Alternate Learning-Based SNR-Adaptive Sparse Semantic Visual Transmission

Siyu Tong, Xiaoxue Yu, *Student Member, IEEE*, Rongpeng Li, *Senior Member, IEEE*, Kun Lu, Zhifeng Zhao, *Senior Member, IEEE*, and Honggang Zhang, *Fellow, IEEE*

***Abstract*— Semantic Communication (SemCom) demonstrates strong superiority over conventional bit-level accurate transmission, by only attempting to recover the essential semantic information of data. Nevertheless, most SemCom works train the whole system in an End-to-End (E2E) way, with the assumption of a differentiable channel which is rare in reality applications. In this paper, to tackle the non-differentiability of channels, we propose an alternate learning-based sparse SemCom system with an SNR-adaptive capability for visual transmission, named SparseSBC-SADM. Specially, SparseSBC-SADM leverages two separate Deep Neural Network (DNN)-based models at the transmitter (TX) and receiver (RX), respectively. It alternates between learning the encoding and decoding processes, rather than the joint optimization commonly found in existing literature, to solve the non-differentiability in the channel. In particular, a "self-critic" training scheme is leveraged for stable training. Moreover, the DNN-based TX generates a sparse set of bits in deduced "semantic bases", by further incorporating a binary quantization module by combining Compressive Sensing (CS) and DNN on the basis of minimal detrimental effect to the semantic accuracy. Furthermore, enlightened from the denoising steps in the Denoising Diffusion Model (DDM), a lightweight, SNR-Adaptive Denoising Module (SADM) is provisionally deployed at RX to improve data reconstruction with a gate mechanism to determine the activation under poor channel conditions. Extensive simulation results validate that SparseSBC-SADM shows efficient and effective transmission performance under various channel conditions, and outperforms typical SemCom solutions.**

***Index Terms*— Sparse semantic communication, visual transmission, non-differentiable channel, alternate self-critic learning, denoising diffusion model.**

I. Introduction

RECENTLY, faced with the advent of the Internet of Intelligence (IoI) [2], Semantic Communication (SemCom) [3], [4], which puts more emphasis on the semantic-level accuracy, emerges as a novel and promising solution and attracts significant research interest, as the classical bit-level accuracy-oriented communication techniques approach the Shannon limit and fail to satisfy stringent requirements in the IoI era. In line with implementation maturity to extract semantic features from the source, SemCom could be generally classified as text transmission [5], [6], [7], speech transmission [8], [9], image transmission [10], [11], [12], [13], [14], [15] and video transmission [16], [17].

Notably, SemCom concentrates on semantics rather than bits' correction in communication transmission, which promotes the application of Artificial Intelligence (AI) technology in communication in recent years. In this regard, a series of Joint Source-Channel Coding (JSCC) communication systems [10], [11], [18] emerge with communication quality getting enhanced than traditional systems. Nevertheless, most SemCom works adopt an End-to-End (E2E) approach to train the corresponding Deep Neural Network (DNN) structures, and implicitly assume the differentiability in the channel layer [10]. In large-scale and realistic wireless communication, the gradient is unable to be fully backpropagated through a multi-path physical channel. Therefore, the strong assumption of the differentiable channel in JSCC schemes might not hold in practice [3], which poses a significant challenge for these models. Instead, alternately learned transmitter (TX) and receiver (RX), which separate the learning procedure in JSCC structure, can be more competent to deal with this challenge in visual transmission. Additionally, this kind of separation fosters an easier implementation means to add or replace modules at both TX and RX sides if necessary.

Besides, given the widespread application demands for images, there has been significant development in signal processing techniques specifically for image compression. As a sub-Nyquist sampling framework, Compressive Sensing (CS) has been introduced into communication for efficient compression [19], [20]. Unfortunately, in the literature, there

Received 14 May 2024; revised 17 September 2024 and 3 December 2024; accepted 3 December 2024. Date of publication 13 December 2024; date of current version 13 February 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2024YFE0200600, in part by the National Natural Science Foundation of China under Grant 62071425, in part by Zhejiang Key Research and Development Plan under Grant 2022C01093, in part by Zhejiang Provincial Natural Science Foundation of China under Grant LR23F010005, in part by the National Key Laboratory of Wireless Communications Foundation under Grant 2023KP01601, and in part by the Big Data and Intelligent Computing Key Laboratory of Chongqing University of Posts and Telecommunication (CQUPT) under Grant BDIC-2023-B-001. An earlier version of this paper was presented in part at the IEEE PIMRC 2023 [DOI: 10.1109/PIMRC56721.2023.10293971]. The associate editor coordinating the review of this article and approving it for publication was F. Chiariotti. *(Corresponding author: Rongpeng Li.)*

Siyu Tong, Xiaoxue Yu, and Rongpeng Li are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: tongsiyu@zju.edu.cn; sdwhyxx@zju.edu.cn; lirongpeng@zju.edu.cn).

Kun Lu was with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. He is now with Huawei Technologies Company Ltd., Shenzhen 518129, China (e-mail: 22060598@zju.edu.cn).

Zhifeng Zhao is with Zhejiang Lab and the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: zhaozf@zhejianglab.com).

Honggang Zhang was with Zhejiang Lab and the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. He is now with the Faculty of Data Science, City University of Macau, Macau, China (e-mail: honggangzhang@zju.edu.cn).

Digital Object Identifier 10.1109/TWC.2024.3512652

TABLE I
SUMMARY AND COMPARISON WITH RELATED LITERATURE

Related Works	Brief Description	Limitations
[23]	It learns the noise distribution of the channel input signal after the MMSE equalizer for denoising.	It trains with differentiable objective functions under the JSCC framework.
[24]	It combines digital communication with generative refinements, achieving bandwidth efficiency.	The reliance on the diffusion process to generate the refined component introduces a level of randomness.
[25]	Its integration of DDM enables the generation of high-quality images from transmitted binary maps.	The generated outputs may not consistently align with the original content.
Ours	Integrating a lightweight SNR-adaptive denoising module with a self-critic strategy, our method reduces randomness in non-differentiable channels through alternately learning sparse semantic embedding.	

shed little light on explicitly investigating the sparsity into semantic image transmission. However, considering the successful applications of CS, it remains worthwhile to effectively combine the latest sparsity-driven techniques and advanced DNNs towards SemCom for image transmission. In addition, RX typically experiences performance loss due to channel noise, making further enhancements to robustness a valuable pursuit. Due to the great achievements of the Denoising Diffusion Model (DDM) [21], [22] in the field of visual generation, numerous valuable applications have emerged in wireless communications [23], [24], [25]. Considering the capability of DDM to fit the observed data to mitigate the negative effects of noise in SemCom [23], continuing exploration in this area is highly worthwhile. Nevertheless, the introduction of large-scale modeling may bring about a reduction in efficiency. Inspired by the task-level Mixture of Experts (MoE) mechanism [26], which splits a system skillfully into multiple independent networks, each corresponding to one task, we can also design a gating mechanism to accommodate the received Signal-to-Noise Ratio (SNR) and decide whether to activate the Denoising Module (DM) or not.

In this paper, we put forward SparseSBC-SADM, an alternate learning-based sparse SemCom framework for visual transmission. In particular, SparseSBC-SADM involves a TX and a RX to learn the transmission scheme by turns, thus obtaining the applicability for both differentiable and non-differentiable channels. In addition, SparseSBC-SADM adopts a "self-critic" scheme to overcome the divergence of semantic decision space in the training procedure. Furthermore, SparseSBC-SADM deduces semantic bases that can describe all semantic embeddings with calibrated DNN-driven modules. On top of that, SparseSBC-SADM quantizes the sparsely represented bits from the space of semantic bases for transmission. Conversely, by jointly taking account of sparse signal recovery and JSCC-based decoding, SparseSBC-SADM reconstructs the image from the received noisy bits. Apart from this, drawing on the idea of task-level MoE, we propose an SNR-adaptive DM (SADM), which can be provisionally activated under poor channel conditions to enhance the decoding performance. Furthermore, a lightweight U-Net [27] for DDM is designed to improve the decoding efficiency. In brief, after tabulating the differences among similar literature in Table I, we summarize the major contribution of our work as follows:

- We put forward SparseSBC-SADM, an alternate learning-based sparse SemCom framework for visual transmission, by effectively combining the concept of CS and the advance of DNNs. In particular, a "self-critic" scheme is introduced into the training procedure to achieve better exploration and a stable learning process.
- Within SparseSBC-SADM, at TX, besides the encoder, we additionally transform semantically encoded bits into the space of semantic bases and quantize the results by building up non-linear mapping for efficient channel transmission. Meanwhile, similar operations are applied to RX as well.
- At RX of SparseSBC-SADM, an SNR-adaptive gating mechanism decides the necessity of activating the DDM-derived DM according to the channel condition, ensuring enhanced efficiency and performance. Besides, it incorporates a lightweight U-Net architecture, which leads to optimized resource utilization.
- Through extensive simulations in various channel conditions, we validate that SparseSBC-SADM shows superior performance than "BPG+LDPC [28]" and JSCC-schemes [13], [29], [30], in terms of the reconstruction performance under the same ratio of the Energy per Bit to Noise Power Spectral Density (E_b/N_0). The ablation experiments demonstrate the denoising ability of SparseSBC-SADM in poor channel environments.

The remainder of the paper is organized as follows. Section II introduces recent related works and demonstrates the necessity of our work. In Section III, we present the framework of SparseSBC-SADM. In Section IV, we explain the details of implementations of DNN-based modules, and show the alternate learning training scheme. Section V gives the corresponding simulation results. Section VI concludes the paper.

II. Related Works

A. Deep Joint Source-Channel Coding

As traditional image compression methods, JPEG [31] and JPEG2000 utilize quantization and entropy coder to compress images for transmitting fewer bits. Meanwhile, HEVC [32] and MPEG adopt the hybrid coding framework based on transform coding and prediction. Better Portable Graphics (BPG) [33] leverages technology from HEVC to achieve superior compression ratios compared to JPEG, without sacrificing image quality. On the contrary, the latest studies in JSCC transmission schemes have promised significant improvements in text [5], [6], speech [8], image [30], [34], [35] and video [16] communication methods. For example, Long Short-Term Memory (LSTM) and Transformer [36]-based natural language processing techniques have been extensively leveraged to design JSCC [29] for text transmission [5], [6], [7]. Meanwhile, to understand emotions and tunes of speech more thoroughly, attention-based squeeze-and-excitation [8] and symbol recognition modules [9] have been added to the basic JSCC structure for speech transmission.

Furthermore, as for visual transmission, limited to the storage of equipment and capability of transmission channel, JSCC-based image transmission methods [10], [11] are often contingent on image compression works. In that regard, swin transformer [34] demonstrates astonishing performance in (lossy) image compression. Besides, by integrating the principles of the information bottleneck method with JSCC, AIB-JSCC [35] reduces the requirements for image transmission rate while enhancing the reconstruction quality. Based on an adaptive deep learning framework and JSCC structure, [37] optimizes data rate and quality based on channel conditions and image content. Nevertheless, contemporary JSCC-based visual communication systems, which directly transmit embeddings, lack superiority in terms of the performance under the same E_b/N_0.

B. Compressive Sensing in Visual Domain

As a classical framework, CS has gained significant attention in the image processing domain and has been employed to capture the sparsity in images and boost the performance of imaging applications [38]. By designing a special measurement matrix for better CS process and encryption, [39] guarantees the reconstruction of high-quality images. On the other hand, CS is useful in communication systems where traditional sampling methods may be costly or unfeasible. Reference [19] proposes an E2E communication system with the integration of CS and traditional image compression techniques, achieving competitive image compression performance, especially in low-bit rate scenarios. By formulating active user detection and channel estimation approaches within the framework of CS, [20] achieves rapid detection of active users and accurate estimation of channels. Nevertheless, the integration of CS with SemCom for image transmission is still relatively under-explored.

