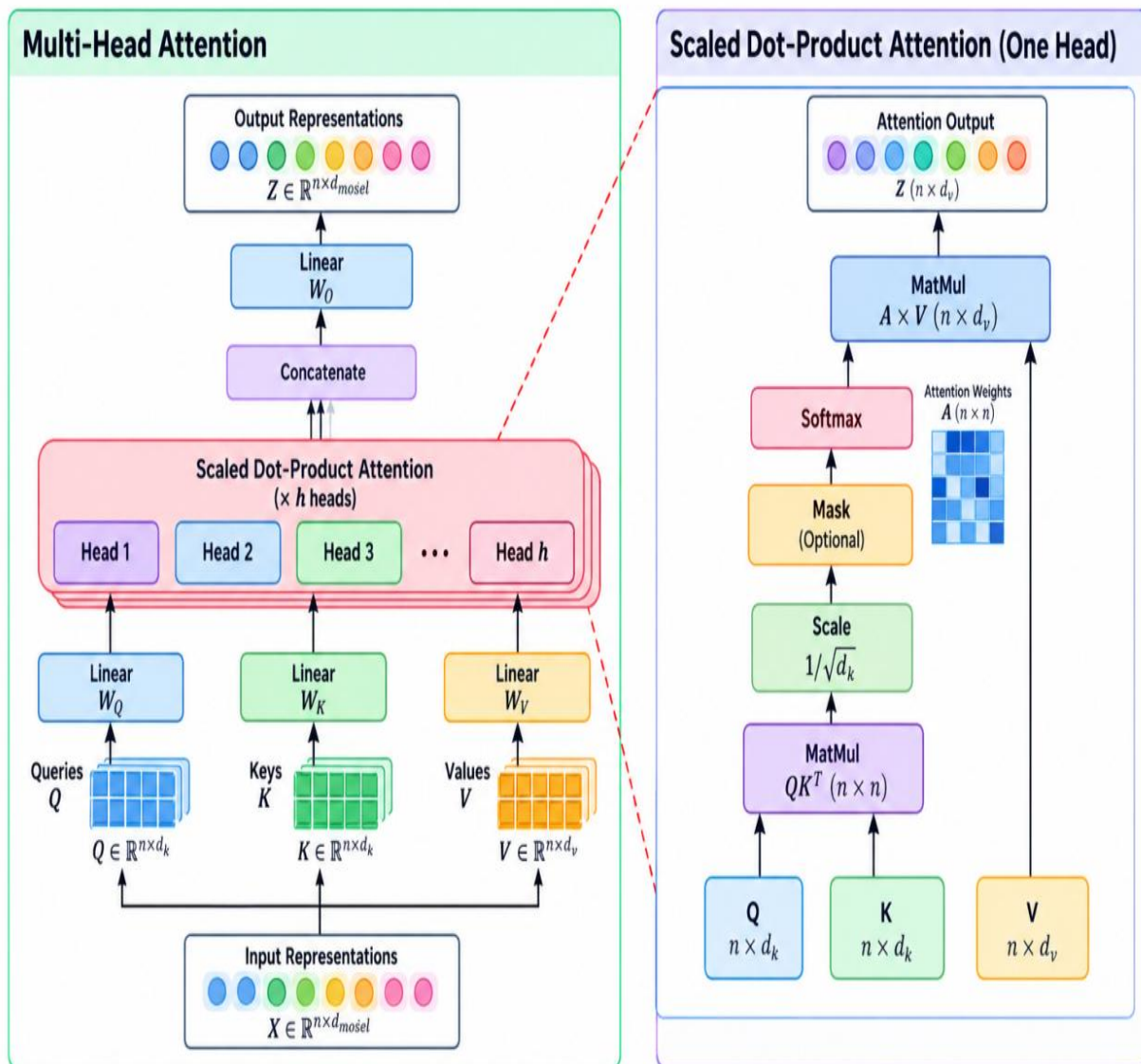


Figure 19: Architecture of Multi-Head Attention, showing the scaled dot-product attention mechanism for each attention head

4.2.4 Transformer-Based Semantic Encoding and Decoding Architectures

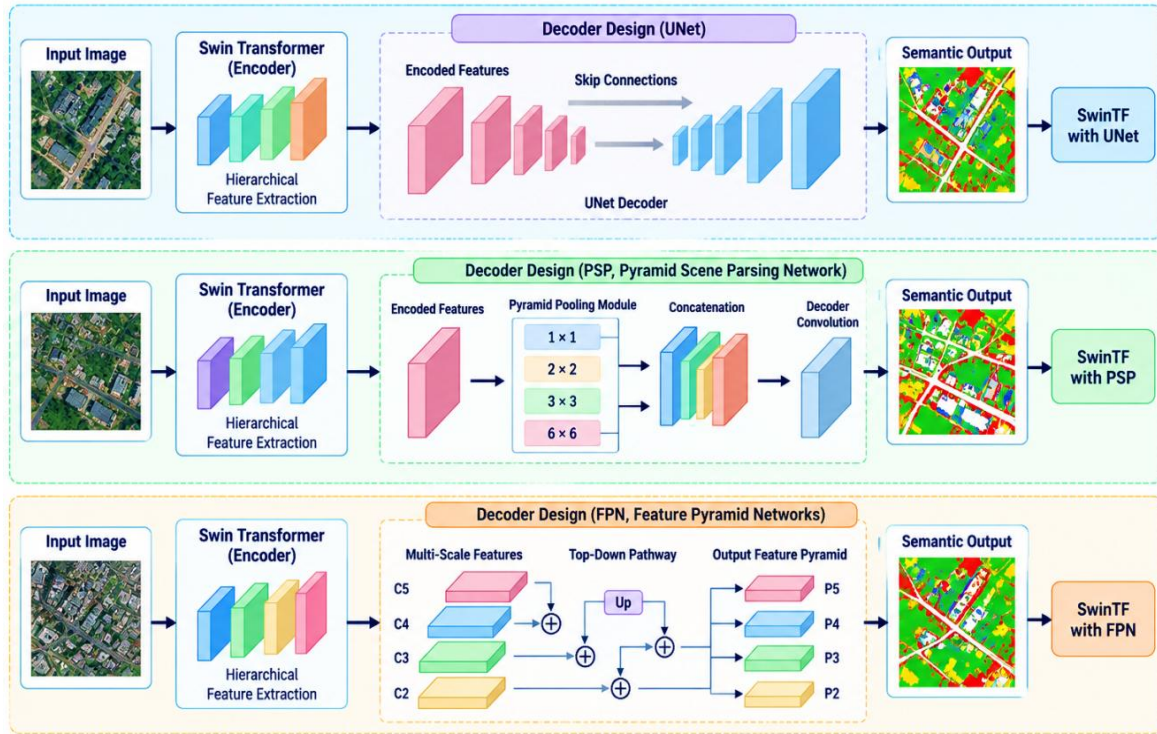


Figure 20: Transformer-Based Semantic Encoding and Decoding Architectures

Three Transformer-based semantic encoding and decoding configurations in which a Swin Transformer serves as the encoder and different decoder architectures are used to transform encoded features into semantic outputs. The Swin Transformer performs hierarchical feature extraction from the input image, generating representations at different levels of abstraction. These encoded features are subsequently processed by the decoder, whose structure determines how information is reconstructed and combined to produce the final semantic representation. The figure therefore emphasizes that Transformer-based semantic communication architectures can combine a common hierarchical Transformer encoder with different decoder designs according to the requirements of the target task.

The first configuration uses a U-Net decoder. Encoded features generated by the Swin Transformer are progressively processed through the decoder while skip connections transfer information from earlier feature stages to corresponding decoder stages. This allows the decoder to combine higher-level semantic information with lower-level spatial details. The resulting representation is transformed into the semantic output shown on the right. This configuration is useful when semantic interpretation requires both abstract contextual information and relatively detailed spatial information.

The second configuration uses a Pyramid Scene Parsing Network (PSP) decoder. Here, encoded features are processed through a pyramid pooling module operating at multiple spatial scales, illustrated using different pooling sizes. The resulting features are combined through concatenation and subsequently processed by decoder convolution. This multi-scale processing allows the architecture to incorporate

information from different receptive-field sizes, supporting semantic interpretation of structures that vary in spatial scale.

4.3 Generative AI and Diffusion Models for Semantic Information Reconstruction

4.3.1 Generative Models for Missing and Incomplete Semantic Information

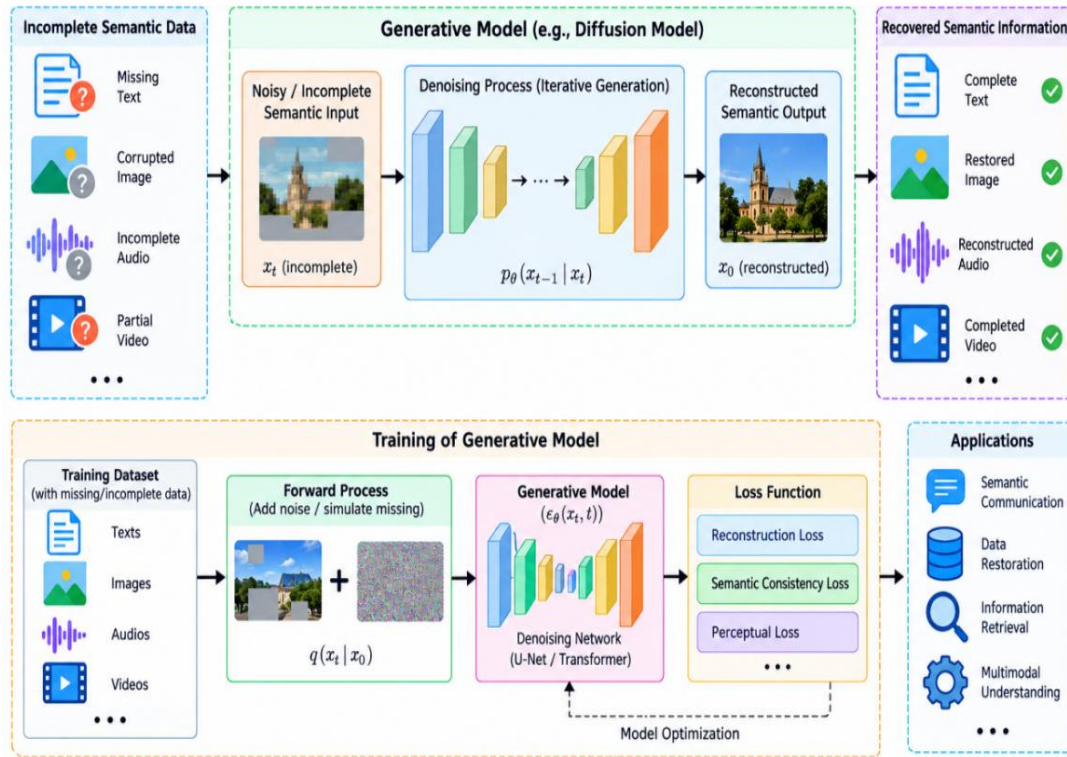


Figure 21: Generative Model-Based Reconstruction of Incomplete Semantic Information

How generative models can reconstruct missing, corrupted, or incomplete semantic information from multimodal sources. The upper portion begins with incomplete semantic data, including missing text, corrupted images, incomplete audio, and partial video. Instead of simply discarding these incomplete inputs, the system processes them through a generative model. The noisy or incomplete semantic input is represented as an initial state, which is progressively transformed through an iterative generation or denoising process. The objective is to recover a complete semantic representation that is consistent with the available information.

The central generative-model stage demonstrates an iterative reconstruction process. In the case of a diffusion-based approach, the model progressively transforms a noisy or incomplete representation toward a reconstructed semantic output. The figure represents this process through a sequence of intermediate transformations, eventually producing a reconstructed output. The resulting information can then be used to recover complete text, restore an image, reconstruct audio, or complete video information. The important principle is that reconstruction is guided by learned patterns and semantic relationships rather than relying only on the directly observed portions of the source.

The lower portion of the figure shows the training process of the generative model. A training dataset containing incomplete or missing multimodal information is first prepared. A forward process introduces noise or simulates missing information, after which a generative model, such as a denoising network based on a U-Net or Transformer architecture, learns to recover the desired information. The model is optimized using several types of loss, including reconstruction loss, semantic consistency loss, and perceptual loss. These objectives encourage the generated output to remain consistent with the available source information while preserving the semantic characteristics and perceptual quality required by the application.

The right side of the figure highlights potential applications of generative semantic reconstruction, including semantic communication, data restoration, information retrieval, and multimodal understanding. This demonstrates that generative reconstruction can operate as more than a conventional data-recovery mechanism. In semantic communication, it can help the receiver infer information that was lost or incompletely transmitted, provided that sufficient contextual and semantic evidence remains available. However, generated information should not automatically be assumed to be identical to the original information; reconstruction quality depends on the learned model, available context, input completeness, and application requirements.

4.3.2 Generative Reconstruction of Semantically Degraded Information

Generative reconstruction of semantically degraded information focuses on recovering meaningful information when the received representation has been affected by communication impairments, incomplete transmission, noise, compression, or other forms of degradation. In conventional communication systems, degradation is commonly addressed by attempting to recover the original signal or data as accurately as possible. Semantic communication introduces a different objective: the reconstructed output should preserve the meaning and task-relevant information contained in the original source. Consequently, a semantically degraded representation may be reconstructed even when exact recovery of the original data is not possible or necessary.

Generative models are particularly useful in this setting because they can learn patterns and dependencies from large datasets and use those learned relationships to infer plausible missing or degraded information. A generative reconstruction system can receive a partially corrupted semantic representation and use the available information, contextual relationships, and learned semantic structure to generate a more complete representation. For example, a degraded image may still contain sufficient information to identify an object even when some visual details have been lost. Similarly, incomplete text may provide enough contextual information for a model to reconstruct the concepts required for a downstream task.

Diffusion models provide one possible mechanism for this reconstruction process. The model can iteratively transform a degraded representation toward a cleaner and more semantically coherent representation. Rather than treating every low-level error as equally important, the reconstruction process can emphasize semantic characteristics that contribute to the intended interpretation. This makes generative reconstruction particularly relevant when communication systems operate under bandwidth limitations or unreliable channel conditions.

The quality of generative reconstruction should be evaluated at multiple levels. Reconstruction quality determines whether the recovered information remains consistent with the available source evidence. Semantic consistency examines whether the generated output preserves the intended concepts and relationships. Task-level performance evaluates whether the reconstructed information remains useful for the application. These criteria are important because a visually or linguistically plausible reconstruction may still contain incorrect semantic information. A generative model should therefore avoid introducing unsupported content that could alter the intended meaning.

Generative reconstruction can also operate together with contextual knowledge and semantic representations. Information already available at the receiver can provide additional constraints for reconstructing degraded content, while task requirements can determine which aspects of the missing information should receive greater priority. This creates a connection between generative AI and knowledge-aware semantic communication. Generative reconstruction provides a mechanism for making semantic communication more resilient to information degradation. Its primary objective is not necessarily perfect reproduction of the transmitted source but recovery of sufficiently accurate and useful meaning. This perspective makes generative models promising components for semantic communication systems operating under incomplete, noisy, or dynamically changing transmission conditions.

4.3.3 Diffusion-Based Semantic Generation and Reconstruction

A diffusion-based framework for generating and reconstructing semantic information from sparse or incomplete point-cloud data. The process begins with an incomplete point-cloud input, which is transformed through voxelization into a three-dimensional voxel representation. The voxel grid contains both observed and missing regions, providing a structured representation of the incomplete scene. This transformation allows subsequent feature-processing modules to operate on a spatially organized representation while preserving information about regions where observations are unavailable.

The voxel representation is then processed by the feature compensation module, which attempts to recover or enhance features associated with missing voxel regions. The detailed architecture shown in the lower portion of the figure uses local voxel blocks, point-wise inputs, sparse convolution, voxel-wise features, and feature-processing layers to generate intermediate and global feature representations. These features are subsequently combined through processing and pooling operations to produce point-voxel features. The objective of this stage is to compensate for missing information while recovering structural characteristics that are useful for subsequent semantic generation.

The compensated features are passed to the diffusion process, where semantic generation occurs through a sequence of transformations from a noisy representation toward a reconstructed semantic representation. The figure labels the progression using (x_T) for the noisy state, intermediate (x_t) states during denoising, and (x_0) for the reconstructed representation. This iterative process allows the diffusion model to progressively refine the semantic representation and generate a more complete description of the scene. The resulting representation is then used to produce the reconstructed semantic scene shown at the output.

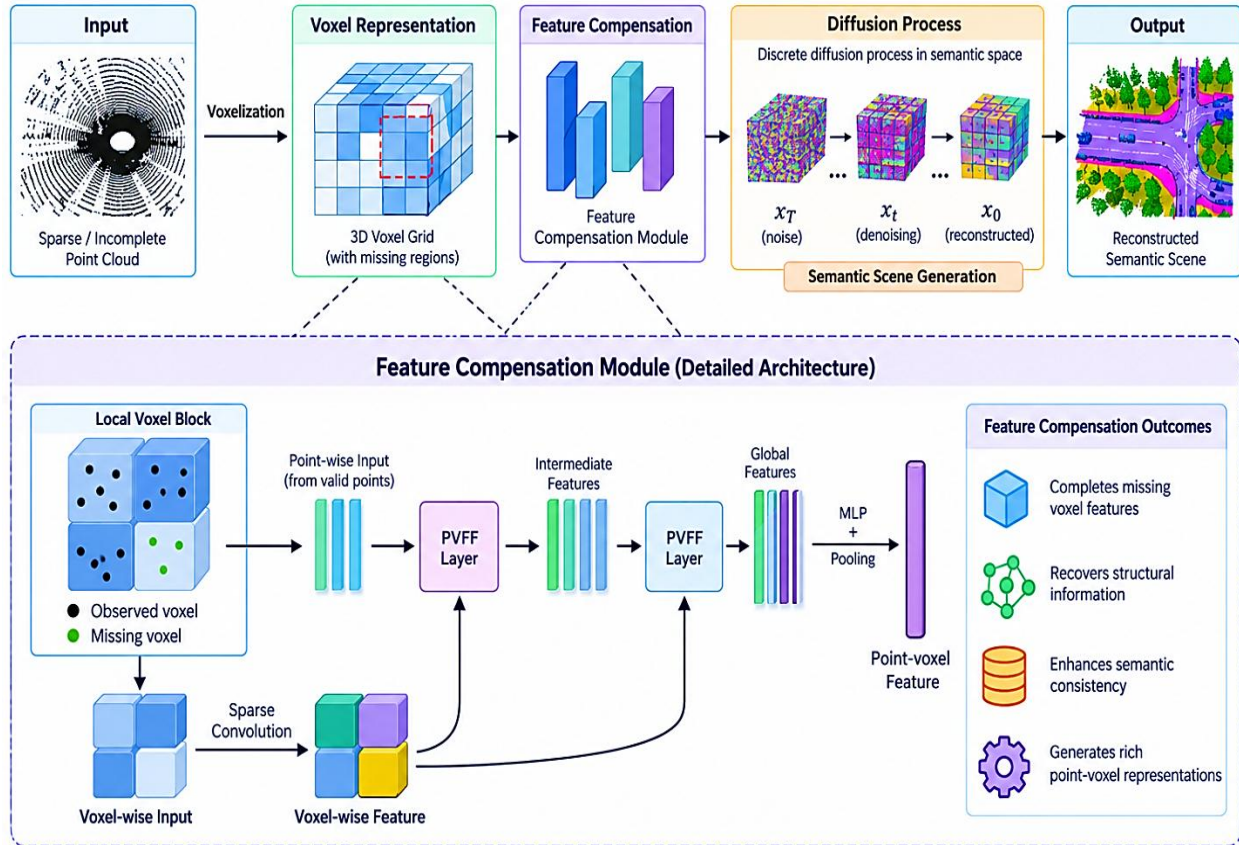


Figure 22: Diffusion-Based Semantic Generation and Reconstruction Framework

The lower feature-compensation architecture also highlights several intended outcomes, including completing missing voxel features, recovering structural information, enhancing semantic consistency, and generating richer point-voxel representations. These functions are important because diffusion-based generation depends on the quality of the representation supplied to the generative process. By combining voxel-based spatial organization, feature compensation, and diffusion-based denoising, the architecture provides a pathway for reconstructing semantic information that may be unavailable in the original observation.

4.3.4 Semantic Consistency and Fidelity in Generative Communication

Semantic consistency and fidelity are critical requirements when generative models are used to reconstruct or generate information in semantic communication systems. Generative models can produce highly plausible outputs even when the original information is incomplete or degraded. However, visual or linguistic plausibility alone does not guarantee that the generated information preserves the intended meaning. A reconstruction may appear realistic while introducing incorrect entities, relationships, attributes, or contextual details. Therefore, generative communication systems must evaluate whether the generated output remains semantically consistent with the source information, available context, and intended communication task.

Semantic fidelity refers to the degree to which the reconstructed information preserves the essential meaning of the original source. In a semantic communication setting, fidelity is not necessarily equivalent to exact source reconstruction. For example, when reconstructing a degraded image for an object-recognition task, preserving the identity, location, and relevant characteristics of the object may be more important than reproducing every pixel. Similarly, for incomplete text, preserving the intended concept or instruction may be more important than reproducing the exact original wording. The required level of fidelity therefore depends on the application and the information that the receiver needs to accomplish its task.

Semantic consistency provides an additional constraint by requiring the generated information to remain compatible with known contextual and domain information. A generative model should avoid producing content that conflicts with established knowledge or with reliable portions of the received information. Knowledge graphs, contextual representations, task information, and previously available semantic information can therefore provide useful constraints during reconstruction. This is particularly important when the generative model must infer information that was not directly observed.

Evaluation of generative semantic communication should consequently consider several complementary dimensions. Reconstruction measures can assess similarity between the generated and reference representations, while semantic measures can evaluate whether important concepts and relationships have been preserved. Task-level evaluation can determine whether the reconstructed information continues to support the intended application. Human or expert assessment may also be valuable for domains where semantic correctness is difficult to capture through a single numerical metric.

Another important consideration is the risk of semantic hallucination, in which a generative model introduces information that was not supported by the available evidence. Such errors can be particularly problematic in safety-critical or decision-oriented applications. Generative reconstruction should therefore distinguish between information that can be confidently inferred and information that remains uncertain. Semantic consistency and fidelity provide essential safeguards for generative communication. They ensure that the objective of generation remains meaning preservation rather than merely producing plausible content. Combining semantic constraints, contextual knowledge, task objectives, and appropriate evaluation methods can make generative reconstruction more reliable and suitable for practical semantic communication systems.

4.4 Intelligent Learning Strategies for Adaptive Semantic Communication

4.4.1 Supervised and Self-Supervised Semantic Representation Learning

A framework for learning semantic representations using both supervised and self-supervised approaches. The upper portion of the figure shows two alternative learning pipelines. In the supervised pathway, labeled training data, including multimodal imagery, are processed through data augmentation and an encoder. The resulting feature representation is then supplied to a classification head, where the model learns task-relevant semantic representations using a supervised loss. This approach directly associates learned features with predefined task labels and can therefore produce representations that are strongly aligned with the target application.

The second pathway illustrates self-supervised semantic representation learning using contrastive learning. Here, the training data do not require explicit labels. Instead, different augmented views of the same input are generated and processed through a shared encoder. The resulting feature embeddings are compared using a contrastive objective, where representations from related views are encouraged to become similar while representations from dissimilar samples are separated. This enables the model to learn useful semantic structures from unlabeled data and can be particularly valuable when obtaining large, high-quality labeled datasets is expensive or impractical.

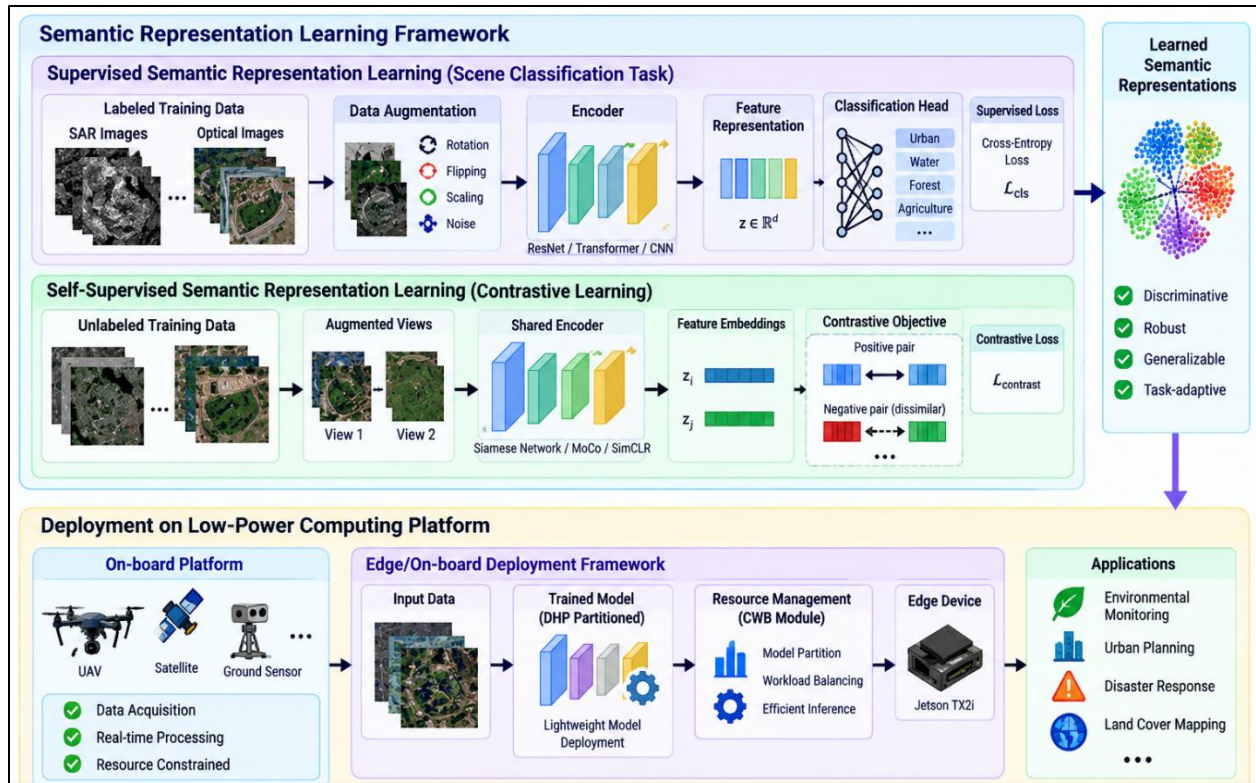


Figure 23: Supervised and Self-Supervised Semantic Representation Learning Framework

The right side of the upper section shows the resulting learned semantic representations, characterized in the figure as discriminative, robust, generalizable, and task-adaptive. These characteristics are important for semantic communication because the learned representation must remain useful when the source data, communication environment, or application requirements change. The figure therefore connects representation learning with the broader objective of producing semantic features that can support downstream tasks rather than simply reproducing the original input.

The lower portion demonstrates deployment on low-power computing platforms. Data can originate from onboard platforms such as UAVs, satellites, and ground sensors, where real-time processing and resource constraints are important. The trained model can subsequently be partitioned or adapted for edge/on-board deployment, while resource-management mechanisms support model partitioning, workload balancing, and efficient inference. The final outputs support applications such as environmental monitoring, urban planning, disaster response, and land-cover mapping.

4.4.2 Reinforcement Learning for Adaptive Semantic Transmission

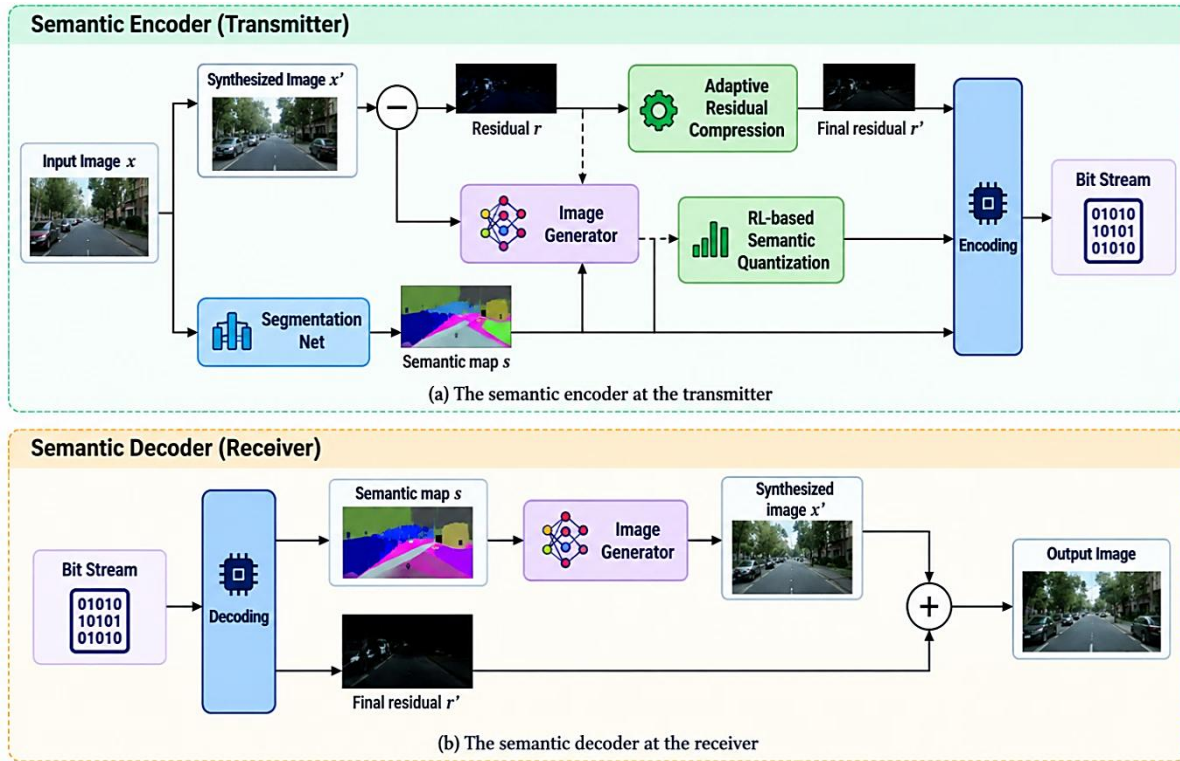


Figure 24: Reinforcement Learning-Based Adaptive Semantic Transmission Architecture

An adaptive semantic transmission architecture in which the transmitter separates an input image into semantic information and residual information before communication. The input image (x) is processed through a segmentation network to generate a semantic map (s), while an image generator produces a synthesized image (x') representing the semantic content. The difference between the original image and the synthesized image forms the residual (r). This separation is important because semantic information and residual information do not necessarily require identical transmission treatment. The semantic representation can be prioritized according to its importance, while residual information can be compressed according to the available communication resources.

The transmitter contains two important adaptive components: adaptive residual compression and RL-based semantic quantization. The residual is processed to generate a final residual representation (r'), while the semantic information is subjected to reinforcement-learning-based quantization. The RL-based component can conceptually determine an appropriate quantization strategy according to the current communication and semantic requirements. The resulting information is then passed to the encoding stage and converted into a bit stream for transmission. This architecture therefore combines semantic extraction, generative image synthesis, residual processing, and adaptive quantization rather than treating the entire image as a uniformly compressed signal. At the receiver, the transmitted bit stream is decoded to recover the semantic map and final residual representation. The recovered semantic map is supplied to the image generator, which produces a synthesized image (x'). The final residual (r') is then combined with the synthesized image to generate the reconstructed output image. This two-part reconstruction strategy allows the receiver to recover the major semantic structure through the semantic representation

while using the residual to restore additional visual information that was not captured by the semantic representation alone.

The figure therefore demonstrates a semantic-plus-residual communication strategy in which reinforcement learning is used specifically for adaptive semantic quantization. The approach can be particularly useful when communication resources vary because the quantization strategy can potentially be adjusted according to the desired balance between semantic quality, residual information, and transmission efficiency. However, the figure should be interpreted as an architectural illustration rather than a complete RL formulation because it does not explicitly show the RL state, action, reward, policy, or training loop.

4.4.3 Continual Learning for Dynamic Semantic Environments

Continual learning enables semantic communication systems to adapt their learned representations and decision mechanisms as communication environments, data distributions, and application requirements change over time. Conventional machine-learning models are commonly trained using a fixed dataset and deployed with relatively stable parameters. However, practical semantic communication environments are rarely stationary. User behavior, task requirements, channel conditions, traffic patterns, sensing data, and environmental characteristics may continuously change. A semantic representation that performs well under one set of conditions may therefore become less effective when the operating environment changes. Continual learning provides a framework for progressively updating the model using newly available information while maintaining previously acquired knowledge.

A major challenge in continual learning is catastrophic forgetting, where adapting a model to new information can cause it to lose knowledge learned from earlier data. This problem is particularly important in semantic communication because previously learned semantic relationships may remain relevant even when new conditions emerge. For example, a system deployed for environmental monitoring may encounter new scenes, seasonal changes, or previously unseen conditions while still needing to recognize information learned from earlier observations. Continual learning therefore aims to balance adaptation and knowledge retention. New semantic features can be incorporated while important existing representations are preserved, allowing the system to remain useful across multiple operating conditions.

Continual semantic learning can also support dynamic task adaptation. Different applications may require different semantic information, and the importance of particular features can change according to the current task. A communication system may consequently update its semantic encoder, representation-selection mechanism, or decoder according to changes in application requirements. Feedback from task performance can provide an important signal for identifying when adaptation is necessary. Similarly, changes in channel quality or available computational resources can motivate adjustments to semantic representation and processing strategies. This creates a learning process in which semantic communication evolves together with its operating environment rather than relying on a permanently fixed model.

Another important consideration is the efficient use of computational and communication resources. Continuous retraining of a large model from scratch can be expensive, particularly for edge devices and

resource-constrained platforms. Continual learning can instead update selected model components or incorporate new information incrementally. This makes it suitable for distributed semantic communication systems in which data are continuously generated at sensors, mobile devices, vehicles, or edge nodes. The system can progressively refine its semantic representations while reducing the need for complete model retraining.

Continual learning provides an important mechanism for achieving long-term adaptability and robustness in dynamic semantic communication environments. By combining knowledge retention, incremental adaptation, task feedback, and resource-aware model updating, continual learning can help semantic communication systems remain effective as their data, users, tasks, and communication conditions evolve. It therefore complements supervised, self-supervised, reinforcement, and generative learning approaches and contributes to the development of semantic communication systems capable of operating reliably in continuously changing environments.

4.4.4 Federated and Distributed Learning for Semantic Intelligence

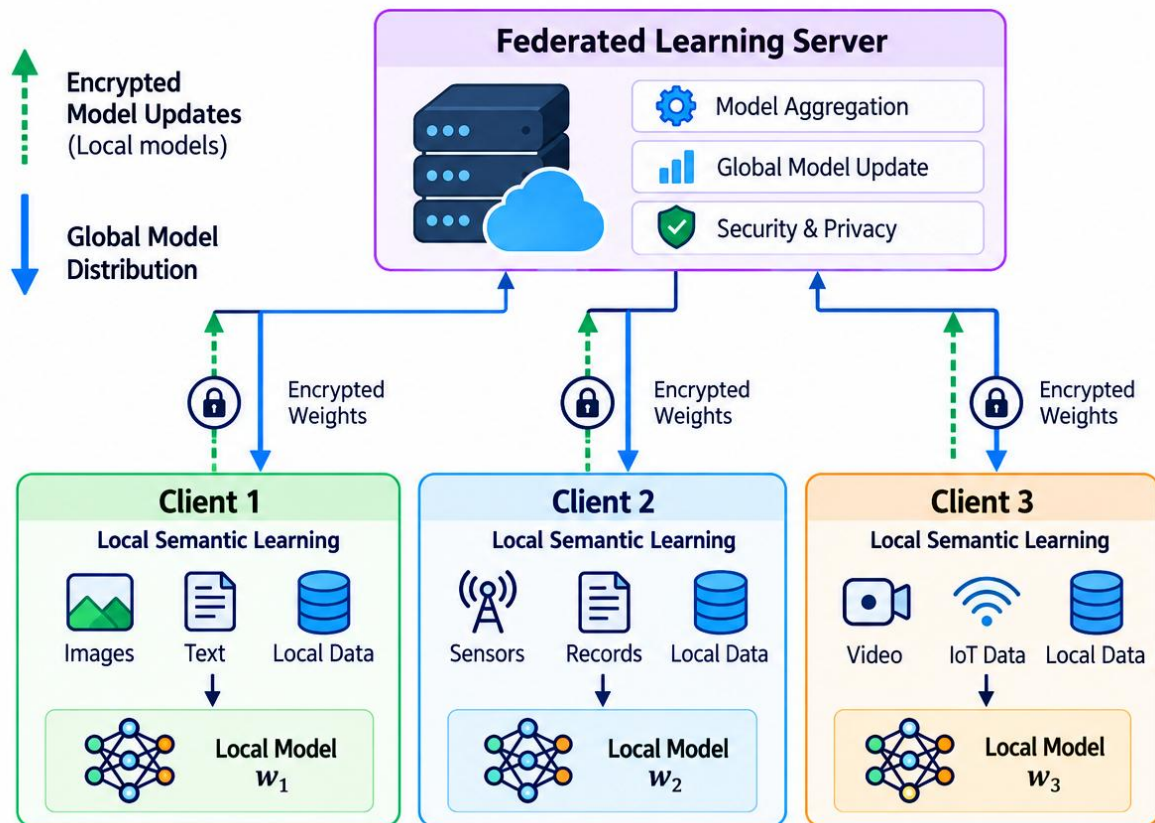


Figure 25: Federated Learning Architecture for Distributed Semantic Intelligence

A federated learning architecture for distributed semantic intelligence, in which multiple clients independently perform local semantic learning while collaborating through a central federated learning server. The figure contains three clients with different types of locally available information. Client 1 uses images, text, and local data; Client 2 uses sensor information, records, and local data; and Client 3 uses

video, IoT data, and local data. Each client processes its locally available information to train a local semantic model, represented by (w_1) , (w_2) , and (w_3) . This arrangement reflects the heterogeneous nature of semantic communication environments, where different devices can observe and process different types of information.

The local semantic models are not directly replaced by a centralized training process. Instead, the clients communicate model updates or encrypted weights to the federated learning server. The server performs model aggregation to combine the contributions from participating clients and generates an updated global model. The global model can then be distributed back to the clients, allowing them to incorporate knowledge learned from other participating devices. This iterative process enables collaborative semantic representation learning while maintaining the separation of local datasets.

The upper portion of the figure represents the federated learning server, which performs model aggregation, generates global model updates, and provides security and privacy mechanisms. The bidirectional communication paths illustrate the exchange between the clients and the server: global model information is distributed from the server to participating clients, while locally trained model updates are returned to the server. The use of encrypted weights in the figure emphasizes that the collaborative learning process can incorporate mechanisms intended to protect model-update communication and reduce direct exposure of locally stored information.

The architecture is particularly relevant to semantic intelligence because different clients may possess complementary knowledge about environments, users, sensing conditions, or application domains. Collaborative training can therefore allow semantic models to benefit from distributed experiences without requiring all raw data to be transferred to a central location. At the same time, practical federated semantic communication must address issues such as heterogeneous data distributions, unequal client resources, communication overhead, model-update privacy, and differences in local learning quality.

Chapter 5

Language-Centric Semantic Communications and Large Language Model Integration

5.1 Natural Language Understanding for Meaning-Aware Communication

5.1.1 Linguistic Representation and Semantic Feature Extraction

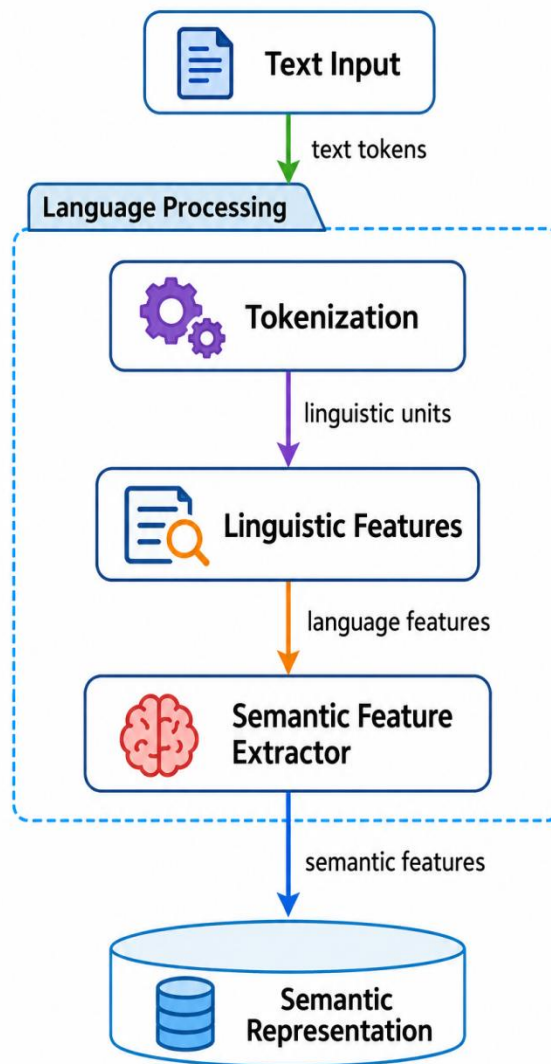


Figure 26: Linguistic and Semantic Feature Extraction Pipeline for Text Communication

The transformation of textual information into a structured semantic representation through a sequence of language-processing stages. The process begins with the text input, which represents the original information that needs to be processed for meaning-aware communication. Rather than directly treating the complete text as a sequence of symbols for transmission, the architecture first converts the input into

smaller linguistic units and progressively extracts increasingly meaningful information. This provides a conceptual connection between conventional language processing and semantic communication.

The first stage within the language processing block is tokenization. Tokenization divides the input text into linguistic units that can be processed computationally. These units may correspond to words, subwords, or other token-level representations depending on the language-processing system. The resulting tokens are then passed to the linguistic feature extraction stage, where characteristics of the language are identified and represented. Such processing provides an intermediate representation from which higher-level semantic characteristics can subsequently be derived.

The semantic feature extractor forms the next major stage of the pipeline. It processes the linguistic features to identify information associated with the meaning of the input rather than focusing only on its surface-level form. The extracted semantic features are then passed to the final semantic representation stage. This representation can provide a more compact and meaning-oriented description of the original text, which is particularly useful in semantic communication systems where the objective is to preserve information relevant to interpretation and downstream tasks.

5.1.2 Contextual Language Understanding and Meaning Representation

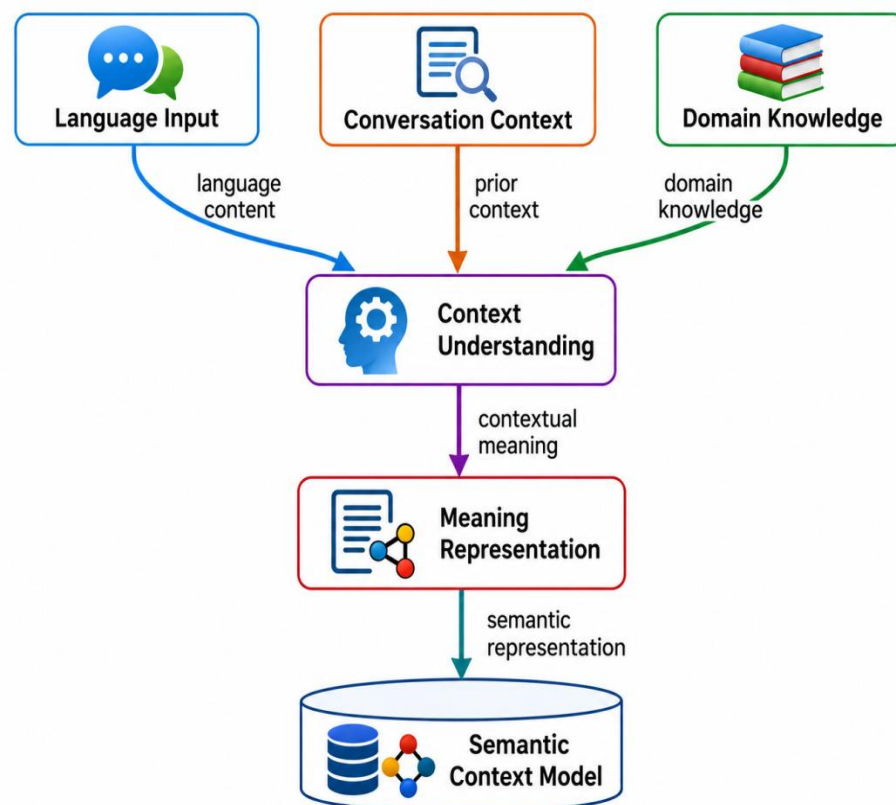


Figure 27: Contextual Language Understanding and Semantic Meaning Representation

Contextual language understanding can be used to transform language input into a meaning-oriented semantic representation. The framework begins with three sources of information: language input,

conversation context, and domain knowledge. Language input provides the immediate content that needs to be interpreted, while conversation context supplies information from the preceding interaction. Domain knowledge contributes knowledge about the relevant subject or application area. Considering these sources together is important because the meaning of a linguistic expression can depend strongly on the surrounding conversation and the knowledge associated with the domain.

The central context understanding stage combines these sources to determine contextual meaning. Rather than interpreting language only according to its surface form, the system considers prior context and domain-specific knowledge when determining what the input means. This is particularly important for semantic communication because the same word or expression can have different meanings depending on the situation in which it is used. Contextual understanding therefore provides a mechanism for distinguishing the intended interpretation and identifying information that is relevant to the communication task.

The resulting contextual meaning is transformed into a meaning representation, which provides a structured representation of the information extracted from the language input. This representation can capture the concepts and relationships that are important for subsequent semantic processing. The figure then shows the resulting semantic representation being incorporated into a semantic context model. Such a model can maintain information about the interpreted meaning together with relevant contextual and domain knowledge, providing a richer basis for subsequent semantic encoding, transmission, and interpretation.

The progression from language content and contextual knowledge to contextual understanding, meaning representation, and semantic context modeling. This architecture demonstrates why contextual information is an important component of language-centric semantic communication. A communication system that considers conversation history and domain knowledge can potentially identify the information that is actually relevant to the intended meaning, rather than treating every linguistic element as equally important. The resulting semantic context model can subsequently support more efficient representation, meaning preservation, and task-oriented communication in language-based semantic systems.

5.1.3 Intent Recognition and Task-Oriented Language Processing

A task-oriented language-processing framework in which natural-language input from a user is transformed into information that can guide a semantic communication objective. The process begins with the user providing natural-language information to the language-understanding stage. Rather than treating the input only as a sequence of linguistic symbols, the system processes the content to identify information relevant to the user's intended task. This provides an important foundation for semantic communication because the information that needs to be transmitted can depend on what the user is actually trying to accomplish.

The input processor forms the central processing stage within the language-understanding block. It receives the natural-language input and produces information that can be used by two complementary components: the intent detector and the entity extractor. Semantic cues are directed toward intent detection, which determines the intended purpose of the user's request. At the same time, relevant entities are extracted from the input. These entities can provide the task with specific information required for

subsequent processing. Separating intent from entities allows the system to represent both what the user wants and the information associated with that request. The detected intent and extracted task parameters are subsequently provided to the task manager. The task manager combines these elements to determine the requirements of the intended task. This stage is important because the communication system does not necessarily need to transmit all information contained in the original language input. Instead, it can determine which information is relevant to the task and establish a corresponding communication objective. This creates a direct relationship between natural-language understanding and semantic communication.

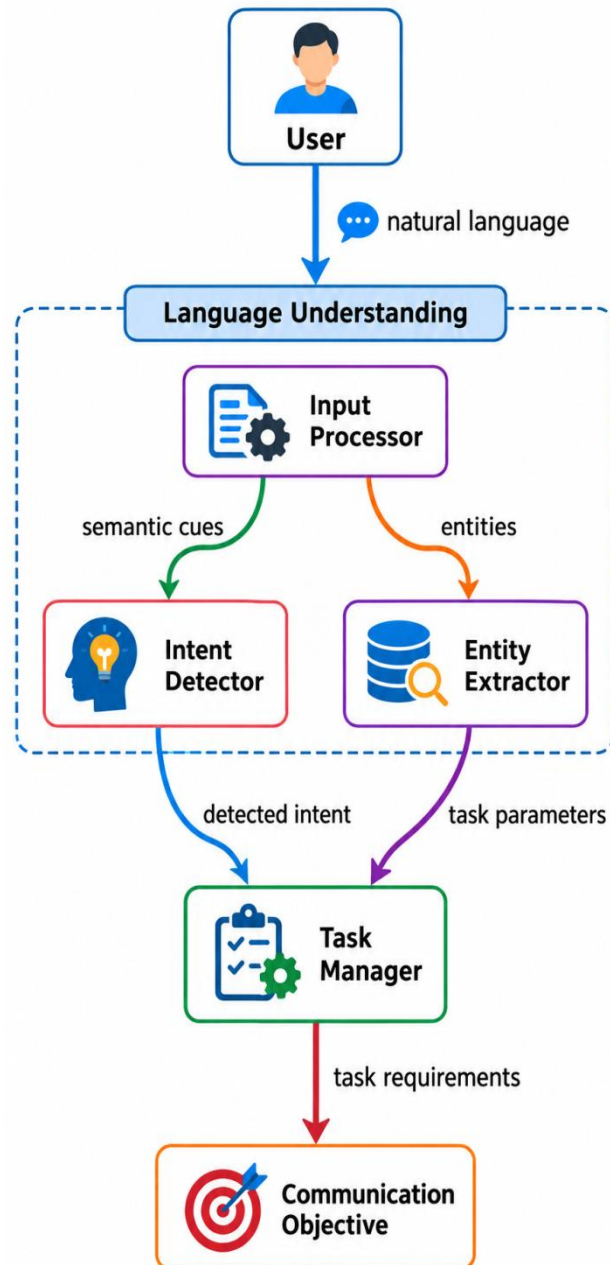


Figure 28: Intent Recognition and Task-Oriented Language Processing Framework

5.1.4 Meaning-Preserving Language Transmission and Reconstruction

An end-to-end framework for meaning-preserving language transmission and reconstruction. The process begins with a source sentence containing the linguistic content that the sender intends to communicate. Instead of transmitting the sentence solely as a sequence of conventional symbols, the information is processed by a semantic encoder that extracts and represents the underlying meaning. This creates a semantic message that focuses on information relevant to the intended interpretation and provides the basis for meaning-aware communication.

