This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TCCN.2026.3732515, IEEE Transactions on Cognitive Communications and Networking

Agentic TokenCom: A Chain-of-Agents Framework for Multimodal Token Communications

Feibo Jiang, *Senior Member, IEEE*, Lei Mao, Li Dong, Kezhi Wang, *Senior Member, IEEE*, Cunhua Pan, *Senior Member, IEEE*, Abbas Jamalipour, *Fellow, IEEE*

***Abstract*—In 6G networks, existing semantic communication systems struggle to achieve composable and generalizable end-to-end optimization under dynamic channel conditions and heterogeneous resource constraints. To address this challenge, we propose Agentic Token Communication (TokenCom), a hierarchical Chain-of-Agents autonomous framework composed of a reasoning chain and an execution chain, which transforms TokenCom structure design and instantiation from a static, one-shot configuration into a self-evolving process that is plannable, evaluable, reflective, and iterative. First, we establish an end-to-end multimodal TokenCom system model and summarize three modes of Large Language Model (LLM) token encoding and decoding. We then formulate the operation of Agentic TokenCom as a joint optimization problem of workflow-structure selection and component-instance selection: supported by a memory module, the upper reasoning chain forms a closed loop of planner, evaluator, and reflector agents to generate and continuously refine the workflow structure; under the resulting structural constraints, the lower execution chain dynamically performs model selection, parameter configuration, and interface alignment across different modalities and task requirements. Experimental results demonstrate the effectiveness and feasibility of the proposed framework.**

***Index Terms*—Token communication, Semantic communication, Agentic AI, LLM, 6G.**

I. Introduction

For emerging 6G applications such as immersive interaction, ultra-reliable and low-latency communications, and massive connectivity, the conventional communication paradigm centered on bit-level error-free transmission is approaching performance limits in terms of spectrum efficiency, energy consumption, and end-to-end latency [1], [2]. Semantic communication is task-oriented and transmits only the semantic information that substantively contributes to understanding, decision making, and control [3]. This principle can effectively reduce redundant overhead and markedly improve end-to-end system utility in weak channels and resource-constrained scenarios. By jointly optimizing source compression, channel coding, and task inference, semantic communication is expected to support a new communication modality that tightly integrates sensing, transmission, and generation and to serve as a key technical foundation for enabling native intelligence in 6G.

However, in the era of large models, existing semantic communication still suffers from critical limitations. On the one hand, semantic representations are often built upon encoders tailored to specific tasks or scenarios, which constrains cross-task generalization and structural composability [4]. On the other hand, a unified and standardized interface between semantic symbols and physical transmission is still absent, complicating end-to-end co-design and limiting adaptability to dynamic environments and multi-agent collaboration [5]. Meanwhile, token sequences within Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) inherently capture hierarchical semantics and enable cross-task reasoning, providing advantages in standardized representation, transmissibility, and reconfigurability [6]. Motivated by this insight, we propose a Token Communication framework, termed TokenCom, which treats tokens as fundamental transmission units and conveys and aligns model-level semantic states over the communication link, thereby enabling task-oriented robust reconstruction and cooperative intelligence.

Consequently, TokenCom and semantic communication differ substantially in the form of information units, and the encoding and decoding mechanisms. TokenCom adopts either continuous or discrete tokens as the fundamental units and directly conveys the semantic compression and reasoning results produced by MLLMs. It enables unified tokenization across text, images, and audio, and can further exploit the context-based generation capability of MLLMs to recover missing tokens, which provides stronger generalization and reconstruction capability in dynamic scenarios [7], [8]. These benefits, however, come at the cost of increased system implementation complexity and more challenging co-design requirements. A comparison between TokenCom and semantic communication is presented in Table I.

Although TokenCom provides a new implementation pathway for cross-modality transmission and generative reconstruction by using tokens to carry model-level semantic states, existing studies remain at an early stage, and a clear gap

This work was supported in part by the National Natural Science Foundation of China under Grants 62572184 and 41604117; in part by the Ministry of Education Humanities and Social Sciences Research Planning Foundation under Grant 25A10554016; in part by the Natural Science Foundation of Hunan Province under Grants 2024JJ5270 and 2025JJ50365.

Feibo Jiang (jiangfb@hunnu.edu.cn) is with the Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing and the Yuelushan Digital Intelligence Laboratory (Artificial Intelligence and International Communication), Hunan Normal University, Changsha 410081, China.

Lei Mao (202520294367@hunnu.edu.cn) is with the School of Information Science and Engineering, Hunan Normal University, Changsha, China.

Li Dong (Dlj2017@hunnu.edu.cn) is with the School of Computer Science, Hunan University of Technology and Business, Changsha 410205, China

Kezhi Wang (Kezhi.Wang@brunel.ac.uk) is with the Department of Computer Science, Brunel University London, UK.

Cunhua Pan (cpan@seu.edu.cn) is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China.

Abbas Jamalipour (a.jamalipour@ieee.org) is with the School of Electrical and Computer Engineering, University of Sydney, Australia, and with the Graduate School of Information Sciences, Tohoku University, Japan.

TABLE I: Comparison Between TokenCom and Semantic Communication.

Comparison Dimension	**TokenCom**	**Semantic Communication**
Information Unit	Tokens serving as semantic symbols, generated via MLLM-based semantic compression	Low-dimensional semantic features or embeddings extracted by task-specific neural networks
Encoding and Decoding	Tokenization and detokenization with generative semantic reconstruction	Semantic encoding and decoding through end-to-end trained models
Relation to MLLMs	Strong dependence on MLLMs for semantic understanding, reasoning, and generation	MLLMs are optional; typically rely on task-driven semantic compression networks
Cross-Modal Support	Native cross-modal support (text, image, audio) via unified token representation	Modality-specific encoders; lack a unified cross-modal semantic representation
Transmission Robustness	High robustness enabled by generative contextual completion and token-level recovery	Robustness achieved mainly via end-to-end training; no explicit generative recovery mechanism
Task Generalization	Strong generalization benefiting from MLLM's knowledge and multi-task reasoning capability	Limited generalization due to task-specific training and weak transferability
System Design Complexity	High complexity involving tokenizers, MLLMs, and token-level decoding pipelines	Moderate complexity with a single semantic encoder–decoder architecture

persists toward sustainable and autonomous optimization for dynamic networks and complex services. This gap becomes particularly evident in 6G scenarios, where channel quality, task objectives, and terminal resources are inherently time varying and heterogeneous. As a result, current TokenCom frameworks exhibit several limitations in dynamic adaptation and system-level coordination:

1) Static Workflow Paradigm: Most existing TokenCom systems adopt a static, one-shot workflow design paradigm, in which the processes of token compression, transmission, and recovery are predefined prior to communication and remain fixed throughout task execution. When the communication environment or service conditions change, such approaches generally lack online performance evaluation and error attribution mechanisms, making it difficult to adaptively refine the predefined workflow during execution.

2) Strong Task–Model Coupling: Conventional TokenCom methods typically fix the modality processing pipeline and task type at the design stage, and then construct dedicated semantic representations and encoding mechanisms for specific modalities and tasks. Under this design pattern, the system cannot dynamically adapt to actual modality characteristics and task requirements, thereby limiting the generality and scalability of TokenCom in multimodal and multi-task scenarios.

3) Lack of Resource-Aware Design: Existing TokenCom systems often assume that token encoding, inference, and decoding are executed at fixed locations and rely on a uniform model scale and execution procedure to complete communication tasks. This design fails to explicitly capture the heterogeneity between the transmitter and receiver in terms of computational capability, latency, and energy consumption, and thus hinders flexible trade-offs between token representation quality and system resource expenditure.

Therefore, we propose an Agentic TokenCom framework based on a chain of agents. By introducing a hierarchical multi-agent collaboration mechanism consisting of a reasoning chain and an execution chain, the proposed framework transforms the structure design and instantiation of TokenCom into an autonomous optimization process that is plannable, evaluable, and iterative. In particular, it can automatically generate and evolve TokenCom system configurations under dynamically varying channel conditions, task objectives, and resource constraints. The main contributions of this work are summarized as follows:

1) Reflection-Driven Self-Evolving Reasoning Chain: At the upper reasoning-chain level, we develop a workflow generation mechanism jointly realized by a planner, an evaluator, and a reflector, which advances TokenCom from static, one-shot structural selection to a reflection-driven self-evolving process. By incorporating workflow-level performance evaluation and error attribution feedback, the system can locally revise the current workflow and perform iterative optimization without altering the task objective, thereby continuously improving semantic communication performance and structural adaptability under time-varying channels and uncertain service conditions.

2) Modality and Task-Adaptive Execution Chain: Subject to the workflow structure produced by the reasoning chain, the lower-level execution chain adaptively selects appropriate models for instantiation according to specific modality types and task requirements. By transforming model selection from a static configuration into a workflow-constrained dynamic decision process, the system can flexibly invoke encoding, projection, and inference models with matched token representation capabilities for different modality characteristics and task forms, ensuring that the instantiated TokenCom system effectively supports multimodal understanding or generation tasks under the prescribed workflow.

3) Resource-Aware Workflow Orchestration: We further introduce a workflow orchestration mechanism that accounts for heterogeneous terminal resources, in which the reasoning chain and the execution chain jointly design key token coding and decoding modules in TokenCom. Specifically, the reasoning chain plans the deployment locations of token encoding and decoding modules based on the resource conditions of the transmitter and receiver, while the execution chain selects models with appropriate scale and complexity to implement input-level, intermediate-layer, or output-level token encoding and decoding. In this manner, the proposed framework enables a dynamic balance between token abstraction levels and

system execution cost.

The remainder of this paper is organized as follows. Section II reviews related work. Section III presents the system model of TokenCom. Section IV elaborates on the key designs of the proposed Agentic TokenCom framework. Section V describes the experimental settings and reports performance evaluation results. Section VI concludes the paper.