C. AIGC in Communication System

Artificial Intelligence Generated Content (AIGC) has been widely applied in image transmission research, which is attributed to its robustness and high-quality image generation ability. A GAN-based method [15] achieves real-time image compression performance based on an adversarial training procedure. Besides, a VAE-based communication system [40] shows greater robustness to degradation than auto-encoders for image transmission in wireless channels. On the other hand, the application of DDM in wireless communication is still being explored. In [25], DDM on RX uses spatially adaptive normalizations from such denoised semantic information, ensuring that the details of the scenes are consistently filled. Reference [24] proposes a hybrid communication system based on DDM and JSCC. In addition to the transmission of coarsely compressed images by traditional methods, noisy images generated by the forward process of DDM are also transmitted to improve reconstructed image quality. Reference [23] focuses on the superiority of the denoising procedure in Denoising Diffusion Probabilistic Models (DDPM) [41] and applies it after channel equalization to learn channel input signal, the distribution of which is further utilized recover images from noisy data. Furthermore, [42] targets finite-precision wireless communications, and designs DDPMs with the account of more realistic channel factors like HardWare Impairments (HWI), channel distortions and quantization errors.

TABLE II
NOTIONS USED IN THIS PAPER

Notation	Definition
TX, RX	The abbreviated terms for transmitter and receiver
$\mathcal{T}, \mathcal{R}, \mathcal{H}$	Abstract transmitter, receiver and noisy channel
$\mathbf{I}, \hat{\mathbf{I}}$	The original image and the received image
d_1, d_2	Image dimensions
$\mathbf{T_s}, \mathbf{T_c}, \mathbf{R_s}, \mathbf{R_c}, \mathbf{D}$	Abstract encoders, decoders and the SNR-adaptive denoising module
ϕ, θ, υ	Parameters for encoder, decoder and denoising module respectively
ϕ^*, θ^*	Optimal policies for TX and RX
$r_\phi^{(l)}, r_\theta^{(l)}$	The learning reward of TX and RX of each batch l
$\mathbf{Q_T}, \mathbf{Q_R}$	Quantization module and dequantization module
M, N	Length of embeddings before and after sparse quantization
$\mathbf{x}, \hat{\mathbf{x}}$	Embeddings before the transmission of the channel
$h, \mathbf{n}$	The channel coefficients and channel noise
$\mathbf{y}, \hat{\mathbf{y}}$	Embeddings after the transmission of the channel
$\hat{\mathbf{z}}$	Embeddings after the SNR-adaptive denoising model
$\mathcal{L}_{(\cdot)}$	Different loss functions
$\mathcal{J}$	The objective function
$\mathcal{N}(\cdot, \cdot)$	Gaussian distribution
α_t	The forward process variances of denoising module
$\boldsymbol{\epsilon}, \boldsymbol{\epsilon}_\upsilon$	Random noise and noise predicted by U-Net
σ	Scale factor
ε	Sparse weights of the transmitted bits
$\boldsymbol{\Theta}$	Semantic similarity metric
B	Batch size
E_i	Epoch numbers of each training stage
lr	Learning rate
m	Number of self-critic samples
$p(\cdot), q(\cdot)$	Predictive and real transfer probability of each step in denoising module
T	Number of diffusion and sampling steps

III. SYSTEM MODEL

A. General Framework of SemCom

As illustrated in Fig. 1, we primarily consider the SemCom framework encompassing an encoder and a decoder as TX and RX respectively, as well as a noisy channel $\mathcal{H}$. Both encoder and decoder, which are implemented by DNNs, are mutually contingent. Specifically, the encoder is logically comprised of

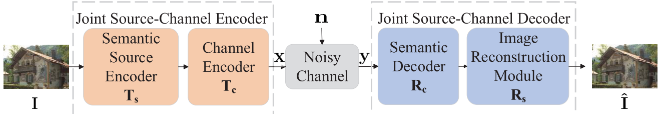

Joint Source-Channel Encoder
Semantic Source Encoder T_s
Channel Encoder T_c
x
n
Noisy Channel
y
Joint Source-Channel Decoder
Semantic Decoder R_c
Image Reconstruction Module R_s
I
Î

Fig. 1. Typical block structure of SemCom.

two DNN-based modules (i.e., a semantic source encoder $\mathbf{T_s}$ and a channel encoder $\mathbf{T_c}$) to extract low-dimensional features from the original image and encode them into symbols for channel transmission, respectively. Without loss of generality, an image $\mathbf{I}$, with $d_1 \times d_2$ pixels, can be downscaled to lower-dimension embeddings $\mathbf{x} \in \mathbb{R}^M$ as

$$\mathbf{x} = \mathbf{T_c}\left(\mathbf{T_s}\left(\mathbf{I}\right)\right). \tag{1}$$

At RX, the received signals $\mathbf{y}$ can be formulated as

$$\mathbf{y} = h\mathbf{x} + \mathbf{n}, \tag{2}$$

where h denotes the channel coefficient, $\mathbf{n}$ represents the independent and identically distributed (i.i.d.) noise in the channel, following a circularly symmetric Gaussian distribution. In that regard, the channel could be modeled as a non-trainable layer to fit the training process. Typically, the Additive White Gaussian Noise (AWGN) channel and the Phase-Invariant Fading (PIF) channel are generally considered in most typical studies [3]. Mathematically, the channel can be considered as an AWGN channel when h is set to a scalar, while in PIF channel h is assumed to follow a constant distribution. Furthermore, 3GPP Technical Report (TR) 38.901 [43] also incorporates a non-differential, more practical Tapped Delay Line (TDL) channel model, which characterizes the propagation characteristics of multi-path signals by adopting a finite number of discrete paths with random time-varying taps and stochastic Doppler shifts.

Conversely, RX utilizes a semantic channel decoder $\mathbf{R_c}$ and an image reconstruction module $\mathbf{R_s}$ to recover from noisy signals. Besides, with a denormalization layer, the value of each pixel for every color channel is restored to $(0, 255)$. Mathematically,

$$\hat{\mathbf{I}} = \mathbf{R_s}\left(\mathbf{R_c}\left(\mathbf{y}\right)\right), \tag{3}$$

where $\hat{\mathbf{I}}$ denotes the reconstructed image. Basically, to further evaluate the quality of the reconstructed image, we adopt the L_1-norm difference between pixels in the original image and the reconstructed one. In other words,

$$\mathcal{L} = \frac{1}{2d_1 d_2} \sum\nolimits_{i,j}^{d_1,d_2} |\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j}|, \tag{4}$$

where $\mathbf{I}_{i,j}$ denotes the pixel (i,j) of a $d_1 \times d_2$ image $\mathbf{I}$, and the smaller $\mathcal{L}$ yields higher semantic similarity between $\mathbf{I}$ and $\hat{\mathbf{I}}$.

B. Framework of SparseSBC-SADM

Fig. 2 presents the framework of SparseSBC-SADM. Notably, compared to SemCom in Fig. 1, SparseSBC-SADM prominently adds a CS-consistent DNN module $\mathbf{Q_T}$ to further map $\mathbf{x}$ to $\hat{\mathbf{x}} \in \mathbb{Z}_2^N$, where $\mathbb{Z}_2 = \{0, 1\}$. Since each image can be fully described by semantic bases, through the process of $\mathbf{Q_T}$, an image can be expressed sparsely in terms of a specific set of bases, and thus transformed into a sequence of bits with a larger compression ratio. Correspondingly,

$$\hat{\mathbf{x}} = \mathbf{Q_T}(\mathbf{x}) = \mathbf{Q_T}\left(\mathbf{T_c}\left(\mathbf{T_s}\left(\mathbf{I}\right)\right)\right). \tag{5}$$

In other words, this quantization step, which implements a code rate $\mathrm{CBR} = k/(d_1 d_2)$, encodes all possible embeddings into a fixed sequence of 0s and 1s, and such a binary quantization method is advantageous for transmission. Notably, such a transformation also bypasses the necessity of pre-channel quantization at the TX.

After receiving the sparse quantization signal $\hat{\mathbf{x}}$, RX uses the DM $\mathbf{D}$ to remove noises from the received signals $\hat{\mathbf{y}} = h\hat{\mathbf{x}} + \mathbf{n}$,[1] and to generate $\hat{\mathbf{z}}$ for dequantization module $\mathbf{Q_R}$, namely

$$\hat{\mathbf{z}} = \begin{cases} \mathbf{D}(\hat{\mathbf{y}}), & \text{if } \textbf{Activate}; \\ \hat{\mathbf{y}}, & \text{otherwise}, \end{cases} \tag{6}$$

where **Activate** is a boolean gate to ascertain whether to activate DM. The implementation details of **Activate** will be discussed in Section IV-B.

Subsequently, the dequantization module and decoders transform $\hat{\mathbf{z}}$ to embeddings and recover the images like the reverse process of the encoder, that is,

$$\hat{\mathbf{I}} = \mathbf{R_s}\left(\mathbf{R_c}\left(\mathbf{Q_R}\left(\hat{\mathbf{z}}\right)\right)\right). \tag{7}$$

In resemblance to CS [38], $\mathbf{Q_T}$ and $\mathbf{Q_R}$ are somewhat equivalent to nonlinear mapping and reconstruction matrices. Besides, for simplicity of representation, we denote TX as $\mathcal{T} = \mathbf{Q_T}\left(\mathbf{T_c}\left(\mathbf{T_s}(\cdot)\right)\right)$ and RX as $\mathcal{R} = \mathbf{R_s}\left(\mathbf{R_c}\left(\mathbf{Q_R}(\mathbf{D}(\cdot))\right)\right)$ hereafter.

Apart from adopting $\mathcal{L}$ (i.e., L_1-norm) to measure the semantic similarity, a sparse factor $\mathcal{L}_\text{s}$ will be used to impose the sparsity of the transmitted bits. In other words,

$$\mathcal{L}_\text{full} = \underbrace{\frac{1}{2N} \sum\nolimits_{i,j}^{d_1,d_2} |\mathbf{I}_{ij} - \hat{\mathbf{I}}_{ij}|}_{\mathcal{L}} + \underbrace{\varepsilon \sum\nolimits_{i=1}^{N} |\hat{x}_i|}_{\mathcal{L}_\text{s}}, \tag{8}$$

where $\hat{\mathbf{x}} = [\hat{x}_1, \cdots, \hat{x}_N]$ and ε denotes the sparsity weight.

In the next section, we will discuss the individual modules and implementation details of SparseSBC-SADM.

IV. Implementation Details of SparseSBC-SADM

In this section, we will talk about how to develop DNN-based modules (e.g., the SNR-adaptive denoising module) of SparseSBC-SADM. We also explain how to use the alternate learning self-critic scheme to bypass the non-differentiable issue of channels.

A. DNN Structure

In SparseSBC-SADM, we adopt the following DNN structure and provide one viable means while leaving the utilization of other DNN structures (e.g., transformer [36]) as future works. Besides, we denote the parameters in the encoder and decoder DNNs (excluding the DM) as ϕ and θ, respectively.

[1] We slightly abuse the notation $\hat{\mathbf{y}}$ with $\mathbf{y}$ in (2) to maintain the statement consistency.

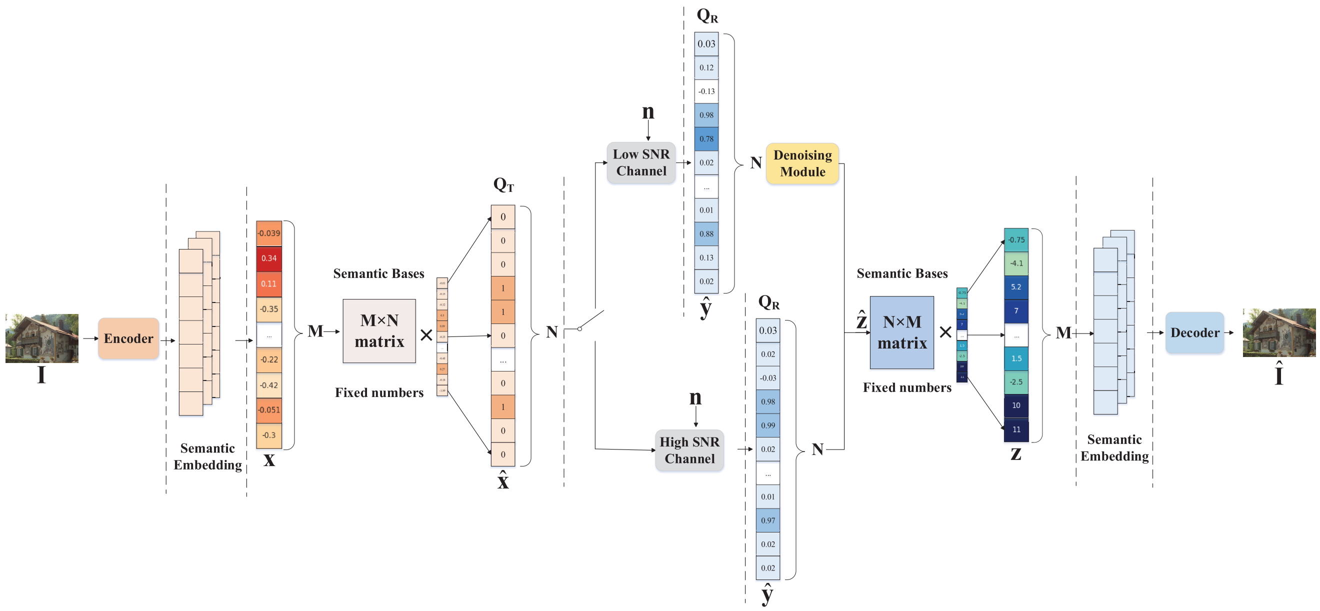


Fig. 2. Framework of SparseSBC-SADM.