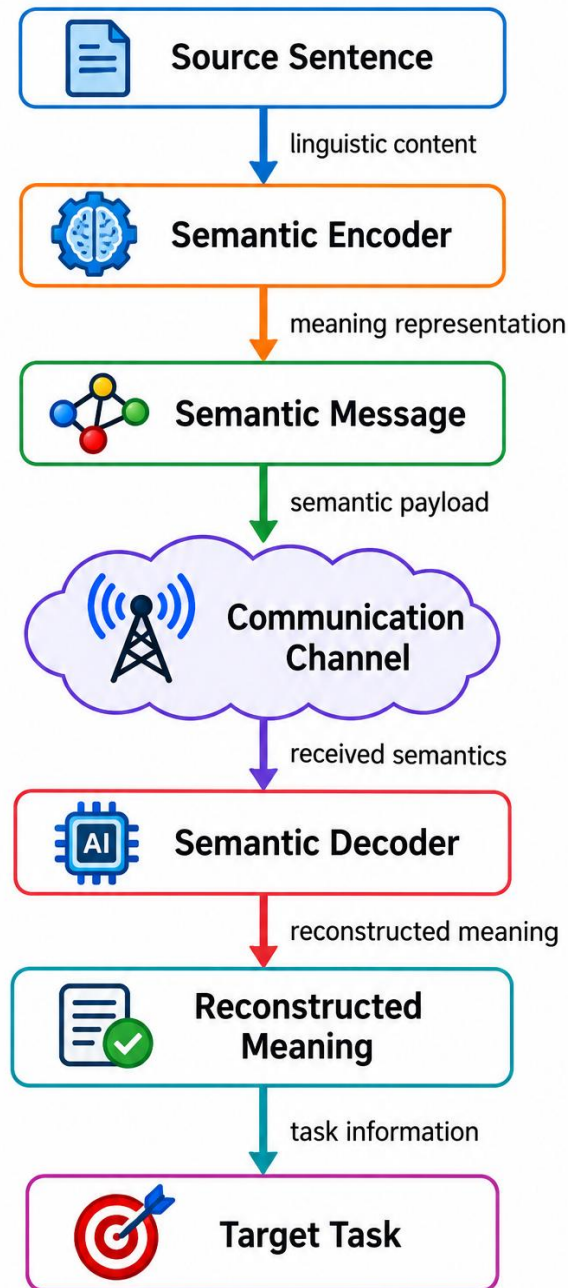


Figure 29: Meaning-Preserving Language Transmission and Reconstruction Framework

The resulting semantic message forms the semantic payload that is transmitted through the communication channel. During transmission, the semantic information may be affected by channel conditions and other communication impairments. At the receiver, the received semantic information is processed by the semantic decoder, which interprets the representation and reconstructs the intended meaning. The architecture therefore separates semantic representation from the exact surface form of the original sentence, allowing the receiver to focus on recovering information that is meaningful rather than necessarily reproducing every linguistic detail. The reconstructed meaning produced by the semantic decoder represents the information recovered at the receiving side. Its objective is to preserve the essential concepts and relationships contained in the source message while remaining suitable for the intended communication context. Depending on the application, the reconstructed output may not need to reproduce the source sentence word for word. Instead, it should retain the semantic information required for successful interpretation or subsequent processing.

The final stage of the framework is the target task, which demonstrates the task-oriented nature of semantic communication. The reconstructed meaning is supplied to the application or task that ultimately determines whether the communication was successful. This establishes a direct relationship between semantic transmission and application-level performance. Overall, the figure represents the complete language-semantic communication process, emphasizing the preservation of meaning from the source sentence through semantic encoding, communication, decoding, reconstruction, and task execution. It provides a strong conceptual basis for discussing semantic fidelity, meaning preservation, task-oriented language communication, and evaluation of reconstructed language information.

5.2 Transformer-Based Natural Language Semantic Communication Systems

5.2.1 Language Transformer Models for Semantic Encoding

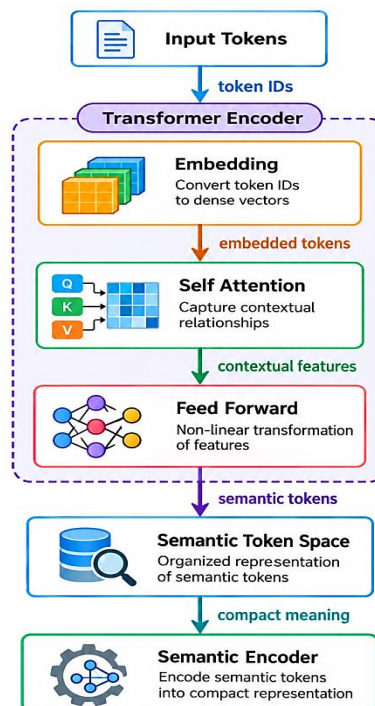


Figure 30: Transformer-Based Language Semantic Encoding Pipeline

Transformer-based semantic encoding pipeline for natural language. The process begins with input tokens obtained from the original language information. These tokens are represented by token identifiers and provided to the Transformer encoder. The first major operation is embedding, where discrete token identifiers are converted into dense numerical vectors. This transformation provides a continuous representation that can be processed by the subsequent neural components while allowing relationships among different linguistic elements to be learned.

The embedded tokens are then processed through the self-attention mechanism, which captures contextual relationships among the elements of the input sequence. The queries, keys, and values generated within the attention mechanism allow the model to determine how strongly different tokens should influence one another. As a result, the representation of each token can incorporate information from other relevant tokens in the sequence. This contextual processing is particularly important for semantic communication because the meaning of a linguistic element often depends on its surrounding words and broader sentence context.

Following self-attention, the contextual features are passed through a feed-forward network, which performs a nonlinear transformation of the learned features. The combination of self-attention and feed-forward processing allows the Transformer encoder to progressively refine the linguistic representation. The output is represented as semantic tokens, which contain contextualized information that is more closely related to the meaning of the original language input than the initial token identifiers. The semantic tokens are subsequently organized within the semantic token space and passed to the semantic encoder. This final stage converts the semantic tokens into a compact semantic representation, which can serve as the representation transmitted by a semantic communication system.

5.2.2 Attention-Based Context and Dependency Modeling

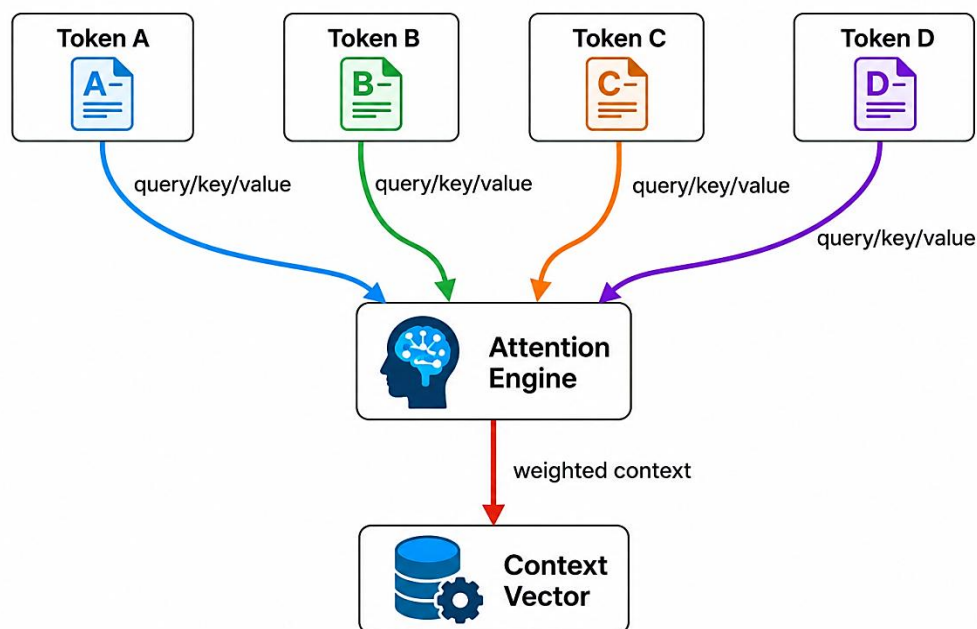


Figure 31: Attention-Based Context and Dependency Modeling

The role of attention-based context and dependency modeling in language-oriented semantic processing. The framework begins with four input tokens, represented as Token A, Token B, Token C, and Token D. Each token contributes query, key, and value information to the central attention engine. Rather than processing each token independently, the attention mechanism considers the relationships among the tokens and determines how information from different parts of the input should contribute to the resulting contextual representation.

The attention engine receives the query, key, and value representations associated with the input tokens. Queries represent the information used to determine what contextual information is relevant, keys provide representations against which the queries can be compared, and values contain the information that is integrated according to the resulting attention relationships. This mechanism allows the model to assign different importance to different tokens when constructing the representation of the input. Consequently, tokens that provide more relevant contextual information can have a stronger influence on the resulting representation.

The output of the attention engine is shown as a weighted context, which summarizes information obtained by combining the contributions of the input tokens according to their learned relationships. This weighted contextual information is then transformed into a context vector. The context vector provides a compact representation of the dependencies and contextual relationships identified within the input sequence. Such a representation is particularly useful when the meaning of a token depends on other words or concepts occurring elsewhere in the sequence.

5.3 Large Language Models as Intelligent Semantic Communication Components

5.3.1 Foundation Language Models for Semantic Representation

A foundation language model-based framework for semantic representation in which textual or multimodal information is progressively transformed into a representation suitable for meaning-aware communication. The process begins with text or multimodal input, which may contain information in different forms. This input is converted into input tokens and provided to the foundation language model. The architecture therefore establishes a connection between general-purpose language modeling and semantic communication, where learned representations can be used to capture information relevant to meaning.

Within the foundation language model, the first major stage is the embedding layer. This layer converts the input tokens into dense numerical representations that can be processed by the model. The embeddings are then passed through a sequence of Transformer blocks, where contextual relationships among the input elements are learned. Through this contextual processing, the model can represent information according to its surrounding linguistic or multimodal context rather than considering individual tokens in isolation.

The output of the Transformer blocks is provided to the context representation stage. This representation captures contextual features learned by the foundation model and provides an intermediate representation from which semantic information can be derived. The figure then shows the contextual representation being transformed into a semantic representation within a semantic space. This stage is particularly

important for semantic communication because it provides a representation focused on concepts and relationships that can be relevant to subsequent communication and application tasks.

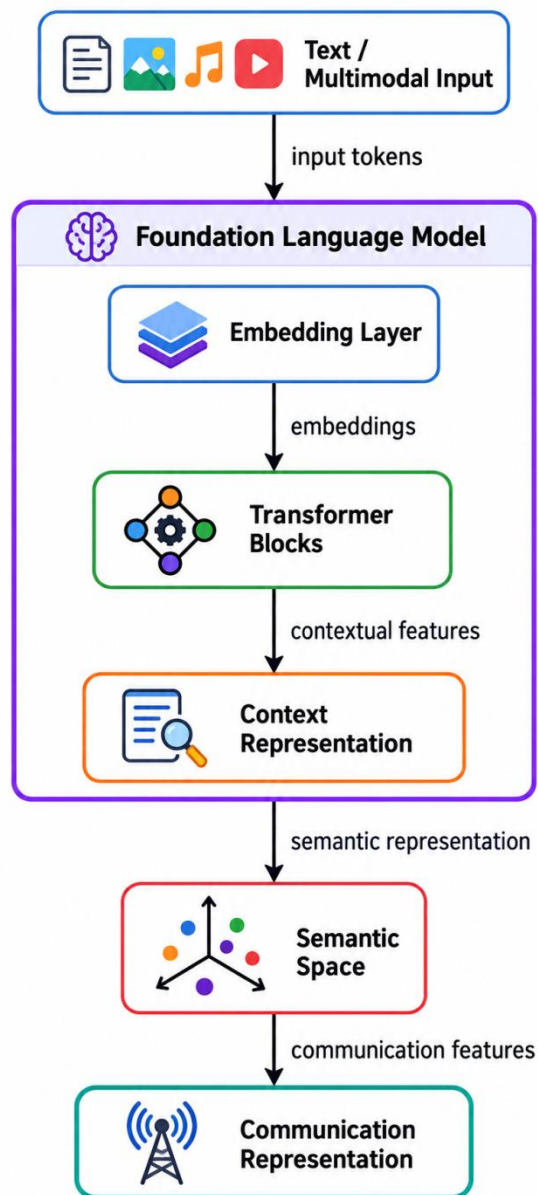


Figure 32: Foundation Language Model-Based Semantic Representation Pipeline

5.3.2 Prompt-Based Semantic Information Extraction and Transmission

Prompt-based semantic information extraction provides a flexible mechanism for identifying and transmitting information that is relevant to a particular communication objective. In conventional communication systems, the source information is generally processed without explicitly specifying which aspects of the content are most important for the intended task. Prompt-based approaches introduce an additional layer of guidance in which a prompt describes the desired information, task, context, or interpretation. A large language model can use this guidance to determine which portions of the input

contain the most relevant semantic information. Consequently, communication can be organized around the intended purpose rather than the complete surface-level content of the source.

A prompt can specify requirements such as extracting particular entities, summarizing important events, identifying relationships, answering a question, or selecting information required by a downstream application. The language model interprets the source together with the prompt and generates a task-relevant semantic representation. For example, when a long textual description contains information about several events but the application requires only information related to a specific event, the prompt can guide the semantic processing system toward the relevant content. This approach can reduce unnecessary information processing and provide a more focused semantic representation for subsequent transmission.

Within a semantic communication architecture, prompt-based extraction can operate before semantic encoding. The source content is first interpreted according to the prompt, after which relevant concepts, entities, relationships, or contextual information are identified. The selected information can then be transformed into a compact semantic representation and passed to the communication module. At the receiver, the transmitted representation can be interpreted using a corresponding prompt, task specification, or contextual instruction. This allows the receiver to reconstruct information according to the intended application rather than simply reproducing the complete original message.

An important advantage of prompt-based semantic communication is its ability to support task adaptability. Different prompts can produce different semantic representations from the same underlying source, allowing the communication system to adjust information selection according to changing application requirements. Prompts can also provide contextual guidance when ambiguity exists, helping the model distinguish between multiple possible interpretations. However, prompt-based approaches require careful control of prompt design, model behavior, contextual consistency, and semantic fidelity. An inappropriate or ambiguous prompt may cause relevant information to be omitted or irrelevant information to be emphasized. Therefore, prompt-based semantic information extraction should be evaluated according to both the quality of the extracted meaning and its effectiveness for the final communication task.

5.3.3 Knowledge-Augmented Semantic Communication with Language Models

Knowledge-augmented semantic communication framework in which language models are supported by external knowledge and contextual information. The process begins with user information and a knowledge base. User information provides the source context and information need, while the knowledge base contains potentially relevant domain information. Instead of relying exclusively on the information contained in the immediate input, the framework retrieves additional knowledge that can support a more complete and accurate interpretation of the intended meaning.

The retriever forms the connection between the user's information need and the available knowledge base. It uses the information need together with relevant knowledge to identify contextual information that can assist the language model. The retrieved context is then provided to the language model along with the source context. This allows the model to interpret the communication content using both the information supplied by the user and additional knowledge obtained from the external knowledge repository.

The language model produces interpreted meaning, which is subsequently processed by the semantic composer. The semantic composer integrates the interpreted information with the retrieved knowledge to produce knowledge-enriched semantics. This stage is important because semantic communication may require more than direct extraction of information from the source. External knowledge can provide missing context, clarify concepts, and support the construction of a richer semantic representation that is aligned with the intended communication objective.

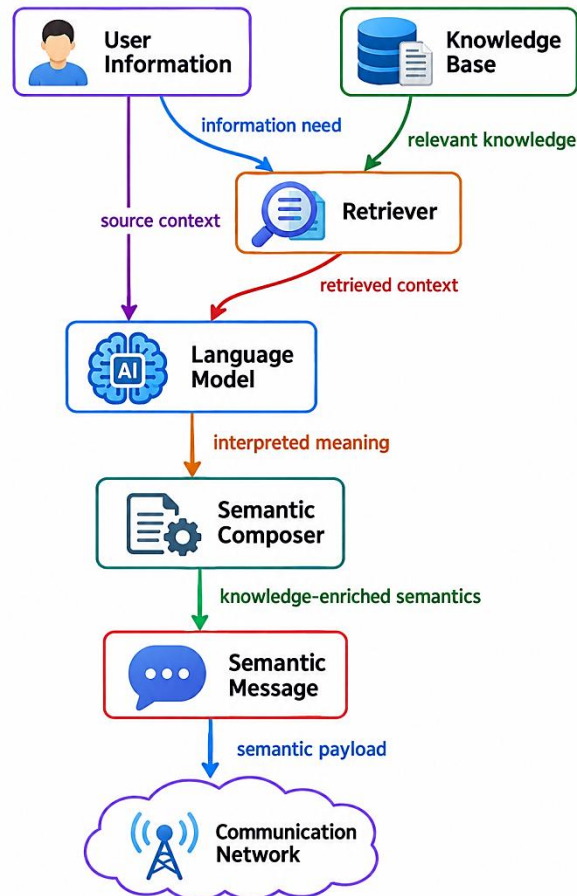


Figure 33: Knowledge-Augmented Semantic Communication Using Language Models

The resulting semantic message represents the knowledge-enriched information that is selected for communication. It is subsequently treated as the semantic payload and provided to the communication network. Overall, the figure demonstrates how user information, source context, knowledge retrieval, language-model reasoning, semantic composition, and communication can be integrated into a unified architecture. It highlights the potential of knowledge-augmented language models to improve semantic interpretation by providing access to relevant contextual knowledge, while also showing that the final transmitted information can be focused on meaning and task relevance rather than transmitting the complete underlying information.

5.3.4 LLM-Assisted Semantic Decoding and Task Completion

Large language models can play an important role at the receiver side of semantic communication systems by assisting in the interpretation and reconstruction of transmitted semantic information. Unlike

conventional decoders that primarily attempt to recover a predefined symbol sequence, an LLM-assisted semantic decoder can interpret a received semantic representation according to its learned language knowledge, contextual understanding, and task requirements. This makes the decoding process more focused on recovering the intended meaning and producing information that is useful for the final application. The approach is particularly relevant when the transmitted representation is compact and does not contain every detail of the original source information.

In an LLM-assisted architecture, the receiver first obtains the semantic representation or semantic message transmitted through the communication channel. The representation is then provided to an LLM together with relevant context, instructions, or task information. The model interprets the received semantic content and reconstructs a meaningful representation of the original information. Context can play an important role because the received semantic representation may be intentionally compressed and therefore require background knowledge or contextual information for correct interpretation. The LLM can use these available sources to resolve ambiguity and generate a coherent semantic reconstruction.

The reconstructed information can then be supplied to a task-completion module or downstream application. The objective is not necessarily to reproduce the source message exactly, but to provide sufficient information for successful execution of the intended task. For example, a receiver may need to identify an object, answer a question, summarize information, classify content, or make an application-level decision. In these situations, successful communication can be evaluated according to whether the reconstructed information enables the required task to be completed correctly. This reinforces the task-oriented principle of semantic communication, in which communication quality is connected to application utility rather than only conventional bit or symbol accuracy.

LLM-assisted semantic decoding also introduces important challenges. The decoder must maintain semantic consistency with the transmitted information and avoid introducing unsupported information during reconstruction. Model-generated content can potentially contain errors, ambiguities, or interpretations that are inconsistent with the source semantics. Therefore, semantic fidelity, contextual consistency, task accuracy, and reliability should be considered when evaluating LLM-assisted decoding. Appropriate receiver-side constraints, contextual information, and task-specific evaluation mechanisms can help control these issues. Overall, LLM-assisted semantic decoding provides a pathway toward intelligent receivers capable of interpreting compact semantic representations and converting them into useful task information, extending semantic communication from simple meaning reconstruction toward meaning-aware task completion.

5.4 Performance Evaluation and Optimization of Language Semantic Communication

Performance evaluation in language semantic communication requires a broader perspective than conventional communication metrics such as bit-error rate, packet loss, or throughput. Since the primary objective is to preserve and communicate meaning, evaluation should determine whether the information reconstructed at the receiver remains semantically consistent with the source and whether it supports the intended application. Language semantic communication systems may intentionally remove redundant linguistic content, compress representations, or reconstruct information using learned models. Therefore, a system can achieve substantial information reduction while still maintaining high communication

utility. Performance evaluation should consequently consider semantic quality, task effectiveness, communication efficiency, and robustness together.

A comprehensive evaluation framework can examine the complete processing chain from source language to semantic representation, transmission, decoding, and task execution. At the source side, the system can be evaluated according to how effectively it extracts information that is relevant to the intended communication objective. During transmission, the compactness and reliability of the semantic representation become important. At the receiver, evaluation should determine whether the reconstructed meaning remains aligned with the source and whether the recovered information is sufficient for the target application. This end-to-end perspective is particularly important for LLM-assisted systems because their ability to interpret and reconstruct semantic information may affect performance beyond the communication channel itself.

Language semantic communication should also be evaluated under different communication conditions and application requirements. A representation that performs well for one task may not provide sufficient information for another. Similarly, semantic quality may vary depending on sentence complexity, contextual ambiguity, domain knowledge, language characteristics, and the severity of channel impairments. Evaluation should therefore consider diverse datasets, communication conditions, and downstream tasks rather than relying on a single metric.

The four dimensions discussed in this section provide a structured framework for evaluating language semantic communication. Semantic similarity and meaning preservation determine whether the intended meaning is retained, while task completion accuracy measures whether the reconstructed information supports the required application. Communication efficiency evaluates the reduction of transmitted information and associated resource requirements. Finally, robustness evaluation determines whether language models and semantic representations remain effective when communication conditions introduce noise, distortion, or incomplete information. Together, these dimensions provide a more appropriate basis for assessing the practical performance of language-centric semantic communication systems.

5.4.1 Semantic Similarity and Meaning Preservation Metrics

Semantic similarity and meaning preservation are fundamental evaluation dimensions for language semantic communication because the objective of the system is to preserve the intended information rather than necessarily reproduce the exact source sentence. A reconstructed sentence can differ substantially from the original wording while still expressing the same meaning. Conversely, a reconstruction may have high lexical similarity but introduce a small semantic change that significantly affects the intended interpretation. Evaluation must therefore distinguish between surface-level textual similarity and deeper semantic correspondence between the source and reconstructed information.

Semantic similarity can be evaluated by comparing representations of the original and reconstructed language. Sentence-level or token-level embeddings can provide vector representations that capture semantic relationships, allowing the similarity between source and reconstructed information to be estimated. A common conceptual measure is cosine similarity between two semantic representations, expressed as

$$\text{sim}(s, \hat{s}) = \frac{s^T \hat{s}}{\|s\| \|\hat{s}\|}$$

where s represents the semantic representation of the source and $\hat{s}$ represents the reconstructed representation. Higher similarity generally indicates stronger semantic correspondence, although the metric should be interpreted according to the application and representation model used.

Meaning preservation should also consider whether important entities, relationships, attributes, and contextual information remain correct after transmission. For example, changing the identity of an entity, numerical value, location, temporal condition, or relationship can produce a substantial semantic error even when the reconstructed sentence appears linguistically fluent. Therefore, semantic evaluation can combine representation-based similarity with structured checks of critical information. This is especially important for task-oriented communication, where a small semantic change may lead to an incorrect downstream decision.

LLM-based evaluation can provide another mechanism for assessing whether reconstructed language preserves the intended meaning. A language model can compare source and reconstructed information according to semantic equivalence, contextual consistency, and task relevance. However, model-based evaluation should be carefully controlled because the evaluator itself may introduce subjective judgments or inconsistencies. Human evaluation remains useful for validating semantic equivalence in challenging cases, particularly when context or domain-specific knowledge is required. Overall, semantic similarity and meaning-preservation metrics should provide evidence that the communication system retains the essential information required for interpretation and application, while avoiding an unnecessary requirement for exact word-level reconstruction.

5.4.2 Task Completion Accuracy and Application-Level Performance

Task completion accuracy evaluates semantic communication according to its ultimate purpose: whether the reconstructed information enables the intended task to be performed successfully. This perspective is particularly important for language semantic communication because the quality of a transmitted message cannot always be determined from the reconstructed text alone. A reconstruction may differ considerably from the source wording but still contain all information required for the application. Conversely, a fluent and semantically similar reconstruction may fail if it omits a critical task-specific detail.

The evaluation process begins by defining the target task associated with the communication system. Tasks may include question answering, text classification, information extraction, decision support, summarization, instruction following, or other application-specific objectives. The reconstructed semantic information is provided to the corresponding task-processing component, and its output is compared with the expected result. The resulting performance provides a direct measure of whether semantic transmission preserved the information necessary for task execution.

Task completion accuracy can be expressed conceptually as the proportion of communication instances for which the receiver produces the correct task outcome:

$$A_{\text{task}} = \frac{N_{\text{successful tasks}}}{N_{\text{total tasks}}}$$

This measure can be supplemented with task-specific metrics when simple accuracy is inappropriate. For example, classification tasks may use precision, recall, or F1-score, while generation-based tasks may require evaluation of factual correctness, relevance, or instruction compliance. The choice of metric should therefore correspond to the actual application rather than relying on a universal measure.

Application-level evaluation also provides an important connection between semantic quality and communication resources. A semantic communication system may deliberately transmit fewer features or tokens than a conventional system. The reduction is useful only if the resulting communication remains sufficient for the target task. Evaluation can therefore examine the relationship between task performance and transmitted information, allowing designers to determine how much semantic information is necessary to maintain acceptable application performance.

This task-oriented perspective is particularly important for LLM-assisted semantic communication. Large language models may reconstruct linguistically coherent information from compact or partially degraded representations, but successful generation alone does not demonstrate successful communication. The reconstructed information must remain consistent with the intended task. Evaluation should therefore consider both semantic correctness and application outcome. Ultimately, task completion accuracy provides an end-to-end measure of whether semantic communication has achieved its practical objective: delivering enough meaning to enable the receiver to perform the required task reliably.

5.4.3 Communication Efficiency and Information Reduction

Communication efficiency is a central motivation for language semantic communication because semantic systems aim to avoid transmitting information that does not contribute meaningfully to the intended task. Conventional language communication may transmit the complete textual representation even when only a small portion of the information is relevant to the receiver. Semantic communication instead attempts to identify important concepts, relationships, and contextual information and represent them in a more compact form. Performance evaluation should therefore quantify how much information can be reduced while maintaining acceptable semantic and task performance.

One important evaluation dimension is the amount of information transmitted by the semantic system compared with a conventional representation. This can include the number of tokens, encoded symbols, representation dimensions, transmitted bits, or other communication resources depending on the architecture. A simple information-reduction ratio can be expressed as

$$A_{\text{reduction}} = 1 - \frac{B_{\text{semantic}}}{B_{\text{conventional}}}$$

where B_{semantic} represents the communication information required by the semantic system and $B_{\text{conventional}}$ represents the corresponding conventional transmission requirement. A higher reduction indicates greater communication compression, provided that semantic quality and task performance remain acceptable.

However, information reduction should not be considered independently from semantic performance. Excessive compression may remove information that is necessary for correct interpretation or task completion. Therefore, evaluation should examine the trade-off between communication efficiency and semantic quality. A useful system should achieve meaningful reductions in transmitted information without causing unacceptable degradation in semantic fidelity or application-level performance. This trade-off can be particularly important for bandwidth-constrained devices, distributed systems, and communication environments where transmission resources are limited.

Communication efficiency can also involve computational and energy requirements. Language semantic systems may reduce transmission overhead while introducing additional computation through Transformer models, embedding generation, retrieval mechanisms, or LLM-based processing. Consequently, an end-to-end assessment should consider both communication savings and the computational resources required to generate and decode semantic representations. Latency, memory requirements, processing complexity, and energy consumption can therefore complement transmission-size measurements.

The effectiveness of information reduction ultimately depends on the application. A highly compressed representation may be sufficient for a simple classification task but inadequate for a task requiring detailed reconstruction. Evaluation should therefore report communication efficiency together with semantic similarity and task completion performance. This establishes whether the reduction is genuinely useful rather than merely producing a smaller representation. The goal is not simply to minimize transmitted information, but to identify the minimum sufficient semantic information required to preserve meaning and achieve the intended task reliably.

5.4.4 Robustness of Language Models under Noisy Communication Conditions

Robustness evaluation examines whether language semantic communication systems can maintain semantic quality and task performance when the transmitted information is affected by communication impairments. In practical environments, semantic representations may experience noise, interference, packet loss, fading, quantization errors, incomplete transmission, or other forms of degradation. Language models can potentially compensate for some missing or corrupted information through contextual understanding and learned linguistic knowledge, but excessive degradation may cause incorrect reconstruction or task failure. Robustness evaluation is therefore necessary to determine the operational limits of semantic communication systems.

A robust evaluation framework should expose the semantic communication system to progressively different communication conditions and observe how semantic similarity, meaning preservation, and task accuracy change. The comparison should include both clean and degraded transmission conditions. This allows researchers to determine whether the system degrades gradually or experiences a sharp reduction in performance once communication quality falls below a particular level. Such analysis is especially useful for identifying operating regions in which semantic communication can maintain acceptable application performance despite imperfect transmission.

Robustness should also be examined across different types of language inputs. Short and simple sentences may remain understandable under substantial degradation, whereas long, ambiguous, or context-

dependent language may be more sensitive to information loss. Domain-specific terminology, numerical information, named entities, and relationships may also require stronger preservation than ordinary linguistic content. Consequently, robustness evaluation should include diverse language characteristics and communication tasks rather than relying on a single dataset or message type.

LLM-based semantic systems introduce additional considerations because a language model may generate fluent content even when some transmitted information has been corrupted. Fluency alone should not be interpreted as successful reconstruction. The generated output must remain faithful to the transmitted semantic information and should not introduce unsupported facts or alter task-critical details. Evaluation should therefore combine semantic fidelity with factual consistency and application-level task performance. Where external knowledge or contextual information is available, its influence on robustness should also be examined.

Overall, robustness should be characterized as the ability to maintain useful meaning and task performance under progressively degraded communication conditions. An effective language semantic communication system should exploit contextual and learned representations to tolerate moderate information loss while maintaining control over semantic errors. Robustness evaluation can consequently guide the design of adaptive semantic encoders, error-resilient representations, intelligent information selection mechanisms, and receiver-side reconstruction strategies. This makes robustness not merely a communication-layer property but an important end-to-end characteristic of meaning-driven language communication.

Chapter 6

Multimodal Semantic Communications for Text, Vision, Speech, and Generative Information

6.1 Vision-Based Semantic Communication and Meaning-Aware Image Transmission

6.1.1 Visual Feature Extraction and Semantic Representation

Visual feature extraction is a fundamental stage in vision-based semantic communication because raw images contain a large amount of information that may not be equally important for the intended communication task. Conventional image transmission generally attempts to preserve the visual signal with high reconstruction fidelity, whereas semantic communication focuses on identifying and representing the visual information that contributes most strongly to the intended meaning or application. This shift requires the communication system to transform an image from its original pixel-level representation into a structured representation containing relevant visual features, objects, relationships, and contextual information.

The visual feature extraction process typically begins with an input image that is processed by a vision-oriented neural architecture. Convolutional neural networks, vision Transformers, or hybrid architectures can learn hierarchical representations from the visual input. Early processing stages may capture relatively local characteristics such as edges, textures, colors, and simple spatial patterns, while deeper stages can represent higher-level structures such as objects, regions, scenes, and their relationships. The resulting features provide a more meaningful representation of the image than the original collection of pixels. Depending on the communication objective, the system can retain only the features that are relevant to the target application.

Semantic representation is subsequently constructed from the extracted visual features. Instead of preserving every visual detail, the representation can emphasize concepts that determine the interpretation of the scene. For example, an image containing vehicles, pedestrians, buildings, and roads can be represented according to the objects and spatial relationships that are important for a traffic-monitoring application. A different application may require a different semantic representation from the same image. This illustrates the task-dependent nature of visual semantic communication and the importance of aligning feature extraction with the intended communication objective.

The resulting visual semantic representation can then be compressed and transmitted through the communication system. Because irrelevant or redundant visual information may be removed before transmission, the approach can potentially reduce communication requirements while maintaining information needed for interpretation or downstream tasks. At the receiver, the semantic representation can be decoded and used either to reconstruct visual content or directly support an application such as image classification, object detection, scene understanding, or decision making. Thus, visual feature extraction establishes the bridge between raw image data and meaning-oriented communication, enabling the system to prioritize semantically important visual information rather than treating every pixel as equally important.

6.1.2 Semantic Image Encoding and Meaning-Preserving Compression

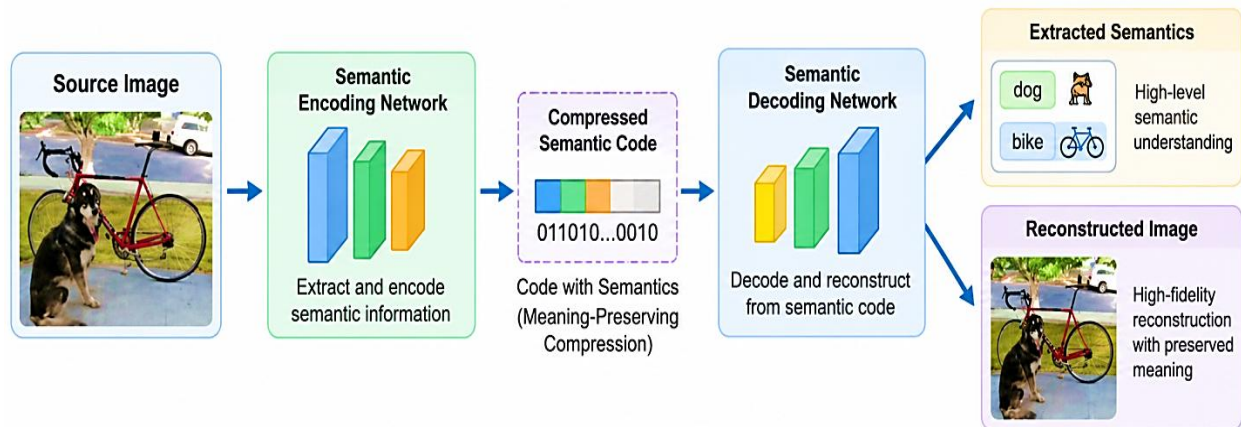


Figure 34: Semantic Image Encoding and Meaning-Preserving Compression Framework

A semantic image communication pipeline that transforms a source image into a compact representation focused on its meaningful visual information. The process begins with the source image, which contains both relevant semantic content and potentially redundant visual details. The image is provided to a semantic encoding network that extracts and encodes information associated with important visual concepts. In the example shown, the image contains recognizable objects such as a dog and a bicycle, demonstrating that the system can focus on high-level visual semantics rather than treating every pixel as equally important.

The semantic encoding network converts the extracted visual information into a compressed semantic code. This stage represents the central distinction between conventional image compression and semantic image communication. Conventional approaches primarily attempt to reduce the size of the visual representation while maintaining acceptable image quality, whereas semantic compression seeks to retain information that is important for interpreting the image or completing a downstream task. The compressed semantic code therefore acts as a meaning-oriented representation that can potentially reduce the amount of information required for transmission.

At the receiver, the compressed semantic code is processed by the semantic decoding network. The figure shows two possible forms of output. The first is extracted semantics, where high-level concepts such as the objects represented in the source image are recovered for semantic understanding. The second is a reconstructed image, in which visual content is generated or recovered while retaining the important meaning of the original scene. These two outputs demonstrate that semantic communication does not necessarily require a single form of reconstruction; the appropriate output can depend on the requirements of the receiving application.

6.1.3 Task-Oriented Image Transmission and Reconstruction

A task-oriented semantic image transmission and reconstruction framework consisting of a front-end transmitter and a back-end receiver. At the transmitter, the input image and current channel state are provided to the processing system. The channel state is represented through channel gain information, indicating that the transmission strategy can take communication conditions into account. This is

important in task-oriented semantic communication because the amount and type of information transmitted can be adapted according to the available communication resources and the requirements of the receiving application. The policy network combines information about the input communication condition with the channel state to determine an adaptive transmission policy. The resulting policy guides the semantic extraction process, which identifies relevant semantic features from the image. These features are then organized into a compressed semantic domain, producing a compact semantic code for transmission. This architecture demonstrates that semantic compression does not have to be fixed; instead, the representation can be adapted according to the communication environment and the desired task.

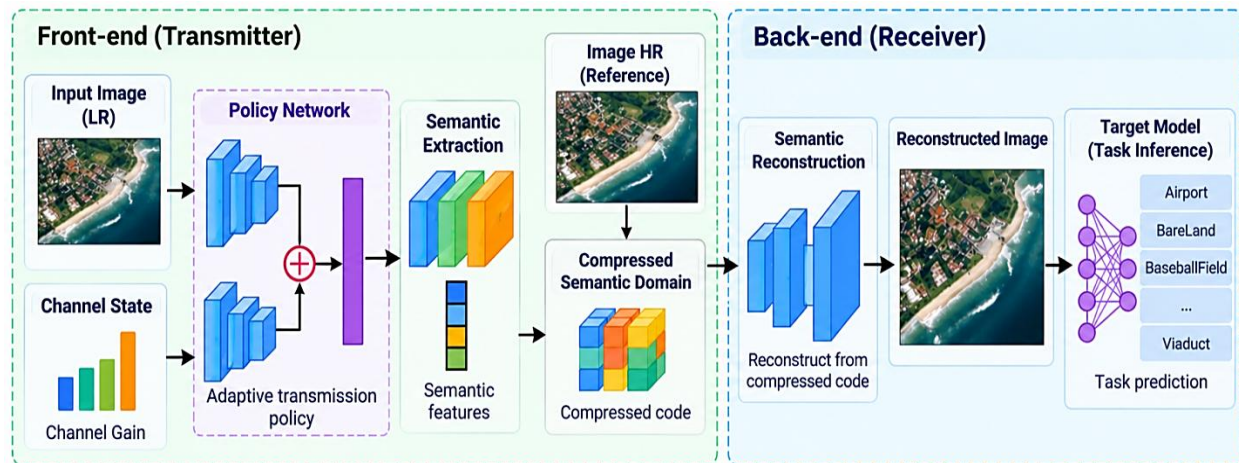


Figure 35: Task-Oriented Semantic Image Transmission and Reconstruction Framework

At the receiver, the transmitted compressed semantic code is processed by the semantic reconstruction module. The reconstruction stage converts the compressed representation into information from which the image can be recovered. The resulting reconstructed image is then provided to the target model, which performs task inference. In the figure, the target model produces task-related predictions such as different scene or land-use categories. This establishes a direct connection between semantic image transmission and the final application rather than evaluating communication solely according to image reconstruction quality.

6.1.4 Robust Semantic Vision Communication under Network Constraints

Robust semantic vision communication focuses on maintaining useful visual meaning when images are transmitted through networks with limited bandwidth, variable channel quality, latency constraints, interference, or unreliable connectivity. Unlike conventional image transmission, where degradation is often assessed through pixel-level reconstruction quality, semantic vision systems should determine whether important visual information remains available for interpretation and downstream tasks. This distinction is important because a reconstructed image may contain visible distortions while still preserving the objects, relationships, and scene characteristics required by the application. Robustness therefore requires the semantic communication system to prioritize information that remains valuable under changing network conditions.

A robust semantic vision architecture can adapt the representation transmitted according to the available network resources. When bandwidth is limited, the system can prioritize task-relevant visual features and remove redundant information before transmission. Under relatively favorable conditions, additional semantic features or visual details can be transmitted when they provide meaningful improvements to reconstruction or task performance. Channel-aware semantic encoding can therefore establish a flexible relationship between network conditions and the amount of information transmitted. Such adaptation is particularly relevant for mobile, edge, Internet of Things, and wireless vision applications where network characteristics can change over time.

Robustness also depends on how the semantic representation responds to information loss and communication errors. A useful semantic representation should retain important concepts even when some transmitted features are corrupted or unavailable. Redundant or complementary semantic features can provide additional resilience, while learned decoding mechanisms can use contextual information to recover useful representations from incomplete transmissions. However, reconstruction mechanisms should avoid introducing unsupported visual content when information has been lost. The objective is therefore not simply to generate a visually plausible image, but to preserve the semantic information necessary for the intended application.

Evaluation of robustness should consider both communication performance and application-level effectiveness. Experiments can examine semantic fidelity, object or scene recognition accuracy, task completion performance, transmitted information volume, latency, and computational requirements under different network conditions. Performance can be measured across progressively degraded bandwidth or channel-quality conditions to determine how gracefully the system degrades. A robust semantic vision system should ideally maintain useful task performance even when conventional image quality deteriorates. This makes robustness an end-to-end property linking visual representation, semantic compression, adaptive transmission, channel conditions, reconstruction, and application requirements. Ultimately, robust semantic vision communication seeks to ensure that the most important visual meaning remains available even when network resources are constrained or communication conditions become unfavorable.

6.2 Speech, Audio, and Acoustic Semantic Communication

6.2.1 Speech Representation and Semantic Feature Extraction

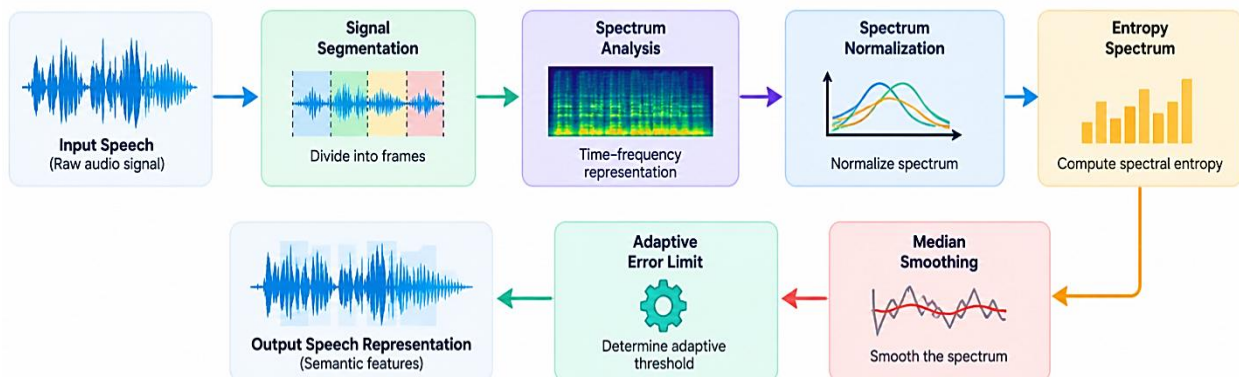


Figure 36: Speech Signal Processing and Semantic Feature Extraction Pipeline

Processing pipeline for converting a raw speech signal into a representation containing meaningful acoustic and semantic characteristics. The process begins with the input speech, represented as a raw audio waveform. Because continuous speech contains variations over time, the signal is first divided into smaller frames through signal segmentation. Frame-based processing makes it possible to analyze local characteristics of speech and capture changes in acoustic patterns that occur during different portions of an utterance.

After segmentation, the system performs spectrum analysis to obtain a time–frequency representation of the speech signal. This representation provides information about how the frequency components of speech vary over time and can expose characteristics associated with different sounds, phonetic structures, and acoustic events. The figure then applies spectrum normalization, which helps place spectral characteristics into a more consistent range and can reduce the influence of variations that are less important for semantic interpretation. Normalization can therefore support more stable downstream feature extraction across different speech signals and recording conditions.

The next stage calculates an entropy spectrum, providing a measure related to the distribution and uncertainty of spectral information across the analyzed speech signal. The resulting spectral characteristics are processed through median smoothing, which reduces irregular fluctuations and helps produce a more stable representation. An adaptive error limit is subsequently used to determine an appropriate threshold for extracting reliable speech characteristics. This adaptive processing is useful because speech signals can vary considerably in intensity, acoustic structure, speaker characteristics, and environmental conditions.

Finally, the processing stages produce an output speech representation containing semantic features. Rather than retaining every detail of the original waveform, the representation emphasizes characteristics that can contribute to meaningful speech interpretation and subsequent semantic communication. This supports the central principle of speech-based semantic communication: transmit the information necessary to preserve the intended meaning while reducing unnecessary acoustic redundancy.

6.2.2 Meaning-Aware Speech Encoding and Transmission

Meaning-aware speech encoding and transmission focuses on extracting and communicating the information that is most relevant to the intended meaning of a spoken message rather than transmitting the complete acoustic waveform. In conventional speech communication, the encoder attempts to preserve a detailed representation of the input signal so that the receiver can reproduce it with high signal-level fidelity. Semantic communication adopts a different objective by identifying linguistic, acoustic, and contextual characteristics that are important for understanding the message. This allows the communication system to reduce unnecessary information while maintaining the semantic content required by the receiving application or user.

The encoding process generally begins with speech preprocessing and feature extraction. Acoustic characteristics such as spectral patterns, temporal structures, phonetic information, and speech activity can be transformed into a compact semantic representation. For example, a speech encoder may identify important linguistic units, speaker-related characteristics, emotional cues, or task-relevant acoustic events depending on the communication objective. Instead of treating every sample of the waveform as equally

important, the encoder assigns greater importance to features that contribute directly to meaning. This meaning-aware representation can then be compressed before transmission, reducing the amount of information that needs to pass through the communication channel.

Transmission of the semantic representation must also consider the characteristics of the communication environment. Channel quality, available bandwidth, latency requirements, and transmission reliability can influence how much semantic information should be transmitted. Under favorable channel conditions, a richer representation may be communicated to preserve additional speech details. Under constrained conditions, the system can prioritize essential semantic features and discard information that has limited impact on the final interpretation. Such adaptive behavior enables semantic speech communication to maintain useful information even when conventional signal-level transmission would experience substantial degradation.

At the receiver, the transmitted semantic representation is decoded and mapped into an interpretable speech representation or directly into task-relevant information. Depending on the application, the receiver may reconstruct speech, recover linguistic content, identify an instruction, or perform another downstream task without reproducing the original waveform exactly. Consequently, the effectiveness of meaning-aware speech transmission should be evaluated using semantic fidelity, recognition accuracy, task performance, communication efficiency, and robustness rather than relying exclusively on conventional waveform-based measures. This approach makes speech communication more suitable for bandwidth-constrained and intelligent communication systems in which preserving useful meaning is more important than reproducing every signal detail.

6.2.3 Contextual Audio Understanding and Semantic Reconstruction

Contextual audio understanding extends semantic speech communication by considering information surrounding the speech signal when interpreting its meaning. The same spoken expression can have different interpretations depending on the conversation, speaker intention, environment, task, and previously exchanged information. Therefore, a semantic communication system should not process speech as an isolated acoustic sequence. Instead, it should integrate contextual information to determine which aspects of the audio are important and how the received representation should be interpreted or reconstructed.

The contextual understanding process can combine linguistic information with acoustic and environmental cues. Linguistic context may include preceding and succeeding utterances, keywords, dialogue history, and inferred intent. Acoustic context can include speaker characteristics, prosody, emphasis, pauses, background sounds, and other temporal properties. Environmental context may provide information about the surrounding situation, such as traffic noise, machinery, public environments, or other sound sources. Combining these sources enables the system to distinguish information that is essential for the communication task from information that is incidental or redundant.

At the receiver, contextual information can support semantic reconstruction even when the transmitted speech representation is incomplete or degraded. Rather than attempting to recover every missing acoustic detail directly, the decoder can use the available semantic features together with contextual knowledge to infer the most likely intended content. For example, if some speech information is lost during

transmission, the receiver may use previously established conversational context and task requirements to recover the intended instruction or message. This approach is particularly valuable in semantic communication because successful reconstruction is defined by preservation of meaning rather than exact reproduction of the original signal.

Contextual reconstruction can also support task-oriented applications in which the final objective is not necessarily to reproduce speech. A receiver may use recovered semantic information for speech recognition, command execution, event detection, dialogue management, or information retrieval. In such cases, the reconstruction stage can produce the representation required by the downstream task instead of generating a complete audio waveform. The quality of contextual audio understanding can therefore be assessed by examining whether the intended meaning, user intent, and task-relevant information are preserved. This establishes a direct connection between contextual interpretation and semantic communication, allowing the system to use available context to improve reliability, reduce unnecessary transmission, and maintain useful performance under incomplete or distorted communication conditions.

6.2.4 Robust Speech Semantic Communication in Dynamic Environments

Robust speech semantic communication addresses the challenges that arise when speech is transmitted through communication environments whose characteristics change over time. Real-world speech communication may encounter fluctuating channel quality, interference, background noise, mobility, changing bandwidth, variable latency, and differences in speaker or environmental conditions. These factors can affect both the extraction of semantic information and its successful delivery to the receiver. A robust semantic communication system therefore needs to maintain meaningful information even when the communication conditions are unstable.