II. Related Work

A. LLM-Empowered Semantic Communication

Zhao *et al.* proposed LaMoSC, a LLM-driven visual semantic communication system. To address inaccurate decoding at low signal-to-noise ratio (SNR) when relying on a single visual feature, LaMoSC introduces multimodal features from both vision and text, and deeply fuses them in an end-to-end encoder and decoder network via an attention mechanism to reconstruct the original visual information [9]. Guo *et al.* investigated LLM-based understanding level semantic communication, where visual data are first converted into natural language descriptions, and a pretrained LLM is then used to quantify semantic importance and perform semantic error correction [10]. Cao *et al.* proposed MLLM-PSC, which leverages an MLLM to support multimodal semantic understanding and generation, using textual semantics as a consistent intermediate representation. In addition, few-shot learning is employed to protect sensitive semantic information, and numerical results demonstrate the effectiveness of the proposed approach [11].

B. Agentic Semantic Communication

Güler *et al.* proposed the Semantic Communication Game framework, which introduced an external agent that may influence the receiver, either adversarially or cooperatively, to characterize the coupling between semantic transmission and decision making under multi-party interactions, thereby providing an analyzable game theoretic modeling paradigm for agent-oriented semantic communication [12]. Zhou *et al.* studied GPT semantic information extraction communication for multi-agent collaboration. To mitigate redundant observations and excessive signaling overhead, they used an LLM to extract key semantic information from observed data and represent it in natural language, enabling more compact semantic message exchange [13]. Yang *et al.* developed an agent-driven generative semantic communication framework, where reinforcement learning enables an adaptive tradeoff between semantic extraction and semantic sampling. Together with cross-modal semantic representations and predictive decoding, agents can organize transmissible semantic content according to task context and channel conditions [14].

C. TokenCom

Qiao *et al.* introduced TokCom, which represents heterogeneous information in a unified token space and exploits inter-token contextual relationships through Transformer-based processing at both communication ends [15]. For multimodal and multitask communications, Jiang *et al.* developed a vision-language-model-based TokenCom system, which combines dual-resolution visual tokenization, bilateral attention fusion, and a KAN-based modality projector to generate compact and semantically aligned visual tokens. A joint VLM–channel coding scheme was further introduced to improve task performance over wireless channels [16]. Lee *et al.* investigated semantic-preserving token grouping and packing and proposed the SemPA-GBeam algorithm to optimize Average Token Similarity [17].

However, existing studies have not systematically summarized the working paradigms of Token encoders in TokenCom, nor have they proposed automated design methodologies. To address this gap, this study develops an Agentic AI–driven TokenCom optimization framework, which aims to collaboratively accomplish the design and optimization of TokenCom through automated reasoning chains and execution chains.

III. System Model

This section presents an end-to-end multimodal TokenCom framework built upon MLLMs. The proposed system treats tokens in MLLMs as the fundamental transmission units. At the transmitter, multimodal tokenization and semantic alignment are jointly performed, and the encoded tokens are transmitted over the physical channel. At the receiver, robust token recovery and decoding are conducted, followed by cross-modal detokenization. In this manner, the proposed framework supports multimodal understanding and generation tasks under challenging channel conditions. The overall architecture of the complete TokenCom system is illustrated in Fig. 1.

A. Notations

We consider a communication system that supports multimodal inputs, where the modality set is defined as $\mathcal{X} = \{\text{Image}, \text{Video}, \text{Audio}, \ldots\}$. For any modality $X \in \mathcal{X}$, the transmitter input consists of two parts: the raw modality signal I_X (e.g., an image, video, or audio signal) and the corresponding textual instruction or query t.

B. Transmitter

The transmitter consists of a modality tokenizer MT_X, an input projector $\Theta_{X\to T}$, an LLM token encoder, and a channel encoder $C_\beta(\cdot)$.

1) Modality Tokenizer: The raw modality input I_X is first encoded into multimodal embedding tokens as follows:

$$\mathbf{F}_X = \mathrm{MT}_X(I_X), \tag{1}$$

where $\mathbf{F}_X$ denotes the token feature embeddings associated with modality X. The modality tokenizer MT_X can be flexibly selected according to the modality type.

2) Input Projector: To map and align the modality features to the textual feature space, we introduce an input projector to produce modality-conditioned soft prompts as follows:

$$\mathbf{P}_X = \Theta_{X\to T}(\mathbf{F}_X). \tag{2}$$

The generated soft prompts $\mathbf{P}_X$ are concatenated with the textual embedding tokens $\mathbf{F}_T$ obtained by feeding t into the text tokenizer, and the resulting sequence is jointly fed into

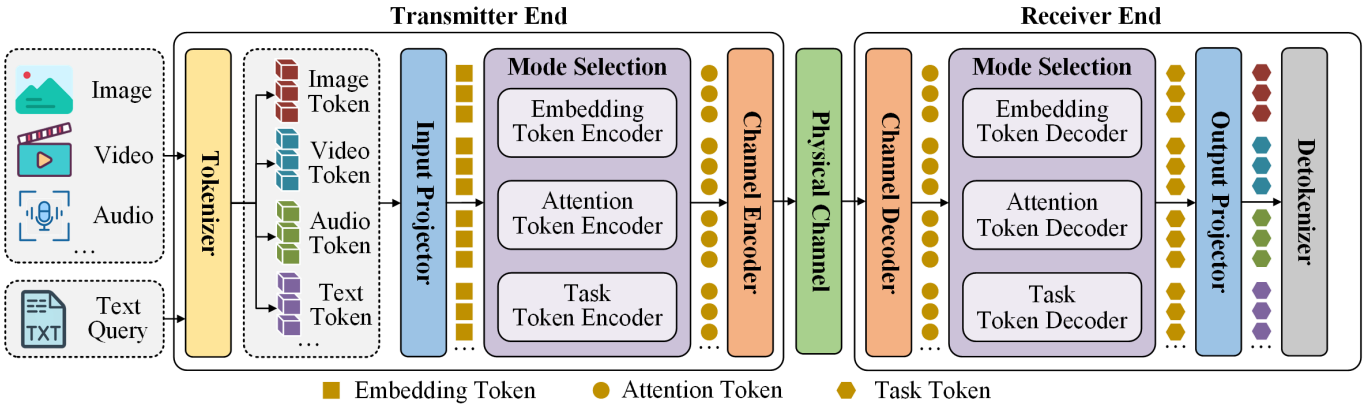

Fig. 1: Components of the TokenCom System.

the LLM backbone. For a multimodal training pair (I_X, t_M), the training objective of the input projector can be formulated as minimizing a conditional text generation loss:

$$\min_{\Theta_{X\to T}} \mathcal{L}_{\text{ip}}\big(\text{LLM}(\mathbf{P}_X, \mathbf{F}_T),\, t_M\big), \tag{3}$$

where $\mathcal{L}_{\text{ip}}$ is typically a token-level cross-entropy loss that encourages the LLM to generate a text output t_M that is semantically consistent with the instruction t under multimodal input I_X.

3) LLM Token Encoder: Upon receiving the multimodal input tokens $(\mathbf{P}_X, \mathbf{F}_T)$, the LLM backbone performs multimodal token encoding, reasoning, and outputs

$$(\tilde{t}, \mathbf{S}_X) = \text{LLM}(\mathbf{P}_X, \mathbf{F}_T), \tag{4}$$

where $\tilde{t}$ denotes the tokens for the text generation task, and $\mathbf{S}_X$ denotes the tokens for multimodal generation with respect to modality X, which will be used for subsequent cross-modal content generation. Depending on the position of the LLM in TokenCom, the LLM token encoder can take three forms.

a) Embedding Token Encoder: When the LLM is deployed at the receiver side, the embedding tokens produced by the projector are not further encoded and are directly fed into channel coding and decoding. In this case, the token encoder is simply constructed from the aligned input semantic conditions, given by

$$\mathbf{Z} = \mathcal{G}_{\text{embd}}(\mathbf{P}_X, \mathbf{F}_T) := \text{SemMap}\{\mathbf{P}_X, \mathbf{F}_T\}, \tag{5}$$

where SemMap denotes an identity mapping and $\mathcal{G}_{\text{embd}}(\cdot)$ represents the embedding token construction operator.

This scheme does not rely on the internal reasoning outcomes of the LLM and transmits only the multimodal embedding tokens, which is suitable for scenarios that emphasize low latency and low computational complexity, while the receiver has sufficient computing capability to perform full LLM inference. The conveyed semantics remain at the tokenization level and do not explicitly include the LLM reasoning and generation results. The corresponding receiver side decoding is given in Eq. (12).

b) Attention Token Encoder: To convey intermediate semantic representations after deep multimodal fusion and partial reasoning within the LLM, hidden states can be extracted from intermediate layers of the LLM as attention tokens. When the LLM is split across the transmitter and receiver, the attention tokens are formed by the hidden states from a set of intermediate layers, given by

$$\mathbf{Z} = \mathcal{G}_{\text{attn}}(\mathbf{P}_X, \mathbf{F}_T) := \{\mathbf{H}^{(l)} = \text{LLM}^{1:l}(\mathbf{P}_X, \mathbf{F}_T)\}, \tag{6}$$

where $\mathbf{H}^{(l)}$ denotes the hidden state output of the lth layer of the LLM, and $\mathcal{G}_{\text{attn}}(\cdot)$ denotes the attention token construction operator.