1) Design of Source & Channel Encoder and Decoder: At the TX side, the joint encoder encompasses multiple convolutional layers with LeakyReLU activation functions to extract semantic features from an image $\mathbf{I}$ and reshape the features to an M-length vector $\mathbf{x}$. Similarly, the decoder at RX contains de-convolutional layers to restore the images. The details of convolutional layers are summarized in Fig. 5.

2) Design of Quantization and Dequantization Module: The DNN for both $\mathbf{Q_T}$ and $\mathbf{Q_R}$ consists of a fully connected layer with Tanh activation function, to transform semantically encoded bits to a sequence of numbers ranging from -1 to 1. Afterward, a binary quantization is performed to obtain N-length binary bits vector $\hat{\mathbf{x}}$ (i.e., containing 0s or 1s only). Notably, such a fully connected layer resembles the mapping and reconstruction matrices (to map and reconstruct from semantic bases) in CS; while the latter Tanh function and quantization module capture non-linear transformations.

3) Lightweight DM Structure: With reference to DDPM, we also choose U-Net [27], [41], an Encoder-Decoder structure network, as the backbone of SADM. Stacked by multiple layers of the same number of encoder blocks and decoder blocks on each side, U-Net also contains attention blocks and residual blocks. The down-sampling block comprises convolutional layers and attention blocks that can better compress data, while in the up-sampling period, the input is derived from both the previous decoder block and the co-level encoder block, ensuring that decoders would not lose information in the inference and reconstruction process. Specifically, time-embeddings are obtained by Multi-Layer Perceptron (MLP). We leave the details in Section IV-B. Considering a large number of parameters in the classical U-Net, we reduce the number of encoder blocks and decoder blocks to fit in the communication system. As illustrated in Fig. 3, we set the number of encoders and decoders to 2 in the developed lightweight U-Net and will clarify its advantage in Section V-B.2.

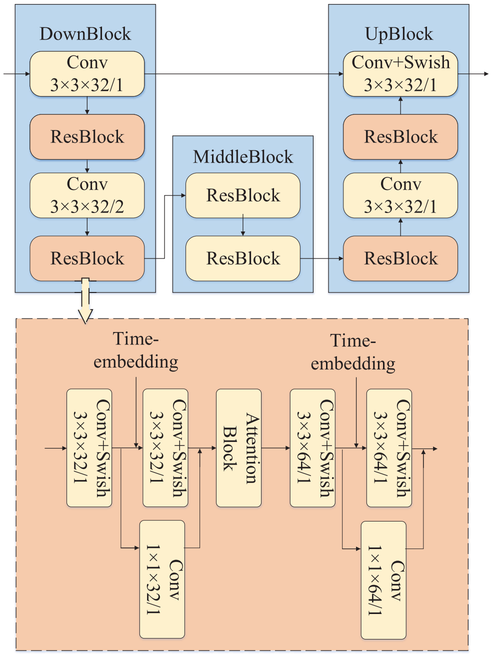


Fig. 3. The structure and details of lightweight U-Net in SparseSBC-SADM.

B. SNR-Aware Denoising Module Design

The output obtained through the noisy channel in (2) can be interpreted as the consequence of a noise-influence process. In other words, within SparseSBC-SADM, we consider the effect of the channel as a sequence of stages where noise is progressively superimposed on the transmitted signal, similar to the way that noise is incrementally added in the forward process of DDPM [41], as illustrated in Fig. 4. Correspondingly, we incorporate a DDPM-alike DM to incrementally and provisionally denoise the noisy signal, thus improving the overall signal quality and reliability of data transmission.

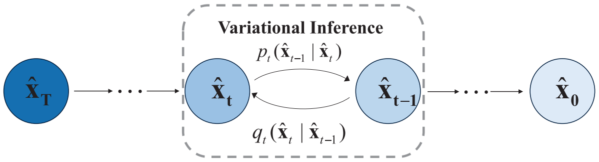

Variational Inference
$p_t(\hat{\mathbf{x}}_{t-1} \mid \hat{\mathbf{x}}_t)$
$\hat{\mathbf{x}}_T$
$\hat{\mathbf{x}}_t$
$\hat{\mathbf{x}}_{t-1}$
$\hat{\mathbf{x}}_0$
$q_t(\hat{\mathbf{x}}_t \mid \hat{\mathbf{x}}_{t-1})$

Fig. 4. The training representation of the DM based SNR-adaptive denoising module.

Specifically, the prediction of each step in the denoising procedure of DM anticipates a Gaussian noise $\boldsymbol{\epsilon}_{\upsilon}(\hat{\mathbf{y}}_t, t)$ parameterized by υ for the subsequent iteration, and the denoising process systematically iterates the data from the noisy state $\hat{\mathbf{y}}_T = \hat{\mathbf{y}}$. After progressively eliminating noise to obtain $\hat{\mathbf{y}}_t$, $\forall t \in \{T, \cdots, 0\}$, the original data is obtained as $\hat{\mathbf{z}} = D(\hat{\mathbf{y}}) = \hat{\mathbf{y}}_0$. Mathematically, we can sample $\hat{\mathbf{y}}_{t-1}$ from $p(\hat{\mathbf{y}}_{t-1}|\hat{\mathbf{y}}_t)$ and $\boldsymbol{\epsilon}_{\upsilon}$ as [41]

$$\hat{\mathbf{y}}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\hat{\mathbf{y}}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\upsilon}(\hat{\mathbf{y}}_t, t)\right) + \sqrt{(1-\alpha_t)}\boldsymbol{\epsilon}_t, \tag{9}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{E})$ denotes the random noise for image reconstruction at each step t, and $\bar{\alpha} = \prod_{i=1}^{t} \alpha_i$ and $\alpha_0 \geq \alpha_i \geq \alpha_T \in (0, 1)$ is a hyper-parameter decreased over t.

As the inference time of DM can be prohibitively long, it impacts the operational efficiency of SemCom. To address this challenge, we draw inspiration from task-level MoE [26], which demonstrates the capability to dynamically allocate computational resources when handling diverse tasks, and adopt a similar approach to the design of DM. Specifically, we introduce a novel SNR-adaptive DM, and the decision to activate DM is based on the performance gain measured by one of the image quality metrics $\Delta\mathbf{S}$. When the average performance gain $\Delta\mathbf{S}$ upon the activation of DM exceeds a predefined threshold k, DM is engaged, that is,

$$\textbf{Activate} = \begin{cases} \texttt{TRUE}, & \text{if } \Delta\mathbf{S} \geq k; \\ \texttt{FALSE}, & \text{otherwise.} \end{cases} \tag{10}$$

where k serves as a performance-driven threshold to decide whether to activate SADM for a better denoising function. Meanwhile, $\Delta\mathbf{S}$, which may include but not limited to Peak Signal-to-Noise (PSNR) [10], Structural Similarity Index Measure (SSIM) [44] and Fréchet Inception Distance (FID) [45], is ascertained from simulation outcomes corresponding to different tasks and channel conditions. $\Delta\mathbf{S}$ is determined computationally during the training phase in Section IV-D, rendering it a constant under the specified channel conditions for practical communications. Subsequently, during the inference phase, we leverage this MoE-alike component to determine the activation of SADM appropriately. In other words, SADM can intelligently adjust itself according to the prevailing conditions by (6) and (10), and the evaluation results are illustrated in Section V-B.2.

C. Integration of Alternate Learning and SemCom

Most typical SemCom schemes discuss the differentiable objective optimization, but such a stringent assumption on transmission channels might not always hold in practice. Therefore, we consider an alternate learning scheme into SemCom to deal with the non-differentiability of random channels and help to turn the whole learning system into a collaborative semantic transceiver.

To better illustrate the idea of alternate learning, we define the input of $\mathcal{T}$ and $\mathcal{R}$ at each batch l respectively. That is, for TX, it encodes $\mathbf{I}^{(l)} \in \mathbb{R}^{d_1 \times d_2}$ to $\hat{\mathbf{x}}^{(l)} \in \mathbb{Z}_2^N$, and at RX, it decodes $\hat{\mathbf{y}}^{(l)} \in \mathbb{R}^N$ to $\hat{\mathbf{I}}^{(l)} \in \mathbb{R}^{d_1 \times d_2}$. Both TX and the RX target reconstruct images as close as the original signals. Meanwhile, TX shall impose the sparsity of encoded embeddings. Hence, if only considering the encoder parameterized by ϕ and the decoder parameterized by θ while temporarily neglecting the impact of the DM, the alternate learning reward can be formulated by semantic similarity as

$$r_{\phi}^{(l)} = \Theta_{\phi}(\mathbf{I}^{(l)}, \hat{\mathbf{I}}^{(l)}, \hat{\mathbf{x}}^{(l)}) = 1 - \mathcal{L}_{\text{full}}, \tag{11a}$$

$$r_{\theta}^{(l)} = \Theta_{\theta}(\mathbf{I}^{(l)}, \hat{\mathbf{I}}^{(l)}) = 1 - \mathcal{L}. \tag{11b}$$

Different from JSCC systems, to maximize the learning reward, we train the encoder and decoder alternately by taking a batch of B samples (i.e., image transmissions) as a mini-batch to optimize the following objective function

$$J = \mathop{\mathbb{E}}_{\hat{\mathbf{x}}^{(1)}, \cdots, \hat{\mathbf{x}}^{(B)}} \left[\sum\nolimits_{l=1}^{B} r^{(l)}\right], \tag{12}$$

where r takes the formula in (11a) and (11b) for ϕ and θ, respectively. As the terminology "alternate learning" implies, we freeze θ and thus the objective function is only parameterized on ϕ, that is, $J(\phi)$ and r_{ϕ}. The convergence of "alternate learning" has been theorectically establised in the following theorem [46].

Theorem 1 (Theorem 3 of [46]): In the context of alternate learning where TX is parameterized by θ and RX is parameterized by ϕ for their policy updates, the reward approaches a steady-state value.

To further overcome the difficulty of divergence in training for high-dimensional semantic space, inspired by the methodology in Reinforcement Learning (RL), we adopt a simpler and quicker scheme named "self-critic" [47] based on the Gaussian policy gradient, which will be given in-depth in Section IV-D.