An important characteristic of robust speech semantic communication is its ability to distinguish between degradation that affects the acoustic signal and degradation that actually changes the intended meaning. A noisy transmission may alter waveform characteristics substantially while leaving the spoken message understandable. Conversely, a small distortion affecting an important keyword or instruction can have a much larger semantic impact. Therefore, robustness mechanisms should prioritize protection of semantically important speech features. Important words, intent-related information, critical commands, and task-specific acoustic cues can receive greater protection or transmission priority than less relevant details.

Adaptation is also essential in dynamic environments. The semantic encoder can adjust the amount and type of information transmitted according to current channel conditions and application requirements. When bandwidth is limited or channel quality deteriorates, the system can reduce semantic redundancy and transmit only the most important information. When conditions improve, additional semantic or acoustic details can be transmitted to enhance reconstruction quality. At the receiver, adaptive decoding can similarly adjust its reconstruction strategy according to the reliability and completeness of the received representation. This creates a communication process that responds continuously to changing environmental and network conditions.

Robustness can be further improved through contextual information, redundancy management, error-tolerant representations, and learning-based adaptation. Context can help the receiver infer missing or

corrupted semantic information, while selective redundancy can provide additional protection for information that is critical to the task. Learning-based systems can also adapt to variations in speakers, acoustic environments, network conditions, and application requirements. The overall performance should consequently be evaluated using semantic preservation, speech understanding, task accuracy, latency, bandwidth efficiency, and reliability. The primary goal is not perfect recovery of the original waveform under every condition, but sustained preservation of the information necessary for successful interpretation and task completion. This makes robust semantic speech communication particularly relevant to mobile, edge, IoT, vehicular, emergency, and other dynamic communication scenarios.

6.3 Cross-Modal Semantic Fusion and Multimodal Knowledge Representation

6.3.1 Joint Representation of Text, Image, Audio, and Video Information

A framework for obtaining a joint semantic representation from text, image, audio, and video information. The process begins with four different information modalities. Text provides linguistic features, images provide visual features, audio provides acoustic features, and video contributes temporal and visual information. Because these modalities have different structures and characteristics, each is processed through a modality-specific encoder. The text encoder transforms linguistic information into text semantics, while the vision, audio, and video encoders generate corresponding semantic representations for their respective modalities.

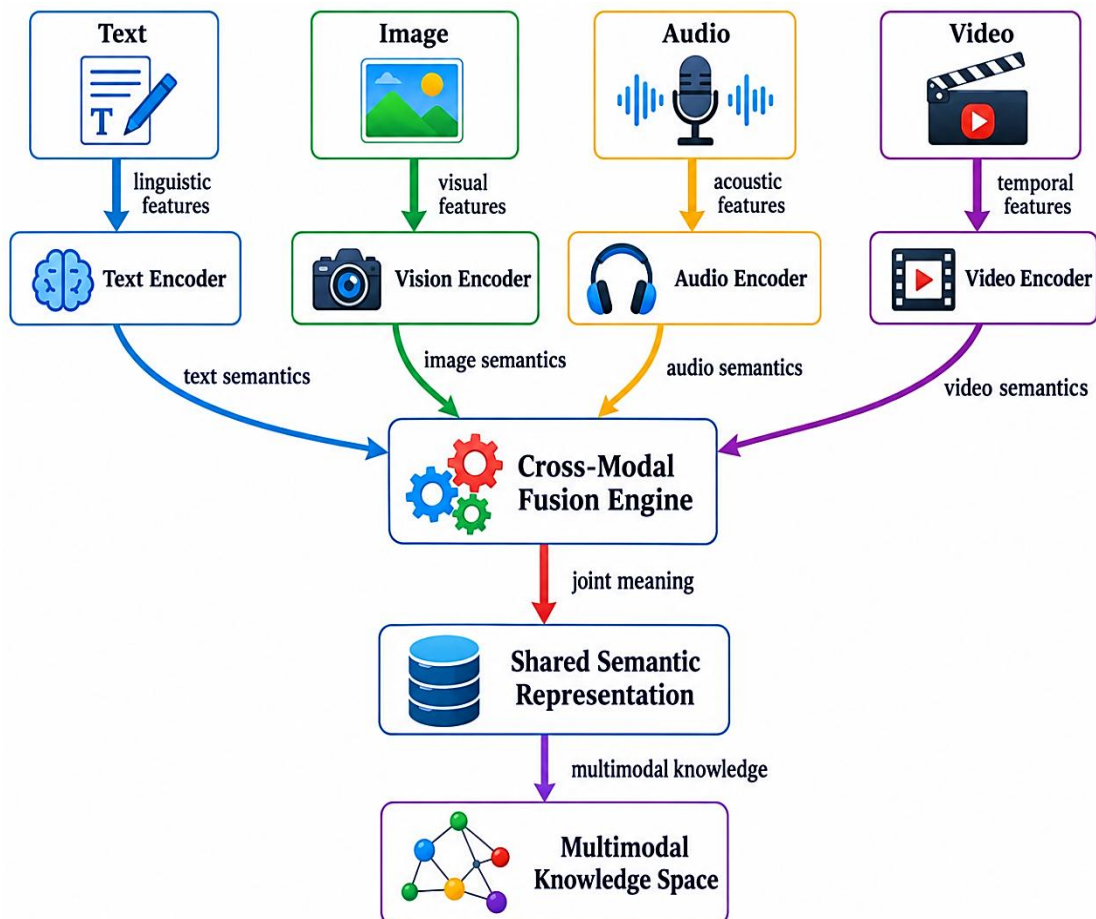


Figure 37: Joint Semantic Representation of Text, Image, Audio, and Video Information

The semantic outputs from the four modality-specific encoders are subsequently provided to a cross-modal fusion engine. This component combines information from the different modalities and identifies relationships between their semantic representations. For example, an object identified in an image can be associated with a textual description, while corresponding audio or video information can provide additional contextual or temporal evidence. The fusion process therefore moves beyond independent processing of each modality and creates a unified interpretation that captures complementary information across modalities.

The fused information is then organized into a shared semantic representation, which provides a common space for representing the meaning derived from multiple modalities. Such a shared representation can support semantic alignment between heterogeneous information sources and reduce the difficulty of communicating or processing each modality independently. Instead of treating text, image, audio, and video as completely separate information streams, the system can represent their common meaning and their relationships within an integrated semantic structure.

6.3.2 Cross-Attention-Based Multimodal Semantic Fusion

Cross-attention-based multimodal semantic fusion provides a mechanism for integrating information from different modalities by allowing one modality to selectively attend to semantically relevant information in another. Unlike simple feature concatenation, cross-attention can identify relationships between heterogeneous representations and assign different importance to individual features. This is particularly useful in multimodal semantic communication because text, images, audio, and video contain complementary information that may describe the same event, object, or situation from different perspectives. The objective is to construct a unified semantic representation that retains the most informative cross-modal relationships while avoiding unnecessary redundancy.

In a typical cross-attention mechanism, one modality provides the queries, while another modality supplies the keys and values. For example, textual representations can act as queries that attend to relevant visual features, allowing the system to identify image regions associated with particular words or concepts. Similarly, visual representations can attend to textual descriptions to identify the language information that is most relevant to the observed scene. Audio and video can also be connected through cross-attention to associate acoustic events with corresponding temporal visual information. This selective interaction enables the system to capture relationships that may not be apparent when each modality is processed independently.

Cross-attention can be incorporated at different stages of a semantic communication architecture. Early fusion can establish relationships between relatively low-level modality features, whereas deeper fusion can operate on more abstract semantic representations. Attention weights can be dynamically determined according to the relevance of each modality to the current communication task. Consequently, a system designed for image captioning may emphasize visual–text relationships, while an event-detection application may place greater emphasis on audio–video interactions. This task-dependent behavior allows multimodal semantic representations to remain focused on the information required by the application.

The resulting fused representation can subsequently be compressed and transmitted as a semantic payload rather than transmitting complete independent representations for every modality. At the receiver, the

shared representation can be decoded and used for reconstruction, multimodal reasoning, or direct task execution. The effectiveness of cross-attention-based fusion should therefore be evaluated according to semantic fidelity, cross-modal consistency, communication efficiency, and downstream task performance. By selectively integrating complementary information across modalities, cross-attention provides an effective foundation for meaning-aware multimodal communication in which the relationships between modalities are considered as important as the individual modality representations.

6.3.3 Multimodal Context and Intent Understanding

Multimodal context and intent understanding focuses on determining the meaning and purpose of information by jointly considering evidence from multiple modalities. Human communication rarely depends on a single information source. A spoken instruction may be accompanied by a visual scene, a gesture, background sounds, or temporal events in a video. Similarly, a text description can provide contextual information that changes how an image or audio signal should be interpreted. Multimodal semantic communication therefore needs mechanisms that can combine heterogeneous contextual information to determine what the user intends to communicate and which information is important for the target task.

The first stage involves collecting contextual information from the available modalities. Text can provide explicit linguistic meaning, while images can describe objects, locations, and spatial relationships. Audio can provide speech, environmental sounds, or acoustic events, whereas video can provide temporal information about actions and changes in the environment. These sources can be processed independently and subsequently integrated through multimodal representation learning. The system can then identify complementary evidence, resolve ambiguity, and construct a richer interpretation than would be possible from any individual modality.

Intent understanding is particularly important for task-oriented semantic communication. The system should distinguish between the information contained in a multimodal input and the purpose for which that information is being communicated. For example, an image accompanied by the instruction identify the vehicle requires different semantic prioritization from the same image accompanied by determine whether the road is safe. In the first case, vehicle-related visual features may receive greater importance, whereas the second task may require information about road conditions, obstacles, traffic, and environmental context. Multimodal intent modeling therefore provides a basis for selecting semantic information according to the requirements of the receiving application.

Context can also assist the receiver when semantic information is incomplete or degraded during transmission. Previously exchanged information, task requirements, environmental conditions, and information from other modalities can provide additional evidence for interpreting partially received representations. This can improve semantic reconstruction without requiring exact recovery of every original signal component. Multimodal context and intent understanding consequently connects semantic representation with communication objectives by determining what information should be preserved, transmitted, reconstructed, and ultimately used. Its effectiveness can be assessed through intent recognition accuracy, multimodal understanding performance, semantic fidelity, task completion, and robustness under incomplete or noisy multimodal inputs.

6.3.4 Cross-Modal Semantic Consistency and Alignment

Cross-modal semantic consistency and alignment address the problem of ensuring that representations generated from different modalities describe compatible meanings. Text, images, audio, and video may represent the same underlying event or concept, but their features exist in different representation spaces and may emphasize different aspects of the information. Without appropriate alignment, multimodal systems can produce inconsistent interpretations or fail to exploit complementary information. Semantic alignment therefore seeks to establish meaningful relationships between representations while preserving modality-specific characteristics.

A central objective of cross-modal alignment is to bring semantically related representations closer within a shared semantic space. For example, an image containing a bicycle and a textual description referring to a bicycle should produce representations that exhibit a strong semantic relationship. Similarly, a video showing an object together with an audio description of that object should maintain compatibility between the visual and acoustic representations. Alignment can be learned using paired or weakly paired multimodal data, where corresponding examples provide information about which representations should be considered semantically related.

Semantic consistency is broader than simple similarity. Two modalities may contain different levels of detail while still communicating the same underlying meaning. A textual description may identify an event explicitly, whereas an image may provide visual evidence without directly expressing the event in language. Therefore, alignment mechanisms should account for relationships, concepts, attributes, temporal dependencies, and contextual information rather than requiring identical representations. Contrastive learning, shared embedding spaces, cross-attention, and multimodal projection mechanisms can be used to establish these relationships while allowing each modality to retain useful modality-specific information.

Maintaining cross-modal consistency becomes particularly important during semantic transmission and reconstruction. If only selected semantic information is transmitted, the receiver should be able to interpret the recovered representation consistently across the available modalities. For example, a reconstructed image should not contradict the semantic information obtained from an associated text or audio representation. Consistency can consequently be evaluated by examining cross-modal similarity, semantic agreement, retrieval performance, alignment accuracy, and downstream task performance. In semantic communication, these measures can be complemented by evaluating whether aligned representations preserve the intended meaning after compression, transmission, and reconstruction. Cross-modal semantic consistency thus provides an important foundation for reliable multimodal communication, enabling different information sources to contribute complementary evidence while maintaining a coherent interpretation of the communicated content.

6.4 Generative and Multimodal Foundation Models for Semantic Communication

6.4.1 Multimodal Generative Models for Semantic Reconstruction

Multimodal generative models provide a powerful approach to semantic reconstruction by combining information from multiple modalities and generating missing or degraded content according to the available semantic representation. In conventional communication systems, reconstruction generally attempts to reproduce the transmitted signal as accurately as possible. In semantic communication,

however, the receiver can focus on recovering the underlying meaning and generating an appropriate representation of that meaning. Multimodal generative models are particularly suitable for this purpose because they can integrate textual, visual, audio, and video information and use relationships among these modalities to support reconstruction.

A multimodal generative reconstruction system typically receives a compact semantic representation rather than the complete original information. The representation may contain concepts, entities, relationships, attributes, contextual information, or task-specific features. A generative model then interprets these features and produces the required output. For example, transmitted semantic information describing an object, its surroundings, and its relationships with other objects can be used to reconstruct a visual scene. Similarly, semantic information extracted from speech can be combined with contextual information to generate an appropriate textual or audio representation. The reconstruction process therefore depends on the semantic information available at the receiver and the learned relationships encoded within the generative model.

The ability to combine modalities is especially useful when one modality is incomplete or degraded. Text can provide constraints for visual reconstruction, while visual information can provide evidence for interpreting ambiguous language. Audio and video can similarly complement each other when reconstructing events or activities. This cross-modal interaction can improve the completeness and contextual relevance of reconstructed information. However, the generative model must distinguish between information that is supported by the received semantics and information that is merely plausible. Generating visually or linguistically realistic content does not necessarily guarantee preservation of the intended meaning.

For semantic communication, multimodal generative reconstruction should therefore be evaluated using semantic fidelity, cross-modal consistency, reconstruction quality, and downstream task performance. The objective is not necessarily to reproduce the exact original signal but to generate an output that remains consistent with the transmitted meaning and supports the intended application. Such models can consequently reduce communication requirements by transmitting compact semantic information while allowing the receiver to generate richer representations. This establishes multimodal generative modeling as an important component of future semantic communication systems, particularly for applications involving incomplete information, heterogeneous data, and task-oriented reconstruction.

6.4.2 Diffusion Models for Visual and Multimodal Information Generation

Diffusion models provide a generative framework for reconstructing and generating complex information through an iterative denoising process. Their application to semantic communication is particularly relevant when the transmitted information is compressed, incomplete, noisy, or represented in a latent semantic space. Instead of requiring the communication system to transmit every detail of an image, video, or other multimodal signal, a semantic representation can provide the information needed to guide generation at the receiver. The diffusion model can then progressively transform a noisy or incomplete representation into a coherent output that reflects the transmitted semantics.

In visual semantic communication, diffusion models can be conditioned on semantic features, object information, scene descriptions, latent representations, or other task-relevant information. The receiver

begins with a noisy representation and performs successive denoising operations to obtain a reconstructed visual output. Because the generation process is conditioned on semantic information, the resulting image can preserve important objects, spatial relationships, and scene characteristics even when detailed pixel information was not transmitted. This provides a natural connection between semantic compression and generative reconstruction, where communication focuses on essential information and generation supplies appropriate visual details at the receiver.

Diffusion models can also be extended to multimodal generation. Textual information can guide image generation, visual information can support language generation, and combinations of audio, video, and textual semantics can provide richer conditioning information. Such multimodal conditioning is useful when different sources provide complementary evidence about the same underlying event. For instance, a semantic description of an object can constrain visual reconstruction, while temporal information from video can help maintain consistency across generated frames. The resulting system can therefore use multiple semantic sources to improve the relevance and coherence of generated information.

Despite these advantages, diffusion-based semantic communication introduces important challenges. Iterative generation can require considerable computational resources and may introduce latency at the receiver. More importantly, a generated output can appear realistic while containing details that were not supported by the transmitted information. Therefore, diffusion-based reconstruction must be controlled by reliable semantic representations and evaluated according to semantic fidelity rather than visual realism alone. Metrics should consider meaning preservation, task accuracy, cross-modal consistency, computational cost, and communication efficiency. Properly integrated with semantic encoding and task-oriented transmission, diffusion models can provide a flexible mechanism for generating high-quality visual and multimodal information from compact semantic representations.

6.4.3 Generative Completion of Missing Semantic Information

Generative completion of missing semantic information addresses situations in which part of the information required for interpretation is lost, compressed excessively, or unavailable during communication. In practical communication environments, packet loss, channel degradation, limited bandwidth, sensor failures, and incomplete source data can prevent the receiver from obtaining a complete semantic representation. Rather than treating missing information as an unrecoverable failure, generative models can use the available semantic features and contextual knowledge to infer information that is consistent with the received content. This creates an important link between semantic communication and intelligent information reconstruction.

The completion process begins with the semantic information that successfully reaches the receiver. This information may include partial descriptions, detected entities, relationships, visual features, linguistic concepts, or task-specific representations. A generative model uses these available elements together with learned patterns and contextual information to estimate missing components. For example, if an image contains missing regions but the receiver knows the scene category and the surrounding objects, a generative model may reconstruct a visually coherent representation of the missing region. Similarly, if part of a textual or spoken message is unavailable, contextual information can help determine a suitable semantic completion.

An important distinction must be maintained between semantic completion and unrestricted content generation. The purpose of completion is to recover information that is compatible with the intended meaning, not simply to generate the most likely output according to general training data. The model should therefore be constrained by the semantic information that was actually transmitted. Context, task requirements, multimodal evidence, and previously established knowledge can provide additional constraints. When the available evidence is insufficient, the system should avoid treating uncertain generated content as confirmed information, particularly in applications where incorrect semantic inference could affect task decisions.

Generative semantic completion can improve communication efficiency because the transmitter does not necessarily need to send information that the receiver can reliably infer from existing context. However, this benefit introduces a trade-off between communication reduction and reconstruction uncertainty. If excessive information is omitted, the receiver may generate content that is plausible but semantically incorrect. Evaluation should therefore examine completion accuracy, semantic consistency, task performance, uncertainty, and the amount of information saved through transmission. Effective systems should preserve essential semantic information while allowing generative models to reconstruct secondary or redundant details. In this way, generative completion can transform missing information from a communication limitation into an opportunity for intelligent, context-aware reconstruction.

6.4.4 Semantic Consistency in Multimodal Generative Communication

Semantic consistency is a fundamental requirement for multimodal generative communication because generated outputs must remain compatible with the meaning conveyed by the transmitted information and with information available across different modalities. A generative model may produce visually realistic images, fluent language, natural speech, or coherent video, but these properties alone do not guarantee that the generated content correctly represents the intended message. Semantic consistency therefore requires the generated information to preserve important concepts, entities, relationships, attributes, and task-relevant information established by the communication process.

In a multimodal system, consistency must be maintained both within individual modalities and across modalities. For example, if transmitted semantics indicate that an image contains a particular object, the reconstructed image should preserve that object and its relevant characteristics. If a textual description accompanies the image, the generated image should remain compatible with the description. Similarly, audio describing an event should not contradict the corresponding visual or textual representation. Cross-modal consistency can therefore help ensure that the receiver develops a unified interpretation rather than producing independent outputs that disagree with one another.

Maintaining semantic consistency becomes more challenging when generative models reconstruct information that was not explicitly transmitted. Generative systems learn broad statistical patterns from training data and may introduce details that are plausible but unsupported by the received semantic representation. This problem is especially important when communication is highly compressed. The fewer constraints provided to the generative model, the greater the potential for ambiguity in reconstruction. Semantic communication systems should therefore distinguish between essential transmitted meaning and optional generated details. Conditioning mechanisms, contextual knowledge,

cross-modal alignment, and task-specific constraints can be used to reduce semantic inconsistencies during generation.

Evaluation should consequently extend beyond conventional perceptual or signal-quality measures. Semantic similarity can determine whether the reconstructed content remains aligned with the intended meaning, while cross-modal consistency can examine whether different generated modalities agree. Task-level accuracy can further determine whether the generated information supports the intended application. Communication efficiency, reconstruction latency, and computational cost are also relevant when evaluating practical deployment. Ultimately, a successful multimodal generative communication system should generate information that is not merely realistic but semantically faithful, contextually appropriate, and consistent across modalities. This principle ensures that generative intelligence complements semantic communication rather than introducing unintended meaning, making semantic consistency a central requirement for reliable multimodal foundation-model-based communication systems.

Chapter 7

Wireless Semantic Communications and Meaning-Aware Radio Networks

7.1 End-to-End Architecture of Wireless Semantic Communication Systems

7.1.1 Wireless Semantic Transmitter and Intelligent Encoding

The wireless semantic transmitter represents the first major processing stage in an end-to-end semantic communication system. Its primary objective is to transform source information into a compact representation that preserves the meaning required by the receiving system or target application. Unlike conventional wireless transmitters, which generally focus on converting source data into bits and protecting those bits against channel errors, a semantic transmitter determines which aspects of the source are important from a meaning-oriented perspective. This makes the transmitter an intelligent component that connects source information, semantic understanding, and wireless resource constraints.

The process normally begins with source data such as text, speech, images, video, sensor observations, or multimodal information. A semantic feature extraction mechanism identifies relevant concepts, entities, relationships, attributes, and contextual information. Depending on the application, deep neural networks, transformer models, semantic embeddings, knowledge graphs, or multimodal foundation models can be used to obtain these representations. The extracted semantic information is then transformed into a compact latent representation. Information that has limited relevance to the communication objective can be reduced or discarded, allowing the transmitter to avoid spending wireless resources on unnecessary details. Intelligent encoding can also incorporate information about the communication task and the current wireless environment. For example, a system supporting object detection may prioritize object-related semantic features, whereas a speech-command application may prioritize intent and command-related information. Similarly, changing channel conditions can influence the amount of semantic information selected for transmission. Under limited bandwidth or poor channel quality, the transmitter may emphasize highly important semantic features and reduce secondary information. Under favorable conditions, additional information can be transmitted to improve reconstruction or task performance.

The resulting semantic representation is subsequently mapped into a wireless transmission format suitable for the communication channel. This stage can involve semantic compression, feature quantization, channel-aware adaptation, and resource-aware encoding. The effectiveness of the transmitter should therefore be evaluated not only through compression efficiency but also through semantic fidelity, wireless resource utilization, reliability, latency, and downstream task performance. The fundamental role of the intelligent semantic transmitter is to ensure that the information entering the wireless channel is compact, relevant, and aligned with the intended meaning and application objective.

7.1.2 Semantic-Aware Wireless Channel Modeling

Semantic-aware wireless channel modeling extends conventional channel modeling by considering not only the physical characteristics of signal transmission but also the effect of channel conditions on the meaning carried by the transmitted information. Traditional wireless channel models describe phenomena such as attenuation, fading, interference, multipath propagation, and noise. In semantic communication,

these physical effects remain important, but their ultimate significance depends on how they affect the semantic representation and the performance of the receiving task. Consequently, channel modeling must establish a connection between wireless impairments and semantic degradation.

A semantic-aware channel can be viewed as an environment in which a transmitted semantic representation is transformed by physical communication effects before reaching the receiver. Changes in signal-to-noise conditions, interference, bandwidth availability, fading, and mobility can alter the reliability of individual semantic features. However, not all feature errors have equal importance. Losing a redundant visual feature may have little effect on interpretation, whereas corruption of an important object identifier or task-critical instruction can significantly reduce application performance. This creates a distinction between conventional signal-level channel quality and semantic-level channel impact.

Semantic-aware modeling can therefore incorporate measures that describe the relationship between channel conditions and semantic performance. A wireless system may characterize how different levels of noise, fading, or interference affect semantic similarity, reconstruction fidelity, intent recognition, or task accuracy. Such models can support adaptive transmission decisions by identifying which semantic information should receive greater protection under difficult channel conditions. The transmitter and receiver can consequently respond to the semantic importance of information rather than treating every transmitted component equally.

Another important aspect is the dynamic nature of wireless environments. Mobile users, changing network loads, variable interference, and fluctuating channel conditions can cause the effective semantic communication environment to change continuously. A semantic-aware model should therefore support adaptation rather than assuming a fixed channel state. Learning-based channel models can potentially capture relationships between wireless conditions and semantic performance and provide useful information for adaptive encoding, modulation, resource allocation, and decoding.

Ultimately, semantic-aware wireless channel modeling provides a bridge between physical-layer communication conditions and application-level meaning preservation. Its purpose is not to replace established wireless channel models but to complement them with semantic information about the consequences of transmission impairments. This perspective enables wireless semantic communication systems to optimize communication according to meaningful outcomes, allowing limited radio resources to be directed toward information whose successful delivery has the greatest impact on the intended task.

7.1.3 Intelligent Semantic Receiver and Meaning Reconstruction

The intelligent semantic receiver is responsible for converting the information received through the wireless channel into a meaningful representation that can support reconstruction, interpretation, or direct execution of the target task. In conventional wireless communication, the receiver typically attempts to recover the transmitted bits as accurately as possible. In semantic communication, the receiver has a broader responsibility because successful communication is determined by whether the intended meaning can be recovered despite possible information loss, noise, compression, or channel distortion.

The receiving process begins with the recovery of the transmitted semantic representation from the wireless signal. A semantic decoder then interprets the received representation using learned models,

contextual information, and, where appropriate, shared semantic knowledge. The decoder does not necessarily need to reproduce every component of the original source. Instead, it attempts to reconstruct the information that is relevant to the communication objective. For image communication, this may involve recovering important objects and scene relationships; for speech, it may involve recovering linguistic content or user intent; and for sensor or multimodal applications, it may involve producing task-relevant information.

Context and prior knowledge can play an important role in meaning reconstruction. When part of a semantic representation has been degraded during transmission, the receiver can use contextual relationships and previously available information to interpret incomplete representations. Generative models can further assist reconstruction by producing missing details that are consistent with the received semantics. However, reconstruction should remain constrained by the information actually supported by the transmitted representation to reduce the possibility of generating plausible but incorrect content.

The intelligent receiver can also operate in a task-oriented manner by bypassing complete source reconstruction when it is unnecessary. If the communication objective is classification, detection, decision-making, or control, the received semantic representation can be directly provided to a task model. This can reduce reconstruction requirements and computational overhead while focusing the receiver on the actual purpose of communication. The receiver can therefore support both meaning reconstruction and task-specific semantic interpretation, depending on application requirements.

The final performance of an intelligent semantic receiver should be evaluated using semantic fidelity, meaning preservation, task accuracy, robustness, latency, and computational efficiency. A receiver that produces a visually or acoustically high-quality output but fails the intended task should not be considered successful. The central objective is to recover sufficiently accurate and useful meaning from the received information. Thus, the intelligent semantic receiver completes the end-to-end wireless semantic communication process by transforming a potentially degraded wireless representation into reconstructed meaning and actionable information for the target application.

7.1.4 End-to-End Semantic Transmission Architecture

The figure illustrates an end-to-end semantic communication architecture in which the communication process begins with an information source containing different types of content, including text, images, audio, and video. The source information is first passed to the semantic processing stage, where the raw information is analyzed to identify and represent its relevant meaning. Instead of directly transmitting the complete source information, the system produces a semantic payload that captures the information required for meaningful communication. This represents the fundamental distinction between conventional bit-oriented transmission and semantic communication, where the objective is to preserve useful meaning rather than necessarily reproduce every source element. The semantic payload is subsequently provided to the wireless transmitter, which encodes the semantic representation into a signal suitable for transmission over the wireless medium. The encoded signal travels through a meaning-aware wireless channel that may introduce noise, fading, and interference. The figure emphasizes that these channel impairments can distort the received signal before it reaches the wireless receiver. The receiver therefore performs semantic decoding to recover the semantic representation from the potentially degraded signal. This stage is important because semantic communication does not simply focus on

recovering an identical sequence of transmitted bits; instead, it attempts to preserve the information that is important for the intended interpretation or application.

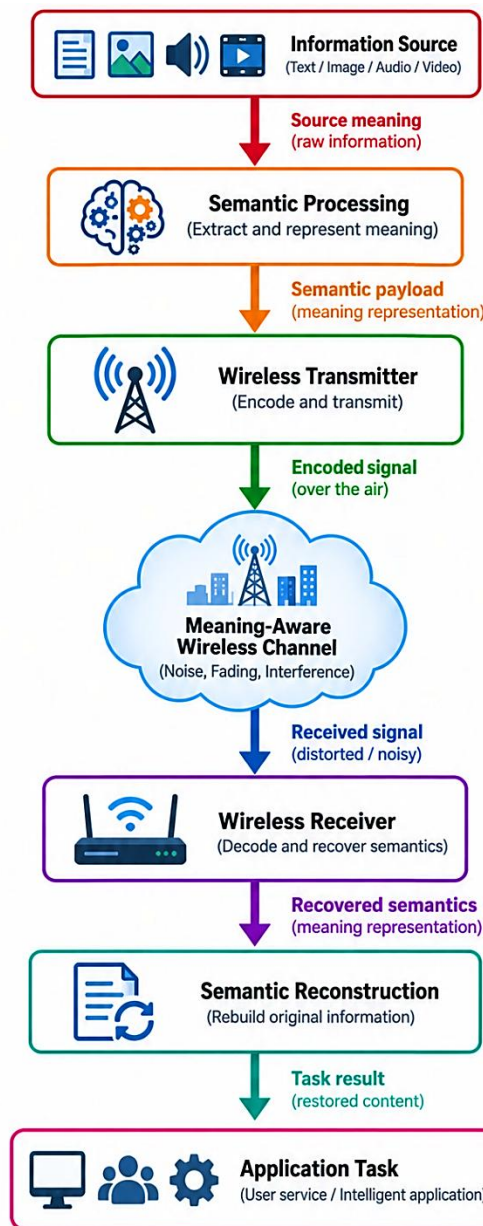


Figure 38: End-to-End Semantic Transmission Architecture for Meaning-Aware Wireless Communication

Following semantic recovery, the reconstructed semantic information is passed to the semantic reconstruction stage, where the original information or an appropriate representation of it is rebuilt. The reconstruction does not necessarily need to reproduce every low-level detail of the original source when those details are irrelevant to the target application. Instead, the reconstruction process aims to retain the essential meaning required by the communication objective. This makes the reconstruction stage closely

connected with semantic fidelity and meaning preservation, particularly when the wireless channel has caused partial information loss.

7.2 Robust Semantic Communication under Wireless Channel Impairments

7.2.1 Semantic Communication over Noisy Wireless Channels

Wireless semantic communication must operate reliably even when the transmitted information is affected by noise. In a conventional communication system, noise primarily causes errors in the received bit sequence, and the communication system attempts to recover the original bits using channel coding and decoding techniques. In semantic communication, however, the effect of noise must also be considered in terms of its influence on the recovered meaning. A small change in the transmitted representation may have little impact on the intended meaning, whereas another small perturbation may remove an important semantic feature. Therefore, robustness against noise requires the semantic encoder, communication channel, and semantic decoder to work together to preserve information that is important for the intended task.

A semantic transmitter typically transforms the source information into a compact semantic representation before wireless transmission. This representation can contain linguistic features, visual features, contextual information, task-relevant features, or learned latent representations. When the representation passes through a noisy channel, the received signal may differ from the transmitted semantic representation. The semantic decoder must therefore infer the intended information from an imperfect observation rather than simply reproduce the received signal. Deep neural networks can learn this relationship during training by exposing the semantic communication model to different levels and types of channel noise. Noise-aware training is particularly important for maintaining stable semantic performance under changing communication conditions. During training, artificial channel impairments can be introduced so that the encoder and decoder learn representations that remain informative after transmission. The model can learn to place greater emphasis on robust semantic features and reduce its dependence on fragile details that are easily corrupted. This approach allows semantic communication systems to tolerate moderate signal degradation while maintaining useful information for downstream applications.

The effectiveness of noisy-channel semantic communication should therefore be evaluated using measures that reflect both communication reliability and semantic usefulness. Semantic similarity, semantic fidelity, task accuracy, reconstruction quality, and information reduction can be considered together. A system that experiences some signal-level degradation but preserves the correct task result may be more useful than a system that achieves accurate signal reconstruction but loses important meaning. Consequently, robustness to noise represents a central requirement for practical wireless semantic communication, particularly in mobile, IoT, autonomous, and resource-constrained environments where channel quality can vary continuously.

7.2.2 Fading, Interference, and Multipath-Aware Semantic Transmission

Wireless channels are rarely affected by noise alone. Fading, interference, and multipath propagation can significantly modify the transmitted signal before it reaches the receiver. Fading can cause variations in received signal strength because of changes in propagation conditions, distance, obstacles, and mobility. Multipath propagation occurs when transmitted signals reach the receiver through multiple paths,

producing constructive and destructive combinations. Interference can additionally arise from other transmitters operating within the same spectral or spatial region. For semantic communication, these impairments are important because they can selectively damage semantic features contained within the transmitted representation.

A multipath-aware semantic communication system should therefore consider the characteristics of the wireless environment during semantic encoding and transmission. Instead of producing a representation independently of channel conditions, the semantic encoder can adapt its representation according to the expected reliability of the communication link. Important semantic features may be given greater protection, while less important information can be represented more compactly or transmitted with reduced priority. Such adaptive behavior enables the communication system to allocate its available resources according to semantic importance rather than treating every transmitted component equally.

Channel-aware learning provides another mechanism for handling fading and interference. During model training, the semantic communication architecture can be exposed to different channel conditions so that the encoder and decoder learn to operate across a range of propagation environments. The learned model can develop representations that remain distinguishable after channel distortion. Channel-state information can also be incorporated into transmission decisions, allowing the system to modify semantic compression, feature selection, transmission power, coding strategy, or representation size according to the current wireless state.

The receiver plays an equally important role in multipath- and interference-aware semantic transmission. Rather than depending exclusively on exact recovery of the received signal, a semantic decoder can use learned relationships, contextual information, and prior knowledge to infer the most likely meaning represented by the corrupted observation. This can provide additional resilience when some semantic features are partially degraded. The overall objective is to maintain semantic fidelity and task performance despite variations in channel quality. Such robustness is particularly valuable in high-mobility networks, wireless sensor systems, autonomous vehicles, aerial communication platforms, and dense wireless deployments, where fading, interference, and multipath effects can change rapidly.

7.2.3 Packet Loss and Semantic Information Recovery

Packet loss represents another important challenge for wireless semantic communication, particularly in networks where transmission reliability, congestion, mobility, or weak channel conditions can cause portions of the transmitted information to disappear. In conventional systems, packet loss can prevent complete recovery of the original data unless retransmission or error-control mechanisms are applied. Semantic communication provides an alternative perspective by distinguishing between information that is essential for understanding and information that is less important. Consequently, the loss of a packet does not necessarily have to result in complete communication failure if the remaining semantic information is sufficient for reconstructing the intended meaning.

Semantic representations can be designed so that important information is distributed across compact features rather than being dependent on a single low-level component. When some semantic information is lost, the receiver can exploit the surviving representation, contextual information, learned priors, or task requirements to infer missing content. This enables semantic information recovery even when the

received representation is incomplete. The recovery process can range from simple reconstruction of missing features to more advanced neural or generative approaches that infer information based on the available semantic context.

Importance-aware transmission can further reduce the effect of packet loss. Semantic features associated with critical objects, entities, relationships, instructions, or task requirements can receive higher transmission priority or stronger protection. Less important information can be transmitted with lower priority or omitted when network resources are limited. This creates a graceful degradation mechanism in which communication quality decreases according to semantic importance rather than experiencing an abrupt loss of usefulness when individual packets are lost.

The effectiveness of semantic recovery under packet loss should be evaluated according to the final communication objective. Reconstruction similarity alone may not adequately describe system performance because a reconstructed message can differ from the original while still preserving its intended meaning. Task accuracy, semantic fidelity, meaning preservation, and recovery consistency can therefore provide more meaningful evaluation criteria. In applications such as remote sensing, intelligent transportation, industrial monitoring, and autonomous systems, the ability to recover task-relevant information from incomplete transmissions can substantially improve communication efficiency. Packet loss thus becomes not only a reliability problem but also a semantic inference problem in which the receiver attempts to recover the most useful meaning from the information that remains available.

7.2.4 Robust Semantic Reconstruction under Unreliable Channels

Robust semantic reconstruction is the final stage through which a semantic communication system converts imperfectly received information into a meaningful representation or task output. In an unreliable wireless environment, the semantic decoder may receive representations affected by noise, fading, interference, packet loss, or other transmission impairments. The reconstruction process must therefore tolerate incomplete and distorted information while maintaining the semantic properties required by the target application. The objective is not necessarily to reproduce every detail of the original source but to recover the information that is relevant to the intended meaning and task.

Neural semantic decoders can support this process by learning relationships between corrupted representations and their corresponding semantic outputs. During training, the model can be exposed to a range of communication conditions so that it learns to reconstruct meaningful information from imperfect inputs. Context and prior knowledge can further improve reconstruction because the decoder can use relationships among entities, features, or concepts to compensate for missing information. This is particularly useful when the received semantic representation contains sufficient contextual evidence to infer information that was degraded during transmission.

Robust reconstruction can also incorporate task-oriented objectives. For example, an autonomous system may not require a visually perfect reconstruction of an image if the primary purpose of communication is to identify a vehicle, detect an obstacle, or determine a road condition. Similarly, a language-based application may prioritize preservation of an instruction or intent over exact reproduction of every word. Designing reconstruction around the target task allows the system to concentrate its computational and communication resources on information that has the greatest practical value.

Evaluation should consequently consider both reconstruction quality and semantic usefulness. Semantic fidelity measures whether the intended meaning has been preserved, while task-level measures determine whether the reconstructed information supports the required application successfully. Robust systems should maintain acceptable performance across different channel conditions rather than achieving high quality only under ideal transmission. This makes robustness an important design objective for practical semantic communication. By combining channel-aware encoding, resilient semantic representations, contextual inference, and task-oriented reconstruction, semantic communication systems can continue to provide useful information even when wireless channels are unreliable.

7.3 Semantic-Aware Radio Resource Management and Network Scheduling

7.3.1 Semantic-Aware Bandwidth and Spectrum Allocation

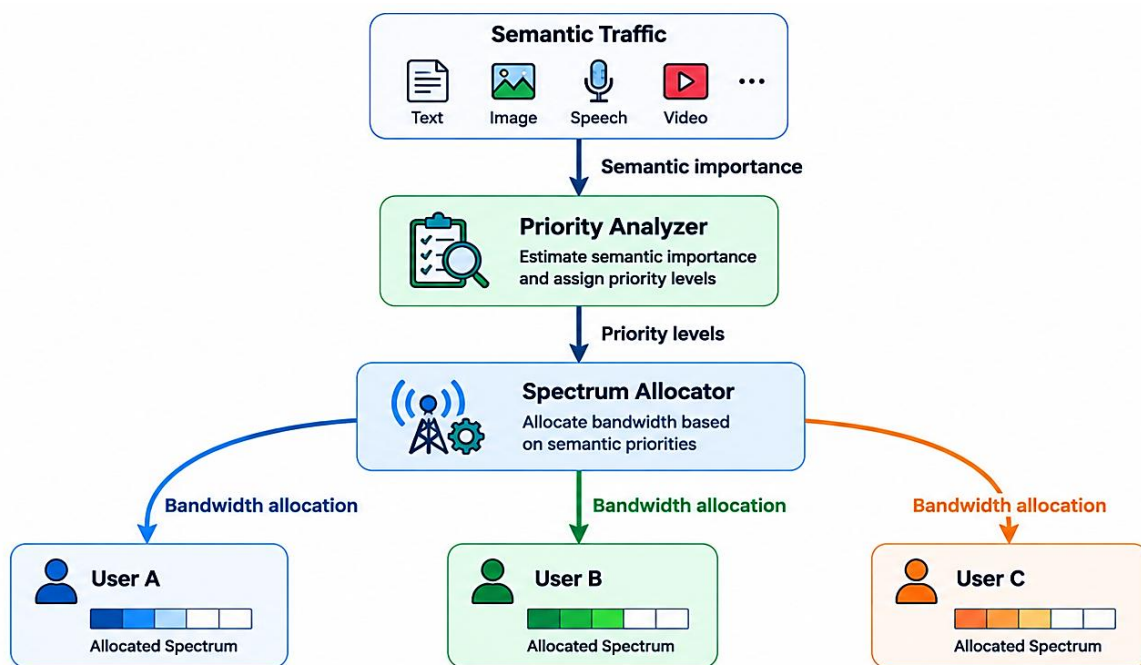


Figure 39: Semantic-Aware Bandwidth and Spectrum Allocation Framework

Semantic-aware radio resource management framework in which different types of semantic traffic, including text, images, speech, and video, enter the communication system as heterogeneous information sources. Instead of assigning communication resources solely according to the amount of data generated by each source, the framework first evaluates the semantic importance of the information. This is particularly relevant to semantic communication because not every piece of transmitted information contributes equally to the final communication objective. Information that is highly relevant to a user's task or application can receive greater priority than information with relatively low semantic significance.

The central component of the framework is the Priority Analyzer, which estimates the semantic importance of incoming information and assigns corresponding priority levels. This stage represents an important difference between conventional resource allocation and semantic-aware allocation. In a conventional wireless network, bandwidth may be distributed using parameters such as channel quality,

data rate, queue length, or service requirements. In the proposed semantic-oriented approach, these factors can be complemented by the value of the information itself. For example, an urgent message, a critical visual object, or speech containing task-relevant instructions may deserve higher transmission priority than less important background information.

Following priority estimation, the Spectrum Allocator determines how available wireless spectrum or bandwidth should be distributed among different users. The figure shows three users, User A, User B, and User C, receiving different portions of the available spectrum. This unequal allocation visually represents the principle that communication resources can be dynamically assigned according to semantic priority. A user carrying highly important semantic information can receive a larger or more reliable resource allocation, while information with lower semantic urgency can operate with fewer resources. Such an approach can improve spectrum utilization while maintaining the information most important to the application.

The overall framework demonstrates the connection between semantic importance and radio-resource allocation. The process begins with heterogeneous semantic traffic, evaluates its importance, and translates the resulting priorities into bandwidth allocation decisions. This makes the radio resource management process more adaptive and application-oriented. In practical wireless semantic communication systems, the allocation decision could additionally consider channel quality, latency requirements, reliability, user requirements, and available spectrum. Thus, the figure provides a clear conceptual foundation for explaining how semantic intelligence can be incorporated into bandwidth and spectrum management to achieve more efficient and meaning-aware wireless communication.

7.3.2 Power Allocation for Energy-Efficient Semantic Transmission

Power allocation is an important component of wireless semantic communication because the transmission of semantic information must be achieved within limited energy budgets. Conventional wireless systems often allocate transmit power according to channel quality, data rate requirements, or user fairness. In semantic communication, power allocation can additionally consider the importance and usefulness of the information being transmitted. This creates an opportunity to provide more transmission power to semantic information that has a greater influence on the final communication task while reducing the energy expenditure associated with less important information. Such an approach is particularly valuable for battery-powered devices, wireless sensors, Internet of Things networks, and mobile terminals.

A semantic-aware power allocation mechanism can operate by first identifying the importance of individual semantic features, messages, or information streams. Highly important semantic features may require stronger protection because their loss can significantly affect semantic fidelity or task accuracy. Conversely, information with relatively low semantic importance may tolerate greater distortion and can therefore be transmitted using lower power. The allocation process can also incorporate channel conditions, since a weak wireless link may require additional power to achieve an acceptable probability of successful semantic recovery. Thus, power allocation becomes a joint decision involving semantic importance and physical-layer transmission conditions.

The relationship between transmit power and semantic performance can be expressed conceptually through an optimization objective such as

$$\max \{P_i\} U_s(\{P_i\})$$

subject to an energy constraint

$$\sum_i P_i \leq P_{\max}$$

where (P_i) represents the power allocated to semantic stream (i) , $(P_{\max})$ is the available power budget, and (U_s) denotes semantic utility. The semantic utility may incorporate semantic fidelity, task accuracy, reconstruction quality, or other application-specific measures. This formulation emphasizes that the objective is not simply to maximize received signal quality but to maximize the usefulness of transmitted information under limited energy.

Energy-efficient semantic transmission can further benefit from adaptive power control. When channel conditions are favorable, the system may reduce transmission power while maintaining adequate semantic quality. During severe channel degradation, power can instead be concentrated on critical semantic information. Combined with semantic compression and priority-aware scheduling, this strategy can reduce unnecessary energy consumption while preserving task-relevant information. Consequently, semantic-aware power allocation provides a pathway toward sustainable and intelligent wireless communication in which energy resources are distributed according to both communication conditions and the value of the information being transmitted.

7.3.3 Semantic Priority-Based Scheduling and Resource Allocation

Semantic priority-based scheduling extends conventional wireless scheduling by considering the importance of information in addition to conventional network parameters. In traditional communication networks, scheduling decisions are commonly influenced by queue length, channel conditions, throughput requirements, latency, or fairness among users. However, these criteria do not explicitly distinguish between information that is highly valuable to an application and information that has limited impact on the final task. Semantic communication introduces the possibility of assigning priority according to the contribution of information to the intended meaning or application objective. This allows network resources to be directed toward the information that provides the greatest practical value.

A semantic scheduler can receive information from a semantic analysis module that estimates the importance of different messages, features, or data streams. Each semantic unit can then be associated with a priority level reflecting factors such as task relevance, urgency, semantic significance, and sensitivity to information loss. High-priority semantic information can be scheduled earlier, assigned more bandwidth, or provided stronger transmission protection. Lower-priority information can be delayed, compressed more aggressively, or transmitted when additional network resources become available. This approach supports a form of graceful degradation in which network congestion affects less important information before critical semantic content.

The scheduling problem can be formulated using a semantic utility function. For example, a resource allocation objective can be represented as

$$\max \sum_i w_i U_i(x_i)$$

where (x_i) represents the communication resources assigned to semantic stream (i) , (U_i) represents the resulting semantic utility, and (w_i) denotes the semantic priority weight. The optimization can additionally include constraints related to bandwidth, transmit power, latency, reliability, and computational resources. The priority weights do not necessarily remain fixed; they can be updated according to user intent, task requirements, network conditions, and changes in the semantic relevance of transmitted information.

Semantic priority-based scheduling is particularly useful in multi-user wireless environments where communication resources are scarce and different applications have different requirements. For instance, an urgent control message may require immediate transmission, while background multimedia information can tolerate delay. Similarly, in autonomous or industrial systems, information directly related to safety or control can receive higher priority than redundant environmental observations. By integrating semantic priority with channel state, queue status, latency requirements, and resource availability, the scheduler can make more intelligent decisions than purely data-centric scheduling methods. This results in communication networks that are not only resource-efficient but also meaning-aware, task-oriented, and adaptive, supporting reliable delivery of the information that matters most.

7.3.4 Joint Communication and Computation Resource Optimization

Joint communication and computation resource optimization is an important approach for semantic communication systems because semantic processing and wireless transmission are strongly interconnected. The provided figure illustrates a framework in which a semantic workload and its task requirements are supplied to a joint optimizer. The semantic workload may involve text, images, audio, video, or multimodal information, while the associated requirements can include latency, accuracy, priority, and energy considerations. Instead of optimizing communication and computation independently, the framework coordinates both resource types to improve the overall quality of the semantic task.

On the communication side, the joint optimizer determines suitable spectrum and transmit-power decisions. Bandwidth allocation influences the amount of semantic information that can be transmitted and therefore affects communication quality. Transmit-power allocation supports reliable wireless transmission while also influencing energy consumption. These decisions can be adjusted according to the semantic workload and channel conditions so that communication resources are concentrated where they provide the greatest benefit to the semantic task. This approach is particularly relevant when wireless resources are limited and different semantic workloads have different priorities.

The framework also considers computation resources, including edge CPUs and AI accelerators. Semantic communication systems often require computationally intensive operations such as semantic feature extraction, representation learning, encoding, decoding, reconstruction, and inference. The optimizer can determine how computational workloads should be distributed across available resources. Edge CPU allocation can influence processing latency, while AI accelerator allocation can improve the

efficiency of neural semantic processing and inference. Consequently, communication decisions and computation decisions should be coordinated because increasing communication resources may reduce transmission time while increasing computational resources may reduce processing or inference time.

The final objective of the framework is to improve semantic task quality rather than optimizing an individual communication or computing parameter in isolation. The figure connects bandwidth, transmit power, edge computation, and AI acceleration to communication quality, energy impact, processing latency, and inference efficiency, which collectively determine the resulting semantic task quality. This joint perspective enables a semantic communication system to adapt its resource allocation according to application requirements. By coordinating radio and computational resources, the system can seek an appropriate balance among accuracy, reliability, latency, and energy efficiency, providing a more integrated foundation for intelligent and resource-aware semantic communication networks.

