

Diffusion-Based Adaptive Channel Denoising for Deep Joint Source-Channel Coding in Semantic Communications

Long V. Nguyen, *Student Member, IEEE*, Hans-Jürgen Zepernick, *Senior Member, IEEE*, Nguyen-Son Vo, Hyundong Shin, *Fellow, IEEE*, and Trung Q. Duong, *Fellow, IEEE*

Abstract—Recent advancements in deep learning have led to the emergence of deep joint source-channel coding (DeepJSCC) as a powerful technique for robust and efficient communication over noisy channels. This approach is regarded as a key enabling technology for semantic communication (SemCom), a paradigm that ensures only relevant information for the underlying tasks is transmitted at the receiver. Currently, diffusion models have demonstrated remarkable success in generative tasks, where they learn a denoising process to restore the original signal. This paper introduces an adaptive denoising diffusion model (ADDM) for DeepJSCC in SemCom, which dynamically learns the channel denoising process based on the corrupted signal and channel state information (CSI). By modifying both forward and reverse diffusion processes and integrating CSI through multi-head attention mechanisms into the U-Net model, our system tailors the diffusion trajectory to the specific noise level, leading to more efficient and accurate signal reconstruction. Extensive simulations on ADE20K and ImageNet datasets demonstrate that our adaptive approach significantly outperforms specialized, fixed-SNR DeepJSCC baselines across a wide SNR range, particularly at low signal-to-noise ratio (SNR) regimes below approximately 5 dB. Additionally, our model does not rely heavily on CSI, which is critical in scenarios where channel information is unavailable or CSI estimation is imperfect.

Index Terms—Deep joint source-channel coding, semantic communication, adaptive channel denoising, diffusion model.

I. INTRODUCTION

Mobile communication systems have evolved dramatically, from the simple analog audio of the first generation to the high-speed, low-latency multimedia services of the fifth generation (5G). Although 5G has used advanced technologies, such as massive multiple-input multiple-output (MIMO) and millimeter wave communications, to meet most current demands, its capabilities are insufficient to handle the intelligent applications beyond-5G (B5G) era. Future B5G networks will need to support multimedia services, including virtual and augmented reality (VR/AR), autonomous driving, and telemedicine. The technical requirements for these applications, such as 10 Gb/s/m³ traffic density, 1 Tb/s data rates, and latency under 0.1 ms, are far beyond what 5G can currently offer [1]. For many decades, traditional communication architectures have relied on Shannon's separation theorem [2], which advocates for independent optimization of source coding (data compression) and channel coding (error correction) components without any loss of optimality. This principle has been the foundation for virtually all modern digital communication systems by allowing each component to be addressed separately. However, the capacity guaranteed by Shannon's theorem is optimal theoretically based on assumptions of infinite block length and unlimited processing delay, which are impractical. As a result, this traditional separation-based approach often fails to deliver the optimal end-to-end performance, particularly in B5G networks, where latency and data rate are critical.

Recently, the advent of deep learning has facilitated a transition toward end-to-end optimized systems, leading to the emergence of deep joint source-channel coding (DeepJSCC) [3]. This scheme is no longer dependent on the modular design of separation-based schemes. Instead, it employs a single, unified neural network, typically structured as an autoencoder, to learn an end-to-end mapping directly from the source data (e.g., image pixels) to the complex-valued symbols transmitted over the channel. By training a single neural network to jointly optimize both compression and error correction, DeepJSCC has demonstrated remarkable performance with gradual signal quality degradation under decreasing signal-to-noise ratio (SNR) values and outperformed traditional separation-based methods. This promising data-driven approach aligns

L. V. Nguyen is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada (e-mail: lnguyen@mun.ca).

H.-J. Zepernick is with the Blekinge Institute of Technology, Sweden (e-mail: hans-jurgen.zepernick@bth.se).

N.-S. Vo is with the Institute of Fundamental and Applied Sciences, Duy Tan University, Ho Chi Minh City, 70000, Vietnam, and also with the Faculty of Electrical-Electronic Engineering, Duy Tan University, Da Nang, 50000, Vietnam (e-mail: vonguyenson@duytan.edu.vn).

H. Shin is with the Department of Electronics and Information Convergence Engineering, Kyung Hee University, 1732 Deogyong-daero, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Republic of Korea (e-mail: hshin@khu.ac.kr).