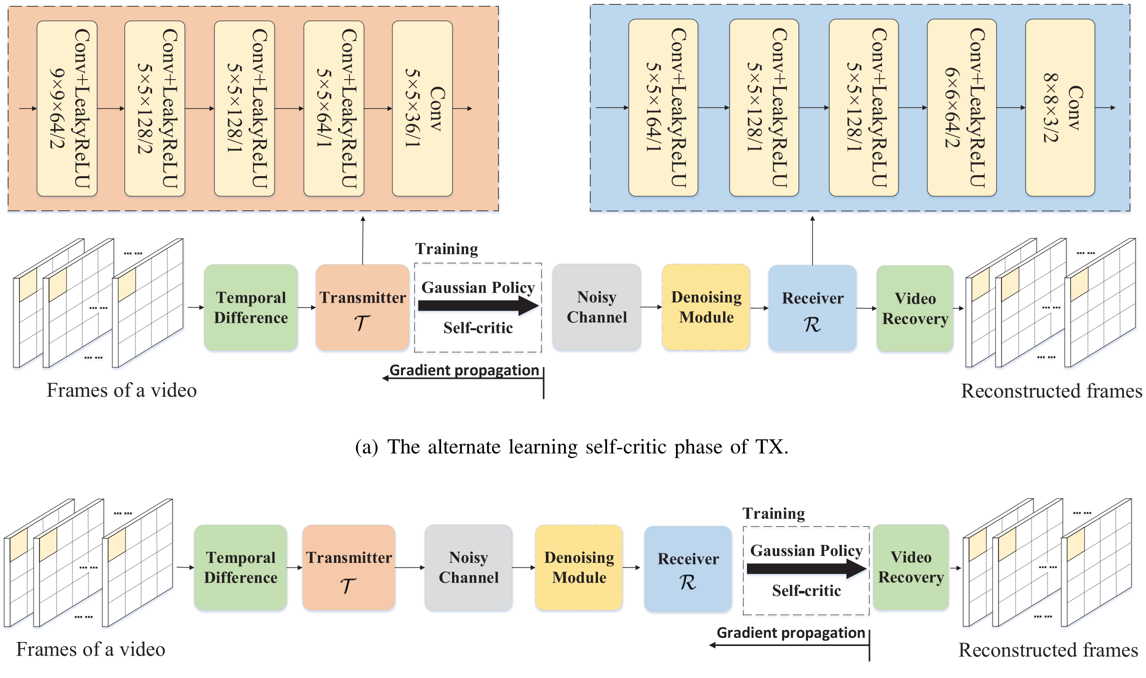
D. Training Procedures

In this part, we will explain the details of the training procedure for the whole system SparseSBC-SADM, which is primarily divided into three stages.

1) Training Encoder and Decoder by Alternate Learning: We first introduce the training procedures of the alternate learning process of SparseSBC, that is, the first training stage. Specifically, we explain TX in-depth as a detailed example. In particular, we attempt to learn an optimal policy ϕ^* for TX:

$$\phi^* = \arg\max_{\mathcal{T}} \Theta_{\phi}(\mathbf{I}, \underbrace{\mathcal{R}(\mathcal{H}(\mathcal{T}(\mathbf{I})))}_{\text{no grad}}). \tag{13}$$

Notably, instead of training the model to estimate the baseline directly, we use the average return from a group of parallel samples as the baseline [47]. Specifically, we repetitively



(a) The alternate learning self-critic phase of TX.

(b) The alternate learning self-critic phase of RX.

Fig. 5. The details of TX and RX in the training period, which both adopt the alternate learning scheme.

generate the encoding results from the input $\mathbf{I}^{(l)}$ m times according to the Gaussian distribution,

$$\hat{\mathbf{x}}_j^{(l)} \sim \mathrm{S}\left(\mathcal{N}\left(\boldsymbol{\mu}_\phi^{(l)} = \hat{\mathbf{x}}^{(l)}, \boldsymbol{\Sigma}\right)\right), \forall j = \{1, \cdots, m\}, \tag{14}$$

where S denotes the parallel sampling following the Gaussian distribution $\mathcal{N}$. $\boldsymbol{\Sigma} = \left(\sigma^2\mathbf{E}\right) \in \mathbb{R}^{N\times N}$ is a covariance matrix set by the identity matrix $\mathbf{E}$ and a scale factor σ, which can be regarded as an exploration factor to get more abundant expression of embeddings. Since a constant σ may not be well realized for exploration and exploitation, we can adjust the value of σ in different stages of the training procedure dynamically, similar to the simulated annealing algorithm [48], wherein the value of σ gradually decreases along with the increase of epochs (denoted as "Annealed"). Furthermore, we also consider $\boldsymbol{\Sigma} = \operatorname{diag}[\sigma_1, \cdots, \sigma_N]$ to be a learnable matrix (denoted as "Learnable"), which can be determined by a sigmoid function of the encoded bits, that is,

$$\sigma_i = \operatorname{Sigmoid}\left(\hat{x}_i\right), i \in \{1, 2, \cdots, N\}. \tag{15}$$

Following the decoding operation at RX, we can obtain m rewards, that is, Θ_j, $j \in \{1, \cdots, m\}$ taking a value Θ_ϕ. Without loss of generality, for any j, by regarding the average of the remaining $m-1$ outputs as the bias term, we calculate the difference between Θ_j and the bias term for its parameter update. On the other hand, following our previous work [49], we have the following theorem to demonstrate the result of the policy gradient propagation for TX.

Theorem 2: Let $\widetilde{\mathcal{T}}(\mathbf{I}^{(l)})$ be one of the multi-sampled embeddings in TX at batch l. With the self-critic Gaussian policy gradient defined in (14), the gradient propagation for TX is given as

$$\nabla_\phi \log\left(\pi_\phi^{(l)}\right) = \left[\widetilde{\mathcal{T}}(\mathbf{I}^{(l)}) - \mathcal{T}(\mathbf{I}^{(l)})\right]^\mathsf{T} \boldsymbol{\Sigma}^{-1}\left[\nabla_\phi \mathcal{T}(\mathbf{I}^{(l)})\right]. \tag{16}$$

Therefore, the calculation of the semantic policy gradient can be summarized as

$$\nabla J\left(\phi\right) \approx \frac{1}{mB}\sum_{j=1}^{m}\left[\sum_{l=1}^{B}\nabla_\phi \log \pi_{j;\phi}^{(l)}\left(\Theta_j - \operatorname*{avg}_{k\sim m;k\neq j}\left(\Theta_k\right)\right)\right], \tag{17}$$

where $\operatorname{avg}(\cdot)$ calculates the mean value. Finally, we update ϕ with a learning rate lr, as

$$\phi \leftarrow \phi - \mathrm{lr}\cdot \nabla \mathrm{J}\left(\phi\right). \tag{18}$$

Notably, such a "self-critic" training scheme, which leverages m outputs parallelly in (17), can alleviate the high variance problem of the plain policy gradient and keep a stable training procedure [47]. It also avoids the need to train additional networks for state value estimation, especially those that are unstable in high dimensional spaces. In addition, the computational complexity for the self-critic method can be denoted as $O(m(N^3 + N^2)B + m^2)$, which only adds a constant order and is on par with conventional gradient descent methods.

Similarly, in the training period of θ, ϕ is frozen as well and the Gaussian policy gradient-based "self-critic" is identically adopted to reconstruct images from the received embeddings.

Afterwards, the objective of RX can be optimized as

$$\theta^* = \arg\max_{\mathcal{R}} \Theta_\theta(\mathbf{I}, \mathcal{R}(\underbrace{\mathcal{H}(\mathcal{T}(\mathbf{I}))}_{\text{no grad}})). \tag{19}$$

Specifically, it can be deemed as a form of unsupervised learning, which is independent of channel influence. It circumvents the complexities introduced by channel, and allows RX to learn directly from the data without the need for labeled outcomes.

2) Training of the Denoising Module: In the second stage, based on the well-trained encoder and decoder, we fix the parameters ϕ, θ and train DM separately, while presumably excluding the impact of channel in the whole system. Specifically, each sample from the dataset intentionally undergoes a noise schedule, incrementally perturbed by Gaussian noise $\boldsymbol{\epsilon}$. Afterward, U-Net-based DM adaptively learns to predict the distribution of channel noise $\boldsymbol{\epsilon}_t$ at each step t and counteracts the emulated channel noise. Notably, the training procedure of DM comprises two processes (i.e., the forward process and the reverse process), which is summarized in Algorithm 1.

- The forward process involves multiple steps to simulate the gradual introduction of noise to the signal $\hat{\mathbf{x}}$, as if the signal were transmitted through the channel and became increasingly noisy. By the reparameterization trick, the transformation [21] of the reverse transfer probability at each step can be expressed as

$$\hat{\mathbf{x}}_t = \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, \tag{20}$$

  where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{E})$ is a Gaussian noise. In other words, the t-th data can be defined by Gaussian distribution as $q(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{t-1}) \sim \mathcal{N}(\hat{\mathbf{x}}_t; \sqrt{\alpha_t}\hat{\mathbf{x}}_{t-1}, (1-\alpha_t)\mathbf{E})$. Therefore, the t-th distribution of noisy data in the forward process can be organized by the original data $\hat{\mathbf{x}}_0$ as

$$q(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_0) \sim \mathcal{N}(\hat{\mathbf{x}}_t; \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0, (1-\bar{\alpha}_t)\mathbf{E}). \tag{21}$$

- The reverse process in DM is consistent with DDPM to counteract the effect of noise, so as to reconstruct the original signal from the noisy observations.
  To regain the original data, DM learns to predict the distribution of each step, that is, $q(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t)$ by Gaussian distribution $p(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t)$ at each step, which can be unraveled by Bayesian inference. To evaluate the prediction of noise, we can use Variational Lower Bound (VLB) to optimize the negative log-likelihood as [50]

$$\begin{aligned}\mathcal{L}_D = \mathcal{L}_{\text{VLB}} &= \mathbb{E}_{q(\hat{\mathbf{x}}_{0:T})}\left[\log\frac{q(\hat{\mathbf{x}}_{1:T}|\hat{\mathbf{x}}_0)}{p(\hat{\mathbf{x}}_{0:T})}\right] \\ &= \mathbb{E}_q[\underbrace{D_{\text{KL}}(q(\hat{\mathbf{x}}_T|\hat{\mathbf{x}}_0) \parallel p(\hat{\mathbf{x}}_T))}_{\mathcal{L}_T} \\ &\quad + \sum\nolimits_{t=2}^{T} \underbrace{D_{\text{KL}}(q(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_0) \parallel p(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t))}_{\mathcal{L}_{t-1}} \\ &\quad - \underbrace{\log p(\hat{\mathbf{x}}_0|\hat{\mathbf{x}}_1)}_{\mathcal{L}_0}],\end{aligned} \tag{22}$$

  where D_{KL} denotes the Kullback–Leibler (KL)-divergence, which is used to measure the similarity between two Gaussian distributions. Moreover, $\mathcal{L}_D$ can also be represented as a sum of $\mathcal{L}_t$ across each timestep t, denoted by the subscript. Since $\mathcal{L}_T$ is a constant and $\mathcal{L}_0$ is considered as a decoder to regain the origin form for the dequantization module, they can both be omitted from consideration during training and be approximated and simplified by a constant $\mathbf{C}$. In this case, by ignoring the weighting term [41], we have

$$\begin{aligned}\mathcal{L}_D &= \sum\nolimits_{t=1}^{T-1} \mathcal{L}_t + \mathbf{C} \\ &:= \mathbb{E}_{t,\hat{\mathbf{x}}_0,\boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\upsilon(\sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\right\|^2\right],\end{aligned} \tag{23}$$

  which guides gradient descent and updates network weights to minimize future prediction errors, iteratively refining the denoising model.

Algorithm 1 Training Algorithm of SADM

```
Input: Training data $\hat{\mathbf{x}}$, hyper-parameter $T$, hyper-parameters
    $\alpha_t$, random noise $\boldsymbol{\epsilon}_t$.
Output: The trained SADM parameters $\upsilon$.
repeat
    Sample $\hat{\mathbf{x}}_0 \sim q(\hat{\mathbf{x}})$ from training data $\hat{\mathbf{x}}$.
    Sample $t$ from set (i.e., $t \sim \mathrm{Uniform}(\{1, 2, \dots, \mathrm{T}\})$).
    Sample random Gaussian noise $\boldsymbol{\epsilon}_t$ (i.e., $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{E})$).

    Take gradient descent step on $\mathcal{L}_D$ referred to (23).
until convergence
return The parameters $\upsilon$.
```

In addition, during the training process, we integrate the early stopping process to mitigate overfitting of DM.