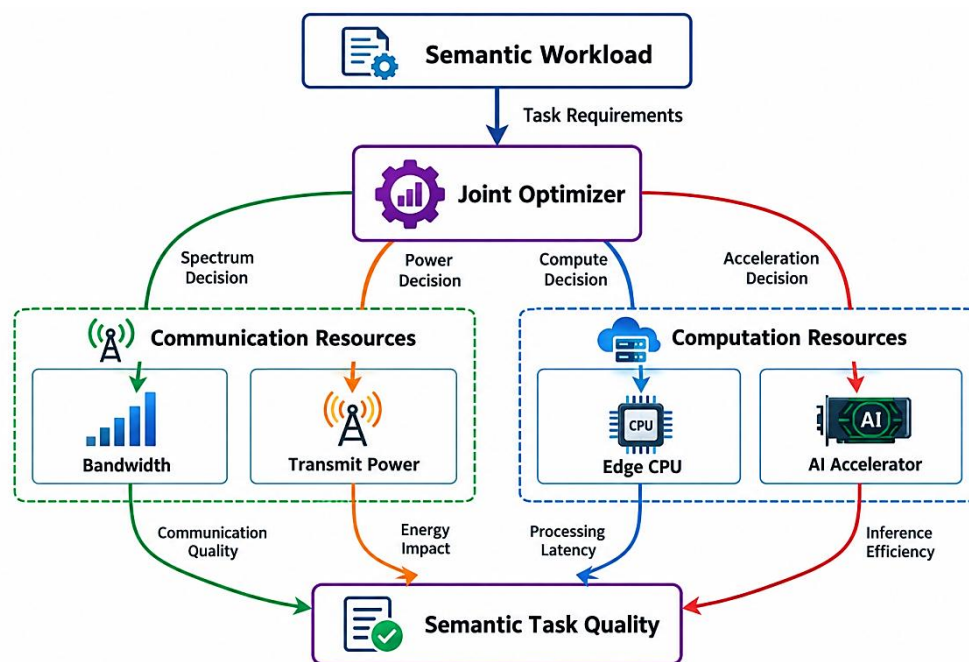


Figure 40: Joint Communication and Computation Resource Optimization for Semantic Communication

7.4 Adaptive Wireless Semantic Networking for Dynamic Environments

7.4.1 Channel-Aware Semantic Adaptation

Wireless communication environments are inherently dynamic because channel conditions can change as a result of user mobility, distance variation, fading, interference, obstacles, traffic fluctuations, and changing network loads. In conventional communication systems, adaptation is generally performed by modifying physical-layer parameters such as modulation, coding, transmission power, or data rate. Semantic communication introduces an additional adaptation dimension in which the semantic representation itself can be modified according to the current channel condition. This enables the system to preserve important meaning even when the available communication resources or channel quality change significantly.

Channel-aware semantic adaptation begins with the estimation of relevant wireless conditions, such as received signal quality, channel gain, interference level, available bandwidth, and link reliability. This information can be supplied to the semantic encoder or an adaptive control mechanism. When the channel is reliable, the system can transmit richer semantic representations with greater detail. When the channel becomes degraded, the encoder can increase semantic compression, remove less important features, reduce representation complexity, or prioritize information that is essential for the target task. Consequently, adaptation does not simply attempt to maintain a fixed data rate; it seeks to maintain useful semantic information under changing communication conditions.

An important advantage of this approach is the possibility of graceful semantic degradation. If channel quality deteriorates, the system does not necessarily need to stop communication or reproduce the complete source information. Instead, it can preserve the most important semantic elements while sacrificing information that has a lower contribution to the application. For example, a vision-based system may prioritize objects, boundaries, or events relevant to a particular task while reducing the transmission of visually redundant background information. Similarly, a language-based system can prioritize intent and task-related concepts over less important linguistic details.

Channel-aware adaptation can therefore establish a continuous relationship between wireless conditions and semantic processing decisions. The encoder, transmission mechanism, and decoder can jointly adjust their behavior according to changing network states. Such an adaptive strategy is particularly suitable for mobile devices, autonomous systems, wireless sensor networks, aerial platforms, and other environments where channel conditions cannot be assumed to remain stable. By incorporating channel information directly into semantic decision-making, wireless semantic networks can achieve more efficient and reliable communication while maintaining the meaning most relevant to the application.

7.4.2 Semantic Quality-Aware Transmission Control

Semantic quality-aware transmission control focuses on maintaining the usefulness and meaning of transmitted information rather than relying exclusively on conventional signal-level performance indicators. In wireless semantic communication, a transmission can experience some degree of physical-layer degradation without causing a significant loss in application performance. Conversely, a relatively small degradation in a critical semantic feature can have a substantial effect on the final task. Transmission control should therefore consider the relationship between communication conditions and the resulting semantic quality.

A semantic quality-aware controller can monitor measures such as semantic fidelity, meaning preservation, reconstruction quality, task accuracy, or application-specific performance. These measures provide feedback regarding whether the current transmission strategy is delivering sufficient semantic information. If semantic quality falls below an acceptable level, the system can modify transmission parameters or semantic processing strategies. Possible adaptations include changing the semantic compression level, allocating additional bandwidth, increasing protection for important features, adjusting transmission power, or selecting a more robust representation. In this way, transmission control becomes feedback-driven and oriented toward the final communication objective.

The approach can also distinguish between semantic information according to its importance. Critical information can receive stronger protection or greater communication resources, whereas information with limited influence on the application can tolerate higher distortion or compression. This allows the system to make more efficient use of constrained wireless resources. Rather than attempting to provide identical quality for every transmitted component, semantic quality-aware control concentrates resources on information that contributes most strongly to the desired task.

Feedback is particularly important in dynamic environments because semantic quality can change as channel conditions, user requirements, and application objectives change. A closed-loop architecture can observe the quality of the reconstructed semantics or task result and communicate this information back to the transmission controller. The controller can then adjust subsequent transmissions accordingly. Such a mechanism allows semantic communication to operate adaptively rather than following a fixed transmission policy. Overall, semantic quality-aware transmission control provides a bridge between network-level resource management and application-level performance. It recognizes that the ultimate purpose of communication is to deliver useful meaning rather than merely accurate signals or bit sequences. By continuously relating transmission decisions to semantic quality, the network can improve resource efficiency, maintain task performance, and respond intelligently to variations in wireless conditions.

7.4.3 Latency- and Reliability-Aware Semantic Networking

Latency and reliability are critical requirements in many wireless semantic communication applications, particularly when information is used for real-time decision-making. Applications such as autonomous systems, industrial monitoring, intelligent transportation, remote control, and emergency response may require information to arrive within a limited time while maintaining sufficient semantic accuracy. Conventional communication approaches often attempt to achieve reliability through retransmissions and stronger error-control mechanisms. However, excessive retransmission can increase latency and consume valuable wireless resources. Semantic networking provides an opportunity to balance these requirements according to the importance of the transmitted information.

Latency-aware semantic networking considers the time required for semantic processing, transmission, decoding, reconstruction, and task execution. Information associated with urgent tasks can be assigned higher scheduling priority and transmitted using communication strategies designed for rapid delivery. Less time-sensitive information can tolerate greater delay or be transmitted when resources become available. Semantic compression can also reduce transmission time by removing redundant information before communication. At the same time, computational complexity must be considered because sophisticated semantic models can improve representation quality but may introduce additional encoding and decoding delay.

Reliability-aware networking similarly focuses on preserving the semantic information required by the application. Instead of treating every bit or packet as equally important, the network can identify critical semantic components and provide them with stronger protection. Important features may receive additional coding protection, greater transmission resources, or higher scheduling priority. Less important information may be transmitted with lower protection when resources are limited. This approach allows

reliability to be defined in terms of successful semantic delivery rather than only successful packet reception.

Latency and reliability can sometimes create competing requirements. Increasing protection or retransmissions may improve reliability but increase delay, while aggressive compression or reduced protection may decrease latency but increase the risk of semantic degradation. An adaptive semantic networking system must therefore select appropriate operating conditions according to task requirements. The resulting decisions can incorporate channel state, semantic priority, application deadlines, available bandwidth, computational resources, and acceptable semantic distortion. The overall goal is to ensure that information reaches the intended application quickly enough and reliably enough to support the required task. This task-oriented perspective allows semantic networking to avoid unnecessary resource expenditure while protecting critical information. By jointly considering latency, reliability, semantic importance, and network conditions, adaptive wireless semantic networks can provide more responsive and dependable communication for dynamic real-time applications.

7.4.4 Adaptive Semantic Policies for Mobile Communication Environments

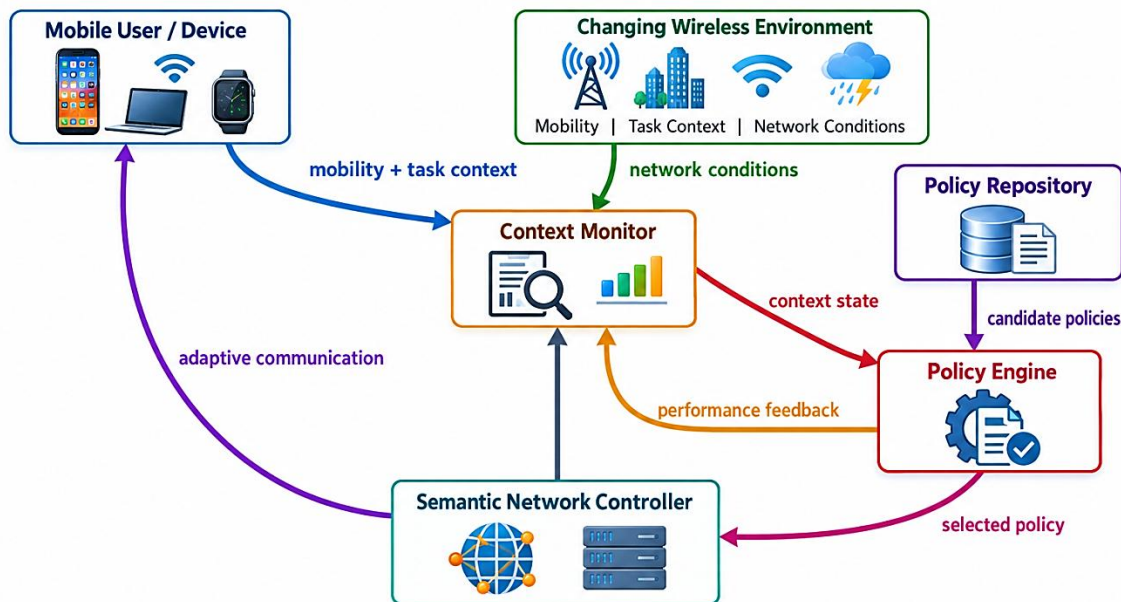


Figure 41: Adaptive Semantic Policies for Mobile Communication Environments

The figure presents an adaptive semantic policy framework designed for mobile communication environments in which both user requirements and wireless conditions can change continuously. At the upper left, the Mobile User/Device represents devices such as smartphones, laptops, or wearable systems. At the same time, the Changing Wireless Environment represents variations associated with mobility, task context, and network conditions. These two sources provide information to the Context Monitor, which observes the current operating situation and creates an updated context state for adaptive communication decisions.

The Policy Repository provides candidate communication policies to the Policy Engine. Based on the context state received from the context monitor, the policy engine selects an appropriate policy for the

current situation. Such policies can determine how semantic information should be compressed, prioritized, transmitted, or adapted according to the current requirements. The figure therefore emphasizes that semantic communication policies should not remain fixed when the operating environment changes. Instead, policy selection can respond to variations in mobility, task requirements, and wireless network conditions.

The selected policy is then delivered to the Semantic Network Controller, which applies the required communication adaptations. The controller can coordinate network behavior so that semantic information is transmitted according to the selected policy. The figure also shows a performance feedback path from the policy engine toward the context-monitoring process. This feedback allows the system to evaluate its communication behavior and use the observed performance to support subsequent policy decisions. Therefore, the framework represents an adaptive closed-loop mechanism rather than a one-time configuration. Monitoring, policy selection, network control, and performance feedback can work together to support reliable semantic communication for mobile users. As mobility and network conditions change, the system can select or modify suitable communication policies instead of relying on a static configuration. This makes the framework particularly suitable for dynamic wireless scenarios where channel conditions, application requirements, and user context may vary over time.

Chapter 8

Semantic Networking in 5G, 6G, and AI-Native Communication Infrastructures

8.1 Integration of Semantic Communication with 5G Network Architectures

8.1.1 Semantic Communication Integration into 5G Infrastructure

The figure illustrates a conceptual framework for integrating semantic communication into a 5G network architecture. On the left side, the Data block represents different types of information that can enter the communication system, including images, speech, text, video, and IoT or sensor data. Rather than treating all these inputs only as conventional bit streams, semantic communication introduces semantic encoding to extract information that is relevant to the intended meaning or application. The semantic encoding stage therefore provides an interface between heterogeneous source data and the 5G communication infrastructure.



Figure 42: Integration of Semantic Communication with 5G Network Infrastructure and Applications

The central portion of the figure represents the 5G Network Infrastructure, which provides the connectivity and computing capabilities required for semantic communication. The architecture includes a 5G base station or gNB, edge computing, multi-access edge computing (MEC), the core network, and network slicing. These components can support semantic processing closer to users and applications, potentially reducing unnecessary data transmission and enabling faster responses. Edge computing is particularly important because semantic extraction, representation, and task-related processing can be performed closer to the data source rather than relying entirely on centralized processing.

The arrows labelled Semantic Encoding and Semantic Transmission emphasize the movement from source information toward meaning-aware communication and subsequently toward application services. Network slicing can provide differentiated network resources for applications with different semantic communication requirements, while MEC can support computationally intensive semantic processing at

the network edge. Consequently, the figure presents 5G not simply as a connectivity layer but as an infrastructure capable of supporting intelligent and application-aware semantic communication services.

On the right side, the figure shows representative 5G Applications, including smart cities, connected vehicles, smart manufacturing, e-health and telemedicine, AR/VR and metaverse services, and smart agriculture. These applications demonstrate why semantic communication can be valuable in 5G environments: different applications have different information priorities, latency requirements, and task objectives. For example, connected vehicles may prioritize safety-critical information, whereas AR/VR applications may emphasize timely visual information. Thus, the figure highlights a complete conceptual path from heterogeneous data, through semantic encoding and 5G infrastructure, to AI-native 5G services and application-specific communication.

8.1.2 Semantic-Aware Network Services and Applications

Semantic-aware network services extend conventional 5G communication by allowing network functions and applications to consider the meaning, relevance, context, and task requirements of transmitted information. In a traditional communication system, network services generally focus on metrics such as throughput, delay, packet loss, and connectivity. Semantic communication introduces an additional layer in which information is evaluated according to its usefulness for a particular application or task. Consequently, network services can prioritize information that has greater semantic importance while reducing the transmission of redundant or less relevant content. This approach is particularly useful in heterogeneous 5G environments where users, devices, sensors, and applications continuously generate large volumes of data.

A semantic-aware network can provide application-specific communication services by adapting transmission strategies according to the requirements of individual tasks. For example, a smart-city application may require important traffic, environmental, or emergency information to be communicated with high reliability, while routine sensor measurements can tolerate greater compression or delay. Similarly, connected vehicles may prioritize information related to road hazards, vehicle positions, and collision warnings rather than transmitting every available sensor measurement. The network can therefore associate different semantic priorities with different data sources and dynamically determine which information should receive greater communication resources. This creates a closer relationship between network-level resource management and application-level objectives.

Semantic-aware services can also benefit from edge computing, network slicing, and intelligent network management. Edge nodes can perform semantic extraction, feature processing, context analysis, and task-oriented information selection close to users, reducing the need to transmit complete raw data toward centralized infrastructure. Network slicing can then support differentiated semantic services for applications with different latency, reliability, bandwidth, and processing requirements. In this architecture, semantic information can be processed according to both the communication state and the intended service objective. Such coordination can improve resource utilization while supporting applications that require rapid and meaningful information delivery.

The application scope of semantic-aware network services is broad across emerging 5G systems. Smart cities can use semantic communication for traffic monitoring, public safety, and environmental sensing,

while smart manufacturing can prioritize machine-status information and abnormal-event notifications. In e-health and telemedicine, clinically relevant information can be assigned higher semantic importance to support timely decision-making. Connected vehicles can exchange safety-critical information with greater priority, whereas smart agriculture can focus transmission on information relevant to crop conditions and environmental monitoring. Thus, semantic-aware network services provide a foundation for moving from conventional data-centric networking toward meaning-driven, application-oriented, and intelligent 5G communication services.

8.1.3 Edge-Assisted Semantic Communication in 5G Networks

The figure presents an edge-assisted semantic communication architecture in which IoT and user devices provide raw information to a dedicated 5G edge layer. The device layer includes smartphones, cameras, connected vehicles, industrial IoT devices, wearables, and smart-home devices, representing the heterogeneous sources that can generate large amounts of information. Instead of transmitting all raw information directly to the cloud, the architecture routes the data toward edge-side semantic processing. This arrangement allows relevant meaning to be extracted closer to the point where the information is generated.

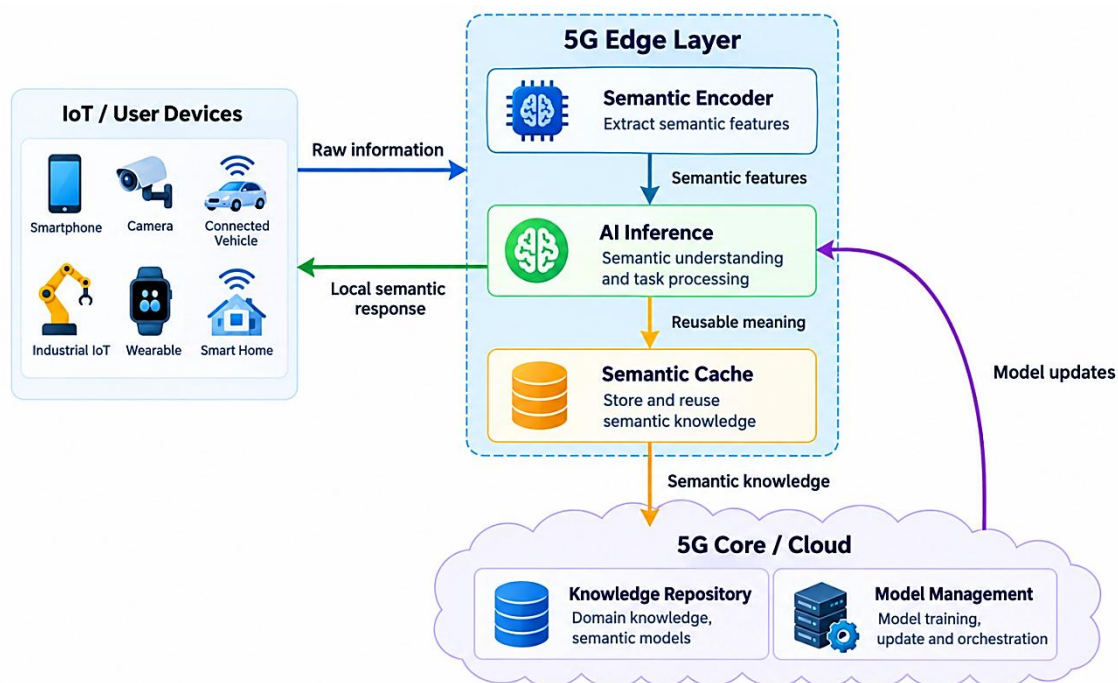


Figure 43: Edge-Assisted Semantic Communication Architecture in 5G Networks

The central 5G Edge Layer contains three important functions: the semantic encoder, AI inference module, and semantic cache. The semantic encoder extracts meaningful features from incoming raw information and converts them into semantic features. These features are then processed by the AI inference component for semantic understanding and task-oriented processing. When a suitable response can be generated locally, the edge layer can return a local semantic response to the connected device. This reduces unnecessary movement of raw data through the network and supports faster application-level processing.

The semantic cache shown below the AI inference component provides another important capability. It stores reusable meaning or semantic knowledge so that previously processed information can potentially be reused when similar requests or situations occur. This reduces repeated semantic processing and can improve the efficiency of edge-assisted communication. The figure therefore illustrates that edge computing is not limited to conventional computation; it can also provide semantic intelligence, knowledge reuse, and task processing close to users.

The lower portion of the figure shows coordination between the 5G Edge Layer and the 5G Core/Cloud. The cloud contains a knowledge repository for domain knowledge and semantic models, together with model management for training, updating, and orchestration. Semantic knowledge can therefore move between edge and cloud resources, while model updates can be delivered back toward the edge AI inference component. This creates a distributed architecture in which computationally intensive learning and model management can remain in the cloud while latency-sensitive semantic processing is performed at the edge. Overall, the figure effectively represents the transition from conventional cloud-centric communication toward distributed, intelligent, and meaning-aware 5G communication supported by edge computing.

8.1.4 Semantic Networking for Advanced 5G Applications

Semantic networking extends the capabilities of 5G networks by enabling communication decisions to consider not only the amount of transmitted data but also its meaning, relevance, context, and importance to the target application. Advanced 5G applications generate heterogeneous and continuously changing information from users, sensors, machines, vehicles, cameras, and intelligent devices. Transmitting all generated information with equal priority can result in unnecessary bandwidth consumption and increased processing requirements. Semantic networking addresses this limitation by identifying the information that is most relevant to a particular application and adapting communication resources accordingly. In this way, network operation becomes more closely connected to application objectives and user requirements.

For connected vehicles and intelligent transportation systems, semantic networking can prioritize safety-critical information such as collision warnings, road hazards, traffic conditions, and vehicle positions. Less important or redundant sensor information can be compressed or transmitted with lower priority. In smart manufacturing, semantic communication can focus on machine states, abnormal operating conditions, predictive-maintenance indicators, and production-related events rather than continuously transmitting every raw sensor measurement. Similarly, smart-city applications can use semantic networking to identify meaningful traffic, environmental, surveillance, and public-safety information. Such application-oriented communication allows network resources to be assigned according to the practical importance of information.

Advanced 5G applications also benefit from the integration of edge computing, artificial intelligence, network slicing, and semantic processing. Edge nodes can extract semantic features and perform task-related inference close to data sources, reducing the amount of information that needs to travel through the core network. Network slicing can provide differentiated communication environments for applications with different requirements, such as ultra-low latency for industrial control or high reliability for safety services. AI-based controllers can further adapt semantic encoding, transmission priority, and resource allocation according to channel conditions, workload, and application requirements. This creates

a flexible networking environment in which communication behavior can dynamically respond to changing circumstances.

The integration of semantic networking with advanced 5G applications therefore supports a transition from data-centric communication toward meaning-driven networking. Its benefits can be evaluated through semantic fidelity, task-completion accuracy, latency, reliability, bandwidth utilization, energy consumption, and computational efficiency. By jointly considering these factors, 5G networks can provide more efficient and application-aware services for smart cities, connected vehicles, industrial automation, healthcare, AR/VR, and other intelligent applications. Semantic networking consequently provides an important foundation for future intelligent communication infrastructures in which network resources are allocated according to the information's contribution to the intended task rather than solely according to its raw data volume.

8.2 Semantic-Native Architectures for Future 6G Communication Networks

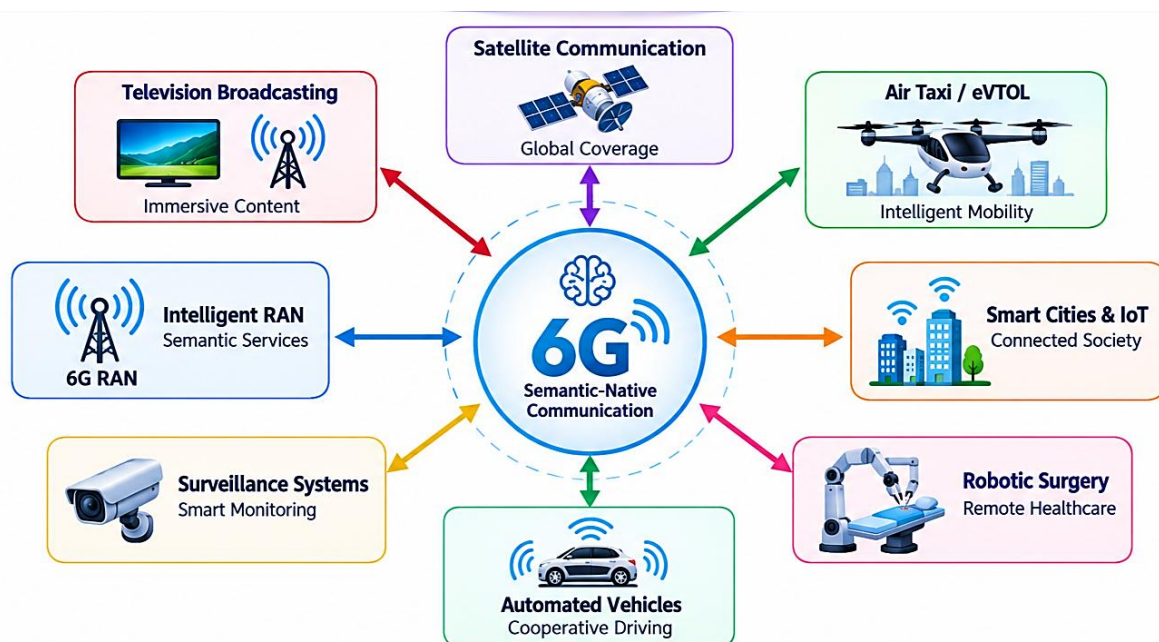


Figure 44: Semantic-Native Architecture and Applications for Future 6G Networks

The figure illustrates a conceptual semantic-native architecture for future 6G communication networks, with 6G Semantic-Native Communication positioned at the center. Unlike conventional communication architectures that primarily focus on reliably transporting data, the semantic-native approach emphasizes the extraction, transmission, and utilization of meaningful information. The central architecture is connected to multiple communication and application domains, demonstrating how semantic intelligence can become an integral component of future 6G infrastructure rather than functioning as an independent application-layer feature.

The Intelligent RAN (6G RAN) shown on the left represents the evolution of radio access networks toward intelligent and semantic-aware operation. Instead of allocating radio resources only according to conventional physical-layer parameters, an intelligent RAN can consider semantic importance,

application requirements, and task objectives when making communication decisions. The figure also includes satellite communication, which highlights the possibility of extending semantic-native 6G services beyond terrestrial networks and supporting global coverage. The integration of terrestrial and non-terrestrial communication can provide connectivity for geographically distributed users and applications.

The surrounding application domains demonstrate the broad scope of semantic-native 6G communication. Automated vehicles can use semantic information to support cooperative driving, while smart cities and IoT can exchange meaningful information among connected infrastructure and devices. Robotic surgery represents highly demanding healthcare applications in which relevant information must be delivered with appropriate reliability and latency. Surveillance systems can use semantic processing to identify important events rather than continuously transmitting unnecessary raw data. Similarly, air taxis/eVTOL systems can require intelligent mobility information, while television broadcasting can use semantic communication to support immersive content delivery.

Overall, the figure emphasizes that future 6G networks can evolve into application-aware, intelligence-driven, and semantic-native communication infrastructures. The bidirectional connections between the central 6G architecture and the different services indicate that semantic communication is expected to interact with diverse network and application environments. This makes the figure particularly useful for introducing the architectural vision of 6G, where intelligent RANs, satellite connectivity, autonomous systems, IoT, healthcare, surveillance, mobility, and immersive media can operate through a common semantic communication foundation.

8.2.1 Architectural Principles of Meaning-Driven 6G Networks

Meaning-driven 6G networks represent a transition from conventional data-centric communication toward communication architectures in which the meaning, relevance, context, and usefulness of information influence how network resources are managed. The fundamental principle is that communication should not necessarily aim to reproduce every transmitted bit exactly when the application can achieve its objective using a smaller or more meaningful representation. In this architecture, semantic information becomes an important design consideration across the communication stack. Semantic encoders can extract task-relevant information from source data, while semantic decoders reconstruct or interpret the intended meaning at the receiver. This approach can reduce unnecessary data transmission and support efficient operation in highly heterogeneous 6G environments.

A second principle is intelligence-native network operation. Future 6G systems are expected to integrate artificial intelligence into communication, computing, and network-management functions. Semantic processing can therefore be coordinated with intelligent radio access networks, edge computing, distributed learning, and adaptive resource management. Network decisions can consider semantic importance together with channel quality, latency, reliability, computational availability, and energy requirements. For example, information associated with a safety-critical task can receive higher transmission priority than redundant or low-value information. Such intelligence enables the network to adapt its behavior according to both environmental conditions and application objectives.

Another important principle is context and task awareness. The same information may have different levels of importance depending on the user, application, environment, and communication objective. Meaning-driven 6G networks should therefore incorporate contextual information when selecting, compressing, transmitting, and reconstructing semantic information. User requirements, task objectives, network conditions, device capabilities, and environmental information can jointly influence semantic processing. This allows communication services to become adaptive rather than relying on a fixed transmission strategy for all users and situations.

Finally, meaning-driven 6G architectures should support heterogeneity, scalability, reliability, and interoperability across terrestrial and non-terrestrial networks, edge and cloud platforms, IoT devices, autonomous systems, and intelligent applications. Semantic information may be exchanged across different network domains while maintaining consistency with the intended task. Robust semantic processing is particularly important in dynamic 6G environments affected by mobility, interference, changing workloads, and variable connectivity. Overall, these principles establish a foundation for semantic-native, intelligent, adaptive, and application-oriented 6G networks, where network performance is evaluated not only by conventional communication metrics but also by how effectively transmitted information contributes to the successful completion of the intended task.

8.2.2 AI-Native Intelligence for Semantic Network Management

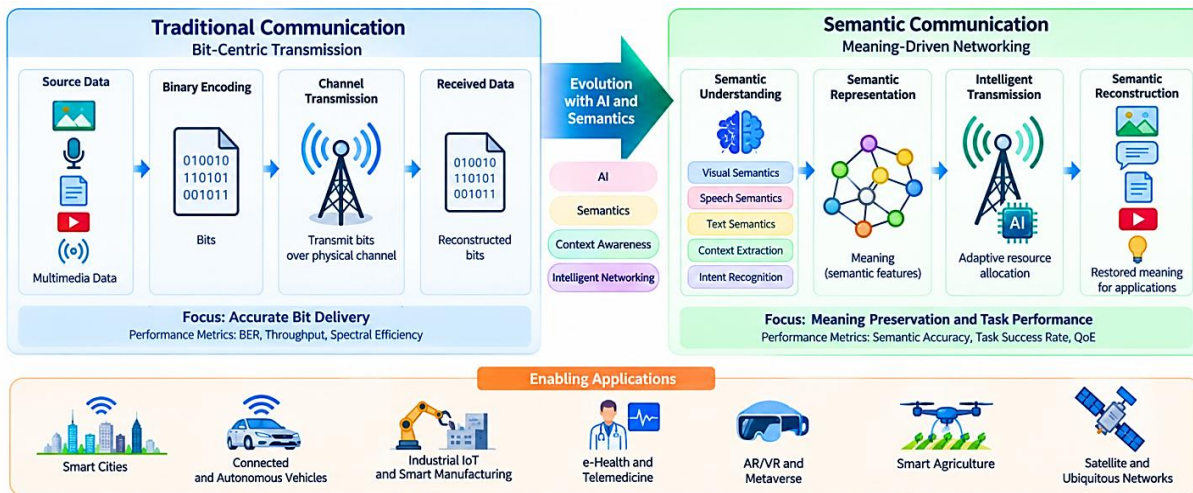


Figure 45: AI-Native Intelligence for Semantic Network Management

The figure illustrates the transition from traditional communication to AI-native semantic communication. The left side represents conventional bit-centric transmission, where source data such as multimedia information is converted into binary bits, transmitted through a physical channel, and reconstructed as received data. The primary objective in this model is accurate bit delivery, with conventional measures such as bit-error rate, throughput, and spectral efficiency used to evaluate performance. This approach treats the transmitted data largely as a sequence of bits, without directly considering how important individual information elements are to the final application task.

The right side presents the semantic communication approach, where AI and semantic processing are incorporated into the communication workflow. Semantic understanding extracts information such as

visual semantics, speech semantics, text semantics, contextual information, and user intent. These elements are transformed into semantic representations containing meaningful features rather than simply transmitting the complete raw information. Intelligent transmission can then adapt communication resources according to semantic importance and network conditions, while semantic reconstruction attempts to recover the intended meaning for the receiving application.

The figure also highlights the importance of AI-native network management through adaptive resource allocation. Instead of allocating communication resources exclusively according to conventional physical-layer measurements, an intelligent semantic network can consider semantic accuracy, task success rate, and quality of experience together with communication constraints. This enables the network to prioritize information that contributes more significantly to the intended application. The transition arrow between the two architectures emphasizes the combined role of AI, semantics, context awareness, and intelligent networking in enabling this transformation.

The lower part of the figure demonstrates the broad application scope of this architecture, including smart cities, connected and autonomous vehicles, industrial IoT and smart manufacturing, e-health and telemedicine, AR/VR and metaverse services, smart agriculture, and satellite or ubiquitous networks. These applications have different requirements for latency, reliability, information relevance, and task performance. Therefore, AI-native semantic network management can dynamically coordinate semantic processing and network resources according to application needs. The figure consequently provides a clear conceptual foundation for explaining how future 6G networks can move beyond reliable bit delivery toward meaning preservation, adaptive intelligence, and task-oriented network performance.

8.2.3 Semantic-Aware Network Slicing and Service Differentiation

Semantic-aware network slicing extends the conventional concept of network slicing by allowing communication resources and service characteristics to be differentiated according to the meaning, importance, context, and task requirements of transmitted information. In conventional 5G and future 6G networks, network slices can be created for different services with specific requirements for bandwidth, latency, reliability, and security. Semantic-aware slicing adds another dimension by considering how much a particular information stream contributes to the intended application. Consequently, two services using similar amounts of network resources may receive different treatment when their semantic importance or task criticality is different.

A semantic-aware slicing framework can begin by identifying the semantic requirements of applications and information flows. Information can be categorized according to its relevance, urgency, accuracy requirements, and expected contribution to the target task. For example, a connected-vehicle slice may assign high priority to collision warnings and road-hazard information, while routine vehicle-status information may receive lower priority. Similarly, an industrial automation slice can prioritize information associated with abnormal machine conditions or safety events. This allows the network to distinguish between information that is essential for successful task execution and information that can tolerate greater compression, delay, or occasional loss.

Service differentiation can further be achieved by dynamically adjusting bandwidth, transmission priority, computing resources, and reliability mechanisms for individual semantic services. Edge computing nodes

can support semantic extraction and task-oriented processing within specific slices, while intelligent controllers can monitor semantic quality and network conditions. When network congestion occurs, less important semantic information may be compressed more aggressively or delayed, whereas critical information can retain stronger transmission protection. Such adaptive differentiation can improve resource utilization while maintaining the quality required by important applications.

The integration of semantic awareness with network slicing is particularly relevant to future 6G heterogeneous networks, where smart cities, autonomous vehicles, industrial IoT, immersive applications, healthcare systems, and ubiquitous connectivity may operate simultaneously. Instead of providing identical communication treatment to every data stream, semantic-aware slicing enables the network to align resources with application objectives. Service quality can consequently be evaluated using not only throughput and latency but also semantic fidelity, task success, information relevance, and quality of experience. This approach transforms network slicing from a mainly connectivity-oriented mechanism into an application-aware and meaning-driven service differentiation framework, supporting efficient and adaptive operation across diverse intelligent communication services.

8.2.4 Intelligent Radio Access Networks for Semantic Services

Intelligent Radio Access Networks (RANs) provide an important foundation for delivering semantic services in future 6G communication systems. Conventional RAN operation primarily focuses on physical and network-layer objectives such as signal quality, spectrum efficiency, transmission power, interference management, and user scheduling. Semantic services require a broader perspective in which the RAN also considers the meaning and importance of transmitted information. An intelligent semantic RAN can therefore coordinate radio resources according to both communication conditions and the requirements of the application or task supported by the transmitted information.

A semantic-aware RAN can incorporate information from semantic encoders, application requirements, channel conditions, user context, and network state when making transmission decisions. For example, when several users compete for limited spectrum, the RAN can identify which semantic information is most important for their respective tasks and assign resources accordingly. Safety-critical information in autonomous transportation may receive higher priority than non-critical background information. Similarly, an industrial monitoring service may require rapid and reliable transmission of abnormal-event information while allowing routine measurements to use fewer resources. Such decisions allow radio resources to be associated with the actual value of information rather than solely with its volume.

Artificial intelligence can further enhance intelligent RAN operation by enabling adaptive resource management and semantic-aware decision making. Learning-based controllers can observe variations in traffic demand, channel quality, mobility, interference, computational availability, and semantic quality, and then adapt transmission policies accordingly. Semantic feedback from receivers or applications can also be used to determine whether the delivered information was sufficient for the intended task. This creates a feedback-driven communication process in which radio parameters and semantic transmission strategies can be continuously adjusted according to observed performance.

Intelligent RANs are particularly important for supporting heterogeneous 6G services involving autonomous vehicles, smart cities, industrial IoT, immersive applications, healthcare, and massive sensor

networks. These services have substantially different requirements for latency, reliability, semantic accuracy, and resource consumption. A semantic-aware RAN can provide differentiated treatment while maintaining efficient use of spectrum and energy. Its performance can therefore be assessed through a combination of conventional wireless indicators and semantic measures such as task accuracy, semantic fidelity, information relevance, and quality of experience. Through this integration of AI, radio intelligence, and semantic awareness, future RANs can evolve from connectivity-oriented infrastructures into adaptive intelligent platforms capable of supporting meaning-driven communication services.

8.3 Semantic-Aware Routing, Forwarding, and Network Orchestration

8.3.1 Meaning-Aware Routing and Semantic Path Selection

Meaning-aware routing extends conventional network routing by considering the semantic importance, relevance, and task requirements of information when selecting communication paths. Traditional routing mechanisms generally determine paths using metrics such as hop count, link cost, bandwidth, delay, congestion, or packet loss. Although these metrics remain important, they do not directly indicate whether the information carried by a packet is critical to the receiving application. In semantic networking, routing decisions can therefore incorporate information value so that important semantic content is delivered through paths that provide appropriate reliability, latency, and resource availability. This approach is particularly relevant to 5G and 6G networks supporting heterogeneous applications with different communication objectives.

A meaning-aware routing process can associate each semantic information flow with characteristics such as semantic priority, task relevance, required fidelity, urgency, and acceptable delay. The network can then evaluate available paths according to both network conditions and semantic requirements. For instance, information related to emergency detection or autonomous vehicle safety may require a highly reliable and low-latency route, whereas routine environmental measurements may tolerate a longer or less resource-intensive path. Consequently, path selection becomes dependent not only on where the information needs to travel but also on how important that information is to the target task.

Semantic path selection can also exploit contextual and application-level information. User requirements, network congestion, device capabilities, edge-computing availability, and current channel conditions can influence the selection process. In a distributed 6G environment, an intelligent routing controller may compare multiple candidate paths and select the one that provides the most appropriate balance between semantic quality and communication cost. If a selected route becomes congested or unreliable, the system can redirect high-priority semantic information through an alternative path. Less important information may be delayed, compressed, or assigned a lower-priority route to preserve resources for critical services.

The effectiveness of meaning-aware routing can be evaluated using both conventional networking metrics and semantic performance measures. Latency, packet delivery reliability, bandwidth consumption, and energy usage can be considered together with semantic fidelity, task accuracy, information relevance, and quality of experience. This integrated approach changes routing from a purely packet-oriented process into an application- and meaning-aware mechanism. By selecting paths according to the contribution of information to the intended task, semantic routing can improve resource utilization and support differentiated services across autonomous transportation, smart cities, industrial IoT, healthcare, and other intelligent 6G applications.

8.3.2 Context-Aware Packet Forwarding and Information Delivery

Context-aware packet forwarding enhances conventional forwarding mechanisms by considering the current context in which information is generated, transmitted, and consumed. In conventional networks, forwarding decisions are commonly based on destination addresses, routing tables, link conditions, and congestion levels. Semantic communication requires a broader forwarding strategy because the usefulness of information can depend on the user, application, environment, task, and network state. Context-aware forwarding therefore aims to deliver information in a manner that reflects its current relevance and importance rather than treating every packet identically.

The forwarding process can incorporate several types of contextual information, including user context, task context, environmental conditions, and network status. User context may represent preferences or service requirements, while task context describes the objective that the communication system must support. Environmental context can include location, surrounding entities, or physical conditions, and network context can include available bandwidth, latency, congestion, and computing resources. By combining these factors, intermediate network nodes can determine which information should be forwarded immediately, which can be compressed or delayed, and which may have limited value under the current circumstances.

Context-aware forwarding is especially useful when network resources are constrained or conditions change dynamically. Consider a connected-vehicle environment in which several information streams are generated simultaneously. A sudden road-hazard notification may become highly important because it directly affects vehicle safety, whereas routine vehicle-status information may have considerably lower urgency. A context-aware forwarding mechanism can therefore prioritize the hazard information and allocate suitable forwarding resources to ensure rapid delivery. Similar principles can be applied to industrial monitoring, emergency response, smart-city sensing, healthcare, and autonomous systems.

Another important feature is adaptive information delivery. Forwarding nodes can use feedback from receivers, applications, or network controllers to determine whether delivered semantic information satisfies the intended task. When contextual conditions change, forwarding priorities and delivery strategies can be updated accordingly. This creates a dynamic process in which information delivery is continuously aligned with application requirements and network conditions. Performance can be assessed through delivery latency, reliability, resource utilization, semantic fidelity, task success, and quality of experience. Thus, context-aware packet forwarding provides a bridge between traditional network-layer forwarding and meaning-driven communication, enabling future 6G networks to deliver information according to what is useful, when it is useful, and for which task it is required.

8.3.3 Intent-Based Network Management and Service Orchestration

The figure presents an intent-based network management and service orchestration framework in which a service user's high-level requirements are transformed into actionable network operations. The process begins with the Service User, which specifies a service intent through an application or tenant interface. The intent is then expressed through the Intent Interface and passed to the Intent Interpreter. This component performs semantic parsing and mapping to transform the user's requirement into an interpreted objective that can be understood by the network management system. This approach reduces the need for users to specify detailed technical network configurations.

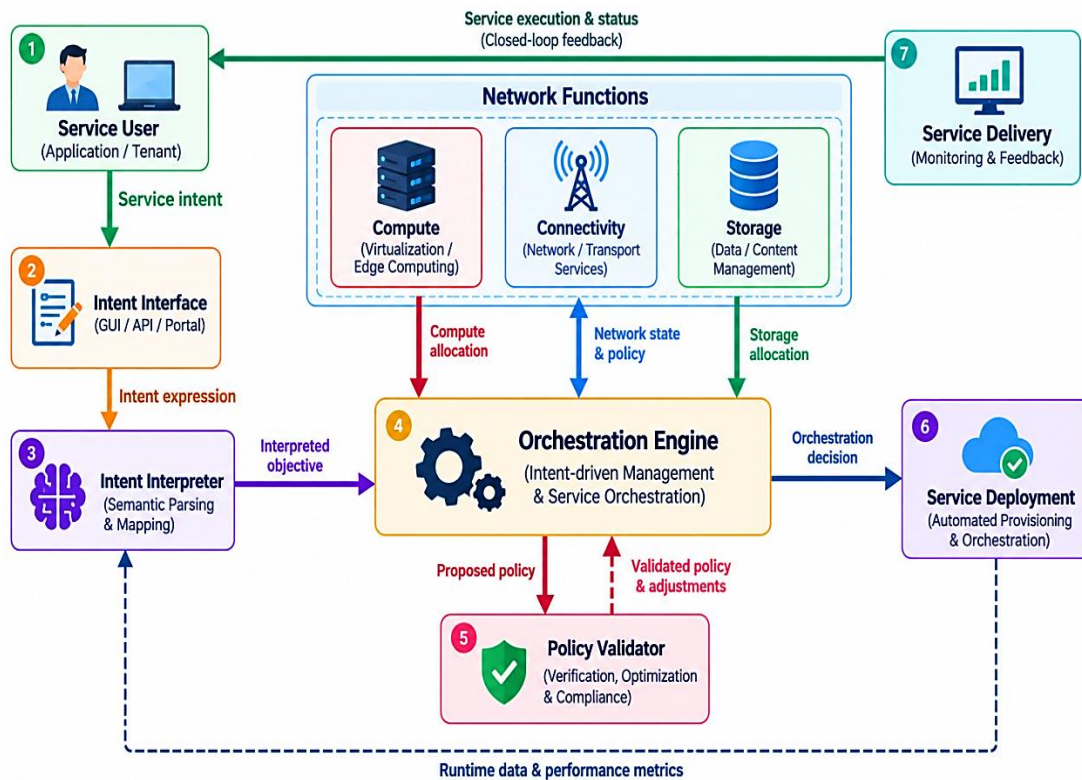


Figure 46: Intent-Based Network Management and Semantic Service Orchestration

The interpreted objective is forwarded to the Orchestration Engine, which forms the central decision-making component of the architecture. The orchestration engine coordinates different network functions, including compute, connectivity, and storage. It considers network state, policies, compute availability, and storage requirements when determining how a requested service should be implemented. In a semantic communication environment, this mechanism can also incorporate application objectives and semantic service requirements when selecting suitable resources and communication strategies.

The figure also includes a Policy Validator, which receives the proposed policy from the orchestration engine and performs verification, optimization, and compliance checking. Validated policies and necessary adjustments are returned to the orchestration engine before the final orchestration decision is implemented. The resulting decision is sent to Service Deployment, where automated provisioning and orchestration establish the requested service. Runtime data and performance metrics can then be returned to the management process, allowing the system to evaluate whether the deployed service is operating according to its requirements.

An important feature of the architecture is the closed-loop feedback mechanism connecting service execution and status back to the service user and management functions. Monitoring and feedback allow the system to observe actual service performance and adjust policies or resources when conditions change. This makes intent-based orchestration particularly suitable for dynamic 5G and 6G semantic networks, where user requirements, network conditions, workloads, and application objectives may change continuously. Overall, the figure demonstrates how intent-based management can transform high-

level service requirements into validated policies, automated resource allocation, service deployment, and continuous performance-driven adaptation, supporting intelligent and meaning-aware network operation.

8.3.4 Semantic Traffic Engineering and Network Adaptation

Semantic traffic engineering extends conventional traffic engineering by considering not only the volume and destination of network traffic but also the semantic importance, relevance, urgency, and task requirements associated with transmitted information. Traditional traffic engineering focuses on distributing network traffic across available paths to reduce congestion, improve utilization, and maintain acceptable latency. In semantic communication networks, however, different information flows can have substantially different values for the application. Therefore, network adaptation can prioritize traffic according to its contribution to the intended task. This enables communication resources to be allocated more intelligently when bandwidth, computing capacity, or network connectivity becomes limited.

A semantic traffic engineering framework can classify information according to its semantic priority and application requirements. Critical information associated with safety, emergency response, industrial control, or healthcare can be assigned higher priority than routine or redundant information. When congestion occurs, high-value semantic traffic can be transmitted through paths offering better reliability and lower latency, while less important information may be compressed, delayed, or routed through alternative paths. This approach prevents network resources from being consumed equally by information streams that have very different impacts on application performance. Semantic traffic engineering can therefore establish a direct relationship between network resource management and the usefulness of transmitted information.

Network adaptation is particularly important in dynamic 5G and 6G environments where channel conditions, user mobility, traffic demand, network load, and application requirements can change continuously. Intelligent network controllers can monitor these conditions and adjust routing, forwarding, bandwidth allocation, transmission priority, and computing resources accordingly. For example, when a mobile user moves into an area with reduced connectivity, the system can increase protection for important semantic information while reducing the transmission of redundant content. Similarly, during periods of network congestion, semantic-aware adaptation can preserve information that is essential for task completion rather than attempting to maintain identical service levels for every data stream.

The effectiveness of semantic traffic engineering can be assessed using a combination of conventional networking and semantic performance measures. Latency, throughput, congestion, packet delivery reliability, and resource utilization can be evaluated together with semantic fidelity, task-completion accuracy, information relevance, and quality of experience. Feedback from applications and receivers can further support closed-loop adaptation, allowing the network to modify its traffic policies according to observed semantic performance. Thus, semantic traffic engineering transforms network adaptation from a primarily data-volume-oriented process into a meaning-aware and task-oriented mechanism, providing a foundation for efficient and intelligent traffic management in future 6G networks.

8.4 Integrated Sensing, Communication, and Semantic Intelligence

8.4.1 Integration of Sensing and Semantic Information Processing

The integration of sensing, communication, and semantic intelligence represents an important direction for future 6G networks. Conventional sensing systems primarily focus on collecting information about physical environments, while communication systems are responsible for transporting the collected data. This separation can result in large volumes of raw sensing information being transmitted even when only a small portion is relevant to the intended application. Integrating semantic information processing with sensing allows the system to identify meaningful information at an early stage and communicate only the information that contributes to a particular task or decision.

In an integrated architecture, sensors and wireless devices continuously observe physical environments and generate information related to objects, movements, locations, activities, and environmental conditions. Semantic processing can then transform these observations into higher-level representations that are more useful to applications. For example, instead of transmitting a complete sequence of raw sensor measurements, a system may identify the presence of an object, a change in environmental conditions, or an abnormal event. This reduces the dependency on transmitting every individual measurement while preserving information that is important for the target application.

The integration also enables joint interpretation of sensing and communication context. Information about the sensing environment can influence communication decisions, while communication conditions can determine how sensing information should be represented and transmitted. If the network is congested, semantic processing can prioritize sensing information that has greater importance for the current task. Similarly, when a sensing event becomes critical, the communication system can allocate additional resources to ensure its timely delivery. This interaction creates a feedback relationship between sensing intelligence and communication intelligence.

Such integration is particularly valuable for smart cities, autonomous vehicles, industrial monitoring, environmental observation, healthcare, robotics, and intelligent infrastructure. Autonomous systems can use semantic sensing to identify objects and events relevant to navigation, while industrial systems can prioritize information indicating equipment abnormalities. Smart-city platforms can extract meaningful traffic or environmental events rather than transmitting all raw sensor observations. Overall, integrating sensing with semantic information processing enables a transition from raw-data collection toward meaning-aware sensing and communication, where information is extracted, represented, and delivered according to its relevance to the intended task.

8.4.2 Semantic Extraction from Wireless Sensing Data

Wireless sensing systems generate information by observing physical environments through radio signals, distributed sensors, connected devices, and other sensing mechanisms. The resulting data can contain substantial redundancy because sensors may continuously collect measurements even when environmental conditions remain relatively unchanged. Semantic extraction addresses this challenge by identifying the meaningful features, events, objects, relationships, and changes contained within wireless sensing data. Instead of treating sensing information only as numerical measurements or signal samples, semantic processing attempts to transform the observations into representations that are directly useful for communication and application-level decision making.