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada, and with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, U.K., and also with the Department of Electronic Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 17104, South Korea (e-mail: tduong@mun.ca).

This paper has been submitted in part for presentation at IEEE 105th Vehicular Technology Conference: VTC2027-Spring, Hamburg, Germany, June 2027.

This work was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109 and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941. The work of H. Shin was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2025-00556064), and by the Ministry of Science and ICT (MSIT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2025-RS-2021-II212046), supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation)

Corresponding authors are Trung Q. Duong and Hyundong Shin.

closely with the goals of semantic communication (SemCom), a paradigm shift beyond the classical Shannon framework that prioritizes the successful transmission of underlying meaning, or the effective execution of a downstream task at the receiver, over bit-perfect accuracy [4], [5]. By training an end-to-end system to optimize for perceptual quality metrics rather than bitwise fidelity, DeepJSCC serves as a powerful and practical enabling technology for realizing the principles of SemCom [6]. It paves the way for more intelligent, efficient, and resilient communication systems, which are envisioned as a cornerstone of sixth-generation (6G) wireless networks [7], [8].

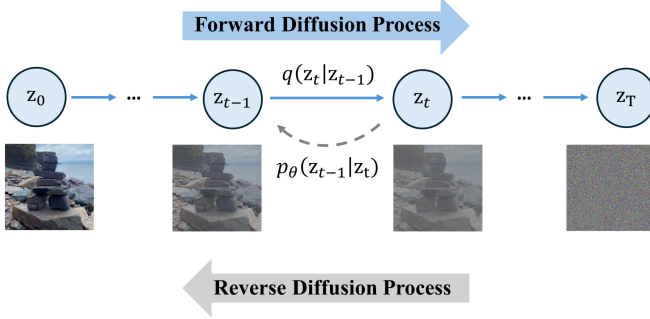


Fig. 1: An overview of the diffusion probabilistic model with two main processes in [9]. The Gaussian noise is added progressively to the original data in the forward diffusion process following a Markov chain with transition kernel $q(z_t|z_{t-1})$. Transitions of this chain are then learned to reconstruct the original in a reverse process with learnable transition kernel $p_\theta(z_{t-1}|z_t)$.

Furthermore, the field of generative modeling has witnessed a breakthrough with the development of denoising diffusion probabilistic models (DDPMs) [9], or diffusion models (DMs) for short. These models achieve unprecedented synthesis quality by learning to reverse a gradual noising process. An overview of a DM is shown in Fig. 1. A DM systematically corrupts data by adding Gaussian noise over multiple steps (the forward process) and then trains a neural network to denoise the data step-by-step for the original signal recovery (the reverse process). The inherent strength of DMs lies in this iterative refinement, which models complex, high-dimensional data distributions through a series of simple denoising steps, making them particularly well-suited for signal processing tasks.

A. Related Work

Semantic communications have attracted intensive attention from the research community recently [3]–[6], [8], [10]–[14]. A key enabling technology for this shifting paradigm is DeepJSCC, first introduced for wireless image transmission [3]. Since this pioneering work, DeepJSCC paradigm has been extended and refined in numerous ways. Recognizing the demand for flexibility in dynamic environments, the authors in [10] developed rate-adaptive DeepJSCC schemes that can adjust their bandwidth utilization based on channel conditions or the complexity of the source image, using techniques like the Gumbel-Softmax trick to enable end-to-end differentiability.

The work in [11] explored the incorporation of channel output feedback, where information about the received signal is sent back to the transmitter to iteratively refine the transmission, demonstrating significant gains in reconstruction quality or reductions in average delay. Furthermore, the framework in [12] integrated the attention mechanism to adapt to various channel conditions by dynamically adjusting the compression ratio according to the SNR. A deep refinement-based JSCC was proposed in [13] to adapt instantaneous channel state information (CSI), which ensures the encoding strategy is optimized with varying channel conditions. Neural image compression was employed in [14] with an accurate and adaptable entropy model to efficiently extract semantic bitstreams for digital semantic communication systems. An innovative semantic adaptive feature extraction (SAFE) network was introduced in [15] to dynamically generate and fuse multiple sub-semantics adapted to diverse channel scenarios. Despite extensive applications of deep learning methods for DeepJSCC in SemCom, existing research has not yet harnessed generative models to enhance further the signal processing and reconstruction, especially diffusion models.