3) Fine-Tuning of Modules in SparseSBC-SADM: After separately training the encoder, decoder and DM, the third training stage typically involves fine-tuning these components to work coordinately. In detail, the previously fixed encoder ϕ and decoder θ during training the DM module are no longer constrained, allowing for simultaneous optimization of all three models. Notably, we still follow an alternative learning approach for fine-tuning, which first immobilizes RX and focuses on fine-tuning ϕ at TX. Afterward, we fix TX to optimize the DM and the decoder θ at the RX. Incorporating DM as part of the RX for SemCom and re-engaging the alternate learning self-critic method for fine-tuning can contribute to performance improvement without sacrificing the possibly unrealistic assumption of differentiability in channels. Specifically, this phase leverages $\hat{\mathbf{y}}$, which may contain noise or corruption and needs to be corrected before being effectively decoded. Therefore, training in this phase involves the sampling steps of DM, which are invoked to progressively denoise the data, as depicted in Algorithm 2.

Finally, the whole training details of SparseSBC-SADM are summarized in Algorithm 3.

V. Performance Evaluation

In this section, we evaluate the performance of SparseSBC-SADM with different metrics under AWGN,

Algorithm 2 Sampling Algorithm of SADM

```
Input: Noisy data $\hat{\mathbf{y}}$, hyper-parameter $T$, hyper-parameters $\alpha_t$.
Output: Denoised data $\hat{\mathbf{z}}$.
$\hat{\mathbf{y}}_T = \hat{\mathbf{y}}$.
for $t = T-1, \ldots, 1, 0$ do
    if $t > 0$ then
        Sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{E})$.
    else
        $\epsilon = 0$.
    end if
    Sample $\hat{\mathbf{y}}_{t-1}$ according to (9).
end for
return $\hat{\mathbf{z}} = \hat{\mathbf{y}}_0$.
```

Algorithm 3 The Training Algorithm of SparseSBC-SADM

```
Input: Batch size $B = 64$, initial learning rate $\mathrm{lr} = 1\mathrm{e}^{-4}$, self-critic samples $m = 5$, epoch $E_1 = 200, E_2 = 100, E_3 = 5$, semantic similarity metric $\Theta$, scale factor $\sigma$.
Output: Encoder parameter $\phi$, decoder parameter $\theta$, DM parameter $\upsilon$.
1:  for epoch $= 1 : \mathrm{E}_1$ do
2:      %Training TX
3:      For each batch, TX encodes each sample image into its sparse binary embedding, on the basis of frozen parameters $\theta$ at the RX.
4:      TX samples $m$ random samples according to (14), and sends the encoded bitstreams through the channel.
5:      RX decodes with the semantic policy gradient (17), thus yielding the objective function with reward (11a).
6:      TX takes gradient propagation towards $\phi$, and updates $\phi$ with lr (18).
7:      %Training RX
8:      For each batch, TX encodes $m$ image samples into its sparse binary embeddings, based on its trained policy.
9:      TX sends encoded bit streams through the channel.
10:     RX samples a sequence of random symbols from the channel with the Gaussian distribution, decodes and calculates the objective function with reward (11b).
11:     RX updates $\theta$ with Gaussian policy.
12: end for
13: for epoch $= 1 : \mathrm{E}_2$ do
14:     %Training DM
15:     For each batch, freeze encoder and decoder parameters $\phi, \theta$. Remove the wireless channel in the whole communication system and train SADM with Algorithm 1.
16: end for
17: for epoch $= 1 : \mathrm{E}_3$ do
18:     %Alternate Learning-Based Fine-Tuning
19:     For each batch, repeat steps 2 to 6 to fine-tune TX on the basis of frozen parameters $\theta$ of decoder and $\upsilon$ of DM at RX.
20:     For each batch, TX transmits $m$ quantified binary bit streams from encoder through the channel, and sends it into DM to sample denoised data with Algorithm 2.
21:     Repeat steps 7 to 11 to fine-tune decoder $\theta$ on the basis of frozen parameters $\phi$ at TX and $\upsilon$ of DM.
22:     DM update $\upsilon$.
23: end for
24: return The parameters $\phi, \theta, \upsilon$
```

PIF and 3GPP TR 38.901 TDL[2] channels respectively, and compare it with previous works like JSCC [10], Multi-Level Semantics-aware Communication system (MLSC-image) [13], SwinJSCC [30] and traditional "BPG+LDPC [28]".

A. Simulation Settings

We adopt the popular dataset Cifar-10 [51] containing $60,000$ RGB images with the fixed sizes of 32×32 and a high-resolution dataset DIV2K [52] with $1,000$ images. Other typical experimental settings are summarized in Table III.

In addition, we evaluate the performance in terms of metrics like the number of transmitted bits, PSNR, SSIM and FID. The number of transmitted bits evaluates transmission efficiency in channel transmission, and the other metrics evaluate the recovered images objectively and subjectively. In particular, PSNR [10] measures the ratio between the maximum possible power of signal and noise that corrupts the signal, and can be defined as

$$\mathrm{PSNR} = 10 \log_{10} \frac{\mathrm{MAX}^2}{\mathrm{MSE}} (\mathrm{dB}), \tag{24}$$

where $\mathrm{MSE} = \frac{1}{d_1 d_2} \sum_{i,j}^{d_1, d_2} \left(\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j} \right)^2$ denotes the mean squared-error, and MAX is the maximum value of pixels in the image of interest (i.e., 255 for 24-bit depth RGB images). Meanwhile, SSIM [44], which can be calculated as

$$\mathrm{SSIM} = \rho_l \big(\mathbf{I}, \hat{\mathbf{I}}\big)^{\lambda_1} \cdot \rho_c \big(\mathbf{I}, \hat{\mathbf{I}}\big)^{\lambda_2} \cdot \rho_s \big(\mathbf{I}, \hat{\mathbf{I}}\big)^{\lambda_3} \in [0, 1], \tag{25}$$

captures luminance, contrast and structural differences between images by ρ_l, ρ_c and ρ_s with exponential coefficients λ_1, λ_2 and λ_3. Besides, FID [45] quantifies the distance between two distributions of feature points extracted from images using a pre-trained Inception model. The FID score is calculated as

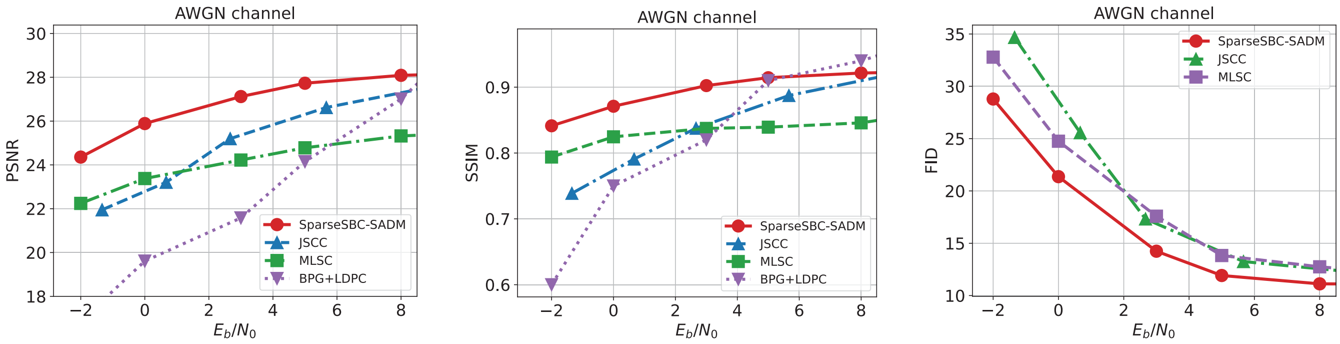
$$\mathrm{FID} = \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| + \mathrm{tr}\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2 \left(\boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2 \right)^{\frac{1}{2}} \right), \tag{26}$$

where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the mean vectors, while $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are the covariance matrices of the two sets of feature points. The FID score provides a comprehensive measure of the similarity between the generated images and the real dataset. Notably, we evaluate the performance (i.e., PSNR and SSIM) of the Cifar-10 dataset for all methods, and exclusively consider FID for comparisons within Deep Learning (DL)-based models, due to its subjectivity.

We conduct the performance comparison between SparseSBC-SADM and the following communication systems including

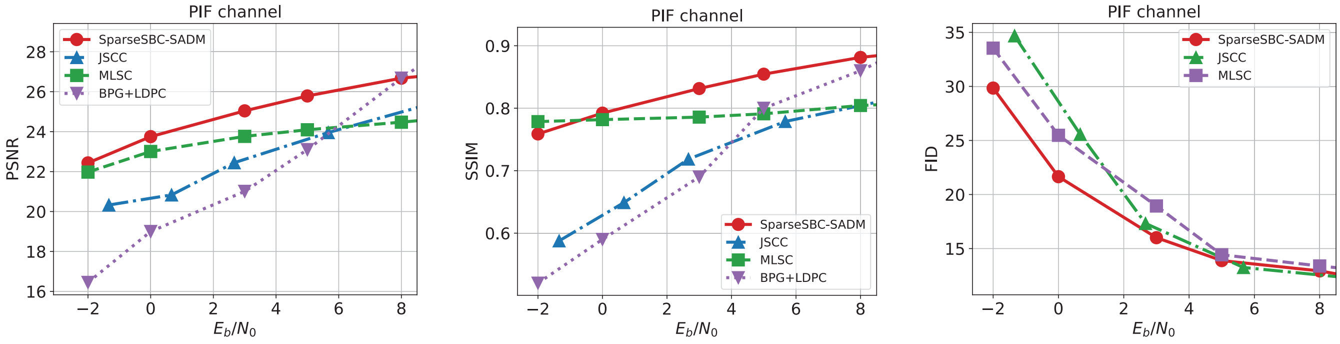
- **JSCC** [10]: JSCC scheme shares the same DNN structure as SparseSBC-SADM, with a joint training procedure of

[2] https://nvlabs.github.io/sionna/api/channel.wireless.html



(a) PSNR (b) SSIM (c) FID

Fig. 6. Performance comparison of SparseSBC-SADM with “BPG+LDPC”, JSCC and MLSC in terms of PSNR, SSIM and FID in AWGN channel of the Cifar-10 dataset.



(a) PSNR (b) SSIM (c) FID

Fig. 7. Performance comparison of SparseSBC-SADM with “BPG+LDPC”, JSCC and MLSC in terms of PSNR, SSIM and FID in PIF channel of the Cifar-10 dataset.

encoder and decoder. Notably, JSCC is a widely used training strategy in semantic communication.

- **MLSC-image** [13]: MLSC-image is a multi-level semantic communication system, and it fully extracts high-level or low-level semantics in images containing text semantics, segmentation semantics and spatial semantics.
- **SwinJSCC** [30]: SwinJSCC integrates the Swin Transformer [53] into deep joint source-channel coding, enhancing the efficiency and adaptability of data transmission in cognitive communication networks.
- **BPG+LDPC** [28]: The BPG codec[3] offers efficient image compression, while the LDPC codec provides powerful error correction capabilities. In the simulation, we follow standard 5G LDPC codes[4] with Quadrature Amplitude Modulations (QAM).

B. Numerical Results and Analysis

1) Comparative Analysis With Baselines: We first testify the performance of SparseSBC-SADM, and present the performance comparison with other baselines like “BPG+LDPC”, JSCC and MLSC in Fig. 6 and Fig. 7. It can be observed from Fig. 6 and Fig. 7 that, under different channels, SparseSBC-SADM achieves better performance compared to JSCC-scheme. Meanwhile, our method outperforms the traditional “BPG+LDPC” method under poor channel conditions. In other words, the “BPG+LDPC” method can not work stably in harsh environments while our method retains the performance to resist poor channels. Furthermore, SparseSBC-SADM leads to superior performance than MLSC, especially in the AWGN channel, which further demonstrates its advantages.