The semantic extraction process can involve several stages, beginning with the acquisition and preprocessing of wireless sensing data. Relevant signal characteristics can be identified and transformed into features that describe physical phenomena. Machine learning and deep learning models can then identify patterns associated with objects, activities, environmental conditions, or abnormal events. Depending on the application, the resulting semantic representation may describe information such as object presence, movement, location, activity type, environmental change, or system status. The objective is not necessarily to reproduce the complete sensing signal but to preserve the information needed by the target application.

Context plays an important role in determining which sensing information should be considered meaningful. The same wireless observation can have different significance depending on the location, user, task, environmental condition, and network state. For example, a change detected by a sensor may be insignificant during routine operation but highly important when an industrial monitoring system is investigating equipment failure. Similarly, an object detected by a sensing system may have greater semantic importance when it is located within a vehicle's immediate path. Context-aware semantic extraction can therefore improve the relevance of the resulting representation.

The extracted semantic information can subsequently be used for task-oriented communication, intelligent monitoring, and adaptive network operation. Rather than transmitting large quantities of raw sensing data, the system can communicate compact representations that retain the information required for inference or decision making. This can reduce communication overhead while improving the efficiency of downstream processing. Semantic extraction from wireless sensing data is therefore an important component of integrated sensing and communication architectures, enabling 6G systems to transform physical observations into meaningful, application-relevant information before transmission.

8.4.3 Joint Optimization of Sensing and Communication Resources

Joint optimization of sensing and communication resources is essential when sensing and communication functions share limited spectrum, energy, computing capability, and network infrastructure. In conventional architectures, sensing operations and communication services may be designed independently, which can result in inefficient resource utilization. Integrated 6G systems require coordinated management in which sensing requirements and communication objectives are considered together. Semantic intelligence adds another dimension by allowing the system to evaluate resources according to the importance of the information generated by sensing activities and its contribution to the target task.

A joint optimization framework can coordinate resources such as spectrum, transmission power, sensing time, bandwidth, computing capacity, and edge-processing resources. The system can determine how these resources should be distributed according to sensing accuracy, communication quality, latency, energy consumption, and semantic requirements. For example, sensing resources can be concentrated on areas or objects that are more relevant to the current application, while communication resources can be assigned according to the importance of the resulting semantic information. This avoids allocating identical resources to sensing observations that have substantially different application value.

Semantic requirements can also influence how sensing information is transmitted. Highly important information may require stronger reliability, lower latency, or greater semantic fidelity, whereas redundant or low-priority sensing information may tolerate greater compression or delayed transmission. Intelligent controllers can continuously observe sensing quality and network conditions and adjust resource allocation accordingly. When channel conditions deteriorate, the system can modify sensing and communication strategies to maintain the semantic quality required by the application rather than attempting to preserve every raw measurement.

The joint optimization of sensing and communication is particularly relevant to autonomous transportation, smart cities, industrial automation, robotics, environmental monitoring, and intelligent infrastructure. In autonomous systems, for example, sensing resources can focus on safety-relevant objects while communication resources prioritize the corresponding information. In industrial environments, abnormal equipment conditions can receive greater sensing and communication attention than routine measurements. Therefore, joint sensing-communication optimization supports a more efficient use of shared resources while maintaining application performance. Combined with semantic intelligence, it enables future 6G systems to move toward task-oriented resource allocation, where sensing and communication decisions are coordinated according to what information is most valuable for achieving the intended objective.

8.4.4 Intelligent Environment Understanding through Semantic Networks

The figure illustrates the overall transformation from a physical environment into an intelligent, semantically interpreted environment. At the top, the Environment represents the physical world containing objects, locations, activities, and changing conditions. These physical observations are collected through Distributed Sensors, including IoT devices, cameras, environmental sensors, and wearable devices. Instead of treating the resulting sensor measurements only as raw numerical data, the architecture transforms them into semantic observations that can describe entities, events, relationships, and environmental conditions in a more meaningful form.

The semantic observations are subsequently processed by the Semantic Network, which performs semantic modeling, entity linking, and reasoning. This layer provides the central intelligence for connecting observations with their meanings and relationships. The Knowledge Graph shown on the left complements this process by maintaining entities, concepts, and relationships that provide contextual knowledge. The interaction between the semantic network and knowledge graph enables the system to interpret observations in relation to previously available knowledge rather than analyzing each observation independently. This is particularly important when multiple sensors provide complementary or partially overlapping information about the same situation.

The resulting semantic events are passed to the Situation Understanding layer, where context analysis, event recognition, and situation awareness are performed. This stage converts individual semantic observations into a more comprehensive understanding of what is happening in the environment. For example, observations from different sensors can collectively indicate a particular environmental condition, activity, or event. The architecture therefore emphasizes the transition from simple sensing to higher-level interpretation, allowing the communication system to operate on meaningful environmental information rather than transmitting every raw measurement.

Finally, the Intelligent Decision layer uses the generated situation awareness to support adaptive actions, resource optimization, and smart responses. The feedback path labeled adaptive request indicates that decisions can influence subsequent semantic processing and sensing requirements. This creates a closed-loop architecture in which the system can request or prioritize additional information when environmental conditions change or when greater semantic certainty is required. Thus, the figure effectively represents the role of semantic networks as an intermediate intelligence layer connecting distributed sensing, contextual knowledge, situation understanding, and adaptive decision-making in intelligent communication environments.

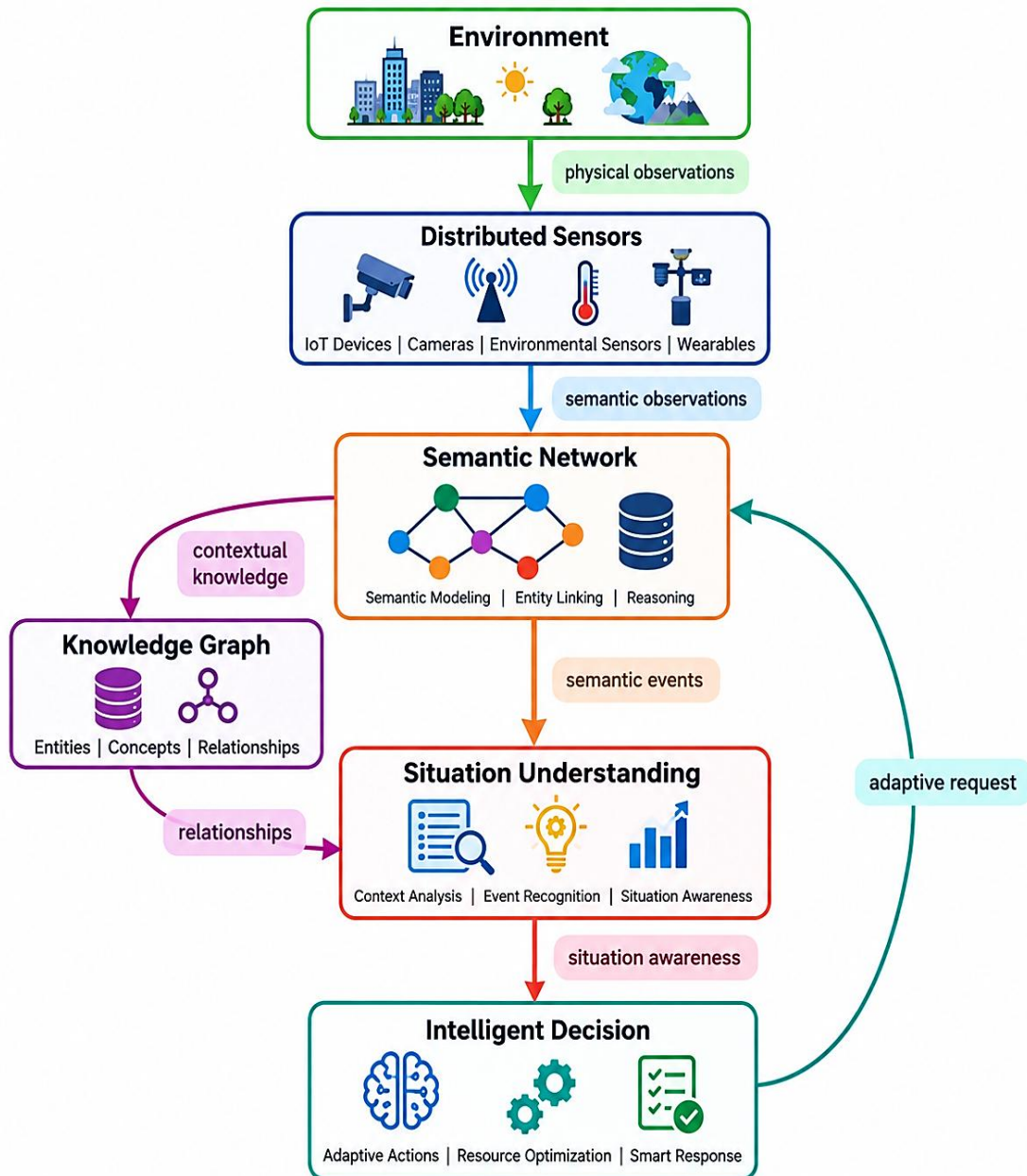


Figure 47: Intelligent Environment Understanding through Semantic Networks

Chapter 9

Edge Intelligence, IoT, and Distributed Semantic Communication Systems

9.1 Edge Computing Architectures for Real-Time Semantic Communication

9.1.1 Distributed Semantic Processing at Network Edge Locations

The figure presents a distributed architecture in which semantic processing is performed across IoT/end devices, edge nodes, and a centralized cloud center. At the bottom, smart devices, vehicles, perception devices, RFID readers, and sensors act as semantic data sources. These devices continuously observe their surroundings and generate information that may require immediate interpretation. Instead of transmitting all raw information to a remote cloud, semantic processing can be performed closer to the source, reducing unnecessary data movement and supporting time-sensitive applications.

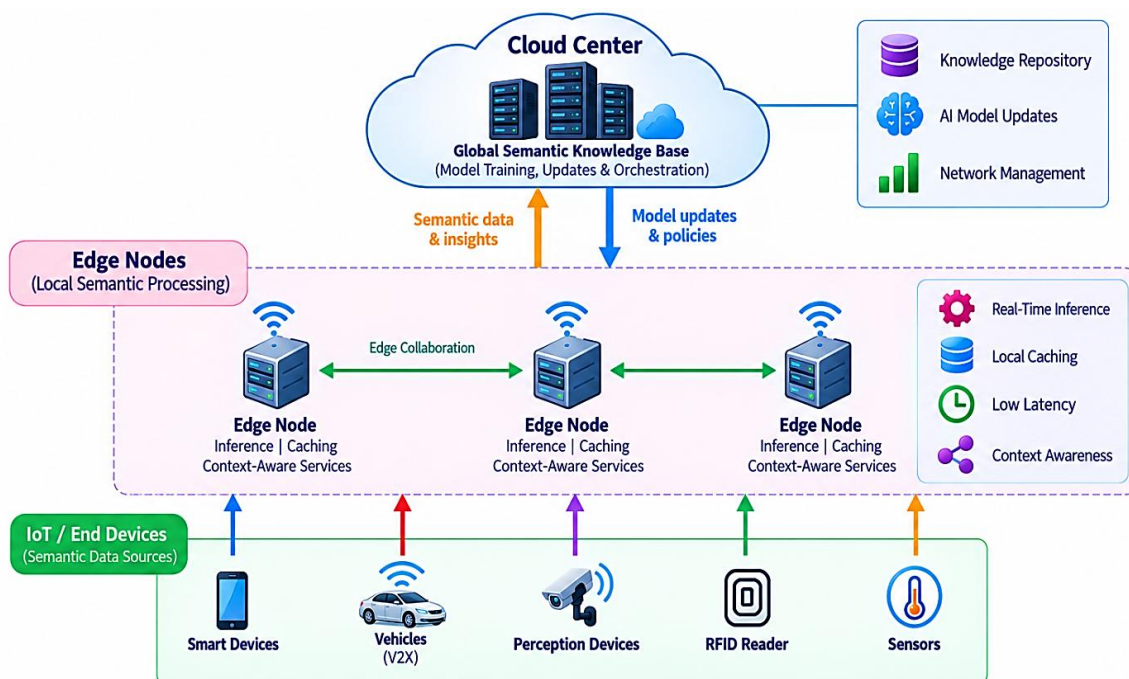


Figure 48: Distributed Semantic Processing Architecture at Network Edge Locations

The central part of the figure shows multiple edge nodes connected through an edge collaboration layer. Each edge node supports inference, caching, and context-aware services, allowing semantic information to be processed locally. Real-time inference enables nearby applications to obtain decisions with reduced delay, while local caching allows frequently required semantic information to be reused without repeatedly retrieving it from the cloud. The collaboration between edge nodes also allows semantic knowledge and processing responsibilities to be distributed across several locations, which is particularly useful when devices are geographically dispersed or network conditions vary. At the top, the Cloud Center provides a global semantic knowledge base responsible for activities such as model training, updates, and orchestration. The cloud exchanges semantic data and insights with edge nodes while also

distributing model updates and policies. This creates a hierarchical relationship between centralized intelligence and localized processing. Edge nodes can respond rapidly to local requirements, whereas the cloud can provide broader knowledge, updated models, and coordinated management across the distributed system. The architecture demonstrates how edge intelligence can complement cloud intelligence in semantic communication systems. Processing semantic information near IoT and user devices can reduce latency, communication overhead, and dependence on continuous cloud connectivity, while centralized cloud resources can maintain global knowledge and coordinate model evolution. This distributed approach provides an effective foundation for real-time semantic services in applications such as autonomous systems, industrial IoT, smart cities, environmental monitoring, and other latency-sensitive intelligent communication scenarios.

9.1.2 Edge-Based Semantic Encoding and Decoding

Edge-based semantic encoding and decoding enables semantic communication functions to be performed close to the devices that generate and consume information. In conventional communication architectures, raw data may be transmitted to centralized servers before meaningful information is extracted. This approach can introduce additional communication overhead and processing delay, particularly for applications involving continuous data generation. Edge-based processing instead places semantic encoders and decoders at nearby edge nodes, allowing information to be transformed into compact meaning-oriented representations before transmission and reconstructed closer to the end user or application.

At the transmitting side, an edge-based semantic encoder receives information from IoT devices, sensors, cameras, vehicles, or user equipment and extracts features relevant to the intended communication task. The encoder can process different modalities, including text, images, speech, video, and sensor observations. Rather than preserving every detail of the original input, semantic encoding identifies the information that contributes most directly to the application objective. The resulting semantic representation can then be compressed and transmitted through the available communication network. Performing this operation at the edge reduces the amount of information that needs to travel through the network and can consequently support more efficient use of communication resources.

At the receiving side, an edge-based semantic decoder interprets the received semantic representation and reconstructs information according to the requirements of the target application. The decoder does not necessarily need to reproduce the exact original data. Instead, it aims to recover the relevant meaning, task information, or application-level representation. For example, an edge decoder supporting an intelligent monitoring system may reconstruct the semantic description of an observed event rather than reproducing every original sensor measurement. This task-oriented reconstruction is particularly useful when communication conditions are constrained or when rapid decisions are more important than exact data recovery.

The distributed placement of semantic encoding and decoding also supports low-latency and context-aware communication. Edge nodes can use locally available information about users, applications, network conditions, and surrounding devices to adapt semantic processing according to current requirements. Frequently used representations can also be cached at the edge, reducing repeated processing and transmission. At the same time, edge nodes can cooperate with cloud infrastructure for

model training, knowledge management, and model updates. Consequently, edge-based semantic encoding and decoding provides a practical bridge between distributed devices and centralized intelligence, supporting efficient, responsive, and task-oriented semantic communication for IoT, autonomous systems, industrial applications, and intelligent services.

9.1.3 Cloud-Edge Collaboration for Semantic Intelligence

Cloud-edge collaboration provides a distributed architecture in which centralized cloud intelligence and localized edge processing work together to support efficient semantic communication. The uploaded figure is well suited to this section because it explicitly illustrates the interaction among IoT/end devices, edge nodes, and the cloud center. The IoT and end devices act as semantic data sources, while edge nodes provide local processing, inference, caching, and context-aware services. The cloud center provides global model training, semantic knowledge management, model updates, orchestration, and network analytics. This division allows semantic intelligence to be distributed according to processing requirements and network conditions.

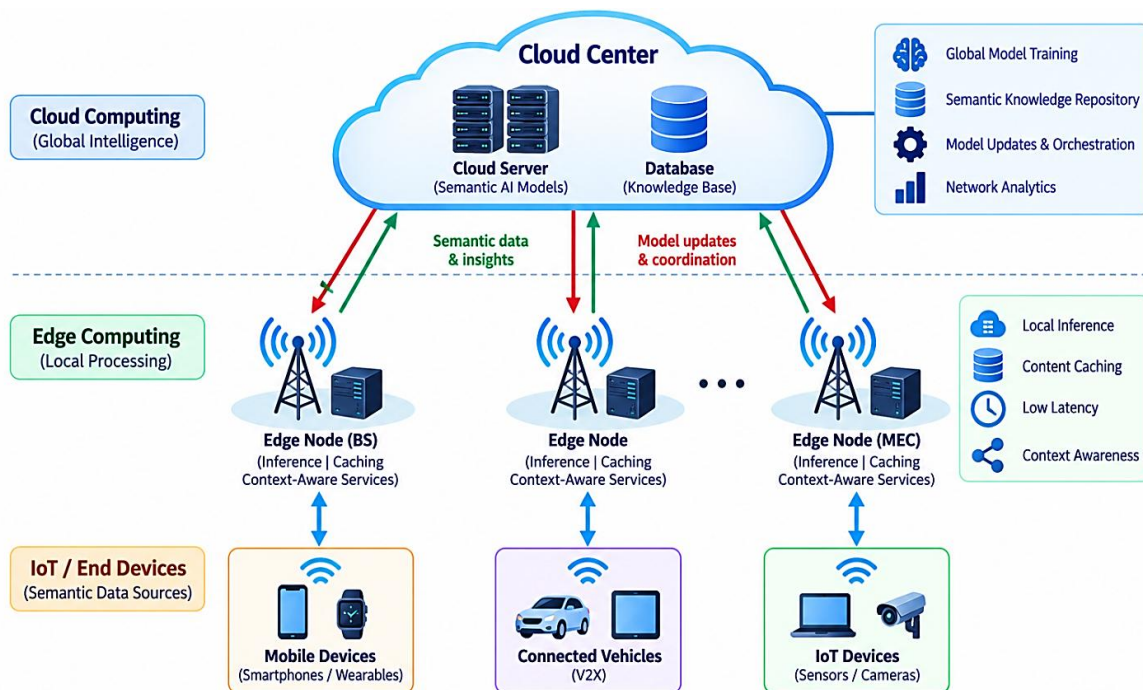


Figure 49: Cloud-Edge Collaboration Architecture for Semantic Intelligence

The figure shows mobile devices, connected vehicles, sensors, and cameras communicating with multiple edge nodes. The edge nodes are positioned between end devices and the cloud, enabling semantic information to be processed closer to where it is generated. Local inference allows edge nodes to make rapid semantic decisions without continuously transferring raw information to the cloud. Content caching can further reduce repeated communication, while context awareness allows semantic processing to consider the characteristics of users, devices, applications, and surrounding environments. These capabilities are particularly important for applications requiring low latency and continuous interaction.

The cloud center complements the edge layer by maintaining broader computational and knowledge resources. As shown in the figure, the cloud contains semantic AI models and a knowledge database, while supporting global model training, semantic knowledge repositories, model updates and orchestration, and network analytics. Semantic data and insights can move from edge nodes toward the cloud, allowing global intelligence to benefit from information collected across distributed locations. In the opposite direction, updated models and coordination information can be delivered from the cloud to edge nodes, enabling local semantic processing capabilities to evolve over time.

The architecture therefore establishes a continuous cloud-edge collaboration loop rather than treating cloud and edge computing as isolated components. Edge nodes provide fast, localized semantic intelligence, whereas the cloud provides global knowledge, model management, and coordinated learning. This arrangement can reduce latency and unnecessary data transmission while maintaining access to powerful centralized intelligence. It is especially suitable for distributed IoT environments, connected vehicles, smart devices, industrial systems, and other applications in which semantic information must be interpreted rapidly while models and knowledge continue to evolve across the network.

9.1.4 Latency-Aware Edge Semantic Processing

Latency-aware edge semantic processing focuses on reducing the time required to transform, transmit, interpret, and reconstruct meaningful information in distributed communication environments. In conventional cloud-centric systems, data generated by end devices may need to travel to a remote cloud server before processing takes place, after which the resulting information is returned to the application. Such a process can introduce communication and processing delays, particularly when networks are congested or devices are geographically distributed. Edge semantic processing addresses this limitation by placing semantic intelligence closer to data sources and users, allowing important information to be processed locally or at nearby edge nodes.

A latency-aware architecture considers the complete semantic processing path rather than communication delay alone. The time required for semantic feature extraction, representation generation, compression, transmission, decoding, and task execution can all influence application responsiveness. Edge nodes can reduce this overall delay by performing computationally intensive semantic operations near the devices that generate the information. For example, cameras, connected vehicles, sensors, and mobile devices can send information to a nearby edge node where relevant semantic features are extracted and interpreted. Only the required semantic representation may then be forwarded to another network component or application, reducing unnecessary data movement.

Latency awareness also requires semantic processing to respond differently to information with different levels of urgency. Information associated with safety-critical or time-sensitive tasks should receive rapid processing, while less important information can tolerate additional delay. An edge controller can therefore prioritize semantic workloads according to application requirements, current network conditions, and computational availability. When network congestion increases, semantic representations can be made more compact or low-priority information can be processed later. Similarly, edge resources can be dynamically assigned to applications that have strict response-time requirements.

Another important aspect is coordination between edge and cloud resources. Although edge nodes provide rapid local processing, the cloud remains valuable for global knowledge management, model training, analytics, and model updates. Latency-sensitive tasks can therefore be executed at the edge, while operations that are less time-critical can be transferred to cloud infrastructure. This division avoids unnecessary dependence on remote processing while preserving the benefits of centralized intelligence. Overall, latency-aware edge semantic processing supports responsive communication by aligning semantic computation, network transmission, and resource allocation with application urgency. It is particularly important for autonomous vehicles, industrial automation, real-time monitoring, smart cities, robotics, and other applications where delayed semantic information can directly affect the quality and timeliness of decisions.

9.2 Distributed and Federated Learning for Semantic Intelligence

9.2.1 Distributed Semantic Model Training across Edge Devices

Distributed semantic model training enables multiple edge devices to collaboratively develop semantic intelligence without relying entirely on a centralized training process. This approach is particularly useful in semantic communication systems because edge devices may observe different environments, applications, users, and data modalities. Instead of transferring all locally collected data to a central cloud, each device can perform model training using its own local data and contribute the resulting model information to a shared learning process. This reduces the need for continuous raw-data transmission and allows semantic models to learn from geographically distributed environments.

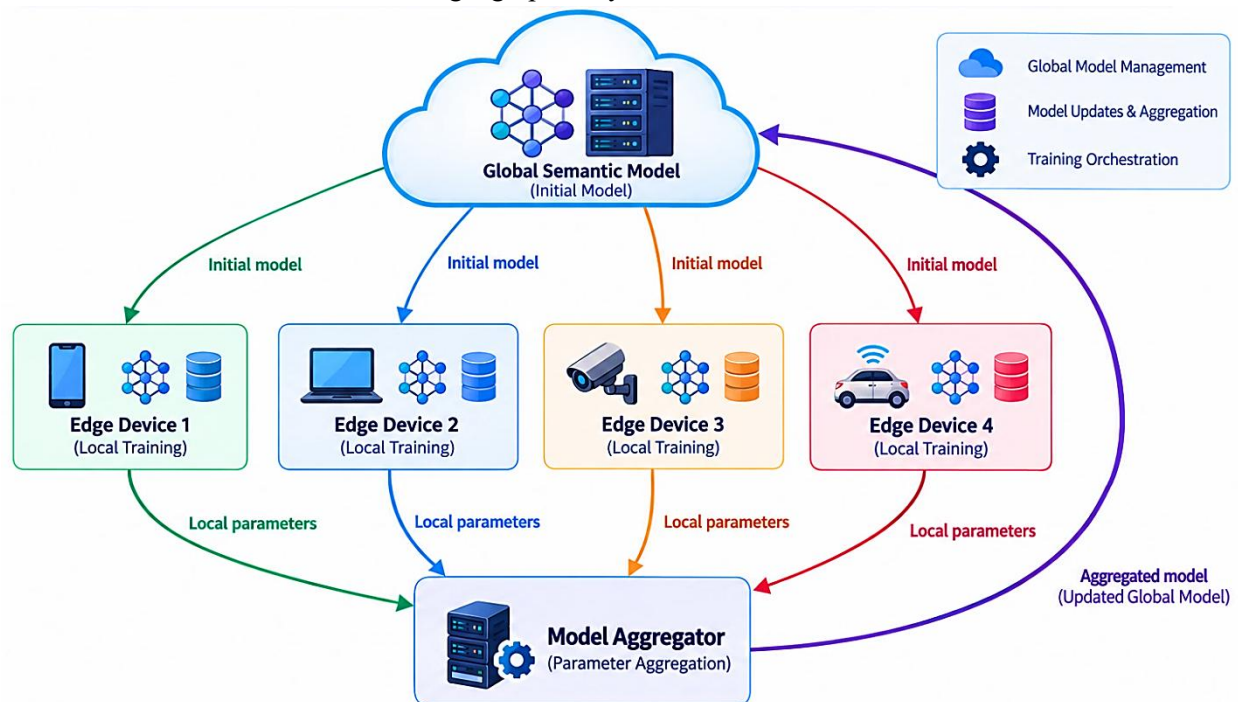


Figure 50: Distributed Semantic Model Training across Edge Devices

The figure illustrates a training architecture beginning with a Global Semantic Model maintained at the cloud level. The initial model is distributed to several edge devices, where local training is performed using device-specific data and computational resources. The edge devices shown in the figure represent different types of sources, including mobile devices, computers, cameras, and connected vehicles.

Because these devices can experience different environments and collect different types of information, their local training processes can capture diverse semantic characteristics. The resulting local parameters are then sent to the Model Aggregator, which combines the contributions from the participating edge devices.

After aggregation, an updated global model is generated and returned to the distributed edge devices. This creates an iterative learning process in which the global semantic model is progressively improved using knowledge obtained from multiple locations. The architecture separates local data processing from global model coordination, allowing devices to contribute to semantic intelligence while keeping their locally collected information at the edge. Such an approach can be beneficial when raw semantic data are large, sensitive, costly to transmit, or highly dependent on local environmental conditions.

Distributed training also supports adaptation to heterogeneous communication and sensing environments. Different edge devices may contribute knowledge associated with different application domains, such as mobile sensing, visual monitoring, connected transportation, or IoT services. The cloud can coordinate model management, updates, and training orchestration, while edge devices provide localized learning. Consequently, the architecture creates a collaborative relationship between global semantic intelligence and local semantic learning, enabling models to improve across distributed environments while reducing dependence on centralized raw-data collection. This makes distributed semantic model training an important foundation for scalable edge intelligence and adaptive semantic communication systems.

9.2.2 Federated Learning for Privacy-Aware Semantic Communication

Federated learning provides a distributed approach for developing semantic communication models while reducing the need to transfer raw data from edge devices to a central server. In conventional centralized learning, data collected from mobile devices, sensors, cameras, vehicles, and IoT systems may need to be transmitted to a common location for model training. This can create substantial communication overhead and raise privacy concerns when the collected information contains sensitive or locally generated content. Federated learning addresses this challenge by allowing participating devices to train semantic models locally and share model updates rather than directly sharing their raw datasets.

In privacy-aware semantic communication, each participating edge device receives a common semantic model and adapts it using its locally available data. The local training process enables the model to learn characteristics associated with the device's environment, users, sensing conditions, or application requirements. Instead of sending the underlying images, text, speech, sensor measurements, or other information to a central server, the device communicates selected model updates to a federated learning coordinator. The coordinator combines contributions from multiple participants to produce an improved global semantic model. The updated model can then be redistributed to the participating devices for further local adaptation.

Federated learning is particularly relevant to semantic communication because semantic models often depend strongly on contextual and application-specific information. A mobile device may encounter different language patterns from an industrial sensor, while a connected vehicle may observe completely different visual or environmental conditions. Federated learning enables these distributed devices to contribute their local knowledge without requiring all raw observations to be collected centrally.

Additional privacy mechanisms can be incorporated into the learning process to reduce the possibility of exposing sensitive information through model updates or participation patterns.

The approach also introduces challenges that must be considered in practical deployments. Edge devices can have different computational capabilities, communication resources, data distributions, and availability. Communication-efficient model updates are therefore important, particularly when many devices participate in training. Security mechanisms are also required to address potentially unreliable or malicious participants. Despite these challenges, federated learning provides an important foundation for privacy-aware semantic intelligence, allowing semantic models to benefit from distributed experiences while keeping locally generated information closer to its source. This makes it suitable for applications involving personal devices, healthcare systems, intelligent transportation, smart homes, industrial IoT, and other environments where both semantic intelligence and data privacy are important.

9.2.3 Collaborative Knowledge Sharing among Distributed Nodes

Collaborative knowledge sharing enables distributed edge nodes to exchange useful semantic information and collectively improve their understanding of an environment. In a large-scale semantic communication system, individual nodes may possess only partial knowledge because their observations are limited by geographical coverage, sensing capabilities, user interactions, or application context. A node responsible for one location may understand local traffic conditions, while another node may have information about nearby environmental events or connected devices. Sharing meaningful knowledge among these nodes allows the distributed system to construct a more comprehensive representation of the environment without requiring every node to independently acquire all relevant information.

Knowledge sharing can involve semantic representations, contextual information, learned features, entity relationships, event descriptions, and application-specific knowledge. Instead of exchanging large volumes of raw data, nodes can communicate information that has already been processed into meaningful representations. For example, an edge node receiving information from cameras and sensors may identify an event and share its semantic description with neighboring nodes. Other nodes can combine this information with their own observations to improve situation understanding. Such cooperation is especially valuable when individual sensing sources provide incomplete or complementary views of the same environment.

A collaborative semantic network can also maintain shared knowledge repositories or distributed knowledge graphs. These structures can represent relationships between entities, events, locations, activities, and contextual conditions. When new information becomes available, participating nodes can update their local knowledge and share relevant changes with other nodes. This enables semantic intelligence to evolve continuously as the environment changes. Knowledge exchange can also support more consistent semantic interpretation across different nodes by allowing them to use common concepts, representations, and contextual information.

However, collaborative knowledge sharing must account for differences in node capabilities, data quality, communication availability, and trustworthiness. Excessive knowledge exchange can increase communication overhead, while inaccurate or outdated information can negatively affect semantic interpretation. Therefore, nodes should determine which knowledge is sufficiently relevant and valuable to share. Mechanisms for knowledge validation, relevance assessment, synchronization, and access

control can improve the reliability of collaborative intelligence. Overall, collaborative knowledge sharing transforms independent edge nodes into a cooperative semantic intelligence system, allowing distributed devices to combine localized observations, improve contextual understanding, and support more effective task-oriented communication across large and dynamic environments.

9.2.4 Adaptive Semantic Learning in Heterogeneous Environments

Adaptive semantic learning addresses the challenge of developing communication models that can operate effectively across environments with different data characteristics, devices, users, applications, and network conditions. Heterogeneity is a fundamental feature of distributed semantic communication systems because edge devices can vary significantly in computational capability, memory, sensing modality, connectivity, and available training data. In addition, the semantic meaning associated with information can change between application domains. A representation that is highly useful for an autonomous vehicle may have limited relevance to an industrial monitoring system or a smart-home application. Adaptive learning enables semantic models to respond to these differences rather than relying on a single static representation.

A key requirement of adaptive semantic learning is the ability to learn from non-uniform and continuously changing data. Different edge nodes may observe different distributions of images, speech, sensor measurements, text, or environmental events. Local models can therefore develop specialized knowledge while maintaining compatibility with a broader semantic model. Training strategies can allow devices to adapt shared models according to their local requirements while incorporating useful knowledge obtained from other nodes. This approach supports personalization without completely isolating each device from the collective learning process.

Adaptation is also required when communication and environmental conditions change. Network congestion, mobility, interference, changing channel quality, and variations in available computing resources can affect how semantic information should be represented and transmitted. An adaptive semantic model can modify its processing behavior according to current conditions. For example, when communication resources become limited, the system can emphasize highly relevant semantic features and reduce less important information. When additional computing resources become available, more sophisticated semantic processing can be performed at the edge. This allows the learning and communication processes to remain aligned with application requirements.

Heterogeneous environments also require attention to model compatibility, resource limitations, and continual adaptation. Lightweight devices may not support the same model complexity as powerful edge servers, while some devices may participate only intermittently because of energy or connectivity constraints. Adaptive learning can address these differences through flexible model deployment, selective participation, knowledge transfer, and resource-aware training. The overall objective is to maintain reliable semantic understanding despite changes in data, devices, users, and network conditions. Consequently, adaptive semantic learning provides a foundation for scalable and resilient distributed intelligence, enabling heterogeneous edge environments to cooperate while retaining the flexibility required for different applications and operating conditions.

9.3 Semantic Communications for Internet of Things and Machine-to-Machine Systems

9.3.1 Meaning-Aware IoT Data Representation and Transmission

Internet of Things (IoT) systems generate large volumes of information from sensors, connected appliances, vehicles, industrial machines, cameras, and wearable devices. Conventional IoT communication generally focuses on transporting measured data reliably from a source device to a destination. However, many IoT applications do not require every generated measurement to be transmitted in its original form. Instead, the application may require only information that is relevant to a particular monitoring, control, or decision-making task. Meaning-aware IoT communication addresses this requirement by transforming raw observations into compact semantic representations before transmission.

At the source device or edge node, semantic processing can identify the features and information that are most relevant to the intended application. Temperature readings, for example, may be interpreted as a normal environmental condition, an abnormal increase, or a possible equipment fault depending on the surrounding context. Similarly, an industrial sensor network can represent machine observations in terms of operating status, detected anomalies, or maintenance requirements rather than continuously transmitting every raw measurement. This approach allows IoT systems to communicate information according to its meaning and application relevance.

Meaning-aware representation can also reduce unnecessary communication between resource-constrained IoT devices. Many sensors operate with limited battery power, processing capability, memory, and wireless bandwidth. Transmitting compact semantic information can reduce communication workload and allow devices to dedicate more resources to sensing and local processing. Semantic representations can be adapted according to the requirements of different applications, allowing safety-critical information to receive greater protection and priority than routine or redundant observations.

At the receiver, the semantic information can be interpreted and reconstructed according to the requirements of the target task. Exact reproduction of the original sensor stream is not always necessary when the application is primarily interested in a decision or event. For example, a smart factory may require information indicating that a machine is operating abnormally rather than the complete sequence of sensor measurements that led to that conclusion. Meaning-aware IoT communication therefore shifts the focus from transmitting all available data toward transmitting useful information for achieving the intended task. This approach can improve communication efficiency, reduce unnecessary network traffic, and support scalable IoT deployments involving large numbers of heterogeneous devices.

9.3.2 Context-Aware Communication among Intelligent Devices

Context-aware communication enables intelligent IoT and machine-to-machine devices to exchange information according to their current environment, operational state, communication conditions, and application requirements. In conventional machine-to-machine communication, devices often exchange predefined messages according to fixed protocols and rules. Such communication may not sufficiently account for changes in device status, network conditions, user requirements, or environmental situations. Semantic communication introduces the ability to interpret the meaning and relevance of exchanged information, allowing devices to make communication decisions based on context rather than data availability alone.

Context information can include device status, location, time, task requirements, environmental conditions, network state, and relationships among communicating devices. An intelligent device can use these factors to determine what information should be transmitted, how frequently it should be exchanged, and which information should receive priority. For example, connected vehicles can exchange safety-relevant information more urgently when detecting a potential collision, while routine status information can be transmitted less frequently. Similarly, industrial machines can communicate abnormal operating conditions immediately while reducing the exchange of repetitive information during stable operation.

Context-aware communication also supports cooperation among heterogeneous devices. IoT environments commonly contain sensors, actuators, cameras, vehicles, appliances, robots, and edge devices with different capabilities and responsibilities. Semantic representations provide a common basis for these devices to exchange meaningful information despite differences in their underlying data formats. A sensing device can provide a semantic description of an observed event, while another intelligent device can interpret that information according to its own task and available context. This enables more coordinated machine-to-machine interaction and reduces dependence on exchanging complete raw datasets.

The communication process can further adapt when the network environment changes. Congestion, mobility, interference, limited battery resources, or changing connectivity can influence the way devices exchange semantic information. Intelligent devices can prioritize essential information, reduce redundant transmissions, or adjust communication frequency according to current conditions. Feedback from receiving devices and applications can also help determine whether additional information is required. Consequently, context-aware communication creates a more flexible interaction model in which IoT devices do not simply exchange data but cooperate through contextually meaningful information. This capability is particularly valuable for smart manufacturing, connected transportation, intelligent buildings, healthcare monitoring, environmental sensing, robotics, and other distributed machine-to-machine applications.

9.3.3 Task-Oriented Machine-to-Machine Semantic Communication

Task-oriented machine-to-machine (M2M) semantic communication enables intelligent machines to exchange information according to the task they need to accomplish, rather than simply transferring complete data between devices. This approach is particularly relevant to industrial IoT, autonomous systems, robotics, and intelligent transportation, where machines frequently need to coordinate actions under strict latency and resource constraints. The uploaded figure is well aligned with this section because it illustrates how a task initiated by one machine is interpreted, converted into a semantic message, transmitted through a semantic network, and finally used by another machine for task execution.

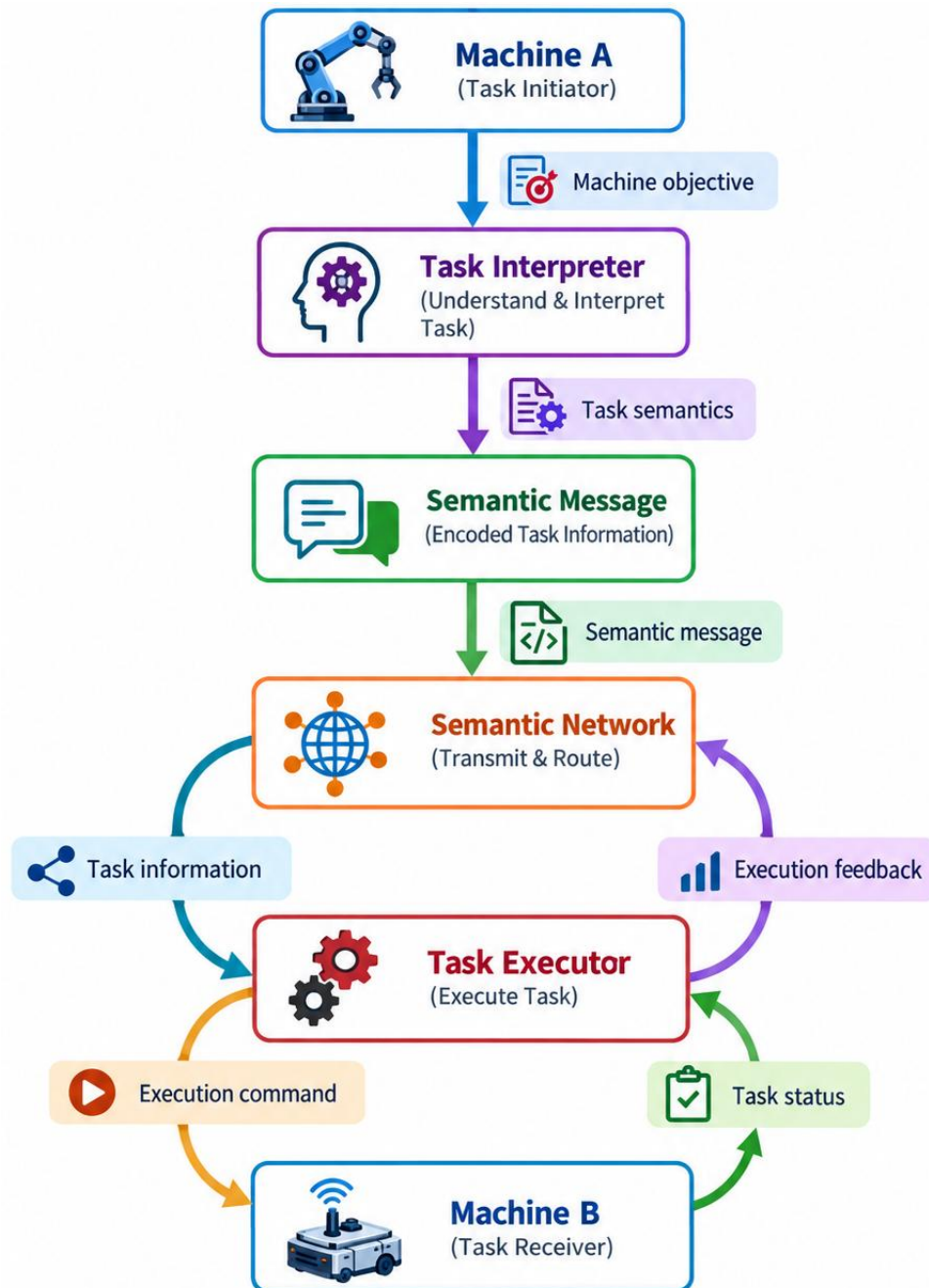


Figure 51: Task-Oriented Machine-to-Machine Semantic Communication

The figure begins with Machine A, which acts as the task initiator and generates a machine objective. The objective is processed by a Task Interpreter, which understands the requested operation and converts it into task semantics. These semantics are then represented as a Semantic Message, providing an encoded description of the information required for the task. This process demonstrates the central principle of task-oriented semantic communication: the transmitting machine does not necessarily need to communicate all of the raw information associated with its objective. Instead, it identifies and communicates the information that is meaningful for accomplishing the intended task.

The semantic message is subsequently delivered through the Semantic Network, which performs transmission and routing of task-related information. At the receiving side, Machine B acts as the task receiver and uses the received task information to perform the required operation through the Task Executor. The figure also illustrates an execution command flowing toward the receiving machine and a task-status response returning from it. This interaction creates a communication loop in which machines can exchange meaningful task information rather than relying only on conventional data transfer.

An important feature of the architecture is the execution feedback returning from the task executor toward the semantic network. This feedback can indicate whether the intended task was successfully performed and can provide information for subsequent communication decisions. If the task is incomplete or additional information is required, the semantic network can support further semantic exchanges between the machines. Thus, the architecture establishes a closed-loop M2M communication process connecting task interpretation, semantic representation, network delivery, task execution, and feedback. Such task-oriented communication can improve efficiency by focusing network resources on information that directly contributes to machine cooperation, making it suitable for autonomous robots, industrial automation, connected vehicles, and other intelligent machine-to-machine systems.

9.3.4 Semantic Coordination among Distributed IoT Systems

Semantic coordination enables multiple distributed IoT systems to cooperate by exchanging meaningful information rather than isolated raw sensor data. In a large IoT environment, different systems may contain heterogeneous sensors, devices, communication technologies, and application objectives. A semantic coordination layer can provide a common mechanism for interpreting local observations, integrating knowledge, and determining how information should be shared among participating systems. This makes the approach particularly suitable for smart cities, industrial automation, connected transportation, environmental monitoring, and other distributed IoT scenarios.

The figure illustrates three independent IoT systems, represented as IoT System A, IoT System B, and IoT System C, connected through a central Semantic Coordination Layer. Each IoT system contains different types of sensors and devices and produces local semantic information based on its observations. The coordination layer performs semantic reasoning, knowledge integration, and coordination management. Rather than requiring every IoT system to exchange all of its raw measurements, the layer can interpret the locally generated semantics and determine which information is relevant for cooperation with other systems.

The architecture also incorporates a Global Context, represented by a cloud or knowledge base. This component provides broader contextual knowledge to the semantic coordination layer, while coordinated knowledge can be distributed back toward the participating IoT systems. The bidirectional exchange shown in the figure allows local observations to contribute to a broader understanding of the environment while global context can influence how individual systems interpret and respond to their observations. Coordination policies are also communicated to the IoT systems so that their local semantic processing and interactions remain consistent with the overall coordination objective.

This architecture supports distributed decision-making while maintaining semantic consistency across heterogeneous IoT domains. For example, information generated by an industrial monitoring system could be combined with transportation or environmental information when these systems participate in a

common intelligent service. The central coordination layer can determine relationships among these semantic observations and support appropriate information sharing without requiring complete raw-data exchange. Consequently, semantic coordination provides a mechanism for connecting independent IoT systems through shared meaning, contextual knowledge, and coordinated policies, improving interoperability and enabling more intelligent distributed IoT services.

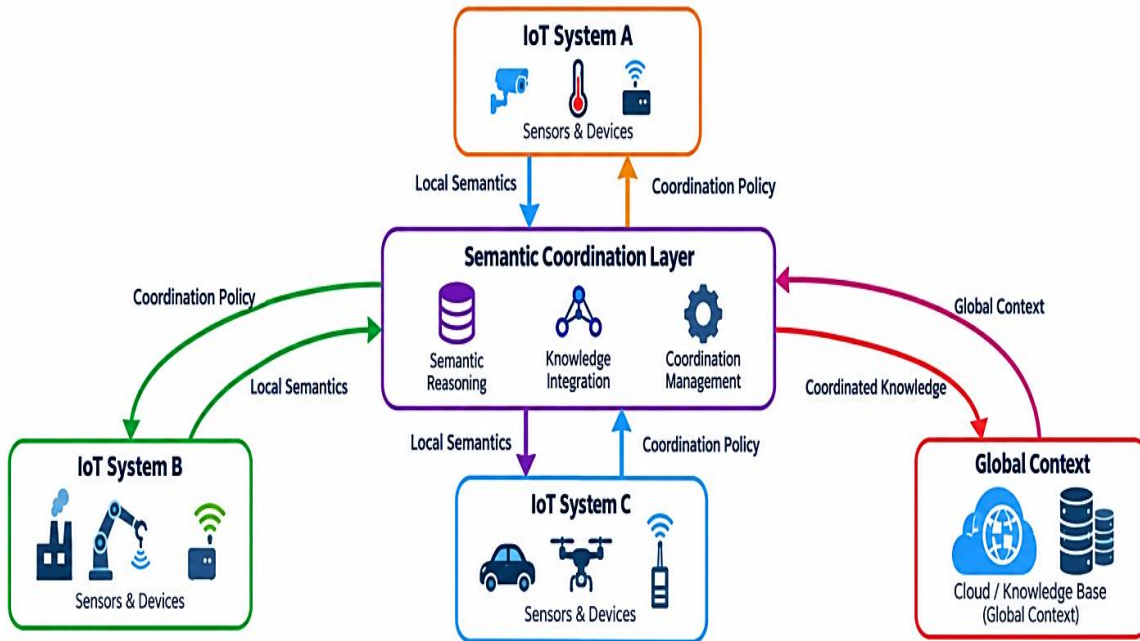


Figure 52: Semantic Coordination among Distributed IoT Systems

9.4 Energy-Efficient and Resource-Constrained Edge Semantic Networking

9.4.1 Computation-Communication Trade-offs in Edge Semantic Systems

Edge semantic communication systems must carefully balance computation cost and communication cost because edge devices typically operate with limited processing power, memory, battery capacity, and network resources. Unlike conventional communication systems, semantic communication introduces additional processing stages for extracting, encoding, interpreting, and reconstructing meaning. These operations can reduce the amount of information that needs to be transmitted, but they also require computational resources at the transmitter, edge node, or receiver. Therefore, improving communication efficiency does not necessarily guarantee lower overall energy consumption. The appropriate balance depends on the complexity of semantic processing and the amount of communication that can be avoided.

A major trade-off occurs when deciding whether semantic processing should be performed locally or whether raw or partially processed information should be transmitted to an edge server. Local semantic encoding can significantly reduce the size of the transmitted information by removing redundancy and retaining task-relevant features. This can reduce transmission time, spectrum usage, and radio-frequency energy consumption. However, sophisticated neural networks, transformer-based encoders, or multimodal semantic models may require substantial computational resources. For battery-powered IoT devices, excessive local processing can offset the energy savings obtained from communication reduction.

Edge computing provides an intermediate solution by transferring computationally demanding semantic operations from constrained devices to nearby edge nodes. An IoT device can perform lightweight preprocessing and transmit compact semantic features to an edge server, where more complex inference or reconstruction is performed. This arrangement can reduce the computational burden on the device while maintaining lower communication overhead than transmitting complete raw data. Nevertheless, transmitting information to the edge still consumes wireless resources, and the edge node itself has finite processing capacity. Consequently, task placement must consider both communication energy and computational energy. The trade-off can also be adapted according to application requirements and network conditions. When the wireless channel is congested or bandwidth is limited, stronger semantic compression and more local processing may be beneficial. When the device battery is critically low, computationally intensive processing may need to be shifted toward the edge. Similarly, latency-sensitive applications may require a carefully selected processing location to avoid delays caused by either excessive local computation or unnecessary data transmission. Thus, energy-efficient edge semantic networking should jointly consider transmission energy, computation energy, latency, semantic quality, and available resources. Rather than minimizing communication or computation independently, the objective is to determine an operating point that achieves the required semantic task performance with the lowest practical overall resource expenditure.

9.4.2 Energy-Aware Semantic Encoding and Model Selection

The figure presents an energy-aware semantic encoding framework in which the task and context are first considered together with the available workload. The system monitors the device's energy availability and uses this information to determine an appropriate encoding level. When the device faces a critical energy constraint, the system can move toward Level 1, Ultra-Low Energy, where maximum semantic compression is applied. This reduces the processing and communication requirements but produces a more compressed semantic representation. When the energy budget is normal, the system can operate at Level 2, Balanced Energy, which aims to provide an appropriate compromise between energy consumption and semantic quality.