The resulting attention tokens already fuse multimodal information and contain partial reasoning traces. Compared with embedding tokens, they provide a stronger semantic representation capability, while avoiding the redundancy introduced by transmitting full reasoning outputs. This design also enables collaborative sharing of LLM computation between the transmitter and receiver. The corresponding receiver side decoding strictly follows Eq. (13).

c) Task Token Encoder: When the LLM is deployed at the transmitter, the task tokens are directly constructed from the output of the final layer, given by

$$\begin{aligned}\mathbf{Z} &= \mathcal{G}_{\text{task}}(\mathbf{P}_X, \mathbf{F}_T): \\ &= \big\{\mathbf{H}^{(L)} = \text{LLM}^{L}(\mathbf{P}_X, \mathbf{F}_T) = \{\tilde{t}, \mathbf{S}_X\}\big\},\end{aligned} \tag{7}$$

where $\mathbf{H}^{(L)}$ denotes the hidden state output of the Lth layer. Here, $\mathcal{G}_{\text{task}}(\cdot)$ denotes the output level token construction operator.

This scheme performs communication directly over task-level output tokens and thus provides the highest degree of semantic abstraction, making it well-suited for downstream task-driven communications. The receiver performs little or no LLM inference and only needs a semantic equivalent mapping, as specified in Eq. (14). Consequently, this mode places more stringent requirements on channel robustness and error control.

d) Unified formulation: The LLM token encoder can be represented by a unified mode selection function:

$$\mathbf{Z} = \mathcal{G}(\mathbf{P}_X, \mathbf{F}_T), \quad \mathcal{G} \in \{\mathcal{G}_{\text{embd}}, \mathcal{G}_{\text{attn}}, \mathcal{G}_{\text{task}}\}. \tag{8}$$

4) Channel Encoder: The channel encoder directly maps the token sequence into a sequence of complex-valued symbols suitable for transmission over the physical channel:

$$\mathbf{Y} = C_\beta(\mathbf{Z}), \tag{9}$$

where $C_\beta(\cdot)$ denotes a channel encoder parameterized by β.

C. Physical Channel

We adopt a baseband equivalent physical channel model. Taking the linear channel model commonly used in TokenCom as an example, the received signal is expressed as follows:

$$\hat{\mathbf{Y}} = \mathbf{H}\mathbf{Y} + \mathbf{n}, \tag{10}$$

where $\mathbf{H}$ denotes the channel gain matrix, and $\mathbf{Y}$ is the transmitted symbol vector satisfying the average power constraint $\mathbb{E}[\|\mathbf{Y}\|_2^2] \leq P$, where P denotes the maximum average transmit power. The noise vector follows $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$, where σ^2 is the noise variance. For an additive white Gaussian noise (AWGN) channel, $\mathbf{H}$ is deterministic and can be normalized to the identity matrix, whereas for Rayleigh or Rician fading channels, its entries are modeled as random fading coefficients.

D. Receiver

The receiver consists of a channel decoder $C_\gamma^{-1}(\cdot)$, an LLM token decoding module, an output projector $\Theta_{T\to X}$, and a modality detokenizer MD_X.

1) Channel Decoder: The received signal is first decoded to recover an estimated token sequence:

$$\hat{\mathbf{Z}} = C_\gamma^{-1}(\hat{\mathbf{Y}}), \tag{11}$$

where $C_\gamma^{-1}(\cdot)$ denotes the channel decoding operator parameterized by γ, which reconstructs the token sequence from noisy physical signals.

2) LLM Token Decoder: Corresponding to the transmitter side LLM token encoder, the receiver side LLM token decoder also takes three forms.

a) Embedding Token Decoder: In this mode, the channel decoded embedding tokens are used as the initial input to the LLM, which is then propagated to the final decoder layer to obtain the task token state:

$$\mathbf{H}^{(L)} = \Psi_{\mathrm{embd}}(\hat{\mathbf{Z}}) := \mathrm{LLM}^{1:L}(\hat{\mathbf{P}}_X, \hat{\mathbf{F}}_T), \tag{12}$$

where $\hat{\mathbf{P}}_X$ and $\hat{\mathbf{F}}_T$ denote the channel decoded multimodal embedding tokens and text embedding tokens, respectively. The output $\mathbf{H}^{(L)}$ is the task token produced by the final LLM layer and serves as the final result of token decoding.

b) Attention Token Decoder: In this mode, the attention tokens already lie in the internal semantic space of the LLM. Therefore, the receiver does not need to re-perform input embedding or alignment, but instead directly continues the forward propagation from this intermediate state through the subsequent layers:

$$\mathbf{H}^{(L)} = \Psi_{\mathrm{attn}}(\hat{\mathbf{Z}}) := \mathrm{LLM}^{l+1:L}(\hat{\mathbf{H}}^{(l)}), \tag{13}$$

where $\hat{\mathbf{H}}^{(l)}$ denotes the channel decoded attention tokens.

c) Task Token Decoder: Task tokens are generated from the output layer of the LLM and thus already correspond to high-level task semantics. Accordingly, the receiver does not apply any additional operations to $\hat{t}$ or $\hat{\mathbf{S}}_X$, and directly treats them as task token representations equivalent to those of the final LLM layer:

$$\mathbf{H}^{(L)} = \Psi_{\mathrm{task}}(\hat{\mathbf{Z}}) := \mathrm{SemMap}(\hat{t}, \hat{\mathbf{S}}_X), \tag{14}$$

where $\mathrm{SemMap}(\cdot)$ denotes an identity mapping. Here, $\hat{t}$ and $\hat{\mathbf{S}}_X$ are the recovered token sequence for the text generation task and the recovered token sequence for multimodal generation with respect to modality X, respectively.

d) Unified formulation: The above three decoding modes can be unified as follows:

$$\mathbf{H}^{(L)} = \Psi(\hat{\mathbf{Z}}), \quad \Psi \in \{\Psi_{\mathrm{embd}}, \Psi_{\mathrm{attn}}, \Psi_{\mathrm{task}}\}. \tag{15}$$

3) Multimodal Output Projection: For multimodal generation tasks, we first extract from $\mathbf{H}^{(L)}$ the task tokens $\hat{\mathbf{S}}_X$ that control the generation of the target modality. Then, an output projector maps these tokens to a conditional feature space that is compatible with the multimodal generator:

$$\hat{\mathbf{H}}_X = \Theta_{T\to X}(\hat{\mathbf{S}}_X), \tag{16}$$

where $\Theta_{T\to X}(\cdot)$ denotes the output projector, which aligns the textual semantic space with the conditional space of modality X.

The training objective of the output projector is given by

$$\min_{\Theta_{T\to X}} \mathcal{L}_{\mathrm{op}}(\hat{\mathbf{H}}_X, \tau_X(t)), \tag{17}$$

where $\tau_X(t)$ denotes the conditional feature representation of the textual instruction t produced by the internal text encoder of the target modality generator, such as the CLIP text encoder. The training objective is to minimize the Mean Squared Error (MSE) between the projector output and the text conditional embedding used by the target modality generator.

4) Detokenizer: For text generation tasks, the detokenizer produces the final textual output based on $\mathbf{H}^{(L)}$, which can be expressed as follows:

$$o = \mathrm{TD}(\mathbf{H}^{(L)}), \tag{18}$$

where TD denotes the detokenizer for the text generation task, which typically consists of a language model head and a subsequent decoding strategy, such as softmax-based decoding, greedy decoding, or beam search. This module projects the final hidden states $\mathbf{H}^{(L)}$ to the vocabulary space and generates the final text output o.

For multimodal generation tasks, a multimodal detokenizer generates the target modality content based on the conditional features:

$$\hat{I}_X = \mathrm{MD}_X(\hat{\mathbf{H}}_X), \tag{19}$$

where MD_X can employ diffusion generators such as Stable Diffusion [18], Zeroscope [19], and AudioLDM [20].

E. End-to-End Optimization Objective

1) Task-Level Semantic Understanding Loss: TokenCom enforces the consistency between the receiver output o and the ground truth label t using a cross-entropy loss, defined as follows:

$$\mathcal{L}_{\mathrm{CE}}(o, t), \tag{20}$$

where $\mathcal{L}_{\mathrm{CE}}(\cdot)$ denotes the standard token-level cross-entropy loss, which characterizes the end-to-end semantic understanding performance of the system.

2) Channel-Level Mutual Information Loss: In addition to the end-to-end task loss, TokenCom introduces a mutual information loss at the channel encoder and decoder to enhance token recoverability over noisy channels. The channel level loss is defined as follows:

$$\mathcal{L}_{\mathrm{MI}} = -I(\mathbf{Z}; \hat{\mathbf{Z}}), \tag{21}$$

where $I(\mathbf{Z}; \hat{\mathbf{Z}})$ denotes a lower bound of the mutual information estimated by a learnable mutual information estimator, such as MINE or a variational lower bound-based estimator. This term measures the mutual information between the transmitted semantic tokens and the recovered tokens. By minimizing $\mathcal{L}_{\mathrm{MI}}$, the system equivalently maximizes the information preservation capability of the token sequence over the physical channel.

3) Joint End-to-End Optimization Objective: By combining the above losses, the end-to-end joint optimization objective of TokenCom is formulated as follows:

$$\min_{\{\beta,\gamma,\Theta_{X\to T},\Theta_{T\to X},\Theta_{\mathrm{LLM}}\}} \lambda_1\mathcal{L}_{\mathrm{CE}} + \lambda_2\mathcal{L}_{\mathrm{ip}} + \lambda_3\mathcal{L}_{\mathrm{op}} + \lambda_4\mathcal{L}_{\mathrm{MI}}, \tag{22}$$

where β and γ are the trainable parameters of the channel encoder $C_\beta(\cdot)$ and the channel decoder $C_\gamma^{-1}(\cdot)$, respectively. $\Theta_{X\to T}$ denotes the parameters of the multimodal input projector that maps modality features into a unified textual semantic space, and $\Theta_{T\to X}$ denotes the parameters of the output projector that maps the multimodal task tokens generated by the LLM into the conditional space of the target modality generator. Θ_{LLM} denotes the subset of parameters in the LLM backbone that are involved in end to end training or finetuning, such as LoRA and adapter modules. The weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4 > 0$ control the tradeoff among task performance, alignment accuracy, and channel robustness.