TABLE III
SIMULATION SETTINGS

Parameter	Value
Learning rate lr	10^{-4}
Batch size B	64
Sparse weight ε	0.1
Length of $\hat{\mathbf{x}}$ N	$5,000$
Length of $\hat{\mathbf{y}}$ M	$2,304$
Self-critic samples m	5
Scale factor σ	0.1
Training epochs $E_1; E_2; E_3$	$200; 100; 5$
Diffusion time T	$1,000$
Image dataset	Cifar-10; DIV2K

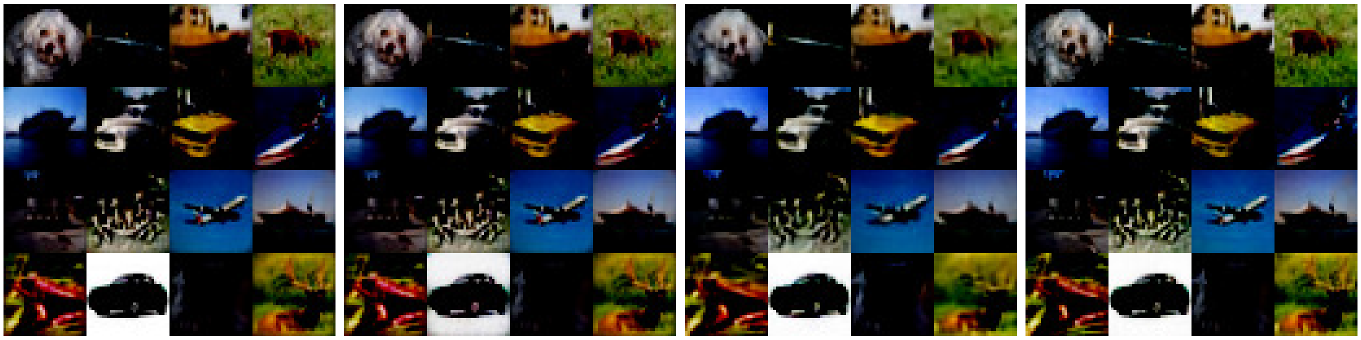
On the other hand, Table IV summarizes the average number of transmitted bits under different techniques in AWGN channels when SNR $= 10$ dB as an example. Specifically, “Float Resolution” refers to the precision of data transmission in terms of the amount of data being transmitted, which can determined by multiplying the amount of embedding data and corresponding float resolution. It can be observed that SparseSBC-SADM compresses every Cifar image to the fixed 625 bytes. As a comparison, JSCC needs approximately

[3] https://github.com/def-/libbpg
[4] https://github.com/NVlabs/sionna/tree/main/sionna/fec/ldpc

TABLE IV

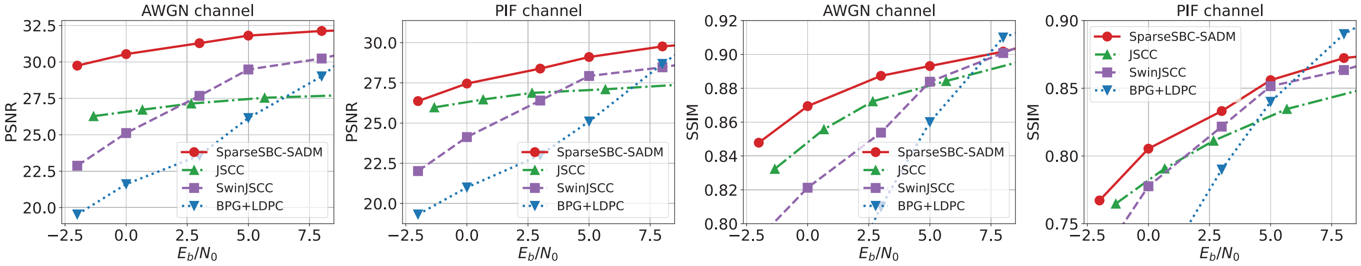
COMPARISON OF SPARSESBC-SADM WITH GENERAL SEMCOM SYSTEM JSCC AND MLSC IN TERMS OF PERFORMANCE AND THE BITS OF ONE IMAGE TO BE TRANSMITTED ($E_b/N_0 = 5$ dB)

	SparseSBC-SADM			SparseSBC	General JSCC					MLSC-image
Float Resolution (Bit)	1			1	1	4	8	16	32	32
No. Transmission Bytes	125	250	625	625	288	$1,152$	$2,304$	$4,608$	$9,216$	$1,536$
PSNR (dB)	24.62	26.17	28.26	27.75	12.83	18.64	26.98	27.06	27.07	24.78
SSIM (%)	83.36	88.53	92.83	92.04	54.83	76.37	88.20	88.48	88.48	83.92



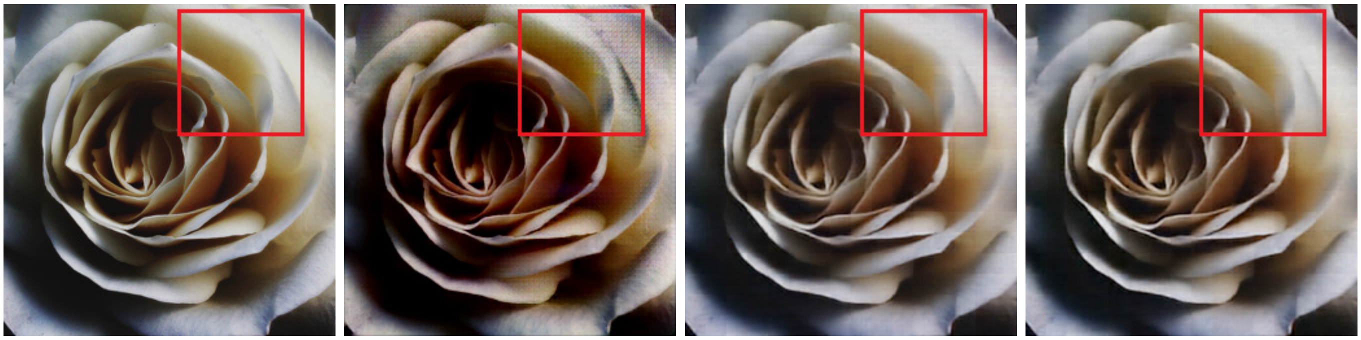
(a) Original/PSNR(dB) (b) JSCC/27.16 (c) SparseSBC/27.66 (d) SparseSBC-SADM/28.18

Fig. 8. Examples of visual comparison of Cifar-10 dataset under AWGN channel at $\text{SNR} = 10$ dB.



(a) PSNR (b) PSNR (c) SSIM (d) SSIM

Fig. 9. Performance comparison of SparseSBC-SADM, SwinJSCC, JSCC and “BPG+LDPC” on the DIV2K dataset under both AWGN and PIF channel conditions.



(a) Original/PSNR(dB)/SSIM(%) (b) JSCC/27.22/88.54 (c) SparseSBC/31.19/90.35 (d) SparseSBC-SADM/32.21/90.68

Fig. 10. Example of visual comparison of DIV2K dataset under AWGN channel at $\text{SNR} = 10$ dB.

$2,300$ bytes to reach a similar PSNR as SparseSBC-SADM. Meanwhile, SparseSBC-SADM outperforms MLSC in both transmitted bits and performance. In a nutshell, SparseSBC-SADM is rather communication efficient. In addition, the visual comparisons are illustrated in Fig. 8, taking AWGN channel with $\text{SNR} = 10$ dB as an example. We further investigate the performance differences between our method and baselines under the same coding rate CBR, and the comparison shown in Table V indicates that SparseSBC-SADM exhibits superior performance under the same CBR.

We also conduct evaluations on the DIV2K dataset to further assess the performance of our models under different channel environments, as illustrated in Fig. 9. To handle the high-resolution dataset, we have incorporated the sliding

TABLE V

COMPARISON OF SPARSESBC-SADM WITH GENERAL SEMCOM SYSTEM JSCC AND MLSC IN TERMS OF PERFORMANCE WHEN CODING RATE CBR $= 0.1$ ($E_b/N_0 = 5$ dB)

	SparseSBC-SADM	SparseSBC	DeepJSCC	MLSC
PSNR (dB)	28.28	27.79	23.51	24.94
SSIM (%)	92.81	92.10	78.30	85.35

TABLE VI

PERFORMANCE OF SPARSESBC-SADM IN 3GPP TR38.901 TDL CHANNEL MODEL

E_b/N_0(dB)	0	3	5	8	10
PSNR (dB)	23.40	24.62	25.33	26.15	26.57
SSIM (%)	77.18	82.07	84.61	87.52	88.49

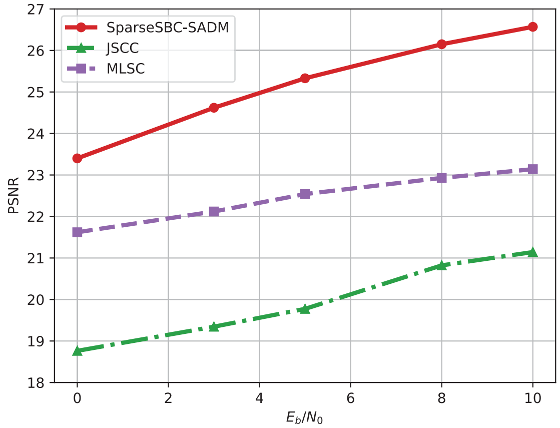


Fig. 11. Performance comparison between SparseSBC-SADM, JSCC and MLSC methods in the 3GPP-compliant TDL channel model.

window [54] to divide the image into segments for compression. The findings indicate that SparseSBC-SADM demonstrates superior performance across all E_b/N_0 than the JSCC system under both AWGN and PIF channels. Besides, SparseSBC-SADM outperforms SwinJSCC in terms of PSNR, especially under low E_b/N_0. Compared with "BPG+LDPC", it also shows increased stability and improved performance at low E_b/N_0 levels. Consistently, the visual contrast in the fine details of performance is showcased in Fig. 10, in which SparseSBC-SADM achieves better visual quality.

Additionally, to substantiate the stability of SparseSBC-SADM, we further conduct simulation experiments on the TDL channel model, as illustrated in Table VI and Fig. 11. Specifically, we approximate the channel as a single-path channel for JSCC and MLSC methods in simulation. It can be observed that the approximation error leads to more severe performance degradation of JSCC and MLSC. In contrast, SparseSBC-SADM manifests itself in the robustness and adaptability of the more real and complex channel.

2) Lightweight Network and SNR-Aware Denoising Module Evaluation: Fig. 12 shows the PSNR performance of the ablation experiment of SparseSBC with DM and without DM in AWGN and PIF channels as examples. The incorporation of DM demonstrates improved performance, particularly under conditions of low SNR ratios. The advantage is not as pronounced in high SNR scenarios, since the transmitted data is not significantly affected by channel noise. Based on the simulation results, we choose $\Delta\mathbf{S} = \Delta\mathbf{PSNR}$, and the activation of DM is triggered when the value of threshold k is set to 0.3. When $\Delta\mathbf{S}$ surpasses k (i.e., $\Delta\mathbf{PSNR} \geq 0.3$), the intervention of DM can enhance the quality and reliability of data transmission; otherwise, a de-activation of DM helps save computational resources. Furthermore, the shaded area represents SparseSBC-SADM, and it can be

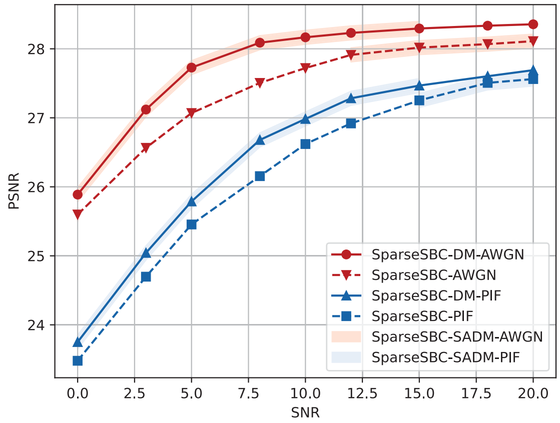


Fig. 12. Ablation experiment of SparseSBC with DM and without DM, take PSNR as an example.

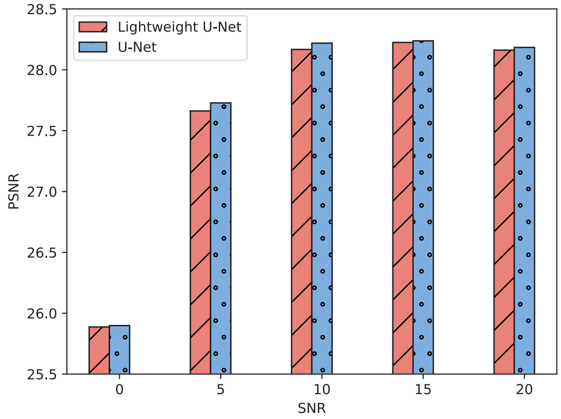


Fig. 13. Comparison of lightweight U-Net and ordinary U-Net in performance.