When sufficient energy is available, the framework can select Level 3, High Accuracy, providing higher-fidelity semantic meaning at the expense of greater computational or communication requirements. The arrows between the three levels illustrate adaptive upgrading when additional resources become available and energy fallback when the energy budget becomes constrained. Therefore, the model-selection process is not fixed; it can dynamically adjust the semantic encoding strategy according to the current operating condition.

The final outputs from the different encoding levels are represented as compressed semantic meaning, optimized semantic meaning, or high-fidelity semantic meaning, which are then delivered as the semantic output. This illustrates an important principle of resource-constrained semantic networking: the best encoding model is not necessarily the most computationally sophisticated model, but the one that provides sufficient semantic quality within the available energy budget. Such adaptive model selection can help battery-powered IoT devices, mobile terminals, sensors, and edge devices maintain useful semantic communication even when their energy resources vary over time.

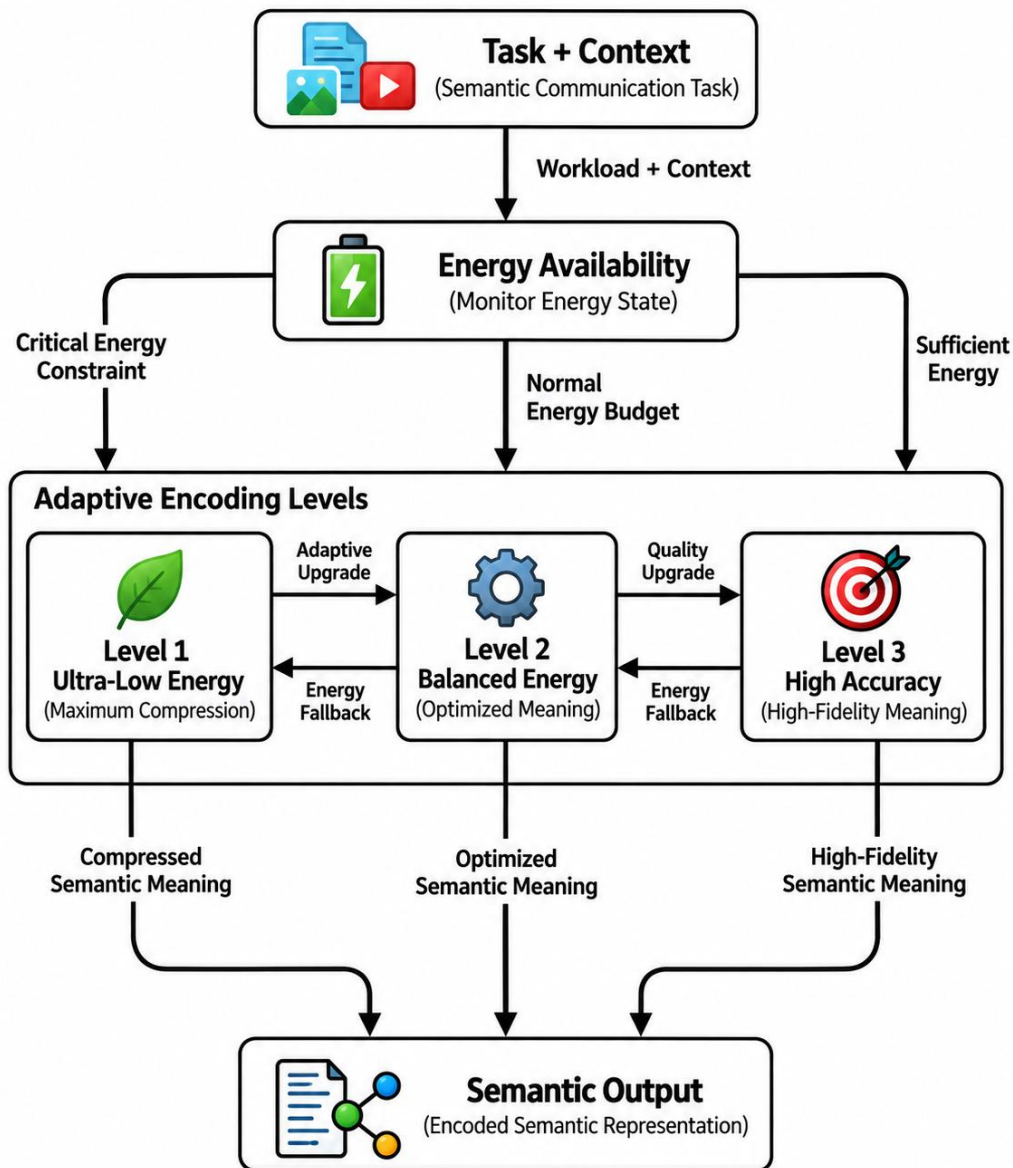


Figure 53: Energy-Aware Semantic Encoding and Model Selection

9.4.3 Adaptive Semantic Processing for Resource-Constrained Devices

Adaptive semantic processing is essential for edge and IoT devices that operate with limited computational power, memory, battery capacity, and communication bandwidth. Unlike conventional systems that may execute a fixed processing pipeline regardless of resource availability, semantic communication systems can dynamically adjust the complexity of semantic extraction, encoding, inference, and reconstruction. The objective is to maintain the semantic information required by the application while avoiding unnecessary expenditure of computational and communication resources. This capability is particularly important for battery-powered sensors, wearable devices, mobile terminals, embedded controllers, and low-power IoT nodes.

A resource-constrained device can select different semantic processing strategies according to its current operating state. When sufficient processing capacity and energy are available, a more sophisticated semantic model can extract richer features and produce higher-quality representations. Under constrained conditions, the device can switch to lightweight neural networks, reduced feature dimensions, lower encoding complexity, or more aggressive semantic compression. The system can therefore prioritize information that has the greatest relevance to the target task while reducing the processing of redundant or low-value information. Such adaptation enables the device to continue providing useful semantic information even when its resources fluctuate.

Adaptive processing can also distribute computational workloads between the device and an edge server. Lightweight preprocessing may be performed locally, while computationally intensive semantic inference or reconstruction can be transferred to a nearby edge node. The decision can depend on processor utilization, battery state, wireless channel quality, latency requirements, and the complexity of the current task. For example, a device with critically limited energy may transmit a compact semantic representation and rely on edge-side processing, whereas a device with sufficient resources may perform more semantic processing locally. This flexible division of computation helps prevent resource exhaustion while maintaining application performance.

Another important consideration is graceful degradation. A resource-constrained semantic system should not fail completely when resources become limited. Instead, it should progressively reduce semantic processing complexity while preserving the information most important to the application. Priority-based feature selection, adaptive model selection, early-exit inference, model compression, and dynamic encoding levels can support this behavior. The resulting system can respond to changing resource conditions without requiring a complete redesign of the communication process. Therefore, adaptive semantic processing provides a practical mechanism for achieving reliable and efficient operation across heterogeneous edge devices while maintaining an appropriate balance among semantic quality, computational complexity, energy consumption, communication overhead, and latency.

9.4.4 Sustainable and Energy-Efficient Semantic Networking

Sustainable semantic networking aims to reduce the overall energy and resource consumption of communication systems while maintaining the semantic quality required by applications. The rapid expansion of IoT devices, edge computing platforms, wireless networks, and intelligent services has increased the amount of data that must be sensed, processed, transmitted, and stored. Continuously transporting raw information can result in substantial network traffic and energy consumption. Semantic communication provides an opportunity to reduce this burden by transmitting information according to its relevance and meaning rather than attempting to reproduce every element of the original data.

Energy efficiency can be improved at several stages of the semantic communication pipeline. At the source, semantic encoders can remove redundant information and generate compact representations. During transmission, semantic importance can be used to determine which information requires greater protection, bandwidth, or retransmission priority. At the edge, intelligent resource management can determine where semantic processing should be performed according to computational and energy availability. At the receiver, semantic reconstruction can focus on recovering information that is

necessary for the target task instead of unnecessarily reproducing irrelevant details. These mechanisms can collectively reduce communication, computation, and storage requirements.

Sustainable networking also requires coordination between communication resources and computational resources. Increasing semantic model complexity may improve representation quality but can increase processor utilization and energy consumption. Conversely, excessive compression can reduce energy consumption while potentially degrading semantic accuracy. An energy-efficient system therefore needs to identify an appropriate operating point that satisfies application requirements without consuming unnecessary resources. Adaptive encoding, lightweight models, edge-assisted inference, intelligent scheduling, and task-oriented transmission can help maintain this balance. Resource allocation can also be adjusted according to traffic demand, device energy levels, network conditions, and the semantic importance of information.

The sustainability objective extends beyond individual devices to the entire network infrastructure. Reducing unnecessary transmissions can decrease wireless resource utilization and the processing workload of network and edge infrastructure. Cooperative processing among devices and edge nodes can further prevent duplicated computation and redundant information exchange. In dynamic environments, energy-aware policies can continuously adapt communication and processing strategies as workloads and resource conditions change. Consequently, sustainable semantic networking represents a shift toward meaning-driven resource utilization, where energy and network resources are devoted primarily to information that contributes to useful tasks. This approach can support the long-term scalability of IoT, edge intelligence, and future wireless semantic networks while reducing the resource burden associated with continuously transmitting and processing large volumes of raw data.

Chapter 10

Intelligent Applications and Emerging Use Cases of Semantic Communications

10.1 Semantic Intelligence for Autonomous and Collaborative Systems

10.1.1 Semantic Perception and Situation Awareness

Semantic perception enables autonomous and collaborative systems to transform raw observations into meaningful representations that support understanding and decision-making. Conventional sensing systems typically collect large amounts of information from cameras, microphones, radar, lidar, and other sensors, after which the data may be transmitted or processed for further analysis. In semantic communication, the objective is to identify the information that is relevant to the current task and represent it according to its meaning. This allows autonomous systems to focus on objects, events, relationships, and environmental conditions rather than processing every element of the sensed data equally.

Situation awareness extends semantic perception by combining observations with contextual information. An autonomous vehicle, for example, may identify vehicles, pedestrians, road boundaries, traffic signals, and potential obstacles from visual and sensing data. However, identifying these individual elements is only one part of understanding the situation. The system must also determine relationships and interactions, such as whether a pedestrian is approaching a crossing, whether another vehicle is changing lanes, or whether road conditions require a change in driving behavior. Semantic representations can combine these observations into a contextual description of the current environment.

Semantic communication is particularly useful when situation information must be exchanged among distributed autonomous systems. Instead of continuously transmitting complete sensor streams, a system can communicate task-relevant semantic information such as detected objects, hazards, intentions, or environmental changes. This can reduce communication requirements while allowing other agents to update their understanding of the shared environment. The usefulness of the transmitted information can therefore be evaluated according to its contribution to the receiving system's task rather than only according to traditional data reconstruction accuracy.

Effective semantic perception and situation awareness also require robustness to incomplete, uncertain, and changing observations. Autonomous systems frequently operate under dynamic conditions in which sensor measurements may be noisy, partially unavailable, or inconsistent. Semantic models can integrate information from multiple sources and use contextual relationships to maintain a useful representation of the environment. Consequently, semantic perception provides the foundation for intelligent autonomous operation by connecting sensing, contextual interpretation, and communication. Its integration with semantic communication allows distributed autonomous systems to exchange meaningful environmental knowledge efficiently and use that knowledge for safer and more informed decisions.

10.1.2 Intent-Aware Communication among Autonomous Agents

Intent-aware communication enables autonomous agents to communicate not only what they observe but also what they intend to do. In collaborative autonomous environments, information about future actions can be as important as information about the current state. A vehicle may need to indicate that it intends to change lanes, a robot may communicate that it is moving toward a particular location, or an autonomous drone may indicate that it is preparing to inspect a specific area. Communicating such intentions allows other agents to anticipate future actions and coordinate their own behavior accordingly.

Traditional communication systems generally focus on transferring predefined messages or sensor information between devices. Semantic communication introduces a task-oriented approach in which the transmitted representation can describe the meaning and purpose of the information. An autonomous agent can extract its current objective, operational state, and intended action and convert these elements into a semantic representation. The receiving agent interprets the representation according to its own context and determines how the information affects its task. This avoids requiring every agent to exchange large quantities of raw observations when a concise description of intent is sufficient.

Intent-aware communication is particularly important in environments containing multiple autonomous agents with interacting objectives. Consider a group of connected vehicles approaching an intersection. Instead of each vehicle continuously transmitting complete sensor measurements, vehicles can exchange semantic information about their intended movement, detected hazards, and priority conditions. Similarly, in collaborative robotics, one robot can communicate its intended trajectory or task while neighboring robots adjust their actions to avoid conflicts and improve cooperation. The semantic representation therefore becomes a mechanism for coordinating behavior among agents.

The effectiveness of intent-aware communication depends on accurate intent recognition and appropriate contextual interpretation. An agent must distinguish between current observations and intended future actions, while the receiver must understand the significance of the communicated intent within its own environment. Intent can also change rapidly as new observations become available, requiring communication policies to adapt accordingly. Reliability is especially important when incorrect or outdated intent information could result in unsafe coordination. Overall, intent-aware semantic communication provides autonomous agents with a richer interaction mechanism in which communication supports prediction, coordination, and task execution. By exchanging meaningful objectives and intended actions rather than unnecessary raw data, autonomous systems can achieve more efficient and responsive cooperation in transportation, robotics, aerial systems, industrial automation, and other multi-agent environments.

10.1.3 Cooperative Decision-Making through Semantic Knowledge Sharing

Cooperative decision-making allows multiple autonomous or intelligent systems to combine their knowledge and make decisions that would be difficult to achieve independently. In distributed environments, each agent generally has access to only a partial view of the overall situation. Different agents may observe different locations, sensing modalities, or aspects of the same event. Semantic communication provides a mechanism for sharing the most relevant knowledge among these agents without requiring the continuous exchange of complete raw datasets. The emphasis is placed on information that contributes directly to a shared task or decision.

Semantic knowledge sharing can involve objects, events, relationships, environmental conditions, task states, and contextual information. For example, autonomous vehicles operating in the same area can share information about detected obstacles, road conditions, traffic events, and anticipated movements. Each vehicle can combine this information with its own observations to form a more complete representation of the surrounding environment. Similarly, multiple robots working in an industrial facility can share information about machine states, task progress, workspace occupancy, and detected anomalies. This allows individual agents to make decisions using both local observations and knowledge obtained from cooperating agents.

A key advantage of semantic knowledge sharing is that it can reduce redundant information exchange. If several agents observe the same event, transmitting complete sensor data from every agent may consume considerable communication resources. Instead, agents can identify relevant semantic information and share only the knowledge that provides additional value. The receiving agent can then integrate the information into its existing semantic representation. Knowledge can also be prioritized according to task importance, urgency, confidence, and environmental conditions. This makes cooperative decision-making compatible with resource-constrained wireless and edge environments.

The quality of cooperative decisions depends on the consistency, timeliness, and reliability of shared knowledge. Agents may have different observations, models, or levels of confidence, and information can become outdated as the environment changes. Semantic knowledge-sharing mechanisms should therefore consider confidence, provenance, contextual validity, and update requirements. Feedback from decision outcomes can further help agents determine whether additional information is needed. Through this process, semantic communication becomes part of a closed-loop collaborative intelligence framework connecting observation, knowledge sharing, decision-making, and feedback. Cooperative semantic decision-making can consequently support autonomous transportation, multi-robot coordination, distributed surveillance, smart manufacturing, emergency response, and other applications in which intelligent agents must cooperate using limited communication resources.

10.1.4 Semantic Coordination of Multi-Agent Systems

Semantic coordination enables multiple autonomous agents to organize their activities by exchanging information about tasks, objectives, capabilities, states, and environmental conditions. Multi-agent systems can contain heterogeneous robots, vehicles, drones, sensors, software agents, or intelligent edge devices, each with different capabilities and responsibilities. Without an effective coordination mechanism, independent agents may perform redundant actions, compete for shared resources, or make decisions that negatively affect other agents. Semantic communication provides a common basis for representing and exchanging the information required to coordinate these distributed activities.

The coordination process can begin with the identification of individual or collective objectives. Agents can communicate task requirements and semantic descriptions of their current states, allowing other agents to understand what actions are being performed or planned. A coordination mechanism can then determine which agent is best suited for a particular task based on capabilities, location, resource availability, workload, and contextual conditions. For example, in a multi-robot system, one robot may be assigned to inspect an area while another transports an object and a third monitors the surrounding

environment. Semantic communication allows these assignments and relevant status information to be exchanged without requiring every agent to share complete sensor information.

Semantic coordination also supports dynamic adaptation when environmental or task conditions change. If an agent becomes unavailable, encounters an obstacle, or completes its assigned task, it can communicate the corresponding semantic state to other agents. The coordination mechanism can then redistribute tasks or modify the overall plan. Similarly, changes in network quality or available computational resources can influence how agents communicate and coordinate. Task-critical information can be prioritized, while redundant updates can be reduced to conserve communication resources. This makes semantic coordination particularly relevant to large-scale autonomous systems operating over wireless and edge networks.

An important requirement is maintaining a consistent understanding of shared tasks and agent states. Different agents may interpret observations differently because they use different models, sensing capabilities, or local contexts. Common semantic representations, contextual knowledge, and coordination policies can help reduce such inconsistencies. Feedback from task execution can further support continuous coordination by indicating whether planned actions achieved their intended outcomes. Thus, semantic coordination transforms a collection of autonomous agents into a cooperative intelligent system in which communication is directly connected to task allocation, knowledge sharing, action planning, and execution. This framework has significant potential for autonomous vehicles, collaborative robotics, UAV networks, smart manufacturing, disaster-response systems, and other emerging multi-agent applications.

10.2 Semantic Communications for Digital Twins and Cyber-Physical Systems

10.2.1 Semantic Data Exchange between Physical and Digital Entities

Semantic data exchange provides a meaningful communication mechanism between physical entities and their corresponding digital representations in digital twin and cyber-physical system (CPS) environments. Conventional data exchange often transfers raw measurements such as temperature, vibration, position, pressure, images, or machine states. Although this information can describe the physical environment, transmitting every measurement continuously can create substantial communication and processing requirements. Semantic communication instead focuses on extracting information that is relevant to the digital twin or CPS task and representing it in a compact, machine-interpretable form.

In a digital twin architecture, physical assets continuously generate observations that must be reflected in their digital counterparts. Sensors installed on industrial equipment, vehicles, buildings, or infrastructure can provide information about operational conditions. A semantic processing layer can interpret these observations and identify meaningful states or events, such as abnormal vibration, excessive temperature, component degradation, equipment failure risk, or changes in operating conditions. Rather than transmitting an entire sensor stream, the system can communicate semantic descriptions that capture the information required to maintain or update the digital representation.

The exchange is generally bidirectional because digital twins and CPSs also generate information that influences physical entities. The digital entity may analyze received observations, combine them with historical knowledge and simulation results, and generate control decisions or recommendations. These decisions can then be communicated back to the physical system. Semantic representations make this

interaction more efficient by allowing the digital and physical entities to exchange task-relevant information rather than repeatedly transferring complete datasets. The communication process can also incorporate context such as time, location, operational state, task requirements, and confidence in the detected information.

An important requirement is maintaining semantic consistency between the physical and digital domains. The receiving digital entity must correctly interpret the meaning of the transmitted information, while the physical system must understand commands or decisions generated by the digital counterpart. Common semantic models, ontologies, knowledge representations, and contextual information can support interoperability between heterogeneous devices and digital platforms. This approach is particularly useful in smart manufacturing, intelligent transportation, energy systems, healthcare, infrastructure monitoring, and industrial automation. By focusing data exchange on meaningful states, events, and task-relevant information, semantic communication can reduce unnecessary traffic while supporting continuous interaction between physical entities and their digital representations.

10.2.2 Meaning-Aware Digital Twin Synchronization

Meaning-aware synchronization enables a digital twin to maintain an appropriate representation of its physical counterpart by prioritizing changes that are semantically important. A digital twin may receive information continuously from sensors, machines, vehicles, cameras, or other physical sources. However, not every measurement represents a meaningful change in the physical system. Repeated transmission of stable or redundant measurements can consume communication bandwidth, processing capacity, and energy without significantly improving the digital twin. Semantic communication addresses this issue by determining which changes have sufficient relevance to justify synchronization.

The synchronization process can begin with semantic analysis of physical observations. The system identifies changes in operational state, detected events, abnormal conditions, component behavior, or environmental context. For example, a manufacturing machine operating within normal parameters may generate numerous measurements without requiring frequent digital-twin updates. If the system detects a significant deviation in temperature or vibration, the corresponding semantic event can be transmitted with higher priority. The digital twin can then update its state based on the meaningful change rather than processing every individual sensor measurement.

Meaning-aware synchronization can also be adaptive to the requirements of the application. Safety-critical systems may require frequent synchronization for important operational variables, whereas less critical information can tolerate longer update intervals. The synchronization policy can therefore consider semantic importance, task requirements, temporal relevance, confidence, and network conditions. If communication resources are limited, the system can prioritize updates that have the greatest impact on decision-making or system control. This creates a semantic synchronization mechanism in which communication frequency and information detail depend on the significance of changes in the physical environment.

The digital twin can also provide feedback to guide future synchronization. After receiving an update, the digital representation can determine whether its current state is sufficiently accurate for the intended task or whether additional information is required. This creates a closed-loop relationship between physical

observations, semantic extraction, transmission, digital-state updating, and feedback. Such a framework can improve communication efficiency while preserving the information necessary for accurate digital-twin operation. Meaning-aware synchronization is particularly valuable for industrial digital twins, autonomous vehicles, smart infrastructure, energy networks, and other CPS applications where physical conditions can change rapidly but only some changes are important for the current task. The fundamental objective is to keep the digital representation semantically synchronized, rather than simply attempting to reproduce every physical measurement continuously.

10.2.3 Semantic State Representation for Cyber-Physical Systems

Semantic state representation provides a structured description of the operational condition of a cyber-physical system by converting heterogeneous physical observations into meaningful system states. CPS environments combine physical processes with computation, communication, sensing, and control components. These systems continuously generate measurements from sensors and devices, but raw measurements alone may not provide an efficient representation of the overall system condition. Semantic state models can combine multiple observations and contextual information to describe higher-level conditions that are directly relevant to monitoring, control, and decision-making.

A semantic state can represent information such as operational mode, equipment condition, detected events, environmental status, resource availability, or system health. For example, a smart manufacturing system may use multiple temperature, vibration, pressure, and current measurements to determine that a machine is operating normally, experiencing an abnormal condition, or approaching a potential failure state. Similarly, an autonomous vehicle can combine camera, lidar, positioning, and vehicle-status information into a semantic state describing road conditions, nearby objects, traffic situations, and vehicle operating conditions. This representation provides a more meaningful basis for communication and control than transmitting every individual measurement independently.

Semantic state representation also supports communication between different components of a CPS. Sensors can provide observations to edge processors, digital twins can maintain corresponding system states, and controllers can use semantic information to determine appropriate actions. Because different components may have different computational capabilities and communication constraints, semantic representations can be designed at different levels of abstraction. Low-level devices may generate compact states, while edge or cloud systems can integrate these states with historical information, domain knowledge, and broader contextual information.

Maintaining an accurate semantic state requires continuous updating as the physical environment changes. The system should distinguish between significant state transitions and minor variations that have little effect on the target task. Semantic communication can therefore prioritize state changes according to their importance and urgency. Confidence information can also be associated with semantic states when observations are incomplete or uncertain. This is particularly important in CPS environments where incorrect state interpretation can influence subsequent control decisions. Overall, semantic state representation creates a bridge between raw physical measurements and higher-level cyber-physical intelligence. It allows CPS components to communicate operational meaning efficiently, supports context-aware decision-making, and provides a common representation for monitoring, control, digital-twin synchronization, and intelligent automation.

10.2.4 Real-Time Semantic Interaction between Digital and Physical Worlds

Real-time semantic interaction enables continuous and meaningful communication between physical systems and their digital representations while responding to rapidly changing environmental and operational conditions. Digital twins and CPSs frequently require information to move in both directions: physical devices provide observations to digital systems, while digital systems generate decisions, predictions, or control instructions for physical devices. Conventional communication can create unnecessary overhead when large volumes of raw data are exchanged continuously. Semantic communication provides an alternative by transmitting information according to its relevance to the current task and system state.

The physical-to-digital direction begins with sensors, machines, vehicles, cameras, and other devices observing the environment. Semantic processing can convert these observations into meaningful events, states, or relationships. A digital twin can use the resulting semantic information to update its representation, perform analysis, detect anomalies, or predict future conditions. For example, an industrial digital twin may receive a semantic description indicating that a machine has entered an abnormal operating state rather than receiving every raw sensor measurement. The digital system can then perform additional analysis and determine whether maintenance, control adjustment, or further sensing is required.

The digital-to-physical direction completes the interaction loop. Based on the updated digital state, simulation results, learned models, or task requirements, the digital system can generate a semantic command or recommendation. The physical system interprets this information and performs an appropriate action. Feedback regarding execution status can then be returned to the digital environment. This creates a closed-loop process connecting physical observation, semantic interpretation, digital reasoning, action generation, physical execution, and feedback. Because the loop can operate continuously, the system must consider latency, reliability, semantic accuracy, and changing network conditions.

Real-time semantic interaction is particularly important for applications where decisions must be made quickly. Autonomous transportation systems may need to respond to changing traffic conditions, industrial CPSs may need to react to equipment anomalies, and smart infrastructure may need to respond to environmental events. In such situations, transmitting only the information necessary for the current task can help reduce communication delay and processing requirements. Adaptive semantic policies can further prioritize urgent events and reduce the transmission of redundant information. Thus, real-time semantic interaction provides a foundation for tightly coupled digital and physical systems in which communication is not simply about transferring data, but about maintaining timely shared understanding and coordinated action between the physical and digital worlds.

10.3 Human–Machine and Human–AI Semantic Communication

10.3.1 Natural Language-Based Human–Machine Interaction

Natural language-based human–machine interaction provides a direct mechanism through which humans can communicate objectives, requests, questions, and instructions to intelligent machines using ordinary language. Conventional machine interfaces often require structured commands, predefined control sequences, or specialized input formats. Such requirements can make interaction difficult for users who are not familiar with the underlying system. Semantic communication enables machines to focus on the

meaning of human input and translate natural-language expressions into representations that can be processed by the target system.

The interaction process generally begins when a user provides a spoken or written request. Language-processing components tokenize and interpret the input, identify relevant linguistic features, and generate a semantic representation. Rather than treating the complete sentence as the final communication payload, the system can extract the information necessary for the intended task. For example, a user may instruct a service robot to move an object to a particular location. The system needs to understand the action, object, destination, and relevant constraints rather than simply reproducing the original sentence. This task-oriented interpretation can make communication more efficient and better aligned with machine capabilities.

Context plays an important role in natural-language interaction. The meaning of a statement can depend on previous dialogue, environmental conditions, user preferences, and the current machine state. A command such as move it there cannot be correctly interpreted without knowing what it and there refer to. Semantic systems can therefore combine linguistic information with contextual knowledge and information from sensors or other system components. This allows the machine to construct a more complete representation of the user's intended meaning before generating a response or executing an action.

Natural-language semantic interaction is also relevant to multimodal human-machine systems. Human instructions may be combined with images, gestures, audio, or environmental observations. A machine can integrate these modalities to determine the intended task and communicate an appropriate response. However, reliable interaction requires accurate language understanding, ambiguity resolution, and robust handling of incomplete or noisy input. The system should also distinguish between literal linguistic content and the actual task objective. By focusing on meaning rather than merely transmitting words, natural language-based semantic communication can support more flexible and intuitive interaction between humans and robots, vehicles, intelligent appliances, industrial systems, and AI-enabled services.

10.3.2 Intent and Context Understanding in Human-AI Communication

Intent and context understanding are fundamental components of effective human-AI semantic communication because the literal content of a message does not always fully describe what a user wants to accomplish. A human request may contain implicit objectives, references to previous interactions, domain-specific terminology, or information that depends on the surrounding environment. An intelligent system therefore needs to determine both the intended task and the contextual conditions under which the request should be interpreted. Semantic communication provides a framework for transforming natural-language input into representations that capture these elements.

Intent recognition identifies the primary objective expressed by the user. For example, a request may correspond to information retrieval, device control, navigation, scheduling, monitoring, or another application-specific task. Entity and parameter extraction can then identify the objects, locations, times, quantities, or other attributes required to complete the task. This creates a structured semantic representation that can be passed to downstream AI components. Such processing allows the

communication system to focus on information that contributes directly to task completion rather than treating every linguistic element as equally important.

Context understanding further improves interpretation by incorporating information from previous conversations, user interactions, environmental observations, and domain knowledge. The same phrase may have different meanings depending on the context in which it is used. For instance, a command involving a vehicle could refer to a particular connected car when the system is operating in a transportation environment, whereas the same term could represent a different entity in another application. Context-aware semantic models can use these relationships to resolve ambiguity and generate more appropriate representations.

Human–AI communication can also operate as a feedback loop. After interpreting the user's intent, an AI system may generate a response, request clarification, or execute an action. The outcome of the interaction provides additional information that can refine subsequent interpretation. When communication resources are limited, semantic representations can prioritize intent, task parameters, and critical contextual information. This is especially valuable for voice assistants, autonomous systems, collaborative robots, intelligent vehicles, and edge AI applications. Effective intent and context understanding therefore allows human–AI communication to move beyond simple language transmission toward goal-oriented interaction, where the communication process is evaluated according to how accurately the AI system understands and supports the user's intended objective.

10.3.3 Personalized Semantic Information Exchange

Personalized semantic information exchange adapts communication according to the characteristics, preferences, requirements, and current context of individual users. In conventional communication systems, information is often transmitted using a general representation that is independent of the recipient. However, different users may have different information needs, levels of expertise, priorities, and preferred forms of interaction. Semantic communication provides an opportunity to identify the information that is most relevant to a particular user and construct a representation that supports the user's specific task.

Personalization can begin with the semantic interpretation of user requirements. A system can consider information such as the user's current task, interaction history, preferred level of detail, and contextual situation when determining what information should be communicated. For example, an expert user may require detailed technical information, whereas another user may need only a concise description of the most important result. In an intelligent transportation system, one user may prioritize travel time while another may be more concerned with safety or energy efficiency. The semantic communication system can therefore adapt the transmitted information according to the target user's requirements.

Personalized exchange is also useful when communication resources are constrained. Instead of transmitting all available information and allowing the receiver to filter it, the system can perform semantic selection at the source or edge. Information with high relevance to the user's current objective can receive greater priority, while redundant or less useful content can be compressed or omitted. This can reduce bandwidth and processing requirements while improving the usefulness of the received

information. Personalization can also be applied to multimodal information by determining whether text, images, audio, or other representations are most appropriate for a particular user and situation.

Privacy and adaptability must be considered when implementing personalized semantic communication. User-specific information can improve semantic relevance, but inappropriate handling of personal context may introduce privacy concerns. Systems should therefore apply appropriate access controls and avoid communicating information that is unnecessary for the intended task. Personalization should also adapt over time as user requirements and contexts change. Feedback from previous interactions can help refine semantic preferences, while current context can override historical assumptions when necessary. Overall, personalized semantic information exchange transforms communication from a generic data-delivery process into a user-oriented information service. By selecting and representing information according to individual requirements, semantic communication can improve relevance, efficiency, and interaction quality across human–AI assistants, healthcare services, smart environments, autonomous systems, and personalized digital applications.

10.3.4 Human-Centric Meaning Preservation and Trust

Human-centric meaning preservation focuses on ensuring that information received by a person or AI-assisted application retains the meaning that was important in the original communication. In semantic communication, exact reproduction of the transmitted data is not always the primary objective. Instead, the system attempts to preserve the information necessary for the intended task. For human–machine and human–AI interaction, however, preserving meaning also requires considering how people interpret information, whether important details are omitted, and whether the resulting representation supports an appropriate decision.

Meaning preservation can be evaluated by examining the relationship between the original intent and the reconstructed or interpreted information. A communication system may compress a message substantially while still preserving its essential objective. For example, a long operational description can potentially be represented by a concise semantic message containing the key event, affected entity, required action, and relevant constraints. However, information that appears secondary from a computational perspective may be important to a human decision-maker. Human-centric semantic systems therefore need to consider task relevance, context, ambiguity, uncertainty, and the potential consequences of information loss.

Trust is closely connected to meaning preservation. Users need confidence that an AI system has correctly interpreted their request and that the information presented in response is consistent with the underlying communication. This is particularly important in applications involving autonomous systems, healthcare, industrial control, transportation, and other situations where incorrect interpretation can have significant consequences. Confidence indicators, provenance information, consistency checks, and explicit clarification mechanisms can help users understand the reliability of semantic interpretations. When uncertainty is high, the system should be capable of requesting additional information rather than presenting an uncertain interpretation as definitive.

Human-centric semantic communication should therefore treat users as active participants rather than passive recipients of reconstructed information. User feedback can be incorporated to determine whether the received meaning was understood correctly and whether the information provided was sufficient for

the intended task. Such feedback can help refine future semantic representations and improve system reliability. Ultimately, human-centric meaning preservation connects semantic accuracy, task relevance, transparency, uncertainty management, and user trust. By ensuring that communication remains understandable and faithful to the intended objective, semantic systems can support more dependable interaction between humans, machines, and AI while maintaining the efficiency advantages of meaning-oriented communication.

Chapter 11

Security, Privacy, Trust, and Robustness in Semantic Communication Networks

11.1 Emerging Security Threats against Semantic Communication Systems

11.1.1 Semantic Information Manipulation and Injection Attacks

Semantic communication systems introduce security challenges that differ from those of conventional bit-oriented networks because the transmitted information is processed according to its meaning, context, and task relevance. In a conventional system, an attacker may attempt to modify bits, packets, or messages. In a semantic system, however, an attacker can target the meaning extracted from the information. Semantic information manipulation occurs when an adversary deliberately modifies transmitted content, semantic features, contextual relationships, or task information so that the receiver interprets a different meaning from the one intended by the legitimate sender.

Semantic injection represents another important threat in which an attacker introduces misleading or unauthorized information into the semantic processing pipeline. Such information may influence semantic encoders, knowledge bases, language models, vision models, or task-oriented decision mechanisms. For example, an attacker could attempt to introduce a false event into a smart-city sensing system or manipulate information used by an autonomous system. Because semantic communication often performs information reduction before transmission, the receiver may rely heavily on a relatively compact representation. Manipulating a small number of highly important semantic features can therefore potentially have a disproportionate effect on the resulting interpretation.

The risk becomes greater when semantic communication systems integrate contextual knowledge and artificial intelligence. A manipulated semantic representation may appear plausible to the receiver because it is consistent with part of the available context. Knowledge-based reasoning systems may consequently propagate an incorrect interpretation to downstream applications. The impact can include incorrect classification, inappropriate resource allocation, erroneous task execution, or degradation of system reliability. Semantic manipulation can therefore affect not only confidentiality or data integrity but also the correctness of application-level decisions.

Protection against semantic manipulation requires security mechanisms that operate across multiple layers. Authentication and integrity protection can help establish whether information originates from a trusted source, while semantic consistency checks can identify contradictions between received information and established contextual knowledge. Confidence assessment, provenance tracking, anomaly detection, and cross-source validation can provide additional protection. Systems can also assign different security levels to semantic information according to its importance and potential impact. Consequently, security in semantic communication must extend beyond protecting bits and packets toward protecting the integrity, authenticity, and intended meaning of information throughout the semantic processing pipeline.

11.1.2 Adversarial Attacks against Semantic Learning Models

Semantic communication systems increasingly rely on machine learning models for semantic representation, feature extraction, encoding, decoding, reconstruction, and task inference. This dependence introduces vulnerabilities to adversarial attacks, in which carefully designed input modifications cause a learning model to produce incorrect semantic interpretations. Unlike conventional communication attacks that primarily target transmitted signals or network protocols, adversarial attacks can directly target the intelligence responsible for understanding and reconstructing meaning.

An adversarial modification may be visually, linguistically, or numerically small while producing a significant change in model output. For example, a slightly altered image could cause a semantic vision model to misclassify an important object, while a carefully modified linguistic input could cause a language model to infer an incorrect intent. In semantic communication, such errors can propagate through the communication pipeline because the semantic encoder may extract incorrect features and transmit them as if they represented legitimate information. The receiving decoder can then reconstruct or interpret the manipulated representation, potentially leading to incorrect task execution.

Adversarial threats can occur at different stages of the semantic learning lifecycle. During inference, attackers may manipulate source inputs to influence semantic extraction. During training, poisoned data can be introduced so that the learned model develops undesirable behavior. Model updates can also become targets when distributed or federated learning is used across edge devices. A compromised participant may provide malicious model updates that influence the global semantic model. These threats are particularly important when semantic models are shared across many devices and applications because a compromised model can potentially affect multiple communication services.

Robustness against adversarial attacks requires a combination of secure model development, robust training, anomaly detection, input validation, and model integrity mechanisms. Semantic models can be trained using diverse and perturbed examples to improve their resilience to unexpected inputs. Detection mechanisms can identify unusual semantic representations or inconsistent model behavior. Critical applications can additionally compare semantic interpretations across multiple models or information sources before executing high-impact actions. Secure model distribution and update mechanisms are also necessary to prevent unauthorized modifications.

Ultimately, adversarial robustness in semantic communication should be evaluated not only by whether the model's prediction remains correct but also by whether the intended meaning and task outcome remain reliable. This makes adversarial security an important research area for AI-enabled semantic communication systems deployed in autonomous vehicles, industrial automation, healthcare, intelligent IoT, and other safety- or mission-critical environments.

11.1.3 Knowledge Base and Model Integrity Attacks

Knowledge bases and AI models form critical components of semantic communication systems because they provide the information and intelligence required to interpret, encode, decode, and reconstruct meaning. A knowledge base may contain entities, relationships, contextual information, domain rules, and semantic concepts, while machine learning models capture patterns required for semantic processing.

Attacks against the integrity of these resources can therefore affect the entire communication pipeline, even when the underlying wireless transmission remains secure.

A knowledge-base integrity attack occurs when an attacker modifies, inserts, deletes, or corrupts semantic knowledge. For example, false relationships may be introduced into a knowledge graph, legitimate entities may be altered, or outdated information may be deliberately retained. Because semantic communication systems can use knowledge bases to interpret incomplete or ambiguous information, corrupted knowledge may lead to incorrect conclusions. An apparently valid received semantic representation could therefore be interpreted incorrectly because the contextual knowledge used by the receiver has been manipulated.

AI models can similarly be targeted through unauthorized modification, malicious model updates, training-data poisoning, or compromised model repositories. A manipulated semantic encoder may extract incorrect features, while a compromised decoder may reconstruct misleading information. In distributed environments, the risk becomes more complex because models may be trained and updated across multiple edge devices. An unauthorized or compromised participant could attempt to introduce malicious updates into a shared learning process. Model integrity is consequently essential for maintaining consistent semantic interpretation across distributed communication nodes.

Protection requires mechanisms for authentication, authorization, provenance, integrity verification, secure model distribution, and controlled updating. Knowledge-base changes should be traceable to trusted sources, while model updates should be validated before deployment. Version control and consistency checking can help detect unexpected changes in semantic knowledge or model behavior. Distributed systems can additionally use secure aggregation and participant validation when collaboratively training semantic models.

Integrity protection should extend beyond the physical storage of models and knowledge toward their functional behavior. A model may remain unchanged at the file level but still produce unexpected outputs after encountering manipulated inputs or incompatible knowledge. Continuous monitoring can therefore assess semantic consistency and detect significant deviations from expected behavior. Maintaining trusted models and knowledge bases is fundamental because semantic communication depends directly on the correctness of the intelligence used to interpret meaning. Strong integrity mechanisms consequently help ensure that semantic representations remain authentic, consistent, reliable, and suitable for downstream task execution.

11.1.4 Denial-of-Service and Semantic Resource Exhaustion

Denial-of-Service (DoS) attacks represent a major availability threat to semantic communication networks because they attempt to overwhelm communication, computation, or service resources and prevent legitimate users from receiving adequate service. In conventional networks, an attacker may generate excessive traffic to consume bandwidth, processing capacity, memory, or connection resources. Semantic communication introduces additional targets because semantic encoding, decoding, inference, reconstruction, knowledge retrieval, and AI model execution can require substantial computational resources. Consequently, an attacker may attempt to exhaust semantic-processing resources even without completely saturating the physical communication channel.

Semantic resource exhaustion can occur when malicious or unnecessary information causes repeated and computationally expensive semantic processing. For example, a large number of complex requests could force edge nodes to execute resource-intensive inference operations. Multimodal inputs can be particularly demanding because processing combinations of text, images, speech, and video may require multiple AI models. If an attacker intentionally generates workloads that trigger expensive semantic operations, edge devices may experience increased latency, energy consumption, and computational overload. This can reduce the resources available for legitimate applications.

The consequences can extend beyond reduced throughput. In latency-sensitive applications, resource exhaustion can prevent critical semantic information from being processed within the required time. Autonomous systems, industrial control, emergency communication, and healthcare applications may therefore experience degraded task performance even if some network connectivity remains available. Semantic-aware systems must consequently distinguish between legitimate high-priority workloads and malicious or excessive requests while maintaining fair access to limited resources.

Protection mechanisms can combine traffic filtering, admission control, workload prioritization, resource quotas, anomaly detection, and adaptive semantic processing. High-priority information can be protected through appropriate resource reservations, while suspicious or unusually expensive requests can be restricted. Lightweight semantic models can also be used when computational resources become constrained, allowing essential services to remain operational. Continuous monitoring of communication traffic and semantic-processing workloads can help identify abnormal patterns before they cause severe degradation.

11.2 Privacy-Preserving Mechanisms for Semantic Information Exchange

11.2.1 Privacy Risks Associated with Semantic Representations

Semantic communication introduces privacy challenges because semantic representations can contain information that is more directly related to the identity, behavior, preferences, activities, or intentions of users than conventional low-level data. A semantic encoder may extract information from text, images, speech, sensor observations, or multimodal inputs and convert it into compact representations. Although this process reduces the amount of transmitted information, the resulting representation may still reveal sensitive characteristics if an unauthorized party can analyze or correlate it with other information. Therefore, reducing data volume does not automatically guarantee privacy.

One important risk is information inference from semantic representations. Even when explicit identifiers are removed, semantic features can contain patterns that allow an observer to infer attributes about the source or user. For example, a representation generated from speech may reveal linguistic characteristics, while information derived from images may indicate locations, activities, or environmental conditions. Similarly, semantic representations generated from IoT devices can reveal behavioral patterns or operational states. The receiver may therefore obtain information beyond what was explicitly intended for the communication task.

Context-aware semantic systems can introduce additional privacy concerns because they may use information about users, locations, historical interactions, device states, and application requirements. Combining these contextual elements can create detailed profiles of users or environments. In distributed

systems, semantic information may also be stored at edge nodes, cached for later use, or shared among multiple devices. Each additional processing or storage location can increase the number of potential exposure points. Privacy protection must therefore consider the complete lifecycle of semantic information rather than only the wireless transmission stage.

Privacy risks can be reduced through data minimization, semantic feature selection, access control, secure storage, encryption, privacy-preserving learning, and controlled information sharing. A semantic system should transmit only the information necessary for the intended task and avoid unnecessarily exposing detailed contextual attributes. Privacy requirements can also influence semantic representation design, with sensitive features suppressed, transformed, or processed locally whenever appropriate. These mechanisms should be balanced against semantic fidelity because excessive privacy protection may remove information required for successful task execution. Consequently, privacy-aware semantic communication requires a careful balance between information usefulness, semantic quality, and protection against unauthorized inference.

11.2.2 Secure Semantic Encoding and Information Transmission

Secure semantic encoding and transmission aim to protect meaningful information throughout the communication process while maintaining sufficient semantic quality for the intended application. In conventional communication systems, security mechanisms commonly focus on protecting transmitted bits through encryption, authentication, and integrity mechanisms. Semantic communication requires these protections to be considered together with the semantic processing pipeline because sensitive information may be exposed during semantic extraction, representation generation, transmission, decoding, or reconstruction.

At the transmitter, semantic encoding can incorporate privacy-aware feature selection and secure representation mechanisms. Instead of transmitting all extracted semantic features, the encoder can select only those necessary for the target task. Sensitive attributes that do not contribute to the application objective can be suppressed or transformed. Local processing can also be used to extract semantic information without sending the underlying raw data to external infrastructure. This approach reduces the amount of information exposed during transmission and can be particularly useful for edge devices handling personal, industrial, or environmental information.

During wireless transmission, conventional security mechanisms such as encryption and authentication remain important. Encryption can protect semantic representations from unauthorized observation, while authentication can help verify the identity of communicating entities. Integrity mechanisms can detect unauthorized modification of transmitted information. However, security should also consider semantic-level manipulation because an attacker may attempt to modify important semantic features without necessarily causing obvious low-level communication errors. Semantic consistency checking and contextual validation can therefore complement conventional communication security mechanisms.

At the receiver, secure semantic decoding should ensure that received representations are interpreted only by authorized components and that reconstructed information remains consistent with the intended communication. Access-control mechanisms can determine which applications, users, or devices are permitted to access particular semantic information. Secure model execution and protected semantic

knowledge repositories can further reduce the risk of unauthorized disclosure. In edge-cloud architectures, security policies should remain consistent across local edge nodes and centralized cloud infrastructure.

Overall, secure semantic encoding and transmission require a multi-layer protection strategy covering source processing, semantic representation, wireless transmission, decoding, and application access. The system should minimize unnecessary information exposure while preserving the semantic information needed for task completion. This is especially important for healthcare, smart-home, industrial IoT, autonomous systems, and human–AI communication, where semantic information can contain highly valuable contextual and behavioral knowledge. Effective security therefore requires protecting not only the communication channel but also the meaning-bearing information and the intelligent processing mechanisms through which that information is generated and interpreted.

11.2.3 Federated and Distributed Privacy-Preserving Learning

Federated and distributed learning provide important mechanisms for developing semantic intelligence while reducing the need to centralize raw training data. Semantic communication systems often rely on machine learning models trained using text, images, speech, sensor measurements, and other potentially sensitive information. Centralizing these datasets can create privacy risks and increase the communication burden associated with transferring large volumes of raw data. Federated learning addresses this challenge by allowing participating devices or edge nodes to train models locally and share model updates instead of directly transmitting their underlying datasets.

In a federated semantic communication architecture, a central coordinator distributes a semantic model to participating edge devices. Each device performs local training using information available within its own environment and then sends selected model updates to the coordinator. The coordinator aggregates contributions from multiple participants to produce an improved global model. The updated model is subsequently distributed to the participating devices. Through this iterative process, semantic intelligence can benefit from diverse distributed data while keeping the raw training information closer to its original source.

Distributed learning can be particularly valuable when semantic information is highly dependent on local context. Different IoT devices, vehicles, mobile users, or industrial systems may observe different environments and generate different types of semantic data. Local training allows models to capture these characteristics while collaborative aggregation provides broader knowledge. Additional privacy-preserving mechanisms can be applied to model updates to reduce the possibility of reconstructing sensitive information from shared parameters. Secure aggregation, access control, participant authentication, and other protection mechanisms can strengthen the learning process.

However, privacy-preserving distributed learning introduces several challenges. Edge devices may have different computational capabilities, data distributions, energy limitations, and communication conditions. These differences can affect training quality and convergence. Furthermore, model updates themselves may reveal information about local datasets if they are not appropriately protected. Malicious or compromised participants may also attempt to manipulate collaborative training. Therefore, privacy protection must be combined with mechanisms for participant validation, update integrity, secure aggregation, and anomaly detection.

Federated and distributed learning consequently provide a useful foundation for privacy-aware semantic intelligence. They allow semantic models to learn from distributed experiences without requiring complete centralization of raw information. When combined with appropriate security and privacy mechanisms, these approaches can support semantic learning across healthcare, intelligent transportation, personal devices, industrial IoT, and smart environments while reducing direct exposure of locally generated data.

11.2.4 Differential Privacy for Semantic Information Protection

Differential privacy provides a mathematical framework for limiting the amount of information that can be inferred about an individual or individual data record from a released dataset, model, or statistical result. Its application to semantic communication is particularly relevant because semantic representations and learned models can contain information derived from user-generated text, images, speech, sensor observations, or behavioral data. Even when raw information is not directly transmitted, semantic representations may reveal characteristics that were present in the original data. Differential privacy can therefore provide an additional protection layer for semantic information processing and learning.

A differential privacy mechanism generally introduces carefully controlled randomness or noise into data, representations, model updates, or query results before they are released. The objective is to make the presence or contribution of an individual data record difficult to determine while retaining sufficient utility for the intended task. In semantic communication, noise can potentially be applied during the training of semantic models, aggregation of distributed updates, or release of semantic statistics. The amount of protection must be carefully selected because excessive perturbation may reduce semantic fidelity and task accuracy.

The central challenge is therefore the privacy–utility trade-off. Stronger privacy protection generally requires greater perturbation, which can reduce the accuracy of semantic representations or downstream inference. On the other hand, insufficient protection may leave semantic information vulnerable to inference attacks. The appropriate privacy level depends on the sensitivity of the information, the application requirements, the type of semantic representation, and the consequences of information disclosure. Systems handling highly sensitive information may accept some reduction in semantic performance in exchange for stronger privacy guarantees.

Differential privacy can complement other privacy-preserving approaches rather than replacing them. For example, it can be combined with federated learning to protect model updates while raw data remain locally stored. It can also complement secure transmission, access control, local semantic processing, and encryption. In edge-based semantic systems, privacy mechanisms can be applied before information leaves a device or during aggregation at an edge or cloud coordinator. This creates multiple opportunities to limit unauthorized inference.

For semantic communication, differential privacy should ultimately be evaluated through both privacy protection and semantic utility. Relevant measures can include task accuracy, semantic similarity, reconstruction quality, communication efficiency, and the strength of the privacy guarantee. The goal is not to eliminate useful information but to prevent unnecessary disclosure while preserving the meaning required by the application. Consequently, differential privacy provides a principled approach for

incorporating privacy into semantic learning and information exchange, particularly in distributed IoT, edge intelligence, human–AI communication, healthcare, and other environments where semantic information may reveal sensitive characteristics.