IV. Agentic TokenCom Framework

Agentic TokenCom adopts a hierarchical chain of agents architecture, consisting of an upper-level reasoning chain and a lower-level execution chain. The reasoning chain plans, evaluates, and reflects on the workflow structure of TokenCom under a given task objective, channel condition, and resource constraint, enabling a self-evolving design of the communication procedure. Subject to the structural constraints produced by the reasoning chain, the execution chain dynamically selects and instantiates concrete models for token encoding, inference, and decoding, thereby converting an abstract workflow into an executable TokenCom system. The architecture of the Agentic TokenCom framework is illustrated in Fig. 2.

A. Overall Framework and Notations

We model the automated system construction problem of TokenCom as follows. Given multiple constraints, including the modality set, the task set, computation and latency budgets, and channel statistics, the goal is to automatically infer a workflow that contains core MLLM components and explicitly inserts channel coding and decoding modules. Furthermore, for each component in the workflow, an appropriate model instance with a suitable scale and functionality is selected. Under communication and resource constraints, the objective is to maximize end-to-end multimodal understanding performance and cross-modal generation performance.

1) System Requirement Specification: We define the system requirement specification as follows:

$$\mathcal{S} = \{\mathcal{X}, \mathcal{T}, \mathcal{B}, \mathcal{C}, \mathcal{R}\}, \tag{23}$$

where $\mathcal{X}$ denotes the modality configuration set (e.g., Image, Video and Audio), $\mathcal{T}$ denotes the task set, which includes text generation tasks, multimodal generation tasks, or their combinations, $\mathcal{B}$ denotes the resource budget constraints that characterize the available computing and storage capabilities at the terminal side or the base station side, $\mathcal{C}$ denotes the channel statistical information, including the SNR, available bandwidth, fading type, and target bit error rate, and $\mathcal{R}$ denotes the communication constraint set, such as the maximum token sequence length and the end to end latency limit.

2) Workflow and Components: A workflow $\mathcal{W}$ denotes a directed sequence of functional modules organized according to the semantic processing and communication procedure in a TokenCom system. In general, it can be expressed as follows:

$$\begin{aligned}\mathcal{W} = \big\{&\{\mathrm{TT}, \mathrm{MT}_X\},\ \Theta_{X\to T},\ \mathrm{LLM}_{\mathrm{Tx}},\ C_\beta,\ \mathrm{H},\\ &C_\gamma^{-1},\ \mathrm{LLM}_{\mathrm{Rx}},\ \Theta_{T\to X},\ \{\mathrm{TD}, \mathrm{MD}_X\}\big\},\end{aligned} \tag{24}$$

where $\{\mathrm{TT}, \mathrm{MT}_X\}$ denote the text tokenizer and the tokenizer for modality X, respectively. $\Theta_{X\to T}$ is the multimodal input projector that aligns modality features to the textual semantic space. $\mathrm{LLM}_{\mathrm{Tx}}$ and $\mathrm{LLM}_{\mathrm{Rx}}$ denote the LLM token encoding and decoding modules deployed at the transmitter and receiver, respectively, where their activation and placement are determined by the TokenCom execution mode. C_β and C_γ^{-1} denote the channel encoder and decoder, respectively. H denotes the physical channel. $\Theta_{T\to X}$ is the output projector that maps the multimodal tokens generated by the LLM to the conditional space of the target modality generator. Finally, $\{\mathrm{TD}, \mathrm{MD}_X\}$ denote the detokenizers for text and for the target modality X, respectively. Hence, the TokenCom pipeline defined in the system model is abstracted as a workflow in the Agentic TokenCom framework, which is generated by the reasoning chain and instantiated by the execution chain.

B. Reasoning Chain: Workflow Generation and Optimization

The reasoning chain consists of a planner, a memory module, an evaluator, and a reflector, where each agent is implemented by an LLM with strong capability for complex inference. With historical experience retrieved from the memory module, the reasoning chain generates candidate workflows that satisfy the system requirements and constraints, and evaluates their end-to-end performance. The reflector then diagnoses and adjusts the workflow structure based on the evaluation outcomes, and writes back the refined structure and its performance into the memory module, thereby driving continuous evolution of the TokenCom architecture and sustained performance improvement.

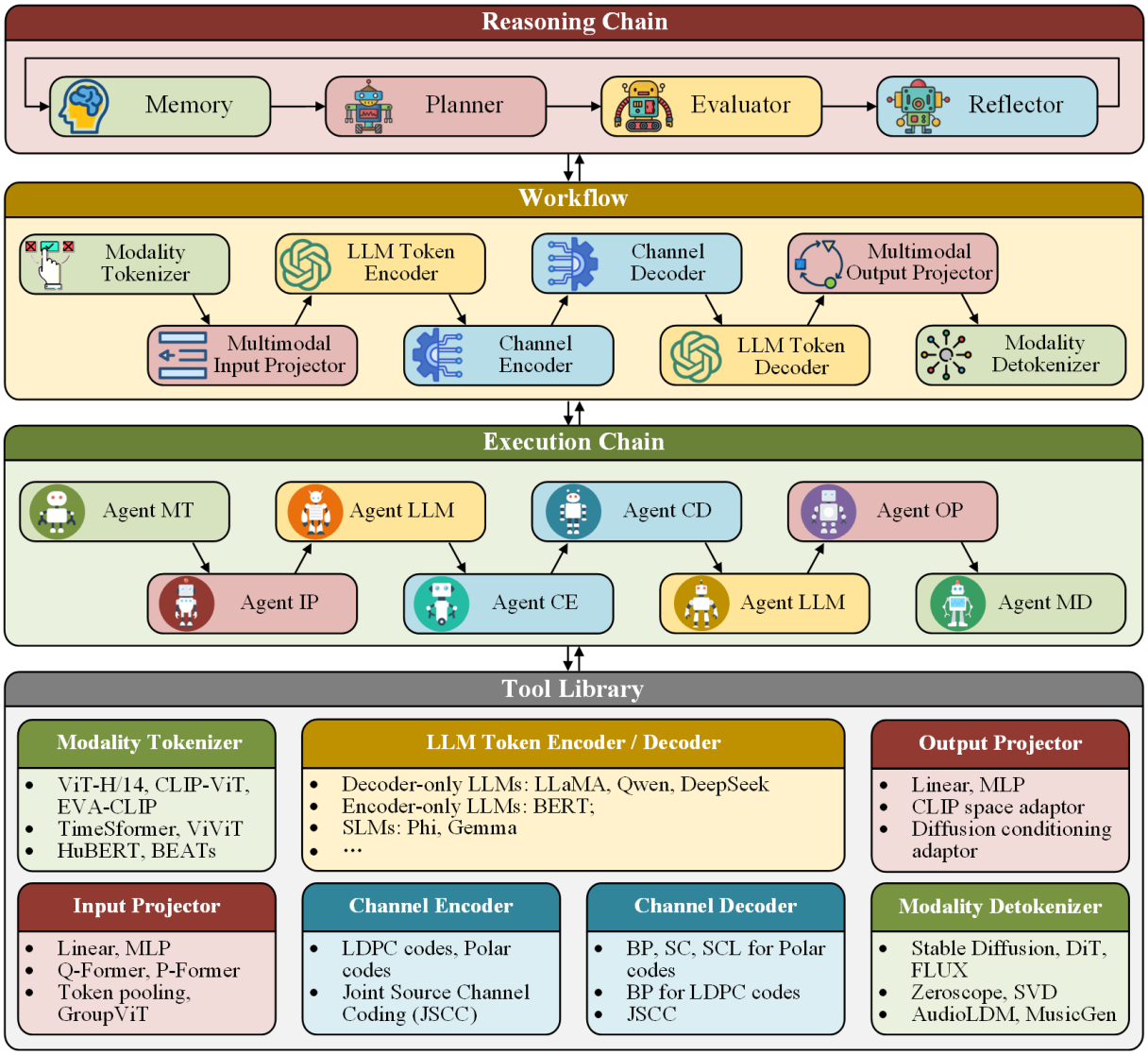

Fig. 2: Architecture of the Agentic TokenCom Framework.

1) Memory Modeling: The memory module $\mathcal{M}$ stores historical TokenCom system construction experiences. Its content includes the mapping from modality configurations, task requirements, channel conditions, and resource budgets to workflow structure selection, as well as performance metrics observed during deployment. This process can be abstracted as the following retrieval operation:

$$\mathbf{m} = \text{Retrieve}(\mathcal{M}, \mathcal{S}), \tag{25}$$

where $\mathbf{m}$ denotes the Top-K historical strategies with the highest similarity to the system requirement specification $\mathcal{S}$.

2) Planner: Under the joint constraints of the system requirement specification $\mathcal{S}$ and the memory prior $\mathbf{m}$, the planner generates the kth workflow as follows:

$$\mathcal{W}^{(k)} = \pi_{\text{plan}}(\mathcal{S}, \mathbf{m}, \xi_k), \tag{26}$$

where π_{plan} denotes the planning policy of the planner, and ξ_k captures the randomness or sampling strategy introduced in the kth generation. The generated workflow specifies not only the components included in the TokenCom system and their execution order, but also the appropriate token mode according to the task requirements, channel conditions, and computational capabilities of the transmitter and receiver. Specifically, the Embedding Token mode is selected when the LLM is primarily deployed at the receiver, the Attention Token mode is adopted for collaborative split inference, and the Task Token mode is used when task processing is mainly performed at the transmitter. The system then enters the second-level execution chain to instantiate the workflow.