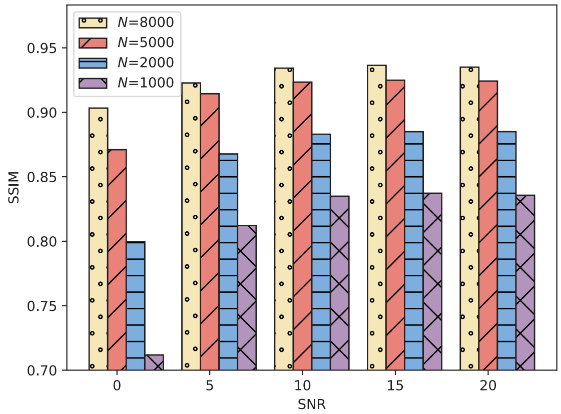
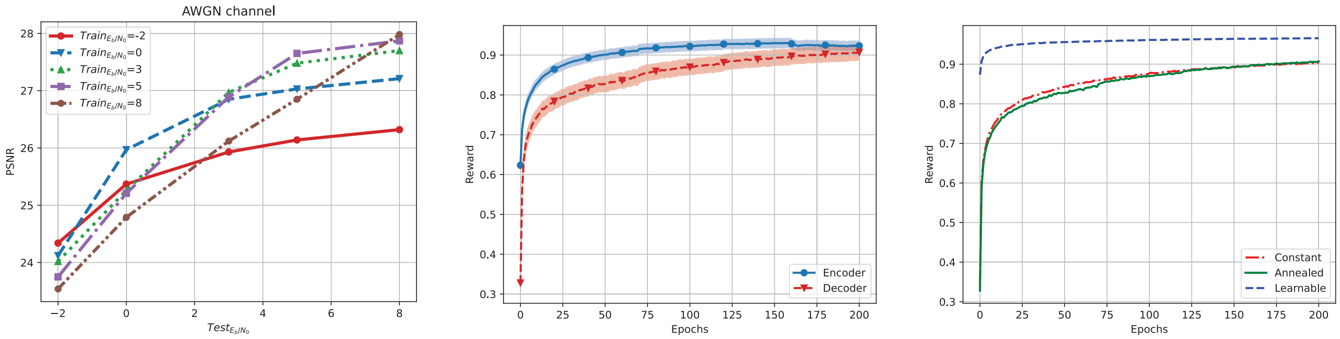


Fig. 14. The SSIM performance of different quantization length N of $\hat{x}$ in SparseSBC-SADM system.



(a)

(b)

(c)

Fig. 15. (a) Testing performance of models trained under some specific channel condition. (b) The training reward of the encoder and decoder. (c) The sensitivity of the scale factor σ.

TABLE VII
COMPARISON OF LIGHTWEIGHT U-NET AND STANDARD U-NET(SNR=10DB)

Model	Parameter	FLOPs	PSNR
U-Net	8.80MB	120.24G	**27.72dB**
Lightweight U-Net	**4.02MB**	**114.07G**	27.66dB

observed from Fig. 12 that under PIF channel, DM is activated when SNR is over 15 dB. Conversely, under AWGN channel, DM is activated when SNR does not exceed 12 dB. Besides, at elevated SNR, the channel noise is sufficiently low, which paradoxically reduces the efficacy of DDM in learning and fitting the noise characteristics. In these instances, the conditional activation strategy for DM can facilitate the balance of the trade-off between the desired performance and computational efficiency. As a result, the performance improvement of SparseSBC-SADM is more pronounced under low SNR conditions, reflecting our deliberate approach to activating DM under such circumstances with potentially significant performance benefits.

Additionally, we also compare the PSNR performance and parameters of lightweight U-Net and ordinary U-Net in Fig. 13. It can be observed from Fig. 13, at intermediate SNR levels of 5 dB to 10 dB, a slight performance drop is observed for the lightweight module, while in other scenarios the performance loss is not significant. Besides, through the lightweight design, the parameter count of U-Net has been significantly reduced from 8.80 MB to 4.02 MB, representing a decrease of storage resource usage over 50%. Moreover, this lightweight design also leads to a decrease in Floating Point Operations (FLOPs), which is shown in Table VII. Specifically, the computational complexity has been reduced from 120.24 GFLOPs to 114.07 GFLOPs. Despite this reduction, the performance only marginally degrades from 27.72 dB to 27.66 dB, indicating that the lightweight U-Net maintains comparable performance while offering substantial improvements in computational efficiency. The simulation results indicate the effectiveness of the lightweight module in maintaining performance with reduced complexity.

3) Performance Sensitivity Evaluation: We perform simulation experiments on different lengths of quantized bit streams N and give the corresponding results in Fig. 14 and Table IV. As Fig. 14 shows, as the value of N increases, the SSIM performance of SparseSBC-SADM improves, especially in poor channel conditions. PSNR and SSIM of different N in Table IV demonstrate the stability of SparseSBC-SADM in small transmission bytes, and validate the superiority of SparseSBC-SADM. Different quantization lengths yield varying performance, while longer length typically correlates with improved transmission performance. When the value of N is set to $5,000$, the performance slightly surpasses that of JSCC methods, with a substantial reduction from $9,216$ to 625 in the number of transmitted bytes. Furthermore, this value is also the one utilized in practical applications.

Next, we conduct experiments to assess the robustness of our model under various channel conditions. Each curve in Fig. 15(a) is derived from training on a particular channel condition, and the performance of each is assessed on test datasets across a range of E_b/N_0 values. When the channel parameters during training do not perfectly align with the actual conditions, the performance of model remains relatively stable without significant fluctuations. Furthermore, to demonstrate the convergence of alternate learning, we illustrate the progression of rewards for the encoder and decoders across training epochs in Fig. 15(b). It can observed that both the encoder and decoder ultimately obtain convergence. We also explore the sensitivity of $\mathbf{\Sigma}$ discussed in Section IV-D, as shown in Fig. 15(c). It can be observed that a learnable $\mathbf{\Sigma}$ which is determined by scale factor σ shows superior performance in the training period, which outperforms the constant and annealed settings.

In addition to the thorough analysis of our self-critic method in various scenarios, we further conduct a comparative study with the Proximal Policy Optimization (PPO)-based algorithm [55], a well-established RL. Our results in Fig. 16(a) indicate that the self-critic method matches the performance of the PPO-based algorithm, but exceeds its efficiency. Fig. 16(b) records transmitted bits for Cifar-10 test images, and clearly demonstrates the sparsity of transmitted bits, since "1"s account for less than 20% of the whole transmitted bits. Besides, Fig. 16(c) demonstrates the relationship between the sparse weight ε and the performance of our model, indicating that higher weights lead to better performance but result in less sparse transmission of bits. Notably, the experiments in this part have been conducted with a fixed random seed value (i.e., 1985) to ensure reproducibility and consistency across different runs.

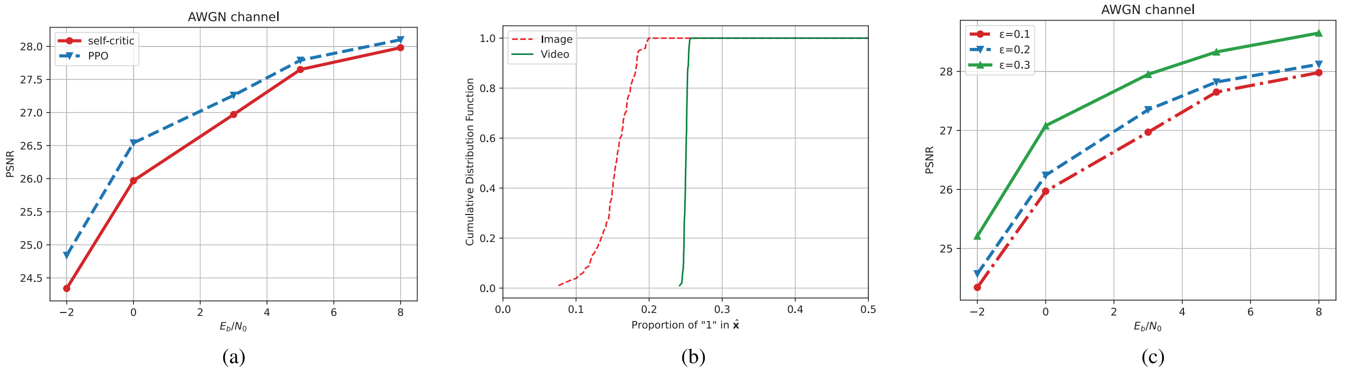
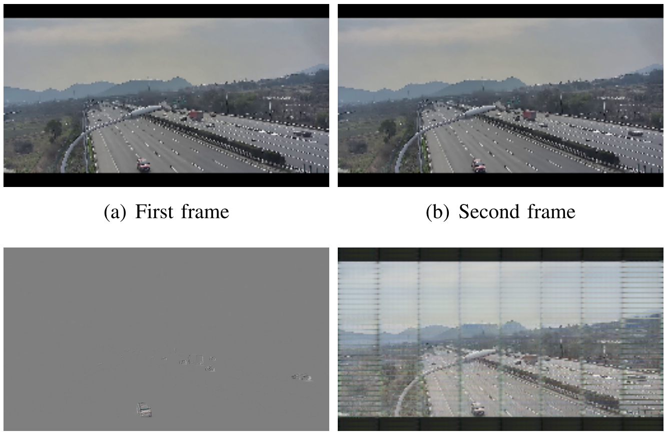


Fig. 16. (a) Performance comparison of self-critic scheme and PPO scheme. (b) The sparsity of transmitted bits in SparseSBC-SADM for image and video transmission. (c) Model performance under different configurations of the sparse weight ε.


(a) First frame

(b) Second frame

(c) Differential image

(d) Reconstructed second frame

Fig. 17. Examples of differential images extracted by temporal difference from the video clip and reconstructed video frame.

4) Extension to Video Transmission: SparseSBC-SADM can be easily adapted to support video transmission by converting a video to a sequence of images [16], [17]. Moreover, as depicted in Fig. 17, subsequent images in a video can be obtained by further computing the difference in the corresponding pixels between the two frames and taking the absolute values. This sequence of differential images, which only records the moving parts in a video, is easier to compress due to the inherent sparsity. Fig. 17 illustrates the process of temporal difference for video transmission and presents the preliminary result to reconstruct the second frame based on temporal difference-involved SparseSBC-SADM, which demonstrates the stable performance. Specifically, when $E_b/N_0 = 5$ dB, our system achieves a PSNR of 29.57 dB and the SSIM reaches a value of 87%. Furthermore, consistent with the image transmission, Fig. 16(b) also unveils a sparsity of less than 30% for the video transmission.

VI. Conclusion

In this paper, we have proposed a sparse SemCom system for visual transmission, named SparseSBC-SADM, which capably learns the DNN-based encoder and decoder deployed on TX and RX alternately, so as to adapt to the non-differentiable channel. In particular, a "self-critic" scheme has been leveraged into the training procedure to guarantee a stable process. In addition, by extracting a set of semantic bases and implementing binary quantization, semantic information is converted into sparse bit streams, thus effectively bridging the potential combination between semantic communications and compressive sensing. Besides, in order to reduce impacts from wireless channel noise, an SNR-aware denoising module inspired by DDPM is employed at the receiver, wherein we specially incorporate a lightweight U-Net model. Moreover, SparseSBC-SADM introduces a performance-driven gating mechanism to adaptively determine the activation of the denoising module, thus improving computational efficiency. Extensive simulation results validate that for visual transmission, SparseSBC-SADM outperforms "BPG+LDPC" and JSCC schemes with efficiency and effectiveness under various channel conditions, and demonstrates its robustness under poor channel conditions. In the future, we will extend our research to full video processing, high-fidelity simulations, and even real-world deployment, with the anticipation of providing a broader spectrum of analyses.