11.3 Trust, Authenticity, and Integrity in Meaning-Driven Networks

11.3.1 Semantic Information Authenticity and Integrity Verification

Semantic communication networks require mechanisms that can verify whether received information is authentic, trustworthy, and unchanged from its authorized source. Unlike conventional communication systems, where integrity commonly focuses on whether transmitted bits have been modified, meaning-driven networks must also consider whether the semantic content remains valid and consistent with the intended information. A message may arrive without conventional transmission errors while still containing manipulated, misleading, or contextually invalid semantic information. Therefore, authenticity and integrity verification must operate across both communication and semantic processing layers.

Semantic authenticity concerns establishing that semantic information originates from a legitimate and authorized source. Devices, users, edge nodes, and intelligent agents may generate semantic messages, and the receiver needs to determine whether the claimed source can be trusted. Authentication mechanisms can establish the identity of communicating entities, while cryptographic signatures or message authentication mechanisms can provide evidence that information originated from an authorized source. These mechanisms are particularly important in distributed semantic networks where information may pass through multiple intermediate nodes before reaching its final destination.

Integrity verification additionally determines whether semantic information has been altered during transmission, storage, processing, or aggregation. Conventional integrity checks can detect modifications to transmitted representations, but semantic systems may require additional validation after decoding. Semantic consistency checks can compare received information with contextual constraints, previously established knowledge, expected system states, or task requirements. For example, an intelligent system may identify a semantic message as suspicious when its content conflicts significantly with trusted contextual information, even if its underlying transmission was technically error-free.

Verification can therefore involve multiple levels, including source authentication, cryptographic integrity protection, semantic consistency analysis, and anomaly detection. The verification process should be integrated with semantic decoding so that potentially unreliable information is not immediately used for decision-making. When suspicious information is detected, the system may reduce its trust level, request additional evidence, or rely on alternative trusted information sources.

Thus, semantic authenticity and integrity verification extend conventional communication security toward meaning-aware protection. The objective is not only to ensure that information has been transmitted correctly but also to establish confidence that the received meaning originates from an authorized source and remains valid for the intended task. This capability is essential for autonomous systems, industrial environments, intelligent transportation, and distributed IoT applications where incorrect semantic information can directly influence system decisions.

11.3.2 Trust-Aware Semantic Communication and Decision Making

Trust is an important component of meaning-driven networks because semantic communication often involves multiple devices, intelligent agents, edge nodes, and information sources with different levels of reliability. A semantic message may be technically valid but still unsuitable for decision-making if its source has a history of unreliable observations or inconsistent information. Trust-aware semantic communication therefore incorporates information about the reliability of sources, models, nodes, and semantic content when determining how information should be transmitted, interpreted, and used.

Trust can be established from several types of evidence. Historical communication behavior, previous semantic accuracy, successful task completion, source authentication, contextual consistency, and feedback from other network participants can contribute to a trust assessment. A node that consistently provides accurate and useful semantic information may receive a higher trust level, whereas repeated inconsistencies or suspicious behavior may reduce its trust. Trust assessments can be updated dynamically as new interactions and observations become available. Trust information can also influence semantic communication decisions. For example, highly trusted semantic information may be transmitted using normal processing, while information from a less-trusted source may require additional verification or supporting evidence. Similarly, a receiver may assign different importance to semantic messages based on both their task relevance and the trustworthiness of their sources. This can help prevent unreliable information from dominating decisions in distributed intelligent systems.

Trust-aware decision-making becomes especially important when multiple sources provide conflicting semantic information. An autonomous system may receive different interpretations of the same environment from several sensors, vehicles, edge nodes, or intelligent agents. Rather than treating all information equally, the system can combine semantic relevance with source trust and contextual consistency to determine which information should have greater influence. Such mechanisms can improve the robustness of decisions under uncertain or adversarial conditions.

However, trust management must itself be protected because attackers may attempt to manipulate trust values, create false identities, or provide misleading feedback. Trust mechanisms should therefore incorporate authenticated interactions, reliable evidence, controlled trust updates, and anomaly detection. Excessive dependence on historical reputation should also be avoided because previously trusted sources may become compromised. Trust-aware semantic communication provides a bridge between secure information exchange and reliable intelligent decision-making. It enables semantic networks to evaluate not only what information means, but also how much confidence should be placed in that information. This becomes increasingly important as communication systems evolve toward autonomous, distributed, and knowledge-driven architectures.

11.3.3 Knowledge Provenance and Semantic Source Verification

Knowledge provenance refers to information describing the origin, history, transformation, and usage of knowledge or semantic information. In meaning-driven networks, provenance is particularly important because semantic information may be generated, transformed, aggregated, and interpreted by multiple distributed entities. Without provenance information, a receiver may be unable to determine where a semantic representation originated, which models or data sources contributed to it, or whether it has undergone unauthorized modification.

Semantic source verification begins by establishing the origin of information. A semantic message may originate from a sensor, intelligent device, user, edge node, cloud service, or another autonomous agent. Source identity can be associated with authentication credentials and other verification information. Provenance records can then describe the subsequent processing history, including semantic transformations, aggregation operations, model versions, and intermediate processing nodes. This creates a traceable relationship between the original information and the semantic knowledge ultimately used by an application.

Provenance is also useful for evaluating the reliability of semantic information. When several knowledge sources contribute to a semantic representation, provenance information can help determine which sources were involved and whether those sources are trusted. It can also support the identification of outdated, inconsistent, or potentially compromised knowledge. For example, if a decision is based on information generated from an obsolete semantic model, provenance records can help identify the model version responsible for that result.

In distributed semantic networks, provenance mechanisms should be protected against unauthorized modification. Cryptographic signatures, authenticated metadata, secure timestamps, access control, and integrity verification can help preserve the reliability of provenance records. Provenance information should also be managed carefully because it may itself contain sensitive information about users, devices, locations, or communication activities. Consequently, provenance protection must balance traceability with privacy.

Knowledge provenance ultimately provides accountability and traceability for semantic information exchange. It enables systems to answer important questions about where information came from, how it was processed, and which entities contributed to its final interpretation. This capability supports semantic verification, trust management, error analysis, and security investigation. As semantic networks become increasingly distributed, maintaining reliable provenance becomes essential for ensuring that knowledge used in automated decisions can be traced back to legitimate and verifiable sources.

11.3.4 Secure Knowledge Sharing across Distributed Semantic Networks

Distributed semantic networks depend on the exchange of knowledge among devices, edge nodes, intelligent agents, cloud platforms, and other network participants. Knowledge sharing enables individual nodes to benefit from information that may not be locally available and supports collaborative semantic understanding. However, unrestricted knowledge sharing can expose systems to unauthorized access, malicious information, knowledge poisoning, and the propagation of incorrect semantic information. Secure knowledge sharing must therefore protect both the knowledge itself and the mechanisms through which it is exchanged.

Access control is a fundamental requirement for secure knowledge sharing. Different participants may have different permissions depending on their roles, capabilities, and application requirements. A device may be permitted to access general environmental information but not sensitive user-related knowledge. Fine-grained access policies can limit knowledge exchange to authorized participants and reduce unnecessary exposure. Authentication mechanisms should verify participating entities before knowledge is accepted or shared.

Knowledge integrity is equally important because distributed participants may modify, replace, or inject information into shared knowledge repositories. Cryptographic integrity mechanisms and authenticated updates can help ensure that knowledge originates from authorized sources and has not been altered without permission. Version management can also help identify outdated or conflicting knowledge. When multiple nodes contribute information, provenance records can preserve the history of contributions and transformations.

Secure knowledge sharing should additionally consider the semantic validity of shared information. A knowledge item may be authentic but incorrect because of faulty sensing, outdated models, or erroneous inference. Semantic validation can compare new knowledge with existing contextual information, domain constraints, and observations from other trusted sources. Trust scores and provenance information can further assist receiving nodes in determining how much confidence should be assigned to shared knowledge.

In large distributed networks, secure knowledge exchange must also remain scalable and efficient. Excessive verification, encryption, or metadata processing can increase communication and computational overhead, particularly for resource-constrained edge and IoT devices. Lightweight security mechanisms, hierarchical trust management, distributed verification, and selective knowledge sharing can help maintain an appropriate balance between protection and efficiency.

Secure knowledge sharing therefore forms a critical foundation for reliable meaning-driven networks. By combining authentication, access control, integrity protection, provenance, trust assessment, and semantic validation, distributed nodes can exchange useful knowledge while reducing the risk of unauthorized disclosure and malicious information propagation. Such mechanisms enable semantic networks to collaborate safely while maintaining confidence in the knowledge used for communication, inference, and autonomous decision-making.

11.4 Robust and Resilient Semantic Communication Architectures

11.4.1 Adversarially Robust Semantic Communication Models

Adversarially robust semantic communication models are designed to maintain reliable meaning transmission when communication systems are exposed to intentional attacks or manipulated inputs. Conventional semantic communication models typically rely on machine learning techniques to extract, encode, and reconstruct task-relevant information. However, these models can be vulnerable to carefully designed perturbations that cause incorrect semantic interpretation even when the transmitted information appears valid at the communication level. Robustness is therefore essential when semantic communication is deployed in safety-critical, autonomous, or highly distributed environments.

Adversarial attacks can target different stages of the semantic communication pipeline. An attacker may manipulate source data before semantic encoding, modify semantic representations during transmission, or influence the receiver's semantic decoder and learning model. Small changes in an input image, speech signal, text sequence, or sensor observation can potentially produce a significantly different semantic interpretation. In addition, malicious training data or model updates can influence the behavior of semantic models before deployment. These vulnerabilities make adversarial robustness an important component of secure semantic communication.

Robust semantic models can be developed using techniques such as adversarial training, robust optimization, input validation, anomaly detection, and model regularization. During training, models can be exposed to representative perturbations so that they learn to maintain stable semantic representations under adverse conditions. At the communication stage, suspicious semantic information can be identified by comparing it with expected contextual patterns or previously established knowledge. Multiple sources can also be used to cross-check critical information before it influences an important decision.

Robustness should be evaluated according to the actual semantic task rather than only conventional signal-level metrics. A model may experience significant numerical changes in its internal representation while still producing the correct task outcome. Conversely, a small representation change may cause a major semantic error. Evaluation can therefore consider task accuracy, semantic consistency, reconstruction quality, robustness under perturbation, and decision reliability. Ultimately, adversarially robust semantic communication aims to ensure that meaning remains dependable even when the system encounters intentionally manipulated information. This requires security considerations to be integrated into semantic model design, training, transmission, and inference rather than being treated solely as an external communication-layer function.

11.4.2 Noise- and Error-Resilient Semantic Representation

Wireless communication systems inevitably experience noise, interference, fading, packet errors, and other transmission impairments. In semantic communication, these impairments can affect the meaning-bearing representations generated by intelligent encoders. Although semantic systems do not necessarily require exact recovery of every transmitted element, excessive corruption can still result in incorrect interpretation or unsuccessful task completion. Noise- and error-resilient semantic representations are therefore required to preserve essential meaning under imperfect communication conditions.

Semantic representations can be designed to tolerate moderate disturbances by emphasizing information that is important for the target task. Instead of treating every feature as equally important, the semantic encoder can prioritize robust and highly informative characteristics while reducing dependence on fragile details. Redundant semantic information can also be incorporated when appropriate so that the receiver can reconstruct the intended meaning even when some information is lost during transmission.

The receiver can further improve robustness through semantic-aware decoding and reconstruction. Contextual knowledge, previously received information, domain constraints, and learned relationships can help infer missing or corrupted semantic information. For example, when part of a semantic message is unavailable, the receiver may use surrounding contextual information to reconstruct the most likely task-relevant interpretation. This allows semantic communication to exploit relationships between information elements rather than relying exclusively on direct symbol recovery.

Different channel conditions may require different levels of semantic protection. Under favorable conditions, the system may transmit compact representations with limited redundancy to reduce communication overhead. When channel quality deteriorates, additional semantic protection, redundancy, retransmission, or stronger error-control mechanisms can be activated for critical information. Such adaptation allows communication resources to be concentrated on information whose loss would have the greatest effect on the application.

The effectiveness of noise- and error-resilient semantic representations should be evaluated using both communication and application-level measures. Semantic similarity, task success, reconstruction quality, latency, and communication efficiency can provide a more meaningful assessment than bit-error performance alone. The objective is to maintain acceptable semantic utility despite unavoidable channel imperfections.

Thus, noise and error resilience represents a fundamental property of robust semantic communication. By designing representations around task relevance, contextual relationships, adaptive protection, and intelligent reconstruction, semantic systems can continue to provide useful information even when the underlying communication channel is unreliable.

11.4.3 Distribution Shifts and Changing Semantic Environments

Semantic communication models are often trained using data that represent particular environments, users, devices, operating conditions, or application scenarios. However, real-world environments are rarely static. Changes in user behavior, sensor characteristics, communication conditions, physical surroundings, application requirements, or available knowledge can cause the distribution of incoming information to differ from the data used during model training. These changes, commonly referred to as distribution shifts, can significantly reduce the reliability of semantic interpretation.

Distribution shifts may occur gradually or suddenly. A semantic model deployed in an intelligent transportation environment, for example, may encounter changes in traffic patterns, weather conditions, road environments, or vehicle behavior. Similarly, an IoT system may experience changes in sensor configurations, device populations, or environmental conditions. Human-oriented semantic systems can also encounter changes in language usage, communication preferences, or contextual requirements. A model that performs well under its original conditions may therefore produce less accurate semantic representations when the surrounding environment changes.

Robust semantic architectures should be capable of detecting and responding to such changes. Continuous monitoring can compare incoming semantic patterns with previously learned distributions and identify significant deviations. Context monitoring can provide additional information about environmental changes, while confidence estimation can identify situations in which the semantic model is uncertain. When substantial changes are detected, the system can adapt its semantic models, update relevant knowledge, or select alternative models better suited to the current environment.

Distributed and edge-based learning can support this adaptation by allowing models to learn from newly observed information without requiring all raw data to be transferred to a centralized location. Model updating should nevertheless be carefully controlled because new data may themselves contain noise, errors, or malicious information. Validation and trust mechanisms are therefore important when incorporating newly learned knowledge into an operational semantic system.

Handling distribution shifts ultimately requires semantic communication architectures to move beyond static model deployment toward continuous environmental adaptation. A resilient system should recognize when its assumptions are no longer valid and adjust its representation, inference, and

communication policies accordingly. This capability helps preserve semantic reliability as users, environments, data distributions, and application requirements evolve over time.

11.4.4 Fault-Tolerant and Self-Recovering Semantic Networks

Fault tolerance is essential for semantic communication networks operating across distributed devices, edge nodes, communication links, and intelligent processing platforms. Failures may result from hardware faults, software errors, unavailable network connections, corrupted models, overloaded edge resources, or temporary communication disruptions. Because semantic networks may support autonomous decisions and real-time applications, the failure of a single component should not necessarily cause complete loss of semantic functionality. Fault-tolerant architectures therefore aim to maintain useful communication and intelligence despite component failures.

A fault-tolerant semantic network can use redundancy and distributed processing to avoid dependence on individual nodes. Critical semantic functions may be replicated across multiple edge devices or cloud resources, allowing another component to continue processing when one node becomes unavailable. Multiple sensing sources can similarly provide alternative information when an individual sensor fails. The network can dynamically redirect semantic processing or communication toward available resources while maintaining the required level of service.

Self-recovery extends fault tolerance by enabling the network to detect failures and initiate corrective actions with limited human intervention. Monitoring mechanisms can observe communication performance, semantic output quality, processing availability, and node behavior. When an abnormal condition is identified, the system can isolate the affected component, select an alternative resource, restore a previous valid model or configuration, and resume semantic processing. Recovery mechanisms can also use previously stored semantic knowledge to reconstruct missing information when complete data are unavailable.

Semantic awareness can make recovery more efficient because not every failure has the same impact on application performance. A network can prioritize recovery of components responsible for critical semantic tasks while temporarily reducing resources assigned to less important functions. Similarly, degraded operating modes can preserve essential semantic services when full functionality cannot immediately be restored. This allows the system to maintain useful task performance rather than requiring complete recovery before communication resumes.

A resilient architecture should also learn from previous failures. Information about faults, recovery actions, semantic degradation, and successful corrective strategies can be incorporated into future network policies. Over time, this can support more effective fault prediction and adaptive resource management.

Consequently, fault-tolerant and self-recovering semantic networks combine redundancy, monitoring, adaptive resource allocation, semantic prioritization, and automated recovery to maintain service continuity. Such capabilities are particularly important for autonomous vehicles, industrial systems, distributed IoT, edge intelligence, and other applications where semantic communication must remain operational despite changing conditions and component failures.

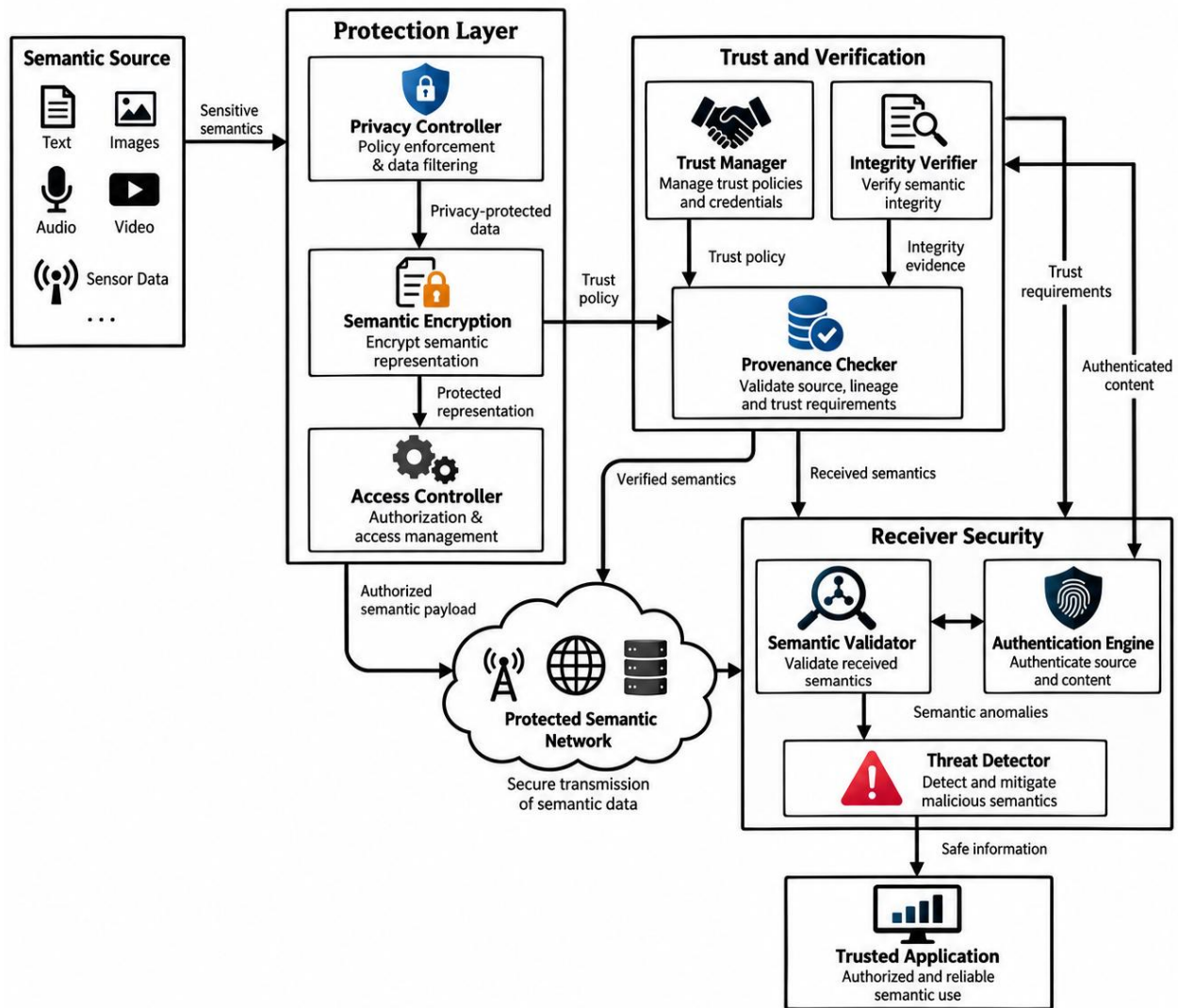


Figure 54: Privacy, Trust, and Integrity Protection Architecture for Semantic Communication Networks

Trust, Authenticity, and Integrity in Meaning-Driven Networks because it integrates privacy protection, semantic encryption, access control, trust management, integrity verification, provenance checking, authentication, and semantic threat detection within a unified architecture. The figure begins with different semantic sources, including text, images, audio, video, and sensor data, demonstrating that security and trust mechanisms must protect heterogeneous semantic information.

The Protection Layer provides the first stage of security. The Privacy Controller applies policy enforcement and data filtering to sensitive semantics, while Semantic Encryption protects the semantic representation during communication. The Access Controller manages authorization and determines which semantic information can be accessed or transmitted. The resulting authorized semantic payload is carried through the Protected Semantic Network, providing a secure communication environment for meaning-bearing information.

The Trust and Verification layer strengthens the reliability of semantic information by combining the Trust Manager, Integrity Verifier, and Provenance Checker. The Trust Manager manages trust policies and credentials, while the Integrity Verifier checks whether semantic information has maintained its integrity. The Provenance Checker validates the source, lineage, and associated trust requirements. This combination is important because semantic communication requires verification not only of the communicating entity but also of the origin and reliability of the meaning being exchanged.

At the receiving side, the Receiver Security layer performs authentication, semantic validation, and threat detection before information reaches the trusted application. The Authentication Engine verifies the source and content, while the Semantic Validator checks the received semantics. Detected semantic anomalies are processed by the Threat Detector to identify and mitigate potentially malicious information. Only information that passes the required security and trust checks is delivered to the Trusted Application, illustrating how privacy, authenticity, integrity, provenance, and threat detection collectively support reliable semantic information use.

Chapter 12

Semantic Network Optimization, Quality Assessment, and Experimental Evaluation

12.1 Multi-Objective Optimization of Meaning-Driven Communication Systems

12.1.1 Semantic Utility Maximization and Communication Efficiency

Semantic communication systems are designed to deliver information that is useful for a particular task rather than reproducing every transmitted element exactly. Consequently, optimization must consider how effectively the available communication resources contribute to the desired semantic outcome. Semantic utility represents the usefulness or task relevance of the information reconstructed at the receiver, while communication efficiency reflects how effectively bandwidth, transmission time, power, and other resources are utilized. Maximizing semantic utility while maintaining communication efficiency is therefore a fundamental optimization objective in meaning-driven networks.

In conventional communication systems, performance is frequently evaluated using measures such as bit error rate, throughput, and spectral efficiency. Semantic communication introduces additional considerations because a reduction in transmitted information does not necessarily imply a reduction in application performance. A semantic encoder can remove redundant or task-irrelevant information while retaining features required by the receiver. The resulting system may transmit fewer data while achieving comparable or even improved task performance. Therefore, optimization should determine which semantic information provides the greatest contribution to the target application under available communication constraints.

Semantic utility can depend on the specific application and may be reflected through task accuracy, semantic similarity, inference performance, reconstruction quality, or decision reliability. Communication efficiency, meanwhile, can be assessed through metrics such as transmitted data volume, bandwidth utilization, latency, energy consumption, and spectral efficiency. These objectives can conflict with one another. For example, transmitting a richer semantic representation may improve task accuracy but increase communication overhead. Conversely, aggressive semantic compression can reduce resource consumption while potentially degrading the quality of the final decision.

An effective optimization strategy therefore seeks an appropriate balance between semantic usefulness and communication cost. Semantic importance can be estimated at the transmitter so that high-value information receives greater protection or representation precision. The system can then adapt the amount of information transmitted according to channel conditions, application requirements, and resource availability. This enables communication resources to be concentrated on information that contributes most significantly to the intended task.

Semantic utility maximization ultimately shifts optimization from accurate bit delivery toward efficient delivery of useful meaning. This perspective provides a foundation for designing communication systems in which resource consumption is evaluated together with the quality and usefulness of the resulting semantic information.

12.1.2 Joint Optimization of Semantic Quality and Network Resources

Meaning-driven communication requires the coordinated optimization of semantic quality and the network resources used to deliver semantic information. Semantic quality describes how effectively the receiver can recover or utilize the intended meaning, while network resources include bandwidth, transmit power, computing capacity, communication time, storage, and processing resources. Optimizing these factors independently can result in inefficient system behavior because improvements in one dimension may negatively affect another.

Semantic quality is strongly influenced by the amount and type of information provided to the receiver. A richer semantic representation may improve task performance but require greater bandwidth or computational resources. Similarly, more powerful semantic models may improve inference accuracy while increasing processing time and energy consumption. Network resource allocation must therefore account for the characteristics of the semantic task. Critical information may require greater resource allocation than background information because its loss can have a larger impact on the final application outcome.

Joint optimization can involve semantic representation selection, bandwidth allocation, transmission power control, computation scheduling, and model selection. For example, an edge-enabled semantic network may dynamically determine how much semantic information should be processed locally and how much should be transmitted to another node. A high-complexity model may be selected when computational resources are abundant, whereas a lightweight model may be preferred when devices are energy constrained. Similarly, additional transmission resources can be assigned to semantic information with high task relevance.

The optimization process must also consider changing network conditions. Channel quality, traffic load, device mobility, available computing resources, and application requirements may vary over time. A static resource allocation strategy can therefore become inefficient under dynamic conditions. Adaptive optimization allows semantic systems to continuously adjust representation quality and resource allocation according to the current operating state.

A useful framework can formulate the problem as a constrained multi-objective optimization in which semantic quality is maximized while resource consumption remains within available limits. Rather than optimizing a single network metric, the system evaluates the combined effect of resource decisions on the final semantic task. This approach helps identify operating points that provide an effective balance between semantic performance and resource expenditure.

Joint optimization consequently represents a major transition from conventional resource management. Instead of allocating resources solely according to data volume or channel conditions, meaning-driven networks can allocate resources according to semantic importance, task requirements, and expected application benefit. This enables more efficient use of network and computing infrastructure while maintaining the semantic quality required by intelligent applications.

12.1.3 Latency, Reliability, and Energy-Aware Optimization

Latency, reliability, and energy consumption are important objectives in semantic communication systems, particularly when they support real-time and resource-constrained applications. Autonomous vehicles, industrial control systems, mobile devices, and IoT networks often require information to be delivered within strict time limits while maintaining sufficient reliability and minimizing energy expenditure. Optimizing only semantic quality may therefore be insufficient because a highly accurate semantic representation may have limited practical value if it arrives too late or requires excessive energy to generate and transmit. Latency-aware optimization focuses on reducing the time required for semantic information to travel through the complete communication and processing pipeline. This includes semantic encoding, channel transmission, edge or cloud processing, decoding, and application-level inference. Processing placement can have a significant influence on latency. For example, computationally intensive semantic processing may be performed at an edge node when local device capabilities are limited. Conversely, lightweight processing may remain on the device when communication with a remote server would introduce excessive delay.

Reliability represents the probability that the semantic task will be completed correctly under the prevailing communication and processing conditions. In semantic communication, reliability does not necessarily require exact recovery of every transmitted element. Instead, it can be related to maintaining an acceptable level of task performance or semantic correctness. High-priority semantic information may therefore receive additional protection, redundancy, retransmission opportunities, or stronger error-control mechanisms when its failure would significantly affect application performance.

Energy efficiency is particularly important for battery-powered and resource-constrained devices. Semantic encoding and machine-learning inference can require considerable computational energy, while wireless transmission also consumes power. An energy-aware system can select appropriate semantic models, compression levels, processing locations, and transmission configurations according to available energy. Lightweight representations may be preferred when energy reserves are low, while more computationally demanding processing can be used when sufficient energy is available.

These objectives frequently compete with one another. Reducing latency may require additional computing or communication resources, while improving reliability may increase redundancy and energy consumption. A practical optimization framework therefore seeks a balanced operating point rather than independently maximizing every metric. Multi-objective strategies can prioritize different requirements according to the application context and dynamically modify the optimization priorities as network conditions change. Latency-, reliability-, and energy-aware optimization thus enables semantic communication systems to operate effectively under practical constraints. It ensures that the system does not evaluate communication performance solely through semantic accuracy, but also considers when information arrives, how reliably it supports the task, and how much energy is required to obtain it.

12.1.4 Adaptive Multi-Objective Semantic Network Optimization

Semantic communication networks operate in environments where channel conditions, network traffic, device capabilities, application requirements, and semantic information characteristics can change continuously. A fixed optimization strategy may therefore become unsuitable as operating conditions evolve. Adaptive multi-objective optimization addresses this challenge by dynamically adjusting

communication, computation, and semantic processing decisions according to the current state of the network and the requirements of the application.

An adaptive optimizer can monitor multiple system parameters, including channel quality, available bandwidth, queue conditions, device energy, computational capacity, semantic task requirements, and observed application performance. These measurements provide the information required to determine whether the current operating configuration remains appropriate. When conditions change, the system can modify semantic representation quality, model selection, resource allocation, transmission parameters, processing placement, or scheduling policies.

The multi-objective nature of the problem is important because several performance goals must be considered simultaneously. Semantic quality may need to be maintained while minimizing latency, energy consumption, and communication overhead. Reliability may become more important during poor channel conditions, whereas energy efficiency may receive greater priority when battery resources are limited. Similarly, latency can become the dominant objective for time-critical applications. Adaptive optimization allows these priorities to change according to the current application and network context.

Machine learning can support adaptive optimization by learning relationships between network states, resource configurations, and semantic performance. Historical observations can be used to predict which configuration is likely to provide an appropriate balance among competing objectives. Reinforcement learning and other intelligent optimization approaches can also enable systems to improve resource decisions through repeated interaction with the environment. However, adaptive mechanisms should incorporate constraints that prevent unstable or unsafe decisions, particularly in critical applications.

The optimization process can be viewed as a continuous feedback cycle in which the semantic network observes its operating environment, evaluates current performance, selects an improved configuration, and measures the resulting outcome. This enables the system to respond to changing conditions rather than relying on predetermined parameters. Adaptation can occur at different timescales, ranging from rapid adjustments to transmission settings to longer-term updates of semantic models and resource-management policies.

Adaptive multi-objective optimization ultimately provides the foundation for self-adjusting meaning-driven networks. By simultaneously considering semantic quality, communication efficiency, latency, reliability, energy consumption, and computational resources, the network can select configurations appropriate to current conditions. Such adaptability is essential for future semantic systems operating across heterogeneous devices, dynamic wireless environments, edge-cloud infrastructures, and intelligent applications with continuously changing requirements.

12.2 Semantic Quality of Service and Quality of Experience

12.2.1 Semantic Quality of Service Requirements and Design Principles

Semantic Quality of Service (QoS) extends conventional network quality measures by considering whether communication delivers information that is sufficiently useful for the intended semantic task. Traditional QoS parameters such as throughput, delay, packet loss, and reliability remain relevant, but they do not fully describe the performance of a meaning-driven communication system. A semantic

network may successfully reduce transmitted data while still providing high application performance if the essential meaning is preserved. Therefore, semantic QoS must connect communication-level conditions with semantic and application-level outcomes.

The requirements of semantic QoS depend strongly on the target application. Real-time intelligent applications may require low latency and rapid semantic reconstruction, whereas applications involving large-scale sensing may prioritize energy efficiency and sustained communication. Safety-critical applications may require high semantic reliability, while less critical applications may tolerate some degradation in exchange for reduced resource consumption. Semantic QoS should therefore be defined according to the importance, relevance, and required accuracy of the information being exchanged.

An important design principle is task awareness. Network resources should be allocated according to the contribution of semantic information to the target task rather than simply according to its data size. Highly relevant semantic features can receive stronger protection and greater resource allocation, while redundant or low-value information can be compressed or omitted. Another principle is context awareness, in which semantic quality requirements are adapted according to environmental conditions, user requirements, and application states.

Semantic QoS should also be adaptive because communication and processing conditions can change over time. Channel quality, traffic load, computational resources, device energy, and semantic complexity may vary continuously. A semantic network should therefore be capable of adjusting representation quality, transmission parameters, processing strategies, and resource allocation to maintain the required service level. Monitoring mechanisms can compare current semantic performance against application-specific QoS requirements and trigger appropriate adaptations. Ultimately, semantic QoS provides a framework for evaluating communication from the perspective of useful information delivery. Its design should integrate network performance, semantic relevance, task requirements, latency, reliability, and resource efficiency. By doing so, semantic communication systems can provide differentiated and application-oriented services rather than treating all transmitted information according to identical communication requirements.

12.2.2 Semantic Quality of Experience for Intelligent Applications

Semantic Quality of Experience (QoE) focuses on how effectively users or intelligent applications benefit from the meaning delivered through a communication system. While semantic QoS describes measurable service requirements and system performance, semantic QoE considers the resulting usefulness, satisfaction, responsiveness, and effectiveness from the perspective of the application or user. This distinction is important because technically efficient communication does not necessarily produce a satisfactory experience if the delivered information is incomplete, misleading, or insufficient for the intended task.

For intelligent applications, QoE can be strongly influenced by semantic accuracy and relevance. A perception system, for example, may receive a compact representation that consumes little bandwidth but fails to preserve information necessary for correct recognition. Conversely, a richer representation may improve application performance but introduce additional latency or resource consumption. Semantic

QoE therefore reflects the balance between the quality of delivered meaning and the practical requirements of the application.

Different applications may interpret semantic QoE differently. In autonomous systems, it may depend on whether the received information supports correct environmental understanding and timely decision-making. In human–AI interaction, it may depend on whether the system correctly understands user requests and produces relevant responses. In multimedia applications, semantic QoE may relate to whether important objects, events, or activities are correctly preserved rather than whether every visual or audio detail is reconstructed. Consequently, QoE assessment must be tailored to the specific application context.

Semantic QoE can incorporate objective and subjective indicators. Objective measures may include task accuracy, semantic similarity, response time, successful inference, decision reliability, and reconstruction quality. Subjective evaluation may consider perceived usefulness, responsiveness, clarity, relevance, and satisfaction when humans directly interact with the system. Combining these perspectives can provide a more comprehensive understanding of semantic communication performance.

QoE can also be affected by changing user expectations and application states. A user may require highly detailed semantic information in one situation but only a concise summary in another. Similarly, an intelligent application may demand higher semantic precision during a critical event and accept reduced quality during routine operation. Adaptive semantic communication can respond to these variations by modifying the information representation and resource allocation.

Thus, semantic QoE shifts evaluation from the question of how well information was transmitted to the question of how effectively the delivered meaning supports the user or intelligent application. This application-oriented perspective is essential for designing semantic communication systems that are not only technically efficient but also practically useful.

12.2.3 Task Success, User Intent, and Application-Level Performance

The effectiveness of semantic communication is ultimately determined by whether the transmitted information enables the intended task to be completed successfully. Unlike conventional communication systems, where accurate reconstruction of transmitted data can serve as a primary performance objective, semantic communication focuses on preserving information that is relevant to the receiver's task. Task success therefore provides an important application-level measure for evaluating whether a semantic communication system is delivering meaningful information.

User intent is particularly important in human-centered semantic communication. Users may communicate an objective without explicitly providing every detail required for its interpretation. A semantic system must use contextual information, prior knowledge, and the characteristics of the interaction to identify the underlying intent. Successful communication occurs when the receiver interprets the intended meaning sufficiently accurately to perform the required action or provide an appropriate response. Consequently, semantic quality cannot always be evaluated through similarity between transmitted and reconstructed representations alone.

In intelligent machine-to-machine and autonomous systems, task success can similarly be defined according to the operational objective. For example, distributed intelligent devices may exchange semantic information to coordinate an action, while an autonomous agent may require semantic observations to make a particular decision. The quality of communication can therefore be assessed by examining whether the resulting decision or action satisfies the application requirement. This makes task-level performance a direct indicator of semantic communication effectiveness.

Several factors can influence task success. Semantic representation quality, contextual understanding, transmission reliability, latency, model accuracy, and resource availability can all affect the final application outcome. A small semantic degradation may have little effect on a non-critical task but may cause a major failure in a safety-sensitive application. Therefore, evaluation should consider the relationship between semantic degradation and the resulting application performance rather than relying on a single universal quality measure.

Application-level evaluation can also support adaptive resource allocation. If the system detects that additional semantic information would significantly improve task success, it can allocate additional communication or computation resources. Conversely, when task performance remains satisfactory despite reduced semantic detail, the system can operate with a more efficient representation. This creates a direct connection between semantic communication decisions and application objectives.

Task success, user intent, and application-level performance therefore provide a practical foundation for semantic evaluation. They establish whether communication has achieved its intended purpose, rather than simply determining whether data have been accurately reconstructed. This task-oriented perspective is central to assessing meaning-driven networks and distinguishing semantic communication from conventional data transmission.

12.2.4 Meaning Preservation and End-to-End Semantic Quality

Meaning preservation is a central requirement of semantic communication because information must remain sufficiently consistent with the intended meaning throughout the complete communication process. Semantic information can undergo several transformations during encoding, transmission, decoding, reconstruction, and application processing. Even when individual communication components operate correctly, accumulated distortions may alter the meaning relevant to the final task. End-to-end semantic quality therefore evaluates the complete pathway from the original semantic information at the source to its final interpretation or use at the destination.

Meaning preservation should not necessarily be interpreted as exact reproduction of the original data. Semantic communication intentionally removes information that is considered irrelevant to the target task. The important question is whether the information that remains after semantic processing is sufficient to support the intended application. For example, an image-based semantic system may not need to reconstruct every pixel if the application only requires reliable identification of an object or event. End-to-end quality should therefore be measured according to the semantic objective rather than low-level data similarity alone.

Several stages can influence end-to-end semantic quality. The semantic encoder determines which information is extracted and represented, while the communication channel can introduce noise, fading, interference, or packet loss. The decoder and semantic reconstruction model determine how received information is interpreted, and the final application determines whether that interpretation is useful. Errors or limitations at any stage can reduce overall semantic performance. This makes coordinated optimization across the entire semantic communication pipeline essential.

End-to-end semantic assessment can combine multiple indicators, including semantic similarity, task accuracy, inference reliability, reconstruction quality, latency, and resource consumption. The appropriate combination depends on the application. A real-time autonomous system may place greater emphasis on timely and reliable task decisions, whereas a multimedia application may place greater emphasis on semantic fidelity and perceptual relevance. Such application-specific evaluation avoids treating semantic quality as a single universal quantity.

Continuous monitoring can also help maintain end-to-end semantic quality. The system can compare the achieved application performance with the required semantic service level and adapt encoding, transmission, decoding, or processing strategies when degradation occurs. Context information can further help distinguish acceptable semantic variation from meaningful errors. End-to-end semantic quality therefore provides a comprehensive perspective on whether meaning survives the complete communication process. By evaluating the relationship between source intent, semantic representation, network delivery, receiver interpretation, and final application performance, it provides a more appropriate foundation for assessing meaning-driven communication systems than isolated communication-layer metrics alone.

12.3 Performance Metrics and Evaluation Frameworks for Semantic Communication

12.3.1 Semantic Similarity and Meaning Preservation Metrics

Semantic similarity is one of the fundamental performance dimensions in semantic communication because the objective is to preserve information that is meaningful for the intended application. Unlike conventional communication systems, where bit-level accuracy can provide a direct indication of transmission quality, semantic communication may intentionally discard redundant or task-irrelevant information. Therefore, evaluation requires metrics that determine whether the reconstructed information retains the essential meaning of the original content.

The appropriate semantic similarity metric depends on the type of information being communicated. For text-based communication, semantic similarity can be evaluated by comparing the meaning of the transmitted and reconstructed text rather than requiring exact word-by-word matching. For images and video, evaluation can focus on whether important objects, activities, attributes, or events are preserved. For speech, semantic evaluation may consider whether the intended message, entities, or commands are correctly understood. Multimodal systems require metrics capable of assessing relationships among multiple information modalities.

Meaning preservation can also be evaluated at the representation level. Semantic embeddings or learned feature representations can be compared to determine whether the reconstructed representation remains sufficiently close to the source representation in a task-relevant semantic space. However, representation

similarity alone may not always correspond to successful application performance because different representations can sometimes produce the same task outcome. Consequently, representation-level metrics should be complemented by task-level evaluation. A comprehensive evaluation framework can therefore combine semantic similarity, reconstruction quality, and downstream task performance. The selected metrics should reflect the intended communication objective rather than applying a single universal measure to every application. Thresholds can also be defined to identify the minimum semantic quality required for successful operation.

Meaning preservation should additionally be evaluated under different communication conditions. Noise, fading, interference, packet loss, compression, and resource limitations can affect semantic representations differently from conventional data. Testing semantic similarity across these conditions helps determine how robustly the system preserves useful information. Thus, semantic similarity metrics provide an important foundation for evaluating meaning-driven communication. They shift performance assessment from exact data reproduction toward determining whether the information required to preserve the intended meaning remains available after communication. Their effectiveness is greatest when combined with application-specific task metrics and conventional network measurements.

12.3.2 Semantic Efficiency and Communication Overhead

Semantic efficiency evaluates how effectively a communication system delivers useful meaning relative to the communication resources consumed. Conventional communication efficiency is often associated with throughput or spectral efficiency, whereas semantic communication introduces the additional consideration of how much task-relevant information is delivered per unit of communication cost. This distinction is important because semantic systems may intentionally transmit substantially less information than conventional systems while maintaining acceptable application performance.

Communication overhead includes the amount of semantic information transmitted, signaling information, control messages, model-related information, metadata, retransmissions, and other communication resources required by the semantic system. A semantic encoder may reduce the payload considerably, but the overall benefit depends on the additional overhead introduced by semantic processing, model synchronization, context exchange, or security mechanisms. Evaluation should therefore consider the complete communication pipeline rather than only the size of the semantic payload.

Semantic efficiency can be assessed by relating communication consumption to the resulting semantic or task performance. For example, a system that achieves similar task accuracy while transmitting substantially less information can demonstrate improved semantic efficiency. However, aggressive compression should not automatically be considered beneficial if it causes significant degradation in task performance. The objective is to identify an appropriate balance between communication savings and useful semantic information.

Different applications may impose different efficiency requirements. Bandwidth-constrained IoT systems may prioritize minimizing transmitted data, while mobile intelligent devices may emphasize a combination of communication and energy efficiency. Real-time applications may accept additional communication overhead if it substantially improves semantic reliability or reduces processing delay. Consequently, semantic efficiency should be evaluated together with latency, energy consumption,

reliability, and task success. Experimental comparisons should also distinguish between the semantic payload and total system overhead. Model updates, semantic dictionaries, contextual information, control signaling, and synchronization data can influence the practical efficiency of a semantic communication architecture. Reporting these components separately provides a more transparent evaluation.

Semantic efficiency therefore provides a broader perspective than conventional data compression. It asks whether communication resources are being used to deliver useful and task-relevant meaning rather than simply measuring how many bits have been removed. A reliable evaluation framework should report both communication savings and the corresponding semantic or application-level performance so that efficiency gains can be interpreted correctly.

12.3.3 Computational and Energy-Efficiency Measurements

Semantic communication systems commonly rely on machine learning models for semantic extraction, representation generation, decoding, inference, and reconstruction. These intelligent operations can introduce substantial computational requirements, particularly when complex neural networks or multimodal models are deployed at edge devices. Computational and energy efficiency are therefore important evaluation dimensions, especially for mobile, IoT, and resource-constrained semantic communication systems.

Computational efficiency can be assessed through processing time, model complexity, memory requirements, number of operations, processor utilization, and inference requirements. Encoding and decoding should be evaluated separately where possible because their computational costs may differ significantly. A semantic system that substantially reduces transmission requirements but requires excessive local computation may not provide an overall efficiency advantage for a resource-constrained device.

Energy efficiency is particularly important when semantic communication is implemented on battery-powered devices. Energy may be consumed during semantic feature extraction, neural-network inference, wireless transmission, reception, and local model training. Measuring only transmission energy can therefore provide an incomplete picture. A complete evaluation should consider the energy consumed by the relevant stages of the semantic communication pipeline.

The relationship between computation and communication must also be considered. Increasing semantic model complexity may reduce the amount of transmitted information but increase local processing energy. Conversely, a lightweight semantic model may require more communication resources because it generates less compact representations. This creates a computation-communication trade-off that should be explicitly examined during experimental evaluation.

Energy measurements can be collected under different operating conditions, including different model configurations, channel conditions, transmission rates, and device workloads. Comparing these measurements with semantic task performance helps determine whether additional computational or energy expenditure produces a meaningful improvement. Model selection can then be optimized according to the available device resources and application requirements.

For fair comparison, researchers should report the hardware platform, model configuration, processing environment, and measurement methodology used during experiments. Computational efficiency observed on a powerful server may not represent performance on an edge device. Therefore, computational and energy-efficiency measurements should be treated as integral components of semantic communication evaluation. They reveal whether semantic intelligence can be deployed practically while maintaining acceptable task performance. A successful system should achieve an appropriate balance between semantic quality, computation, communication, and energy consumption, particularly in distributed and resource-constrained environments.

12.3.4 Reliability, Latency, and Task-Oriented Performance Metrics

Reliability and latency are critical performance dimensions for semantic communication because useful information must not only be correct but also available when it is required. Conventional reliability metrics often focus on packet delivery or transmission errors, whereas semantic communication requires consideration of whether the delivered information remains sufficient for successful task execution. Similarly, latency should account for the complete semantic processing pipeline rather than only the physical transmission time.

Semantic reliability can be evaluated through successful task completion, semantic reconstruction accuracy, inference correctness, decision consistency, or the probability that the received information satisfies a predefined semantic requirement. These measures can be combined with conventional indicators such as packet loss and communication reliability. This provides a more complete view of whether channel impairments actually affect the final application.

Latency should include the time required for semantic encoding, transmission, processing, decoding, reconstruction, and application-level inference when these stages contribute to the final response. Edge processing may reduce communication delay but introduce additional computation time, while cloud-based processing may provide greater computational capability at the cost of additional network latency. Measuring end-to-end latency therefore provides a more meaningful evaluation than measuring individual communication stages in isolation.

Task-oriented metrics provide a direct connection between communication performance and application objectives. Depending on the application, these may include classification accuracy, detection success, decision accuracy, successful coordination, response correctness, or control effectiveness. A semantic communication system can then be evaluated according to whether it maintains acceptable task performance under different network conditions. Reliability and latency often involve trade-offs. Additional redundancy or stronger error protection may improve reliability but increase communication overhead and delay. More sophisticated semantic processing may improve task accuracy while increasing computational latency. Evaluation should therefore examine these metrics jointly rather than optimizing one metric independently.

Experimental studies should test semantic communication under different channel qualities, traffic conditions, processing loads, and resource constraints. Performance curves can reveal how rapidly task success degrades as latency increases or communication conditions become less reliable. Such analysis is particularly valuable for identifying operational limits. Ultimately, reliability, latency, and task-oriented

metrics connect semantic communication performance to real application outcomes. They help determine whether a semantic system can deliver sufficiently accurate information, within an acceptable time, to support the intended task. This application-centered evaluation is essential for meaningful comparison between semantic communication architectures.

12.4 Experimental Platforms, Datasets, and Benchmarking Methodologies

12.4.1 Semantic Communication Dataset Development and Selection

Datasets provide the foundation for training, testing, and comparing semantic communication systems. Because semantic communication focuses on meaning and task relevance, dataset selection must consider not only the availability of raw data but also the semantic information, contextual relationships, and application objectives associated with those data. A suitable dataset should therefore provide sufficient information for evaluating both semantic representation quality and downstream task performance.

Different semantic communication applications require different data modalities. Text datasets can support natural language understanding and semantic transmission, while image and video datasets can support object recognition, scene understanding, and event interpretation. Speech datasets are useful for evaluating semantic voice communication, whereas sensor and IoT datasets can support context-aware and machine-to-machine communication. Multimodal datasets are particularly valuable when semantic information must be extracted from multiple sources simultaneously.

Dataset development should include appropriate training, validation, and testing partitions. The testing data should provide an independent basis for evaluating generalization and should not be used during model development. Where possible, datasets should contain diverse conditions that represent the environments in which the semantic communication system is expected to operate. This can include variations in users, environments, channel conditions, device characteristics, and task requirements. Context and task annotations can substantially improve semantic evaluation. A dataset may include labels describing objects, events, user intent, environmental states, or task outcomes. Such information allows researchers to evaluate whether a semantic representation preserves the information required for a specific application. When semantic communication is evaluated using only raw reconstruction quality, important differences in task usefulness may remain hidden.

Dataset quality also depends on consistency and documentation. Information about data sources, collection procedures, preprocessing, annotation methods, modality characteristics, and intended use should be clearly recorded. Privacy and ethical considerations are especially important when datasets contain human-generated or sensitive information. Dataset selection should ultimately be aligned with the objective of the semantic communication system. A dataset that is suitable for general semantic similarity may not be appropriate for evaluating autonomous decision-making or industrial control. Consequently, experimental studies should clearly justify their dataset choices and explain how the selected data represent the intended communication task. Well-designed datasets provide the basis for reliable, meaningful, and reproducible evaluation of semantic communication technologies.

12.4.2 Wireless Channel and Network Simulation Environments

Wireless channel and network simulation environments are important for evaluating semantic communication systems under controlled and repeatable communication conditions. Real wireless

environments can be difficult to reproduce because channel quality, interference, mobility, traffic, and network load change continuously. Simulation allows researchers to systematically vary these factors and examine how semantic performance responds to different operating conditions.