3) Evaluator: Given an instantiated workflow $\mathcal{W}^{(k)}$, the evaluator computes an overall score $J(\mathcal{W}^{(k)}; \mathcal{S})$ to characterize its performance from a multiobjective optimization perspective. This score jointly accounts for task quality predictors, such as understanding accuracy, generation quality, PSNR, CLIPScore, and semantic consistency, as well as data compression efficiency, end-to-end latency, and penalty terms for violating system constraints. The resulting score provides a quantitative basis for subsequent reflection and structural

refinement.

4) Reflector: Based on the evaluation outcomes, the reflector performs error attribution for the current candidate workflow and outputs refinement instructions:

$$\Delta^{(k)} = \pi_{\mathrm{ref}}\big(\mathcal{S}, \mathcal{W}^{(k)}, J^{(k)}, \mathrm{diag}^{(k)}\big), \tag{27}$$

where π_{ref} denotes the reflection policy of the reflector, and $\mathrm{diag}^{(k)}$ represents diagnostic information, such as "An overly long token sequence causes the LLM inference latency to exceed the budget," "Channel errors reduce semantic consistency," and "The generation model is too large to be deployed at the terminal." The planner then updates its planning policy according to the feedback produced by the reflector:

$$\pi_{\mathrm{plan}} \leftarrow \mathrm{Update}\big(\pi_{\mathrm{plan}}, \Delta^{(k)}\big). \tag{28}$$

Meanwhile, the experience sample obtained in this round of reasoning and evaluation is written back to the memory:

$$\mathcal{M} \leftarrow \mathrm{FIFO}(\mathcal{M} \cup \{\mathcal{S}, \mathcal{W}^{(k)}, J^{(k)}, \mathrm{diag}^{(k)}\}). \tag{29}$$

In practice, the memory module is implemented as a bounded First-In, First-Out (FIFO) queue rather than an indefinitely growing repository. Once the queue is full, the earliest record is removed before a new one is stored.

Through multiple iterations of the above procedure, the reasoning chain gradually establishes a closed-loop optimization mechanism of generation, evaluation, reflection, and refinement, and ultimately outputs a high-quality feasible workflow structure under the given constraints:

$$\mathcal{W}^{\star} = \arg\max_{\mathcal{W}\in\mathbb{W}} J(\mathcal{W}; \mathcal{S}), \tag{30}$$

where $\mathbb{W}$ denotes the feasible workflow space, namely the set of workflows that satisfy the system requirement specification and various structural, resource, and constraints.

C. Execution Chain: Workflow Instantiation

The execution chain consists of a set of component agents. In contrast to the structural decisions made by the reasoning chain, the execution chain addresses implementation decisions. Given a workflow generated by the reasoning chain, the execution chain instantiates each component in sequence by performing model selection, parameter configuration, interface alignment, and deployment mapping, thereby producing an executable TokenCom system instance.

1) Model Selection by Component Agents: For the components in a workflow $\mathcal{W}$, we define a candidate implementation set as $\mathcal{Z}(\mathcal{W})$. Given the system requirement specification $\mathcal{S}$ and the workflow structure $\mathcal{W}$, the component agents select the best component implementation as follows:

$$\mathbf{z}^{\star} = \arg\max_{\mathbf{z}\in\mathcal{Z}(\mathcal{W})} \tilde{J}(\mathbf{z}; \mathcal{S}), \tag{31}$$

where $\tilde{J}$ is a utility function that follows the multiobjective scoring mechanism of the evaluator in the reasoning chain. It quantifies the contribution of each candidate model when used as a component, including its quality gain and the risk of constraint violations.

To ensure that the system structure is interpretable, verifiable, and reproducible, the execution chain instantiates the workflow $\mathcal{W}$ in the following order:

$$\begin{aligned}\mathrm{MT}_X \to \Theta_{X\to T} \to \text{LLM Backbone} \to C_\beta \to \mathrm{H} \\ \to C_\gamma^{-1} \to \Theta_{T\to X} \to \mathrm{MD}_X\end{aligned} \tag{32}$$

where once the LLM backbone model is selected, its internal structure is fixed, including four parts, namely the TT, the LLM token encoder, the LLM token decoder, and the TD. Therefore, selecting the LLM backbone automatically determines the specific structures of these four parts.

2) Candidate Model Libraries and Constraints: Each component agent selects the most suitable model implementation from its candidate library according to the workflow requirements and system resource constraints. The constraints of each module, such as the token budget, model input output consistency, and alignment quality, directly affect the scale and architecture of the selected model.

a) Agent MT: Agent MT instantiates the modality tokenizer. Its candidate model library $\mathcal{Z}_1$ covers representative multimodal encoding architectures:

- Image: ViT-H/14, CLIP-ViT, EVA-CLIP, ConvNeXt;
- Video: TimeSformer, ViViT, InternVideo-Enc;
- Audio: HuBERT, BEATs, Whisper-Enc, wav2vec2.

Key constraint: The token granularity and sequence length must satisfy the token budget specified by the workflow to avoid semantic redundancy and uncontrolled downstream inference cost.

b) Agent IP: Agent IP instantiates the input projector. Its candidate model library $\mathcal{Z}_2$ includes widely used projector architectures:

- Linear, MLP, Cross Attention Projector;
- Q-Former, P-Former, MQ-Former;
- Token pooling, GroupViT.

Key constraints: On the one hand, the length of the produced soft prompts must be compatible with the context window of the selected LLM. On the other hand, the cross-modal alignment quality must meet the semantic consistency requirement specified by the evaluator.

c) Agent LLM: Agent LLM instantiates the text tokenizer, token encoder, token decoder, and text detokenizer. Its candidate model library $\mathcal{Z}_3$ covers mainstream LLMs and Small Language Models (SLMs):

- Decoder only LLMs: LLaMA, Qwen, DeepSeek;
- Encoder only LLMs: BERT;
- SLMs: Phi, Gemma.

Key constraints: The structure must ensure exact correspondence between the transmitter side token encoder and the receiver side token decoder. The semantic representation capability must cover the requirements of the task set $\mathcal{T}$. The model scales and inference latency of the token encoder and token decoder must satisfy the computation and energy budgets at both the transmitter and receiver.

d) Agent CE: Agent CE instantiates the channel encoder. Its candidate model library $\mathcal{Z}_4$ includes mainstream channel coding schemes:

- LDPC codes, Polar codes;

- Joint Source Channel Coding (JSCC).

Key constraint: The design must satisfy channel bandwidth, SNR, target bit error rate, and the specified code rate and spectral efficiency constraints.

e) Agent CD: Agent CD instantiates the channel decoder. Its candidate model library $\mathcal{Z}_5$ includes mainstream channel decoding methods:

- BP, SC, SCL for Polar codes;
- BP for LDPC codes;
- JSCC.

Key constraint: The decoded token structure must be consistent with the representation produced by the transmitter side token encoder, and must align with the input mode of the subsequent LLM token decoder so that it can be fed as embedding level, attention level, or task level tokens.

f) Agent OP: Agent OP instantiates the output projector. Its candidate model library $\mathcal{Z}_6$ includes representative adaptor designs:

- Linear, MLP, Transformer adaptor;
- CLIP space adaptor;
- Diffusion conditioning adaptor.

Key constraint: This module must ensure that the output conditional space matches the dimensionality required by the selected generator, and it must satisfy the generation consistency metrics specified by the evaluator.

g) Agent MD: Agent MD instantiates the modality detokenizer. Its candidate model library $\mathcal{Z}_7$ covers mainstream diffusion based generators:

- Text to image: Stable Diffusion, DiT, FLUX;
- Text to video: Zeroscope, SVD, Video Diffusion;
- Text to audio: AudioLDM, MusicGen.

Key constraint: Subject to the computation and latency budgets at the deployment side, this module should achieve generation quality that meets the task requirements, measured by metrics such as PSNR, FID, CLIPScore, and subjective consistency.

For new modalities such as 3D point clouds or tactile signals, dynamic integration is enabled by registering the corresponding tokenizer, projector, and detokenizer, together with standardized interface and resource descriptions. After compatibility verification, these components are added to the candidate library for online selection and instantiation according to task, channel, and resource conditions.

D. Unified Optimization Objective

The reasoning and execution chains form a memory-assisted alternating optimization process. Given the system requirement specification $\mathcal{S}$, the Planner retrieves relevant historical experience from the Memory and generates a candidate workflow $\mathcal{W}$. The execution chain then selects specific component implementations $\mathbf{z} \in \mathcal{Z}(\mathcal{W})$ and instantiates the workflow into an executable TokenCom system. The resulting feasibility, task performance, inference latency, and resource consumption are evaluated by the Evaluator and Reflector to guide subsequent workflow updates. Meanwhile, the workflow, performance results, and diagnostic information are stored in the Memory for optimization. This process can be formulated as follows:

$$(\mathcal{W}^\star, \mathbf{z}^\star) = \underset{\substack{\mathcal{W}\in\mathbb{W} \\ \mathbf{z}\in\mathcal{Z}(\mathcal{W})}}{\arg\max}\ \widetilde{J}(\mathbf{z}, \mathcal{W}; \mathcal{S}), \tag{33}$$

where $\mathbb{W}$ denotes the feasible workflow space, while $\mathcal{Z}(\mathcal{W})$ represents the set of component implementations compatible with workflow $\mathcal{W}$. The objective therefore jointly optimizes the workflow structure and its component-level instantiation under the requirements specified by $\mathcal{S}$.

V. Simulation Results

A. Simulation Settings

To comprehensively evaluate the performance of the proposed Agentic TokenCom framework, we conducted systematic simulation studies across multimodal and multi-task scenarios under heterogeneous resource constraints and dynamic channel conditions. All experiments were carried out on a high-performance server equipped with an Intel Xeon Gold CPU (2.6 GHz) and eight NVIDIA A800 GPUs (80 GB VRAM each). We implemented all learning components in PyTorch and adopted DeepSpeed ZeRO-2 for training.