References

[1] S. Tong, X. Yu, R. Li, K. Lu, Z. Zhao, and H. Zhang, "Alternate learning based sparse semantic communications for visual transmission," in *Proc. IEEE 34th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Toronto, ON, Canada, Sep. 2023, pp. 1–6.

[2] R. Li, Z. Zhao, X. Xu, F. Ni, and H. Zhang, "The collective advantage for advancing communications and intelligence," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 96–102, Aug. 2020.

[3] K. Lu et al., "Rethinking modern communication from semantic coding to semantic communication," *IEEE Wireless Commun.*, vol. 30, no. 1, pp. 158–164, Feb. 2023.

[4] Z. Lu et al., "Semantics-empowered communications: A tutorial-cum-survey," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 1, pp. 41–79, 1st Quart., 2024.

[5] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, 2021.

[6] H. Xie and Z. Qin, "A lite distributed semantic communication system for Internet of Things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, Jan. 2021.

[7] Q. Zhou, R. Li, Z. Zhao, C. Peng, and H. Zhang, "Semantic communication with adaptive universal transformer," *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 453–457, Mar. 2022.

[8] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2434–2444, Aug. 2021.

[9] G. Shi et al., "A new communication paradigm: From bit accuracy to semantic fidelity," 2021, *arXiv:2101.12649*.

[10] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sep. 2019.

[11] D. B. Kurka and D. Gündüz, “Bandwidth-agile image transmission with deep joint source-channel coding,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8081–8095, Dec. 2021.

[12] A. Prakash, N. Moran, S. Garber, A. Dilillo, and J. Storer, “Semantic perceptual image compression using deep convolution networks,” in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Apr. 2017, pp. 250–259.

[13] Z. Zhang, Q. Yang, S. He, M. Sun, and J. Chen, “Wireless transmission of images with the assistance of multi-level semantic information,” in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Zhejiang, China, Oct. 2022, pp. 1–6.

[14] G. Toderici et al., “Full resolution image compression with recurrent neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5435–5443.

[15] O. Rippel and L. Bourdev, “Real-time adaptive image compression,” in *Proc. ICML*, Sydney, NSW, Australia, Jan. 2017, pp. 2922–2930.

[16] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, “Wireless semantic communications for video conferencing,” *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 230–244, Jan. 2023.

[17] S. Wang et al., “Wireless deep video semantic transmission,” *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 214–229, Jan. 2023.

[18] D. B. Kurka and D. Gündüz, “DeepJSCC-f: Deep joint source-channel coding of images with feedback,” *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, May 2020.

[19] X. Yuan and R. Haimi-Cohen, “Image compression based on compressive sensing: End-to-end comparison with JPEG,” *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 2889–2904, Nov. 2020.

[20] M. Ke, Z. Gao, Y. Wu, X. Gao, and R. Schober, “Compressive sensing-based adaptive active user detection and channel estimation: Massive access meets massive MIMO,” *IEEE Trans. Signal Process.*, vol. 68, pp. 764–779, 2020.

[21] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *Proc. ICML*, Lille, France, Jan. 2015, pp. 2256–2265.

[22] J. Song et al., “Denoising diffusion implicit models,” in *Proc. ICLR*, Vienna, Austria, May 2021, pp. 1–22.

[23] T. Wu et al., “CDDM: Channel denoising diffusion models for wireless semantic communications,” *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 11168–11183, Sep. 2024.

[24] X. Niu, X. Wang, D. Gündüz, B. Bai, W. Chen, and G. Zhou, “A hybrid wireless image transmission scheme with diffusion,” in *Proc. IEEE 24th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Shanghai, China, Sep. 2023, pp. 86–90.

[25] E. Grassucci, S. Barbarossa, and D. Comminiello, “Generative semantic communication: Diffusion models beyond bit recovery,” 2023, *arXiv:2306.04321*.

[26] S. Kudugunta et al., “Beyond distillation: Task-level mixture-of-experts for efficient inference,” in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2021, pp. 3577–3599.

[27] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Munich, Germany, 2015, pp. 234–241.

[28] T. Richardson and S. Kudekar, “Design of low-density parity check codes for 5G new radio,” *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 28–34, Mar. 2018.

[29] N. Farsad, M. Rao, and A. Goldsmith, “Deep learning for joint source-channel coding of text,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 2326–2330.

[30] K. Yang, S. Wang, J. Dai, X. Qin, K. Niu, and P. Zhang, “SwinJSCC: Taming swin transformer for deep joint source-channel coding,” *IEEE Trans. Cognit. Commun. Netw.*, early access, Jul. 8, 2024, doi: 10.1109/TCCN.2024.3424842.

[31] G. K. Wallace, “The JPEG still picture compression standard,” *Commun. ACM*, vol. 34, no. 4, pp. 30–44, Apr. 1991.

[32] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[33] F. Bellard. (2015). *Better Portable Graphics (BPG) Image Format*. Accessed: Aug. 30, 2024. [Online]. Available: https://bellard.org/bpg/

[34] K. Yang, S. Wang, J. Dai, K. Tan, K. Niu, and P. Zhang, “WITT: A wireless image transmission transformer for semantic communications,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[35] L. Sun, Y. Yang, M. Chen, C. Guo, W. Saad, and H. V. Poor, “Adaptive information bottleneck guided joint source and channel coding for image transmission,” *IEEE J. Sel. Areas Commun.*, vol. 41, no. 8, pp. 2628–2644, Aug. 2023.

[36] A. Vaswani et al., “Attention is all you need,” in *Proc. NeurIPS*, Long Beach, CA, USA, Jun. 2017, pp. 5998–6008.

[37] M. Song, N. Ma, C. Dong, X. Xu, and P. Zhang, “Deep joint source-channel coding for wireless image transmission with adaptive models,” *Electronics*, vol. 12, no. 22, p. 4637, Nov. 2023.

[38] V. Kravets and A. Stern, “Progressive compressive sensing of large images with multiscale deep learning reconstruction,” *Sci. Rep.*, vol. 12, no. 1, p. 7228, May 2022.

[39] Z. Gan, X. Chai, J. Zhang, Y. Zhang, and Y. Chen, “An effective image compression–encryption scheme based on compressive sensing (CS) and game of life (GOL),” *Neural Comput. Appl.*, vol. 32, no. 17, pp. 14113–14141, Sep. 2020.

[40] A. H. Estiri, M. R. Sabramooz, A. Banaei, A. H. Dehghan, B. Jamialahmadi, and M. J. Siavoshani, “A variational auto-encoder approach for image transmission in wireless channel,” 2020, *arXiv:2010.03967*.

[41] J. Ho, A. N. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Proc. NeurIPS*, vol. 33, Jan. 2020, pp. 6840–6851.

[42] M. Letafati, S. Ali, and M. Latva-Aho, “Conditional denoising diffusion probabilistic models for data reconstruction enhancement in wireless communications,” 2023, *arXiv:2310.19460*.

[43] *Study on Channel Model for Frequencies From 0.5 to 100 GHz, Version 16.1.0*, document TR 38.901, 3GPP, Dec. 2019. [Online]. Available: http://www.3gpp.org/DynaReport/38901.htm

[44] U. Sara, M. Akter, and M. S. Uddin, “Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study,” *J. Comput. Commun.*, vol. 7, no. 3, pp. 8–18, Jan. 2019.

[45] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” in *Proc. ICML*, Sydney, NSW, Australia, Jan. 2017, pp. 214–223.

[46] Z. Lu, R. Li, M. Lei, C. Wang, Z. Zhao, and H. Zhang, “Self-critical alternate learning based semantic broadcast communication,” *IEEE Trans. Commun.*, vol. 72, no. 3, pp. 1533–1546, Mar. 2024.

[47] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical sequence training for image captioning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1179–1195.

[48] D. Bertsimas et al., “Simulated annealing,” *Stat. Sci.*, vol. 8, no. 1, pp. 10–15, 1993.

[49] K. Lu, R. Li, X. Chen, Z. Zhao, and H. Zhang, “Reinforcement learning-powered semantic communication via semantic similarity,” 2021, *arXiv:2108.12121*.

[50] A. Q. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” in *Proc. ICML*, Jul. 2021, pp. 8162–8171.

[51] A. Krizhevsky et al., “Learning multiple layers of features from tiny images,” in *Handbook of Systemic Autoimmune Diseases*, vol. 1, Amsterdam, The Netherlands: Elsevier, Apr. 2009.

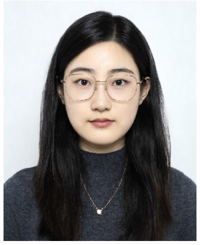
[52] E. Agustsson and R. Timofte, “NTIRE 2017 challenge on single image super-resolution: Dataset and study,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 126–135.

[53] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[54] A. Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition at scale,” in *Proc. ICLR*, May 2021, pp. 1–22.

[55] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017, *arXiv:1707.06347*.

Siyu Tong received the B.E. degree from Zhejiang University, Hangzhou, China, where she is currently pursuing the M.E. degree with the College of Information and Electronic Engineering. Her current research interests include semantic communications and deep learning in wireless networks.



Xiaoxue Yu (Student Member, IEEE) received the B.E. degree in communication engineering from Xidian University, Xi'an, China. She is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. Her research interests include communications in distributed learning and multiagent reinforcement learning.

Rongpeng Li (Senior Member, IEEE) received the B.E. degree from Xidian University, Xi'an, China, in June 2010, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in June 2015. From August 2015 to September 2016, he was a Research Engineer with the Wireless Communication Laboratory, Huawei Technologies Company Ltd., Shanghai, China. He was a Visiting Scholar with the Department of Computer Science and Technology, University of Cambridge, Cambridge, U.K., from February 2020 to August 2020. He is currently an Associate Professor with the College of Information Science and Electronic Engineering, Zhejiang University. His current research interests include networked intelligence for communications evolving.

Kun Lu received the B.E. degree from Shanxi University in 2020 and the M.E. degree from Zhejiang University in 2023. He is currently working on large generative models with Huawei Technologies Company Ltd., Hangzhou, China. His research interests include machine learning and artificial intelligence.

Zhifeng Zhao (Senior Member, IEEE) received the B.E. degree in computer science, the M.E. degree in communication and information systems, and the Ph.D. degree in communication and information systems from the PLA University of Science and Technology, Nanjing, China, in 1996, 1999, and 2002, respectively. From 2002 to 2004, he was as a Post-Doctoral Researcher with Zhejiang University, Hangzhou, China. From 2005 to 2006, he was as a Senior Researcher with the PLA University of Science and Technology. He was an Associate Professor with the College of Information Science and Electronic Engineering, Zhejiang University, from 2006 to 2019. He is currently with Zhejiang Lab, Hangzhou, as the Chief Engineering Officer; and with Zhejiang University, as an Adjunct Professor. His research interests include software-defined networks, wireless networks in 6G, computing networks, and collective intelligence.

Honggang Zhang (Fellow, IEEE) is currently a Professor with the Faculty of Data Science, City University of Macau, Macau, China. He was a Professor with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. He was an Honorary Visiting Professor with the University of York, York, U.K., and the International Chair Professor of Excellence with the Universite Europeenne de Bretagne and Supélec, France. He has co-authored and edited two books: *Cognitive Communications: Distributed Artificial Intelligence (DAI), Regulatory Policy and Economics, Implementation* (John Wiley and Sons) and *Green Communications: Theoretical Fundamentals, Algorithms, and Applications* (CRC Press). His research interests include cognitive radio networks, semantic communications, green communications, machine learning, artificial intelligence, intelligent computing, and internet of intelligence.

Dr. Zhang is a co-recipient of the 2021 IEEE Communications Society Outstanding Paper Award and the 2021 IEEE Internet of Things Journal Best Paper Award. He was the Founding Chief Managing Editor of *Intelligent Computing* and *Science Partner Journal*. He was the leading Guest Editor of the Special Issues on Green Communications of the *IEEE Communications Magazine*. He served as a Series Editor for the *IEEE Communications Magazine* (Green Communications and Computing Networks Series) from 2015 to 2018 and the Chair for the Technical Committee on Cognitive Networks of the IEEE Communications Society from 2011 to 2012. He is the Associate Editor-in-Chief of *China Communications*.