A suitable simulation environment should represent the communication characteristics relevant to the target application. These may include additive noise, fading, interference, path loss, bandwidth limitations, packet loss, variable channel quality, and user mobility. Different network configurations can then be used to determine whether semantic communication maintains useful information under favorable and adverse conditions.

Simulation should not be limited to the physical communication channel. Semantic communication involves interactions among semantic encoding, transmission, decoding, and application processing. Therefore, the experimental environment should allow communication parameters to be connected with semantic model performance. For example, simulated packet loss can be introduced into semantic representations, followed by evaluation of reconstruction quality and task success. Similarly, changing bandwidth can be used to examine the trade-off between semantic compression and application performance.

Network-level experiments can also investigate resource allocation and multi-user scenarios. Multiple semantic users may compete for bandwidth, transmit power, edge computing resources, or processing capacity. Simulation environments can therefore be used to evaluate scheduling, semantic prioritization, adaptive transmission, and joint communication-computation optimization.

For meaningful comparisons, simulation configurations should be clearly documented. Important parameters include channel models, bandwidth, transmission power, packet size, network topology, mobility assumptions, traffic patterns, and simulation duration. Random seeds or repeated experiments can be used where appropriate to reduce the influence of random variation.

Simulation results should ideally be complemented by real-world experiments when practical. Simulation provides controlled evaluation, whereas physical testbeds reveal implementation effects that may not be captured by idealized models. Wireless and network simulation environments consequently provide a controlled foundation for testing semantic communication architectures. By systematically varying network conditions and connecting them with semantic and task-level measurements, researchers can determine how communication impairments influence the actual usefulness of transmitted meaning rather than evaluating network behavior independently from semantic performance.

12.4.3 Comparative Evaluation of Semantic Communication Models

Comparative evaluation is necessary to determine the relative advantages and limitations of different semantic communication models. Because semantic systems may use different architectures, learning methods, representations, and optimization strategies, direct comparison requires a carefully defined experimental framework. The objective is not simply to identify the model with the highest semantic accuracy, but to understand the balance among semantic quality, communication efficiency, computational requirements, reliability, latency, and energy consumption.

A fair comparison should use consistent datasets, preprocessing procedures, channel conditions, evaluation tasks, and experimental configurations wherever possible. Differences in hardware, training data, model capacity, or communication resources can otherwise make results difficult to interpret. Baseline systems should also be included to establish whether the semantic approach provides measurable advantages over conventional communication or simpler semantic processing strategies.

Model comparison should consider both semantic and conventional communication metrics. Semantic similarity and task success can indicate whether the intended meaning is preserved, while transmitted data volume, spectral efficiency, latency, and energy consumption reveal communication and implementation costs. Reporting several complementary metrics prevents a model from appearing superior simply because it has been optimized for one particular measure.

Different semantic models may also have different strengths under different channel conditions. A representation that performs well in a high-quality channel may degrade rapidly under packet loss or interference, while another model may sacrifice some compression efficiency in exchange for greater robustness. Comparative experiments should therefore include multiple operating conditions rather than relying on a single ideal channel configuration.

Statistical analysis can further strengthen comparisons. Repeated experiments, confidence intervals, variance measurements, and appropriate significance analysis can help determine whether observed differences are meaningful rather than caused by experimental randomness. Hyperparameter selection and training procedures should also be reported sufficiently to support fair interpretation.

A useful comparative framework should additionally examine scalability. Models that perform well on small datasets or powerful computing platforms may become impractical when deployed across large numbers of edge devices. Memory requirements, inference time, communication overhead, and model-update costs should therefore be considered alongside semantic performance. Comparative evaluation ultimately provides evidence for selecting appropriate semantic communication architectures for particular applications. A rigorous study should clearly identify the conditions under which each model performs well or poorly, rather than declaring a single universal best approach. This application- and environment-aware comparison is essential for advancing semantic communication from conceptual models toward practical deployment.

12.4.4 Reproducibility, Benchmarking, and Experimental Validation

Reproducibility is essential for establishing confidence in experimental results obtained from semantic communication systems. The complexity of modern semantic models, combined with variations in datasets, hardware, communication conditions, and training procedures, can make experimental results difficult to reproduce if implementation details are not adequately documented. A reproducible evaluation framework should therefore provide sufficient information for independent researchers to understand, repeat, and validate the reported experiments.

Benchmarking requires clearly defined datasets, baseline methods, evaluation metrics, communication conditions, and experimental procedures. A common benchmark should specify the semantic task being evaluated, the available communication resources, channel assumptions, model configurations, and

required performance measurements. Consistent benchmarks allow different semantic communication approaches to be compared under equivalent conditions and reduce ambiguity in reported improvements.

Experimental validation should examine whether a proposed approach performs consistently across multiple scenarios. Testing under different channel qualities, device capabilities, traffic loads, semantic tasks, and computational constraints can reveal whether performance improvements are robust or dependent on a particular configuration. Real-world validation is also valuable because practical systems may experience hardware limitations, timing effects, sensor imperfections, and environmental variability that are difficult to represent fully in simulation. Reproducibility also depends on transparent reporting of implementation details. Researchers should document model architectures, training procedures, dataset partitions, preprocessing methods, hyperparameters, hardware platforms, channel configurations, and evaluation criteria. Where possible, implementation resources and experimental configurations should be made available so that other researchers can independently reproduce the results.

Experimental validation should additionally distinguish between improvements caused by semantic communication itself and improvements resulting from differences in model size, training data, computational resources, or other experimental factors. Appropriate baselines and controlled experiments are therefore necessary. Ablation studies can help determine which components of a semantic communication architecture contribute most strongly to the observed performance.

A strong benchmarking methodology should report both improvements and limitations. Negative results, failure conditions, computational costs, and degradation under adverse channels provide valuable information for future research and practical deployment. Reproducible evaluation consequently strengthens the credibility of semantic communication research. Overall, reproducibility, benchmarking, and experimental validation establish a systematic foundation for assessing meaning-driven communication technologies. They ensure that reported improvements are measurable, comparable, repeatable, and supported by controlled evidence, enabling semantic communication research to progress from individual demonstrations toward reliable and standardized experimental evaluation.

Chapter 13

Future Directions toward Fully Meaning-Driven Intelligent Networks

13.1 From Semantic Communication Systems to Fully Semantic-Native Networks

13.1.1 Evolution from Bit-Centric Networks to Meaning-Centric Networks

Conventional communication networks have traditionally been designed around the reliable transmission of bits. Their primary objective is to transport digital information from a source to a destination with high accuracy, throughput, and reliability. Although this approach has enabled the development of modern Internet and wireless infrastructures, it becomes increasingly inefficient when intelligent applications do not require every transmitted bit. Autonomous systems, IoT devices, intelligent agents, and AI applications are often interested in the meaning, intent, or task relevance contained within information rather than its exact representation. This motivates the transition from bit-centric communication toward meaning-centric networking.

Semantic communication introduces an important step in this evolution by extracting task-relevant information before transmission. Instead of treating all data equally, the communication system identifies information that contributes to the intended application and represents it in a compact semantic form. However, many existing semantic communication approaches still operate within fundamentally conventional network infrastructures. Future networks are expected to move beyond this limited integration toward architectures in which semantic understanding becomes a fundamental networking capability.

A meaning-centric network would allow communication decisions to depend on semantic importance, contextual relevance, user intent, and application objectives. Routing, scheduling, resource allocation, caching, and transmission control could all be influenced by the value of information rather than only by packet size or network-layer characteristics. Network nodes could determine which information needs to be delivered, which information can be compressed or omitted, and which information should receive priority based on its contribution to the final task.

This evolution also changes the role of network endpoints. Devices would increasingly function as intelligent semantic entities capable of understanding, transforming, and exchanging knowledge. Edge nodes could provide distributed semantic processing, while cloud infrastructure could maintain broader knowledge and foundation models. Communication would consequently become more closely integrated with sensing, computing, reasoning, and decision-making.

The transition toward meaning-centric networks therefore represents more than an improvement in data compression. It reflects a fundamental change in the objective of networking: from transporting information as accurately as possible toward delivering the information that is most useful for achieving an intended purpose. This evolution provides the conceptual foundation for fully semantic-native intelligent networks.

13.1.2 Architectural Principles of Fully Semantic-Native Communication

Fully semantic-native communication requires a fundamental redesign of network architecture so that semantic understanding is integrated into communication, computing, sensing, and application processes. Rather than adding semantic processing as an independent function to an existing bit-oriented infrastructure, semantic capabilities would become intrinsic components of network operation. The architecture would therefore need to support meaning-aware information representation, processing, transmission, routing, resource management, and decision-making.

One important architectural principle is task-oriented communication. Network operations should be driven by the objective for which information is being exchanged. The same source information may require different semantic representations depending on whether it is intended for classification, control, monitoring, prediction, or human interaction. A semantic-native network should therefore understand application requirements and adapt information exchange accordingly.

Another principle is context awareness. Meaning depends on the environment, previous information, user requirements, device capabilities, and application state. Semantic-native nodes should maintain relevant contextual knowledge and use it when interpreting or generating information. This can reduce redundant transmission because information already known to the receiver does not necessarily need to be transmitted again. Semantic-native architectures should also integrate communication and computation. Semantic extraction, model inference, knowledge retrieval, and reconstruction may be distributed between devices, edge nodes, and cloud platforms according to latency, energy, and computational requirements. This creates an architecture in which communication resources and computing resources are jointly managed according to semantic objectives.

Trust, privacy, and security must also be incorporated into the architecture from the beginning. Semantic information can reveal sensitive context and may influence autonomous decisions. Authentication, provenance, integrity verification, access control, and privacy-preserving mechanisms therefore need to operate together with semantic processing. Finally, semantic-native networks should support interoperability among heterogeneous models, devices, and knowledge sources. Common semantic representations, interfaces, metadata, and model-management mechanisms will be important for enabling different network components to exchange meaningful information. Thus, fully semantic-native communication requires an architecture built around meaning, context, tasks, intelligence, computation, security, and adaptability. These principles can transform semantic communication from a specialized transmission technique into a fundamental architectural paradigm for future intelligent networks.

13.1.3 Autonomous Semantic Network Management and Optimization

Future semantic networks are expected to manage communication and computing resources with increasing levels of autonomy. Conventional network management typically depends on predefined policies, centralized controllers, and manually configured parameters. Such approaches become difficult to maintain when networks contain large numbers of heterogeneous devices, intelligent applications, edge resources, and dynamically changing communication conditions. Autonomous semantic network management addresses this challenge by allowing network components to understand semantic requirements and make appropriate operational decisions with limited human intervention.

A semantic-aware management system can continuously observe network conditions, application requirements, semantic traffic characteristics, and resource availability. Information such as channel quality, traffic priority, task importance, device energy, computing capacity, and semantic quality can be combined to determine suitable network configurations. Rather than allocating resources according to packet volume alone, the system can prioritize information according to its contribution to application objectives.

Artificial intelligence can support autonomous decision-making at different levels. Learning models can predict traffic requirements, identify changing channel conditions, estimate semantic quality, and select appropriate transmission or processing strategies. Optimization mechanisms can then determine resource allocation, routing, scheduling, model selection, and processing placement. Feedback from application performance can be used to evaluate whether these decisions successfully maintained the required semantic service quality.

Autonomous management should also support continuous adaptation. If an edge device becomes overloaded, semantic processing can be transferred to another available resource. If channel conditions deteriorate, the system can modify semantic representation or protection levels. If an application becomes more time-sensitive, network resources can be reallocated toward lower-latency processing and transmission. Such decisions can occur dynamically without requiring manual reconfiguration.

However, autonomy must operate within clearly defined constraints. Incorrect decisions can have significant consequences in autonomous vehicles, industrial systems, healthcare applications, and other critical environments. Semantic network controllers should therefore incorporate security, trust, policy constraints, explainability, and safe fallback mechanisms. Autonomous semantic network management ultimately aims to create networks that can understand their operational requirements and intelligently manage themselves. By combining semantic awareness with AI-driven optimization, future networks can continuously align communication and computing resources with application needs while improving efficiency, responsiveness, and resilience.

13.1.4 Self-Adaptive and Self-Evolving Meaning-Driven Networks

Self-adaptive and self-evolving semantic networks represent a further development beyond autonomous network management. An autonomous system can make decisions without direct human intervention, whereas a self-evolving system can also modify aspects of its models, policies, representations, and operational strategies as its environment changes. This capability is particularly important for future meaning-driven networks because the meaning and relevance of information may change with users, environments, applications, and available knowledge.

Self-adaptation enables a semantic network to respond to immediate changes. For example, a system may adjust semantic compression when bandwidth becomes limited, select a lightweight model when device energy decreases, or modify processing placement when an edge node becomes overloaded. These adaptations allow the network to maintain acceptable semantic performance despite changing operating conditions.

Self-evolution introduces longer-term changes. Semantic models may need to learn new concepts, update knowledge representations, adapt to new communication patterns, or accommodate previously unseen environments. Distributed learning and continual learning mechanisms can support this process by incorporating new observations into existing models. However, uncontrolled learning can introduce incorrect or malicious knowledge. Model validation, provenance tracking, trust assessment, and controlled deployment are therefore essential.

A self-evolving network should also recognize when its current semantic assumptions are no longer appropriate. Monitoring mechanisms can detect changes in data distributions, task requirements, user behavior, or environmental conditions. When significant changes occur, the network can initiate model adaptation, knowledge updates, policy modification, or resource reconfiguration. The system should preserve stable knowledge while selectively adapting components affected by environmental change.

Such networks will require mechanisms for balancing adaptation with reliability. Constant model changes can make system behavior difficult to predict and validate. Future architectures may therefore employ controlled evolution, where proposed changes are evaluated before being incorporated into operational components. Self-adaptive and self-evolving networks ultimately represent a vision of communication infrastructure that can learn, adjust, and improve throughout its operational lifetime. By combining semantic awareness, continual learning, autonomous optimization, contextual reasoning, and secure model management, meaning-driven networks can remain effective even as their environments and application requirements evolve.

13.2 Foundation Models, Generative AI, and Next-Generation Semantic Intelligence

13.2.1 Foundation Models as General-Purpose Semantic Communication Engines

Foundation models provide an important future direction for semantic communication because they offer broad capabilities for understanding and generating information across diverse tasks and modalities. Traditional semantic communication systems are often developed for specific applications, such as image classification, speech recognition, or sensor interpretation. Such specialization can limit their flexibility when the communication environment or application changes. Foundation models provide an opportunity to develop more general-purpose semantic communication engines capable of supporting multiple tasks using shared learned knowledge.

A foundation-model-based semantic system could perform semantic extraction by identifying the information most relevant to the receiver's task. For text, the model could identify concepts, intent, entities, and contextual relationships. For images and video, it could identify objects, activities, and environmental conditions. Multimodal foundation models could combine information from several modalities and construct richer semantic representations.

At the receiver, the same or a compatible foundation model could interpret the transmitted semantic representation and use contextual knowledge to reconstruct or reason about the intended information. This could reduce dependence on task-specific communication pipelines. Instead of designing an independent semantic encoder and decoder for every application, a common foundation model could provide reusable semantic capabilities across multiple services. Foundation models could also support dynamic task adaptation. A semantic representation generated for one task may not be appropriate for another. By

incorporating task descriptions, user requirements, or application context, a foundation model could adapt the information it extracts and transmits. This creates a more flexible relationship between communication and application objectives.

However, foundation models introduce substantial computational, energy, storage, and security requirements. Large models may be difficult to deploy directly on resource-constrained devices. Model compression, efficient inference, edge deployment, distributed execution, and specialized accelerators will therefore be important. In addition, foundation models may generate incorrect interpretations or inherit biases from training data, making validation and trust mechanisms necessary. Foundation models consequently offer the possibility of transforming semantic communication into a general-purpose intelligence layer for information exchange. Their ability to understand diverse modalities, tasks, and contexts could enable semantic networks to move beyond narrowly designed communication models toward flexible and reusable meaning-driven communication engines.

13.2.2 Transformer-Native Architectures for Future Semantic Networks

Transformer-based architectures have become an important foundation for modern intelligent information processing because they can model relationships among elements within complex sequences and multimodal inputs. Their ability to capture contextual dependencies makes them particularly relevant to semantic communication, where meaning often depends on relationships among words, objects, events, sensor observations, and contextual information. Future semantic networks may therefore increasingly adopt transformer-native architectures for semantic extraction, representation, decoding, and decision-making.

A transformer-native semantic communication system could use attention mechanisms to identify the most relevant components of an input according to the target task. Instead of assigning equal importance to all elements, the model can emphasize information that contributes strongly to the intended meaning. This provides a natural mechanism for semantic prioritization and selective transmission. Transformers can also support multimodal semantic communication. Text, speech, images, video, and sensor information can be represented within compatible learned spaces, allowing the system to reason about relationships among different modalities. Such capabilities are particularly valuable for intelligent environments where semantic understanding depends on combining observations from multiple sources.

Future transformer-native networks may also integrate communication decisions directly into the model architecture. The model could jointly determine what information should be transmitted, how it should be represented, and how the receiver should interpret it. This could reduce the separation between semantic processing and communication control that exists in many current systems. However, transformer architectures can require significant computational resources and memory. Efficient attention mechanisms, model compression, quantization, pruning, knowledge distillation, and hardware acceleration will therefore be important for edge and mobile deployment. Distributed transformer execution may also divide model processing across devices, edge nodes, and cloud infrastructure according to available resources.

Security and reliability present additional challenges. Transformer-based semantic systems may be vulnerable to adversarial inputs, manipulated context, model poisoning, or misleading information.

Robust training, provenance verification, secure model updates, and semantic validation will therefore be necessary. Transformer-native architectures have the potential to provide a common intelligent foundation for future semantic networks. Their contextual and multimodal capabilities can enable communication systems to understand relationships and relevance rather than treating information as independent data units, supporting more adaptive and application-oriented semantic information exchange.

13.2.3 Generative AI and Diffusion Models for Intelligent Communication

Generative AI introduces new possibilities for semantic communication by enabling systems not only to interpret transmitted information but also to generate missing, compressed, or task-relevant content. Conventional communication systems primarily focus on delivering information generated at the source, whereas generative semantic communication can use learned models and contextual knowledge to reconstruct information that was not explicitly transmitted. This capability can significantly reduce communication requirements when the receiver possesses sufficient knowledge to generate an appropriate representation of the intended content.

Generative models can be used to reconstruct semantic information from compact representations. Instead of transmitting a complete image, video sequence, or other high-dimensional signal, a semantic transmitter may communicate information describing the important concepts or conditions. A generative model at the receiver can then use this information and its learned knowledge to produce a representation consistent with the transmitted semantics. The objective is not necessarily exact reproduction but preservation of the information required by the application. Diffusion models provide one potential framework for such generative communication because they can generate complex data through iterative refinement. In future systems, semantic information could guide generation so that the receiver produces content consistent with the intended meaning. This may be useful for multimedia communication, virtual environments, digital twins, and other applications where complete source reconstruction is not always necessary.

Generative AI can also support semantic error recovery. When part of a semantic representation is lost or corrupted, the receiver may use contextual information and generative models to estimate plausible missing information. Such capabilities can improve robustness under unreliable channels. However, generated content may contain details that were not present in the original information. This creates a critical distinction between semantic correctness and generative plausibility.

Therefore, future generative semantic communication systems will require mechanisms for controlling hallucination, maintaining consistency with source information, and evaluating task-level correctness. Security and trust are also important because manipulated semantic prompts or malicious contextual information could cause incorrect generation. Generative AI and diffusion models thus offer a promising direction for communication systems in which compact semantic information guides intelligent reconstruction. Their successful integration will depend on balancing communication efficiency with semantic fidelity, computational cost, reliability, and control over generated information.

13.2.4 Distributed Foundation Models for Network-Wide Semantic Intelligence

Distributed foundation models represent a promising direction for creating network-wide semantic intelligence across devices, edge nodes, and cloud infrastructure. Large foundation models can provide broad semantic understanding but may require computational and memory resources beyond the capabilities of individual edge devices. Distributing model execution, training, storage, and knowledge across network resources can therefore provide a practical approach to deploying advanced semantic intelligence.

In a distributed architecture, different components of a foundation model may operate at different locations. Resource-constrained devices can perform initial sensing and lightweight semantic processing, while edge nodes can execute more computationally demanding inference. Cloud platforms can maintain large-scale models, global knowledge, and long-term learning processes. This arrangement allows semantic intelligence to be positioned according to latency, energy, privacy, and computational requirements. Distributed foundation models can also support collaborative semantic understanding. Multiple devices may observe different aspects of the same environment and contribute information to a shared semantic model. The network can combine these observations to construct a broader understanding than any individual device could achieve. Such collaboration can support intelligent transportation, industrial environments, smart cities, distributed robotics, and large-scale IoT systems.

Privacy and security are critical considerations. Raw data do not necessarily need to be transferred to a central model when local or edge processing can extract relevant semantic information first. Federated learning and privacy-preserving model updates can further reduce exposure of local data. At the same time, distributed models require mechanisms for authenticating participants, validating model updates, maintaining provenance, and preventing malicious contributions. Model consistency and resource management also present challenges. Different nodes may have different model versions, computational capabilities, and available resources. Efficient model synchronization and adaptive model placement will therefore be necessary. The network may select different model sizes or components depending on the capabilities and requirements of individual nodes.

Distributed foundation models ultimately provide a pathway toward network-wide semantic intelligence, where intelligence is not concentrated in a single centralized platform but distributed throughout the communication infrastructure. By combining edge processing, cloud intelligence, collaborative learning, privacy protection, and adaptive resource management, future semantic networks could develop a shared and continuously evolving understanding of their environments and applications.

13.3 Semantic Digital Twins and Intelligent Network Ecosystems

13.3.1 Digital Twin-Based Modeling of Semantic Communication Networks

Digital twins provide a promising framework for modeling and managing future semantic communication networks by maintaining a digital representation of physical communication, computing, sensing, and semantic-processing resources. In a semantic communication environment, a digital twin can represent not only network topology and resource states but also semantic traffic, application requirements, model configurations, and task-level performance. This enables network operators and intelligent controllers to observe and analyze the behavior of a meaning-driven network within a virtual environment.

A semantic network digital twin can integrate information from physical devices, wireless links, edge nodes, cloud resources, semantic models, and intelligent applications. The physical network continuously provides operational information to its digital counterpart, including channel conditions, resource utilization, semantic quality, latency, and task performance. The digital representation can then maintain an updated view of the network and provide a basis for analyzing how changes in configuration may affect semantic communication performance. One important advantage of digital twins is the ability to evaluate network decisions before applying them to the physical system. For example, different semantic encoding strategies, model configurations, resource allocations, routing policies, or processing locations can be evaluated within the virtual environment. This can reduce the risk associated with directly testing new configurations on operational networks. The digital twin can also support what-if analysis by examining the expected impact of changing traffic patterns, channel conditions, device availability, or application requirements.

Digital twins can further support coordination between communication and computation resources. Since semantic processing may occur across devices, edge infrastructure, and cloud platforms, the digital representation can model these resources jointly. This allows intelligent controllers to evaluate whether a semantic task should be processed locally, transferred to an edge node, or handled by cloud infrastructure.

The effectiveness of a semantic digital twin depends on the accuracy and timeliness of its representation. Outdated network states or inaccurate semantic models can lead to inappropriate optimization decisions. Secure synchronization and reliable data collection are therefore important. Digital twin-based modeling can provide a foundation for continuous observation, prediction, experimentation, and optimization of semantic communication networks. By connecting physical network behavior with an intelligent virtual representation, digital twins can support more efficient and adaptive management of increasingly complex meaning-driven network ecosystems.

13.3.2 Semantic Network Simulation and Predictive Optimization

Semantic network simulation provides a controlled environment for studying how communication, computing, semantic processing, and application performance interact under different operating conditions. Traditional network simulation generally focuses on parameters such as throughput, delay, packet loss, and resource utilization. Semantic network simulation extends this perspective by incorporating semantic representations, task requirements, intelligent models, and application-level outcomes into the evaluation process.

A semantic simulation environment can reproduce different combinations of channel conditions, network traffic, device capabilities, semantic models, and application requirements. Researchers can then examine how these factors influence semantic quality and task performance. For example, the simulation can evaluate how semantic task accuracy changes when bandwidth decreases, channel reliability deteriorates, or computational resources become constrained. Such experiments can reveal operating conditions that may not be practical or safe to test directly in physical networks.

Predictive optimization builds upon simulation by using the virtual environment to estimate the consequences of future network decisions. An intelligent controller can evaluate candidate configurations before selecting an operational strategy. Possible decisions may involve semantic compression, model

selection, bandwidth allocation, processing placement, routing, or transmission parameters. The system can estimate the expected semantic quality, latency, energy consumption, and reliability associated with each alternative.

Machine learning can further enhance predictive optimization by learning relationships between network states and semantic performance. Historical observations generated by physical networks or simulations can be used to predict future system behavior. This allows the network to anticipate congestion, resource shortages, channel degradation, or changes in semantic workload and adjust its configuration proactively rather than responding only after performance has deteriorated. Simulation and predictive optimization should nevertheless account for the limitations of virtual models. An overly simplified simulation may fail to represent hardware constraints, real wireless interference, processing overhead, or unexpected environmental conditions. Validation against real-world measurements is therefore important when simulation results are used to guide operational decisions.

Semantic network simulation and predictive optimization consequently provide a foundation for proactive rather than purely reactive network management. By evaluating possible configurations and predicting their semantic consequences before deployment, future networks can make better-informed decisions while reducing unnecessary resource consumption and improving task-level performance.

13.3.3 Intelligent Semantic Network Orchestration and Coordination

Intelligent semantic network orchestration involves coordinating communication, computing, sensing, semantic models, and application services according to the requirements of meaning-driven tasks. Conventional orchestration generally focuses on deploying network functions and allocating infrastructure resources. In semantic networks, orchestration must additionally consider what information is needed, how it should be represented, where semantic processing should occur, and how different network components should cooperate to achieve an application objective.

A semantic orchestration framework can receive requirements from intelligent applications, users, or autonomous agents and translate them into communication and computing policies. For example, an application may require highly reliable semantic information with low latency, while another may prioritize energy efficiency and reduced communication overhead. The orchestration system can interpret these requirements and determine suitable semantic models, network resources, processing locations, and communication strategies.

Coordination becomes particularly important when semantic tasks span multiple distributed nodes. An IoT application may require information from several sensors, edge processors, and cloud services. Each component may possess only partial knowledge of the environment, making coordinated semantic processing necessary. Orchestration mechanisms can determine which nodes should contribute information, how semantic results should be combined, and where intermediate knowledge should be stored or processed. Intelligent orchestration can also adapt to changing network conditions. If an edge node becomes overloaded, processing can be migrated to another node. If network conditions deteriorate, the orchestration system can select a more compact semantic representation or modify communication priorities. If an application changes its requirements, the corresponding semantic and resource policies can be updated dynamically.

Security and trust must be integrated into orchestration decisions. Not every node should automatically receive access to sensitive semantic information or participate in critical tasks. Authentication, authorization, trust assessment, provenance, and integrity verification can help ensure that only appropriate resources participate in semantic workflows. Intelligent semantic orchestration therefore provides a mechanism for coordinating heterogeneous network components around application meaning and task objectives. Rather than managing communication, computing, and intelligence as separate resources, orchestration can treat them as interconnected capabilities that collectively support semantic tasks. This integrated approach is important for future edge-cloud systems, autonomous networks, distributed IoT environments, and intelligent multi-agent applications.

13.3.4 Autonomous Evolution of Semantic Network Ecosystems

Future semantic networks are expected to evolve continuously as devices, applications, communication technologies, semantic models, and environmental conditions change. Autonomous evolution refers to the ability of a network ecosystem to modify its models, policies, resources, and interactions in response to these changes without requiring continuous manual intervention. This represents a progression from adaptive communication systems toward intelligent network ecosystems capable of improving their behavior over time. An evolving semantic ecosystem can monitor changes in network conditions, semantic workloads, application requirements, device populations, and available knowledge. When significant changes are detected, the network can update its semantic models, modify resource allocation strategies, adjust communication policies, or introduce new processing capabilities. Such adaptation enables the network to remain useful even when its original operating assumptions become outdated.

Autonomous evolution can also occur through distributed learning and knowledge sharing. Edge devices and intelligent agents can contribute observations and locally learned information to a broader semantic knowledge structure. The ecosystem can use these contributions to improve its understanding of the environment. Continual learning mechanisms can allow semantic models to incorporate new concepts and changing patterns while preserving previously acquired knowledge. However, autonomous evolution creates significant challenges in reliability and control. A system that changes its own models or policies may behave differently from its originally validated configuration. Incorrect training data, malicious model updates, or unexpected environmental changes can also lead to undesirable behavior. Evolution therefore requires controlled model updates, validation mechanisms, provenance tracking, trust management, and secure rollback capabilities.

Another important requirement is interoperability. As the ecosystem evolves, new devices, models, and services may need to interact with existing components. Common semantic interfaces and standardized representations can help ensure that newly introduced capabilities remain compatible with the broader network. Autonomous evolution ultimately aims to create semantic networks that can learn from operational experience, adapt to environmental change, and improve their capabilities throughout their lifetime. Such ecosystems could become increasingly self-managing and context-aware, reducing the need for static configuration while maintaining appropriate safeguards for security, reliability, and responsible operation.

13.4 Open Research Challenges and the Roadmap toward Meaning-Driven Networking

13.4.1 Semantic Communication Standards and Cross-Domain Interoperability

The development of semantic communication requires standards that enable different devices, models, network infrastructures, and applications to exchange meaningful information consistently. Conventional communication systems benefit from well-established protocols and interfaces that define how information is represented, transmitted, and interpreted. Semantic communication introduces additional challenges because meaning can depend on the task, context, knowledge base, representation method, and learning model used by communicating entities. One major challenge is the absence of universally accepted semantic representations. Different semantic communication models may encode the same information using different learned representations, making direct interoperability difficult. A receiver may not be able to interpret a representation generated by an incompatible model. Future standards will therefore need to establish mechanisms for describing semantic representations, model capabilities, context, task requirements, and associated metadata.

Cross-domain interoperability is equally important. Semantic information generated in one domain may need to be used by another domain with different terminology, models, or knowledge structures. For example, information generated by an industrial sensing system may need to interact with transportation or energy-management systems. Semantic translation and common knowledge structures can help bridge these differences. Standards should also address security, privacy, trust, and provenance. Interoperable semantic systems need mechanisms for identifying information sources, verifying integrity, describing provenance, and enforcing access policies. Privacy requirements must be considered because standardized semantic metadata could potentially expose sensitive contextual information.

Another challenge is interoperability across heterogeneous hardware and software platforms. Semantic communication may involve mobile devices, IoT sensors, edge processors, cloud systems, intelligent agents, and foundation models. Standardized interfaces can allow these components to exchange semantic information without requiring identical internal architectures. Therefore, semantic communication standards should focus not only on how information is transmitted but also on how meaning, context, intent, trust, and model capabilities are described and exchanged. Developing such standards will be essential for moving semantic communication from isolated research prototypes toward interoperable, cross-domain meaning-driven networks.

13.4.2 Generalization, Scalability, and Adaptability of Semantic Models

Semantic models must operate across diverse users, environments, devices, tasks, and communication conditions if semantic communication is to become a general networking paradigm. Many existing approaches are developed for specific datasets and applications, which can limit their ability to generalize to unseen environments. A model trained under one set of conditions may perform poorly when the communication task, data distribution, language, sensing environment, or device characteristics change.

Generalization therefore represents a fundamental research challenge. Future semantic models should learn representations that capture important meaning while remaining robust to variations in the input environment. Multimodal and foundation-model approaches may provide broader generalization capabilities, but their effectiveness must be evaluated across diverse semantic tasks rather than only within a single benchmark. Scalability presents another challenge. A semantic network may contain

thousands or millions of devices, each with different computational capabilities, energy constraints, models, and application requirements. Centralized semantic processing may become impractical at such scales. Distributed intelligence, edge processing, model partitioning, and hierarchical knowledge management can help distribute the computational workload.

Adaptability is also essential because semantic environments evolve over time. New concepts, users, devices, applications, and environmental conditions may appear after a model has been deployed. Continual learning and adaptive model selection can help maintain semantic performance without requiring complete retraining from the beginning. However, adaptation must avoid catastrophic forgetting and uncontrolled changes in model behavior.

Another important issue is computational efficiency. Highly capable semantic models may require substantial memory, processing power, and energy. Model compression, lightweight architectures, efficient inference, and hardware acceleration will therefore be important for practical deployment. Future research must consequently develop semantic models that are generalizable across environments, scalable across large networks, and adaptable to changing conditions. Achieving these properties will require coordinated progress in machine learning, distributed computing, communication systems, knowledge representation, and network optimization.

13.4.3 Ethical, Social, and Governance Considerations in Meaning-Aware Networks

Meaning-aware networks introduce ethical and governance challenges because they can process information related to human behavior, preferences, intentions, environments, and decisions. Unlike conventional communication systems that primarily transport data, semantic networks may interpret information and determine which aspects are important for a particular task. This creates questions regarding privacy, autonomy, transparency, accountability, and appropriate use of intelligent communication technologies.

Privacy is a major concern because semantic representations can reveal sensitive information even when raw data are not transmitted. A system that extracts user intent, behavioral patterns, or contextual information may unintentionally expose characteristics that users did not explicitly provide. Privacy-preserving semantic processing, data minimization, access control, and appropriate governance mechanisms are therefore necessary. Transparency and explainability are also important. Semantic communication systems increasingly rely on machine-learning models whose internal decision processes may be difficult to interpret. When semantic interpretations influence autonomous decisions, users and operators may need to understand why particular information was considered important or why a particular action was selected. This is especially relevant in high-impact applications.

Accountability must also be clearly defined. Errors may arise from semantic models, communication channels, knowledge bases, data sources, or autonomous network decisions. Determining responsibility for incorrect semantic interpretation or automated action requires appropriate provenance and auditing mechanisms. Secure records of semantic processing and decision-making can support accountability. Fairness and representation should also be considered when semantic models are trained on data that may not adequately represent all users or environments. Models that perform well in one population or context may produce poorer semantic interpretations elsewhere. Continuous evaluation and responsible model

development are therefore necessary. Governance frameworks should establish how semantic information can be collected, processed, shared, stored, and used. They should also define appropriate levels of human oversight for autonomous semantic networks. Ethical and governance considerations are therefore not secondary issues but fundamental components of future meaning-driven networking. Responsible development requires balancing technical capability with privacy, transparency, fairness, accountability, human control, and societal benefit.

13.4.4 Research Roadmap toward Fully Intelligent Meaning-Driven Communication

The development of fully intelligent meaning-driven communication requires a gradual transition from task-specific semantic communication systems toward integrated semantic-native network ecosystems. This transition involves advances across semantic representation, intelligent communication, network architecture, distributed computing, security, evaluation, and standardization. A clear research roadmap can help organize these developments and identify the capabilities required at each stage. An initial priority is establishing reliable semantic representations and evaluation methodologies. Semantic communication systems need well-defined task-oriented metrics, benchmark datasets, standardized experimental environments, and reproducible evaluation procedures. These foundations will allow researchers to compare different approaches and identify which techniques provide genuine improvements in semantic efficiency and application performance.

The next stage involves deeper integration of semantic intelligence with network infrastructure. Semantic-aware routing, scheduling, resource allocation, edge computing, and network orchestration can transform semantic communication from an application-layer technique into a network-wide capability. Communication and computation should increasingly be optimized jointly according to semantic requirements. Advanced AI technologies will then provide broader semantic intelligence. Foundation models, multimodal models, generative AI, distributed learning, and intelligent agents can enable networks to understand diverse information and adapt to different tasks. These technologies should be developed together with mechanisms for model efficiency, robustness, privacy, trust, and secure distributed learning.

The longer-term objective is a self-adaptive and self-evolving semantic network ecosystem. Such a network would continuously observe its environment, understand application requirements, optimize communication and computation resources, update its semantic knowledge, and recover from failures. Digital twins and predictive optimization could support this evolution by allowing network decisions to be evaluated before physical deployment. Standardization and governance must progress alongside technical development. Interoperable semantic representations, secure interfaces, provenance mechanisms, privacy frameworks, and responsible AI principles will be necessary for large-scale deployment. Ultimately, the roadmap toward fully intelligent meaning-driven communication is not a single technological transition but a progressive convergence of semantic intelligence, communication, computing, networking, security, and autonomous decision-making. The long-term vision is a network that does more than transport information: it understands what information means, determines what is useful, adapts to changing objectives, and intelligently coordinates resources to support human and machine tasks.

BIBLIOGRAPHY

- [1] A Comprehensive survey on Semantic Communication in Non-Terrestrial Networks: Architectures, Methodologies, and Challenges. <https://arxiv.org/html/2606.05216v1>
- [2] A survey on Semantic Communications for intelligent wireless networks. (2022). In *Wireless Personal Communications* (Vols. 0123456789–0123456789, pp. 1–3) [Journal-article]. Springer Science+Business Media, LLC. <https://orbilu.uni.lu/bitstream/10993/54329/1/s11277-022-10111-7.pdf>
- [3] Ansere, J. A., Le, Q. N., & Nguyen, T. T. (2026). From Bits to Meaning: A survey of semantic communications for 6G networks. *EAI Endorsed Transactions on Tourism Technology and Intelligence*, 3(1). <https://doi.org/10.4108/eettti.11434>
- [4] Best Readings in Semantic Communications | IEEE Communications Society. (2025, July 14). IEEE Communications Society. <https://www.comsoc.org/publications/best-readings/semantic-communications>
- [5] Boursoulatz, E., Kurka, D. B., & Gunduz, D. (2019). Deep Joint Source-Channel coding for wireless image transmission. *IEEE Transactions on Cognitive Communications and Networking*, 5(3), 567–579. <https://doi.org/10.1109/tccn.2019.2919300>
- [6] Chaccour, C., Saad, W., Debbah, M., Han, Z., & Poor, H. V. (2022a, November 25). Less data, more knowledge: Building next generation semantic communication networks. *arXiv.org*. <https://arxiv.org/abs/2211.14343>
- [7] Chaccour, C., Saad, W., Debbah, M., Han, Z., & Poor, H. V. (2022b). Less data, more knowledge: Building next generation semantic communication networks. *alphaXiv*. <https://www.alphaxiv.org/abs/2211.14343>
- [8] Chaccour, C., Soorki, M. N., Saad, W., Bennis, M., Popovski, P., & Debbah, M. (2022). Seven Defining features of Terahertz (THZ) wireless Systems: a fellowship of communication and sensing. *IEEE Communications Surveys & Tutorials*, 24(2), 967–993. <https://doi.org/10.1109/comst.2022.3143454>
- [9] Checking your browser - reCAPTCHA. <https://pmc.ncbi.nlm.nih.gov/articles/PMC12027053/>
- [10] Chen, W., & Qin, Z. (2026). Real-World applications. In *Textbooks in telecommunication engineering* (pp. 179–191). https://doi.org/10.1007/978-3-032-11005-3_13
- [11] Dai, J., Wang, S., Tan, K., Si, Z., Qin, X., Niu, K., & Zhang, P. (2022). Nonlinear transform Source-Channel coding for semantic communications. *IEEE Journal on Selected Areas in Communications*, 40(8), 2300–2316. <https://doi.org/10.1109/jsac.2022.3180802>

- [12] Deep learning enabled semantic communications. Google Books. https://books.google.com/books/about/Deep_Learning_Enabled_Semantic_Communica.html?id=vA1g0QEACAAJ
- [13] Fan, D., Ji, G., Xu, P., Cheng, M., Sakaridis, C., & Van Gool, L. (2023). Advances in deep concealed scene understanding. *Visual Intelligence*, 1(1). <https://doi.org/10.1007/s44267-023-00019-6>
- [14] 3GPP – the mobile broadband standard. 3GPP. <https://www.3gpp.org/technical-specifications-and-releases/releases>
- [15] Feng, H., Yang, Y., & Han, Z. (2023, November 23). Scalable AI generative content for vehicular network semantic communication. *arXiv.org*. <https://arxiv.org/abs/2311.13782>
- [16] Fernandes, G., Fontes, H., & Campos, R. (2024, June 2). Semantic Communications: the New Paradigm Behind Beyond 5G Technologies. *arXiv.org*. <https://arxiv.org/abs/2406.00754>
- [17] Foundations of Semantic Communication networks. Google Books. https://books.google.com/books/about/Foundations_of_Semantic_Communication_Ne.html?id=qu05EQAAQBAJ
- [18] Foundations of Semantic Communication networks. (2025). <https://doi.org/10.1002/9781394247912>
- [19] Gunduz, D., Qin, Z., Aguerri, I. E., Dhillon, H. S., Yang, Z., Yener, A., Wong, K. K., & Chae, C. (2022, July 19). Beyond transmitting bits: context, semantics, and Task-Oriented communications. *arXiv.org*. <https://arxiv.org/abs/2207.09353>
- [20] Guo, S., Wang, Y., Zhang, N., Su, Z., Luan, T. H., Tian, Z., Xuemin, & Shen. (2024, May 2). A survey on Semantic Communication Networks: Architecture, security, and Privacy. *arXiv.org*. <https://arxiv.org/abs/2405.01221>
- [21] Introduction to Semantic Communications. (2025). In *Foundations of Semantic Communication Networks*, First Edition. John Wiley & Sons, Inc. <https://catalogimages.wiley.com/images/db/pdf/9781394247882.excerpt.pdf>
- [22] Iyer, S., Khanai, R., Torse, D., Pandya, R. J., Rabie, K. M., Pai, K., Khan, W. U., & Fadlullah, Z. (2022). A survey on Semantic Communications for intelligent wireless networks. *Wireless Personal Communications*, 129(1), 569–611. <https://doi.org/10.1007/s11277-022-10111-7>
- [23] Ji, J. (2025). Semantic Communications for 6G Networks: Theory, architectures, and challenges. *Academic Journal of Science and Technology*, 17(1), 66–75. <https://doi.org/10.54097/vwsfa141>
- [24] Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2021). A survey on Knowledge Graphs: Representation, acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2), 494–514. <https://doi.org/10.1109/tnnls.2021.3070843>

- [25] Kao, T., Jensen, K. T., De Ven Gido M, V., Bernacchia, A., & Hennequin, G. (2021, June 15). Natural continual learning: success is a journey, not (just) a destination. arXiv.org. <https://arxiv.org/abs/2106.08085>
- [26] Lan, Q., Wen, D., Zhang, Z., Zeng, Q., Chen, X., Popovski, P., & Huang, K. (2021, October 1). What is Semantic Communication? A View on Conveying Meaning in the Era of Machine Intelligence. arXiv.org. <https://arxiv.org/abs/2110.00196>
- [27] Lu, Z., Li, R., Lu, K., Chen, X., Hossain, E., Zhao, Z., & Zhang, H. (2023). Semantics-Empowered Communications: A Tutorial-Cum-Survey. *IEEE Communications Surveys & Tutorials*, 26(1), 41–79. <https://doi.org/10.1109/comst.2023.3333342>
- [28] Lyu, T., Noor-A-Rahim, M., O'Driscoll, A., & Pesch, D. (2024, July 24). Semantic Vehicle-to-Everything (V2X) communications towards 6G. arXiv.org. <https://arxiv.org/abs/2407.17186>
- [29] Meng, R., Gao, S., Fan, D., Gao, H., Wang, Y., Xu, X., Wang, B., Lv, S., Zhang, Z., Sun, M., Han, S., Dong, C., Tao, X., & Zhang, P. (2025, January 1). A survey of Secure semantic Communications. arXiv.org. <https://arxiv.org/abs/2501.00842>
- [30] Nguyen, L. X., Raha, A. D., Aung, P. S., Niyato, D., Han, Z., & Hong, C. S. (2025). A contemporary survey on semantic communications: theory of mind, generative AI, and deep joint Source-Channel coding. *IEEE Communications Surveys & Tutorials*, 28, 2377–2417. <https://doi.org/10.1109/comst.2025.3616973>
- [31] Qin, Z., Tao, X., Lu, J., Tong, W., & Li, G. Y. (2021, December 30). Semantic Communications: Principles and challenges. arXiv.org. <https://arxiv.org/abs/2201.01389>
- [32] Saad, W., Bennis, M., & Chen, M. (2019). A Vision of 6G Wireless Systems: applications, trends, technologies, and open research problems. *IEEE Network*, 34(3), 134–142. <https://doi.org/10.1109/mnet.001.1900287>
- [33] Semantic communications in 6G: coexistence, multiple access, and satellite networks. <https://arxiv.org/html/2506.11779v1>
- [34] Seo, H., Park, J., Bennis, M., & Debbah, M. (2021, August 12). Semantics-Native Communication with Contextual Reasoning. arXiv.org. <https://arxiv.org/abs/2108.05681>
- [35] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- [36] Strinati, E. C., & Barbarossa, S. (2021). 6G networks: Beyond Shannon towards semantic and goal-oriented communications. *Computer Networks*, 190, 107930. <https://doi.org/10.1016/j.comnet.2021.107930>
- [37] Towards effective and interpretable semantic communications. <https://arxiv.org/html/2408.04825v1>

- [38] Weng, Z., & Qin, Z. (2021). Semantic Communication systems for speech transmission. *IEEE Journal on Selected Areas in Communications*, 39(8), 2434–2444. <https://doi.org/10.1109/jsac.2021.3087240>
- [39] Xie, H., Qin, Z., Li, G. Y., & Juang, B. (2021). Deep learning enabled semantic communication systems. *IEEE Transactions on Signal Processing*, 69, 2663–2675. <https://doi.org/10.1109/tsp.2021.3071210>
- [40] Xin, G., Fan, P., & Letaief, K. B. (2024). Semantic Communication: A survey of its theoretical development. *Entropy*, 26(2), 102. <https://doi.org/10.3390/e26020102>
- [41] Yan, X., Xiumei, F., Yau, K. A., Zhixin, X., Rui, M., & Gang, Y. (2025). A review of Reinforcement Learning for Semantic Communications. *Journal of Network and Systems Management*, 33(3). <https://doi.org/10.1007/s10922-025-09927-y>
- [42] Yang, W., Du, H., Liew, Z., Lim, W. Y. B., Xiong, Z., Niyato, D., Chi, X., Shen, X. S., & Miao, C. (2022, June 10). Semantic Communications for Future Internet: fundamentals, applications, and challenges. *arXiv.org*. <https://arxiv.org/abs/2207.00427>
- [43] Yun, W. J., Park, S., Lee, R., Park, J., Ko, Y., & Kim, J. (2023). Semantic Communications and Networking. In *Signals and communication technology* (pp. 733–746). https://doi.org/10.1007/978-3-031-37920-8_29
- [44] Zhang, M., Abdi, M., Dasari, V. R., & Restuccia, F. (2025). Semantic Edge Computing and Semantic Communications in 6G networks: A unifying survey and research challenges. *Computer Networks*, 270, 111531. <https://doi.org/10.1016/j.comnet.2025.111531>
- [45] Zhang, P., Xu, X., Sun, M., Gao, H., Ma, N., Wang, X., Zhang, R., Wang, J., & Niyato, D. (2025, September 16). Towards Native AI in 6G Standardization: The Roadmap of Semantic Communication. *arXiv.org*. <https://arxiv.org/abs/2509.12758>
- [46] Zhang, Z., Xiao, Y., Ma, Z., Xiao, M., Ding, Z., Lei, X., Karagiannidis, G. K., & Fan, P. (2019). 6G Wireless Networks: vision, requirements, architecture, and key technologies. *IEEE Vehicular Technology Magazine*, 14(3), 28–41. <https://doi.org/10.1109/mvt.2019.2921208>

Semantic Communications: From Bit Transmission to Meaning-Driven Networking examines a fundamental shift underway in how communication systems are conceived — moving beyond the century-old goal of faithfully reproducing bits at a receiver, toward networks designed to convey meaning relevant to a specific task or goal. Since Claude Shannon's foundational work on information theory, communication engineering has been built around a single, powerful abstraction: transmit symbols so accurately that the receiver can reconstruct exactly what the transmitter sent, while deliberately setting aside any question of what those symbols actually mean. This framing enabled decades of progress in digital communication, but it is increasingly at odds with how modern networks are actually used. Sensor networks, autonomous systems, machine-to-machine coordination, and AI applications at the edge frequently transmit data that is redundant, irrelevant to the task at hand, or reconstructible from context — consuming bandwidth and energy to move information that was never necessary in the first place. This book explores what becomes possible when networks are redesigned around a different question: not "how do we move these bits accurately?" but "how do we convey the meaning that matters, efficiently?" The book's central premise is that the next major advance in communication systems will not come from transmitting bits faster, but from transmitting meaning smarter. As networks increasingly serve machines, sensors, and AI systems operating under real constraints on bandwidth, energy, and latency, semantic communications offers a framework for building networks that prioritize what actually matters to the task at hand — a shift the book argues is not merely incremental, but foundational to the next generation of networking.

Krishnan Visweswaran is a technology professional, researcher, and engineering leader based in California, USA, with extensive experience in wireless communications, software quality engineering, artificial intelligence, embedded systems, and next-generation networking technologies. Throughout his career, he has worked with leading technology organizations including Meta, Amazon, Qualcomm, and Cisco, contributing to complex hardware-software systems, automation frameworks, intelligent platforms, and large-scale system validation.

He pursued graduate studies in Electrical Engineering at California State University, Long Beach, with a focus on communication systems, VLSI, RF and microwave engineering, digital signal processing, computer networks, and embedded systems.

His research interests include 6G wireless systems, AI-native networking, machine learning, edge intelligence, cloud computing, cybersecurity, and intelligent automation. His academic and professional projects span artificial intelligence, radar systems, robotics, VLSI design, embedded systems, wireless technologies, communication networks, and software automation.

His work focuses on bridging artificial intelligence and next-generation communication technologies to develop intelligent, secure, resilient, and scalable systems for the future of global connectivity.



ISBN: 978-93-49-9291-4



9 789349 929142

 **SCIENCE** TECH
X P L O R E