In the upper reasoning chain, the Planner, Evaluator, and Reflector agents were driven by DeepSeek R1 to support the generation and optimization of the workflow configuration. In the lower execution chain, all component agents were implemented using GPT-4o.

We built a candidate model library for TokenCom, with modality tokenizers including ViT-H/14, CLIP-ViT, and ConvNeXt for visual encoding, and HuBERT and Wav2Vec2 for audio encoding. The input projector offered alignment modules with varying complexity, such as Linear, MLP, GroupViT, and Q-Former. The core LLM backbone could be dynamically chosen from LLaMA3-8B, Vicuna-7B, or the lightweight Gemma-2B. Output projectors included Linear, MLP, and Transformer modules. Modality generators on the output side included generative models like Stable Diffusion, Zeroscope, and AudioLDM. The channel coding and decoding module supported both conventional LDPC and Polar codes, as well as deep learning-based JSCC neural codecs. The component agent, guided by structural constraints from the reasoning chain, automatically selected and instantiated the optimal configuration from the candidate model library.

To validate the effectiveness of the proposed framework, we evaluated visual question answering (VQA) tasks using VQAv2 [21] and CLEVR [22], and used COCO-Caption [23], MSR-VTT [24], and AudioCaps [25] to assess image captioning, video captioning, and audio captioning, respectively. For content generation, we adopted the same three datasets to evaluate text-to-image (T2I), text-to-video (T2V), and text-to-audio (T2A) generation quality.

Finally, we employed a multi-dimensional evaluation suite. For understanding tasks, VQA performance was measured by Accuracy, which quantifies the consistency between predicted answers and ground-truth annotations. For image captioning and video captioning, we reported METEOR [26] and BLEU-4 (B@4) [27], which characterize semantic matching and

fluency as well as precise 4-gram overlap with reference descriptions, respectively. For audio captioning, we used SPIDEr [28] and CIDEr [29], where CIDEr measures the consensus between generated captions and human annotations, and SPIDEr further integrates semantic similarity, such as SPICE, on top of CIDEr to more comprehensively reflect semantic consistency and language quality. For generation tasks, we used Fréchet Inception Distance (FID) for T2I to quantify the distance between the distributions of generated and real images, where lower values indicate better quality. For T2A, we adopted Fréchet Audio Distance (FAD) to measure the distribution discrepancy between generated and real audio in a deep feature space, where lower values are better. For T2V, we used CLIP similarity (CLIPSIM) to assess semantic alignment between generated videos and text prompts in the CLIP embedding space, where higher values indicate better alignment.

B. Multimodal Performance Evaluation

This experiment aims to validate the multimodal understanding capability of the workflow constructed by Agentic TokenCom. We benchmark Agentic TokenCom against representative baseline methods on image, video, and audio captioning tasks. The evaluation metrics include CIDEr, METEOR, SPIDEr, and BLEU-4 (B@4). The compared methods include static TokenCom systems built upon representative large models, including GIT [30], Oscar [31], BART [32], and CoDi [33]. The communication channel parameters follow the settings in [34].

In this experiment, the workflow automatically generated and instantiated by Agentic TokenCom is given by

$$\mathcal{W}_{\text{cap}} = \big\{\{\text{TT}, \text{MT}_{\text{X}}\}, \Theta_{\text{X}\to T}, \text{LLM}_{1:l}, C_\beta, \text{H}, C_\gamma^{-1}, \text{LLM}_{l+1:L}, \text{TD}\big\}, \tag{34}$$

where the multimodal tokenizer adopts ImageBind (ViT-H/14), which represents multimodal inputs in a unified embedding space. The input projector uses Grouping to map multimodal tokens into text tokens that are compatible with the LLM. The LLM backbone is instantiated with Vicuna-7B.

As shown in Table II, for multimodal understanding tasks, the workflow instance generated by Agentic TokenCom consistently outperforms static end-to-end models based on other large models, such as GIT, Oscar, BART, and CoDi, across key metrics including METEOR, BLEU-4 (B@4), CIDEr, and SPIDEr. These results demonstrate that Agentic TokenCom achieves stronger cross-modal semantic representation and understanding compared to other models. The performance gains primarily result from the agentic framework, which adaptively selects a high-capacity multimodal tokenizer, ViT-H/14, along with a more powerful LLM, Vicuna-7B, under structural constraints. Meanwhile, the alignment module effectively maps multimodal features into the LLM's text space, better utilizing the LLM for language understanding and generation, and producing more accurate and semantically faithful captions.

C. Multi-Task Performance Evaluation

This experiment aims to validate the multimodal generation capability of the workflow constructed by Agentic TokenCom under a unified text input. We evaluate Agentic TokenCom on three generative tasks, namely T2I, T2A, and T2V, and compare it with representative baseline methods. The evaluation metrics are configured as follows: FID for T2I, FAD for T2A, and CLIPSIM for T2V. The compared methods include static TokenCom systems built upon representative large models, including UIO-2 [35], CogVideo [36], and CoDi [33]. The communication channel parameters follow the settings in [34]. In this experiment, Agentic TokenCom automatically generates and instantiates the following workflows:

$$\mathcal{W}_{\text{img}} = \big\{\text{TT}, \text{LLM}_{1:l}, C_\beta, \text{H}, C_\gamma^{-1}, \text{LLM}_{l+1:L}, \Theta_{T\to\text{img}}, \text{MD}_{\text{img}}\big\}, \tag{35}$$

$$\mathcal{W}_{\text{aud}} = \big\{\text{TT}, \text{LLM}_{1:l}, C_\beta, \text{H}, C_\gamma^{-1}, \text{LLM}_{l+1:L}, \Theta_{T\to\text{aud}}, \text{MD}_{\text{aud}}\big\}, \tag{36}$$

$$\mathcal{W}_{\text{vid}} = \big\{\text{TT}, \text{LLM}_{1:l}, C_\beta, \text{H}, C_\gamma^{-1}, \text{LLM}_{l+1:L}, \Theta_{T\to\text{vid}}, \text{MD}_{\text{vid}}\big\}, \tag{37}$$

where Vicuna-7B is selected as the LLM backbone to parse text instructions and produce modality-aware signal tokens. The output projector adopts a four-layer Transformer to map the signal tokens into conditional embeddings that can be consumed by diffusion models. On the modality generation side, Stable Diffusion is used for image generation, AudioLDM is used for audio generation, and Zeroscope is used for video generation, thereby enabling unified execution for T2I, T2A, and T2V tasks.

The results are summarized in Table III. The workflow instances generated by Agentic TokenCom achieve the lowest FID on T2I, the lowest FAD on T2A, and the highest CLIPSIM on T2V. These results indicate that the proposed framework can adaptively select high-performing generators in the execution chain according to task requirements. Moreover, it benefits from the semantic enhancement capabilities of the LLM, such as instruction understanding and prompt refinement, which further improve generation quality and lead to clear advantages over the fixed configurations of single static models. In addition, introducing the LLM as the core semantic understanding module in token communication helps maintain semantic consistency between text instructions and generated content, suppresses semantic drift during transmission, and

TABLE II: Multimodal Understanding Task: Static TokenCom with Large Models vs. Agentic TokenCom.

Task (Modality)	Model / TokenCom	Metric1	Metric2
Video Captioning	GIT	METEOR: 31.6	B@4: 52.9
	CoDi	METEOR: 31.4	B@4: 49.9
	Ours	**METEOR: 38.4**	**B@4: 56.1**
Image Captioning	Oscar	METEOR: 29.0	B@4: 35.45
	CoDi	METEOR: 29.6	B@4: 38.6
	Ours	**METEOR: 32.6**	**B@4: 42.9**
Audio Captioning	BART	SPIDEr: 0.442	CIDEr: 0.724
	CoDi	SPIDEr: 0.457	CIDEr: 0.753
	Ours	**SPIDEr: 0.508**	**CIDEr: 0.768**

TABLE III: Multimodal Generation Task: Static TokenCom with MLLMs vs. Agentic TokenCom.

Task (Modality)	Model / TokenCom	Metric1	Metric2
Text-to Image	UIO-2	FID: 13.86	–
	CoDi	FID: 11.85	–
	Ours	**FID: 11.46**	–
Text-to Audio	UIO-2	FAD: 2.72	–
	CoDi	FAD: 1.84	–
	Ours	**FAD: 1.78**	–
Text-to Video	CogVideo	-	CLIPSIM: 25.14
	CoDi	–	CLIPSIM: 27.73
	Ours	-	**CLIPSIM: 30.18**

consequently mitigates the semantic loss commonly observed in conventional semantic communication paradigms.

D. Performance Comparison of Different Token Encoders

This experiment aims to systematically analyze the tradeoff between computational cost allocation and noise robustness among three semantic-granularity token encoding modes in the TokenCom framework, namely Embedding Token, Attention Token, and Task Token. We evaluate the three modes on the VQAv2 dataset using Accuracy as the performance metric. For a fair comparison, all three modes share the same workflow structure and identical component configurations, and only the token encoding mode is varied. The three workflows are given as follows:

$$\mathcal{W}_{\text{embd}} = \big\{\{\text{TT}, \text{MT}_{\text{img}}\}, \Theta_{\text{img}\to T}, \emptyset, C_\beta, \text{H}, C_\gamma^{-1}, \text{LLM}_{1:L}, \text{TD}\big\}, \tag{38}$$

$$\mathcal{W}_{\text{attn}} = \big\{\{\text{TT}, \text{MT}_{\text{img}}\}, \Theta_{\text{img}\to T}, \text{LLM}_{1:l}, C_\beta, \text{H}, C_\gamma^{-1}, \text{LLM}_{l+1:L}, \text{TD}\big\}, \tag{39}$$

$$\mathcal{W}_{\text{task}} = \big\{\{\text{TT}, \text{MT}_{\text{img}}\}, \Theta_{\text{img}\to T}, \text{LLM}_{1:L}, C_\beta, \text{H}, C_\gamma^{-1}, \emptyset, \text{TD}\big\}. \tag{40}$$

In all three modes, the modality tokenizer adopts ViT-H/14 (1.2B), the input projector uses Grouping, and the LLM backbone is instantiated with Vicuna-7B. We further profile and compare the semantic levels involved at both communication ends, the parameter differences, and the VQA accuracy under different SNR settings.