Diffusion models have been investigated immensely for a wide range of image and signal restoration tasks [16], [17]. These applications reframe restoration as a conditional generation problem, where the degraded signal (e.g., a blurry or low-resolution image) guides the reverse diffusion process to reconstruct a high-quality counterpart. A typical work in [18] demonstrated photo-realistic results in image super-resolution by conditioning the reverse process on a low-resolution input. The authors in [19] introduced a denoising diffusion restoration model (DDRM), a framework that established a versatile, unsupervised approach for solving various linear inverse problems, including deblurring and inpainting, without requiring task-specific training. An additional neural network was proposed in [20] to incorporate spatial conditioning controls into large pre-trained diffusion models, enabling effective control over image generation and enhancement. Beyond images, these principles have been successfully applied to audio restoration for tasks like speech enhancement and music restoration [21]. For example, the works in [22], [23] successfully designed a progressive transformation between clean and distorted speech signals, leveraging the advancements in image restoration to investigate the design space for speech enhancement. In [24], a data-driven generative audio model was presented, which was used for solving various inverse problems in audio processing and music restoration. This demonstrated an impressive capability in reversing complex degradation processes and reconstructing high-fidelity signals. Nevertheless, there is limited research on alleviating channel noise and signal reconstruction for communication systems.

Given their potential denoising capabilities, it is natural to investigate DMs as a tool for mitigating channel noise in semantic communications. Currently, the most direct application of DMs is as a post-processing module to enhance the output of an existing communication system [25]–[29]. In this setup, a conditional DM is utilized to refine a noisy or distorted reconstruction. The received signal, after initial decoding, serves as a conditioning input to guide a modified

TABLE I: Summary of Related Studies Versus Our Work

Studies	Forward Process Modified	Reverse Process Design	CSI Integration	Adaptive Channel Denoising	End-to-end Trained DeepJSCC
[28]	×	×	✓	×	×
[29]	×	×	✓	×	✓
[30]	✓	×	✓	✓	✓
[31]	×	✓	✓	✓	✓
[32]	×	✓	✓	✓	✓
[33]	×	×	×	✓	×
[34]	×	×	✓	✓	✓
Our work	✓	✓	✓	✓	✓

reverse diffusion process. The DM effectively learns a “noisy-to-clean” mapping to refine the distorted reconstruction and improve perceptual quality. While effective, these approaches consider the DM a separate component, decoupled from the end-to-end communication process. The JSCC encoder is not trained to produce a latent representation that is optimized for the DM’s generative reconstruction capabilities, leading to suboptimal performance.

Recent studies utilize DMs to learn the statistical properties of the wireless channel itself, treating the DM as a powerful generative prior. For instance, DM has been successfully employed for complex tasks such as MIMO channel estimation [30], where they learn the channel distribution in a sparse domain to provide accurate estimates with low complexity. The work in [31] utilized DM to learn a differentiable channel model from samples, which is a critical challenge for training end-to-end channel codes when the true channel model is unknown or non-differentiable. These approaches highlight the power of DMs for modeling complex channel physics but do not directly address the end-to-end source signal reconstruction problem that is central to DeepJSCC.

To exploit the denoising mechanism, the works in [32]–[36] have adapted DMs directly into the DeepJSCC pipelines, where a DM is used at the receiver to denoise the latent features before they are passed to the source decoder. The authors in [32] considered the signal transmission through the wireless channel as the forward process in DM, and a reverse process learned to denoise Gaussian noise under bandwidth constraints. In [33], a joint semantic equalizer and denoiser was introduced at the receiver based on an inverse problem solved by latent diffusion models to eliminate the channel impact. The work in [34] proposed a channel denoising diffusion model after channel equalization to learn the channel input signal distribution for channel denoising. Additionally, in [35] and [36], a latent diffusion model was utilized to intelligently design a denoiser for transmitted semantic latent representations. In this way, channel noises are modeled in a diffusion manner and progressively mitigated by the inverse diffusion process. However, leveraging noise as a constructive element rather than an unwanted factor

and seamlessly integrating wireless channel impairments into both diffusion processes to enhance the signal reconstruction remain underexplored. A comparison of the related diffusion-based methods is summarized in Table I, highlighting their contributions and limitations.