As reported in Table IV, the three encoding modes exhibit a characteristic tradeoff among semantic abstraction depth, end-side computational allocation, and noise robustness. Under a high-SNR condition (SNR = 9 dB), the VQA accuracies of the three modes are close, indicating that when the channel is reliable, higher-level semantic compression does not noticeably degrade task performance. In contrast, under a low-SNR condition (SNR = 0 dB), the performance gap becomes substantial. The Embedding Token mode yields the highest accuracy by shifting the majority of computation to the receiver, suggesting that transmitting shallower representations is more robust under severe noise. This can be attributed to the fact that even when the received embedding tokens are partially corrupted, the receiver-side LLM can leverage semantic priors to perform partial recovery and error correction. By comparison, the Task Token mode completes deeper task-oriented encoding at the transmitter, but its accuracy drops to 51.78% at low SNR, implying that task tokens are more sensitive to channel perturbations. Once critical semantics are corrupted, they are difficult to reconstruct at the receiver.

TABLE IV: Performance Comparison under Different Token Encoding Modes.

Token Mode	Semantics Level	Tx Param	Rx Param	ACC (9 dB)	ACC (0 dB)
Embedding Token	Shallow (Feature)	1.2B	7B	**60.83%**	**54.67%**
Attention Token	Middle (Context)	4.7B	3.5B	60.54%	53.23%
Task Token	Deep (Symbolic)	8.2B	31M	59.92%	51.78%

E. Performance Evaluation of the Chain of Agents

This experiment evaluates the performance of the Chain of Agents in a BS-to-UAV communication scenario. In this setting, the BS assigns a VQA task to the UAV. The link initially operates at a relatively high SNR, after which the UAV enters an interference region and the wireless channel quality degrades significantly. The goal is to verify whether Agentic TokenCom can achieve an adaptive closed-loop optimization process that is plannable, executable, and evolvable through multi-agent collaboration, under the joint constraints of limited onboard computation at the UAV and dynamically fluctuating wireless channels. We use VQA Accuracy as the primary metric to reflect task completion quality under resource constraints and dynamic channel conditions.

a) Phase 1: Initial Planning. In the initial phase, since the downlink is dominated by the BS and the UAV has constrained computational resources (4 GB VRAM), the reasoning chain tends to place the major token encoding and inference workload on the BS side. Under this constraint, the Planner selects the Task Token encoding mode, where the BS performs the full token encoding and inference with the LLM, while the UAV only receives and processes the Task Token. The corresponding workflow is expressed as follows:

$$\mathcal{W}^1_{\text{B2U}} = \{\{\text{TT}, \text{MT}_{\text{img}}\}, \Theta_{\text{img}\to T}, \text{LLM}_{1:L}, C_\beta, \text{H}, C_\gamma^{-1}, \emptyset, \text{TD}\}, \tag{41}$$

where the execution chain is instantiated as follows. On the BS side, ViT-H/14 is used for input projection, and Vicuna-7B serves as the token encoder to perform full inference. On the UAV side, no LLM is deployed; the UAV only executes the text detokenizer to satisfy the onboard resource constraint.

b) Phase 2: Environmental Change and Replanning. When the UAV enters the interference region and the SNR decreases, the Evaluator detects a noticeable degradation in VQA accuracy at the UAV. The Reflector attributes this degradation to the fact that both token encoding and inference are fully

TABLE V: Performance Evaluation of the Chain of Agents.

Stage Method	Token Mode	BS Model	UAV Model	ACC
Phase 1 (Initial)	**Task Token**	Vicuna-7B (Full)	None	**55.65%**
Phase 2 (Replanning)	**Embedding Token**	None	Vicuna-7B	Fail (Overloaded)
Phase 3 (Optimized)	**Embedding Token**	None	Gemma-2B	50.48%

centralized at the BS, and the transmitted Task Token is highly sensitive to channel noise, providing limited semantic-level recovery and error correction capability. To prevent task failure, the inference and decoding functionality should be migrated to the UAV to enhance its ability to semantically reconstruct corrupted tokens. Based on this diagnosis, the Planner revises the original workflow and replans it as follows:

$$\mathcal{W}^2_{\mathrm{B2U}} = \{\{\mathrm{TT}, \mathrm{MT}_{\mathrm{img}}\}, \Theta_{\mathrm{img}\to T}, \emptyset, C_\beta, \mathrm{H}, C_\gamma^{-1}, \mathrm{LLM}_{1:L}, \mathrm{TD}\}. \tag{42}$$

c) Phase 3: Execution Chain Optimization. When the UAV is required to undertake the inference workload, its 4 GB VRAM makes it infeasible to deploy Vicuna-7B. Therefore, the execution chain triggers a model selection and downgrading mechanism and selects a deployable SLM, namely Gemma-2B, from the candidate model library. Accordingly, the execution chain is re-instantiated as follows. The BS uses CLIP-ViT as the input projector, while the UAV deploys a quantized version of Gemma-2B as the token decoder, thereby satisfying the onboard resource constraint.

As shown in Table V, Agentic TokenCom can effectively avoid task failure under performance degradation and resource constraints by dynamically adjusting the workflow configuration. Compared with static configurations, the proposed framework demonstrates two key advantages in dynamic scenarios. First, when channel conditions fluctuate or task performance degrades, the system can adaptively switch the token encoding mode to maintain semantic consistency and task executability across the communication endpoints. Second, when high-complexity models are infeasible on the UAV, the framework can introduce lightweight model instances through online replanning, which substantially reduces resource consumption while recovering and stabilizing task performance.

F. Performance Comparison with Semantic Communications

This experiment evaluates the VQA task of Agentic TokenCom in comparison with multiple semantic communication systems, including U-DeepSC [37] and DeepSC-VQA [38]. We conduct evaluations on the CLEVR [22] dataset, and use VQA Accuracy as the performance metric. The workflow form generated by Agentic TokenCom is consistent with Eq. (42). To ensure a fair comparison, considering that conventional semantic communication systems typically rely on relatively small-scale models, we configure the workflow instance with a low-resource setup. Specifically, the modality tokenizer uses CLIP-ViT, the input projector adopts an MLP, and the LLM backbone is instantiated with Gemma-2B. Under Rayleigh fading, the results are shown in Fig. 3.

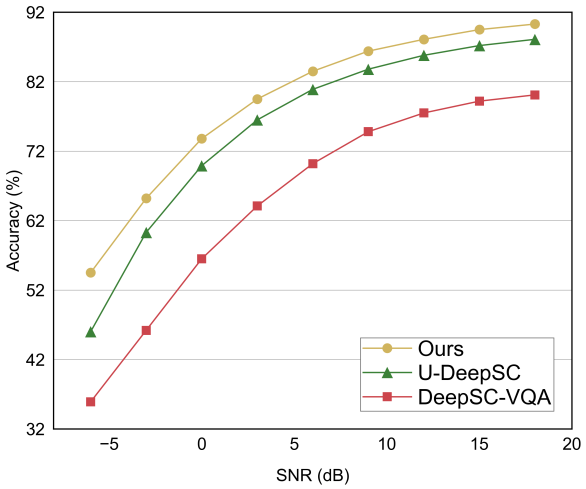

Fig. 3: Token vs. Semantic Communication on CLEVR Dataset.

Overall, as the SNR increases, all methods exhibit a monotonic improvement in VQA accuracy, indicating that better channel quality effectively mitigates distortion in semantic information transmission and directly benefits VQA performance. On both datasets, Agentic TokenCom consistently outperforms the competing methods across the entire SNR range, demonstrating that the generated workflow provides stronger semantic understanding capability and higher robustness to channel noise for VQA.

VI. Conclusion

We propose an Agentic TokenCom framework based on a Chain of Agents, transforming static TokenCom structure design and model instantiation into autonomous optimization through reasoning and execution chains. The reasoning chain integrates Planner, Evaluator, and Reflector agents to iteratively refine workflows based on performance feedback, while the execution chain dynamically selects and deploys task- and modality-aware models under workflow constraints. A resource-aware orchestration mechanism further balances token abstraction and resource consumption. Experiments demonstrate stable executability and strong performance under joint channel, task, and resource constraints.

However, the substantial computational overhead, inference latency, and deployment cost of large models may constrain the real-time applicability of the proposed framework in URLLC and resource-limited 6G scenarios. Meanwhile, tokens may contain rich contextual semantics and potentially sensitive information, making their transmission over open wireless channels vulnerable to eavesdropping, adversarial reconstruction, and tampering. Future work will therefore investigate domain-specific small models, knowledge distillation, model

quantization, and edge–cloud collaborative deployment, while establishing explicit security mechanisms based on token encryption, representation sanitization, differential privacy, and adversarial training.

REFERENCES

[1] W. Yang, H. Du, Z. Q. Liew, W. Y. B. Lim, Z. Xiong, D. Niyato, X. Chi, X. Shen, and C. Miao, "Semantic communications for future internet: Fundamentals, applications, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 213–250, 2023.

[2] F. Jiang, L. Dong, L. Mao, K. Wang, X. Wang, and A. Jamalipour, "Slm, llm, or agentic ai? toward intelligent uav-enabled wpt systems in low-altitude economy networks," *IEEE Journal on Selected Areas in Communications*, vol. 44, pp. 5267–5281, 2026.

[3] Z. Lu, R. Li, K. Lu, X. Chen, E. Hossain, Z. Zhao, and H. Zhang, "Semantics-empowered communications: A tutorial-cum-survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 41–79, 2024.

[4] F. Jiang, C. Pan, L. Dong, K. Wang, M. Debbah, D. Niyato, and Z. Han, "A comprehensive survey of large ai models for future communications: Foundations, applications, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 4731–4764, 2026.

[5] E. Zeydan, L. Blanco, C. J. Vaca-Rubio, M. Caus, and K. Dev, "Semantic non-terrestrial communications with open ran-enabled 6g," *IEEE Communications Standards Magazine*, vol. 9, no. 4, pp. 72–78, 2025.

[6] F. Jiang, C. Pan, K. Wang, P. Michiardi, O. A. Dobre, and M. Debbah, "From large ai models to agentic ai: A tutorial on future intelligent communications," *IEEE Journal on Selected Areas in Communications*, vol. 44, pp. 3507–3540, 2026.

[7] H. Wei, W. Ni, W. Wang, W. Xu, D. Niyato, and P. Zhang, "Token communication in the era of large models: An information bottleneck-based approach," *IEEE Wireless Communications Letters*, vol. 15, pp. 186–190, 2026.

[8] F. Jiang, S. Tu, L. Dong, K. Wang, K. Yang, R. Liu, C. Pan, and J. Wang, "Flashsam: Lightweight vision model for multi-uav token communication in low-altitude wireless networks," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–14, 2026.

[9] Y. Zhao, Y. Yue, S. Hou, B. Cheng, and Y. Huang, "Lamosc: Large language model-driven semantic communication system for visual transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 6, pp. 2005–2018, 2024.

[10] S. Guo, Y. Wang, J. Ye, A. Zhang, P. Zhang, and K. Xu, "Semantic importance-aware communications with semantic correction using large language models," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 3, pp. 232–245, 2025.

[11] D. Cao, J. Wu, and A. K. Bashir, "Multimodal large language models driven privacy-preserving wireless semantic communication in 6g," in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2024, pp. 171–176.

[12] B. Güler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 787–802, 2018.

[13] L. Zhou, X. Deng, Z. Wang, X. Zhang, Y. Dong, X. Hu, Z. Ning, and J. Wei, "Semantic information extraction and multi-agent communication optimization based on generative pre-trained transformer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 2, pp. 725–737, 2025.

[14] W. Yang, Z. Xiong, Y. Yuan, W. Jiang, T. Q. S. Quek, and M. Debbah, "Agent-driven generative semantic communication with cross-modality and prediction," *IEEE Transactions on Wireless Communications*, vol. 24, no. 3, pp. 2233–2248, 2025.

[15] L. Qiao, M. B. Mashhadi, Z. Gao, R. Tafazolli, M. Bennis, and D. Niyato, "Token communications: A large model-driven framework for cross-modal context-aware semantic communications," *IEEE Wireless Communications*, vol. 32, no. 5, pp. 80–88, 2025.

[16] F. Jiang, S. Tu, L. Dong, X. Li, K. Wang, C. Pan, Z. Han, and J. Wang, "Tokencom: Vision-language model for multimodal and multitask token communications," *arXiv preprint arXiv:2603.00482*, 2026.

[17] S. Lee, J. Park, J. Choi, and H. Park, "Semantic packet aggregation for token communication via genetic beam search," in *2025 IEEE 26th International Workshop on Signal Processing and Artificial Intelligence for Wireless Communications (SPAWC)*, 2025, pp. 1–5.

[18] E. Cho, J. Bang, and M. Rhu, "Characterization and analysis of text-to-image diffusion models," *IEEE Computer Architecture Letters*, vol. 23, no. 2, pp. 227–230, 2024.

[19] Z. Luo, D. Chen, Y. Zhang, Y. Huang, L. Wang, Y. Shen, D. Zhao, J. Zhou, and T. Tan, "Videofusion: Decomposed diffusion models for high-quality video generation," *arXiv preprint arXiv:2303.08320*, 2023.

[20] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 21 450–21 474.

[21] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6904–6913.

[22] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2901–2910.

[23] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[24] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5288–5296.

[25] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.

[26] D. Rothenpieler and S. Amiriparian, "Meteor guided divergence for video captioning," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 1–7.

[27] D. A. Hafeth, S. Kollias, and M. Ghafoor, "Semantic representations with attention networks for boosting image captioning," *IEEE Access*, vol. 11, pp. 40 230–40 239, 2023.

[28] X. Xu, Z. Xie, M. Wu, and K. Yu, "Beyond the status quo: A contemporary survey of advances and challenges in audio captioning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 95–112, 2024.

[29] F. Gontier, R. Serizel, and C. Cerisara, "Spice+: Evaluation of automatic audio captioning systems with pre-trained language models," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[30] J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang, "Git: A generative image-to-text transformer for vision and language," *arXiv preprint arXiv:2205.14100*, 2022.

[31] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *European conference on computer vision*. Springer, 2020, pp. 121–137.

[32] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 7871–7880.

[33] Z. Tang, Z. Yang, C. Zhu, M. Zeng, and M. Bansal, "Any-to-any generation via composable diffusion," *Advances in Neural Information Processing Systems*, vol. 36, pp. 16 083–16 099, 2023.

[34] F. Jiang, S. Tu, L. Dong, C. Pan, J. Wang, and X. You, "Large generative model-assisted talking-face semantic communication system," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 12, pp. 4152–4165, 2025.

[35] J. Lu, C. Clark, S. Lee, Z. Zhang, S. Khosla, R. Marten, D. Hoiem, and A. Kembhavi, "Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 439–26 455.

[36] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang, "Cogvideo: Large-scale pretraining for text-to-video generation via transformers," *arXiv preprint arXiv:2205.15868*, 2022.

[37] G. Zhang, Q. Hu, Z. Qin, Y. Cai, G. Yu, and X. Tao, "A unified multi-task semantic communication system for multimodal data," *IEEE Transactions on Communications*, vol. 72, no. 7, pp. 4101–4116, 2024.

[38] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2584–2597, 2022.

Feibo Jiang received his B.S. and M.S. degrees in School of Physics and Electronics from Hunan Normal University, China, in 2004 and 2007, respectively. He received his Ph.D. degree in School of Geosciences and Info-physics from the Central South University, China, in 2014. He is currently an associate professor at the Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, China. His research interests include machine learning, Agentic AI, semantic communication, and mobile edge computing.

Lei Mao is currently pursuing the M.S. degree with the School of Computer Science and Technology, Hunan Normal University. His main research interests include semantic communication and Agentic AI-enabled wireless communications.

Li Dong received the B.S. and M.S. degrees in School of Physics and Electronics from Hunan Normal University, China, in 2004 and 2007, respectively. She received her Ph.D. degree in School of Geosciences and Info-physics from the Central South University, China, in 2018. She is currently a professor at Hunan University of Technology and Business, China. Her research interests include machine learning, Internet of Things, and mobile edge computing.

Kezhi Wang received the Ph.D. degree from the University of Warwick, U.K., funded by the Chancellor's International Scholarship. He is currently a Professor with the Department of Computer Science, Brunel University London, U.K. He is also a Royal Society Industry Fellow and has been recognised as a Highly Cited Researcher by Clarivate Web of Science and a Top 2% Scientist of the World. He has published over 150 papers in IEEE journals and has received several prestigious awards, including the IEEE Communications Society Leonard G. Abraham Prize, Heinrich Hertz Award, and Fred W. Ellersick Prize. His research interests include semantic communications, machine learning for communication systems, mobile edge computing, and wireless networks. He serves as an Associate Editor for IEEE Transactions on Vehicular Technology, IEEE Wireless Communications Letters, and IEEE Communications Letters, and has been actively involved in numerous IEEE conferences and editorial roles.

Cunhua Pan received the B.S. and Ph.D. degrees from the School of Information Science and Engineering, Southeast University, Nanjing, China, in 2010 and 2015, respectively. He was a Research Associate with the University of Kent, Canterbury, U.K., from 2015 to 2016. He held a postdoctoral position with the Queen Mary University of London, London, U.K., from 2016 and 2019, and was a Lecturer from 2019 to 2021. He has been a Full Professor with Southeast University since 2021. He has published over 120 IEEE journal papers. His research interests mainly include reconfigurable intelligent surfaces (RIS), intelligent reflection surface, ultrareliable low-latency communication, machine learning, UAV, Internet of Things, and mobile-edge computing.

Abbas Jamalipour received the Ph.D. degree in electrical engineering from Nagoya University, Nagoya, Japan in 1996. He is the Professor of Ubiquitous Mobile Networking at the University of Sydney. He has authored nine technical books, eleven book chapters, over 650 technical papers, and five patents, all in wireless communications and networking. He is the recipient of several prestigious awards such as the 2026 IEEE James Evans Avant Garde Award, Engineers Australia 2025 Neville Thiele Eminence Award, IEEE ComSoc Harold Sobol Award, the IEEE ComSoc Best Tutorial Paper Award, as well as over fifteen Best Paper Awards. He is currently serving as the IEEE ComSoc Vice President-MGA and was the Editor-in-Chief IEEE WIRELESS COMMUNICATIONS, Vice President-Conferences, and a member of Board of Governors of the IEEE Communications Society. Within IEEE Vehicular Technology Society, he has held positions of President, Executive Vice-President, elected member of the Board of Governors, and the Editor-in-Chief of IEEE TRANSACTION ON VEHICULAR TECHNOLOGY and VTS Mobile World. He is a Fellow of the Institute of Electrical, Information, and Communication Engineers (IEICE), the Institution of Engineers Australia (IEAust), and the International Artificial Intelligence Industry Alliance (AIIA), and a Visiting Fellow of the UK Royal Academy of Engineering.