B. Motivation and Contributions

In this article, we introduce an adaptive denoising diffusion model (ADDM) for DeepJSCC to handle different channel noise scenarios dynamically. The main contributions of this work are summarized as follows:

- *Adaptive Channel Denoising:* We propose a novel ADDM framework, where the denoising process is not only guided by the source signal but also by the varying CSI. Particularly, the channel conditions are fed into the U-Net architecture by multi-head attention (MHA) layers to tailor the reverse process to a specific noise level. This mechanism enables a single, unified model to learn a variety of specialized denoising strategies more efficiently and obtain optimal performance across diverse channel conditions without requiring retraining or relying on multiple specialized models.
- *Jointly Modified Diffusion Processes:* We design an adaptive denoising mechanism through both modified forward and reverse diffusion processes that incorporate the degraded signals. In this way, even in scenarios where CSI is unavailable or imperfectly estimated, our system still demonstrates considerable performance.
- *End-to-end Integration of DM into DeepJSCC:* By proposing a novel ADDM module, we can adapt the diffusion model into the DeepJSCC baseline in an end-to-end fashion. This framework is trained jointly, allowing the encoder to learn a latent representation optimally suited for generative reconstruction by the diffusion-based decoder. As a result, we achieve a system with a powerful generative prior that offers superior resilience to channel impairments to enhance reconstruction fidelity for wireless image transmission.
- *Simulation Validation:* Extensive simulation on ADE20K and ImageNet datasets shows that our ADDM-JSCC

system outperforms the baselines across a wide range of SNRs under both additive white Gaussian noise (AWGN) and Rayleigh slow fading channels.

C. Paper Structure and Notations

The remainder of this paper is structured as follows. Section II provides an overview of the system model, ADDM-JSCC, outlining the roles of the encoder, the physical channel layer, and the decoder. Section III delves into the theoretical principles of the proposed ADDM, beginning with the foundational concepts of DDPMs and then introducing the mathematical formulation of our adaptive denoising framework. Section IV presents the extensive simulation settings, numerical results, and a thorough discussion of our proposed approach's performance compared to baselines under various channel conditions. Finally, Section V concludes the paper, summarizing the key findings and suggesting directions for future research.

Notations: Throughout the paper, we denote the source image as x , the encoded latent representation as z , the received noisy latent vector as $\hat{z}$, and the reconstructed image as $\hat{x}$. The CSI is defined as y and the refined latent representation after forwarding the ADDM module is represented by $\tilde{z}$. The parameters of the JSCC encoder, JSCC decoder, and ADDM's U-Net architecture are denoted by φ , ϕ , and θ , respectively. The notation $\mathbb{E}[\cdot]$ expresses the expected value of a random variable, whereas $\mathbb{R}$, $\mathbb{C}$ represent a set of real and complex numbers, respectively. $\mathcal{CN}(\cdot, \cdot)$ indicates a complex circularly symmetric Gaussian distribution, $\mathbf{I}$ denotes an identity matrix, and the symbol $\|\cdot\|$ refers to the Euclidean norm of a vector.

II. SYSTEM MODEL

In this section, we provide a brief description of the system model, namely ADDM-JSCC, as shown in Fig. 2. The JSC encoder at the transmitter first encodes the input signal into a real-valued semantic vector, which is then mapped to complex-valued channel input symbols to transmit over the wireless channel. The ADDM is utilized to refine the noisy symbols iteratively, and the JSC decoder is employed to reconstruct the original signal with minimal distortion at the receiver.

A. Joint Source-Channel Encoder

At the transmitter, an RGB image with a size of H (height) and W (width) is represented by a vector $x \in \mathbb{R}^m$, where $m = 3 \times H \times W$, first undergoes the joint source-channel (JSC) encoder $\mathcal{E}_\varphi$ to become a real-valued semantic vector $z \in \mathbb{R}^{2k}$ as follows:

$$z = \mathcal{E}_\varphi(x), \quad (1)$$

where φ denotes the encoder's parameters and k denotes the length of the channel input vector. The real-value semantic vector z should be converted into a k -length complex-valued vector z_c , which can be transmitted through the wireless channel. The i -th transmitted symbol of $z_c \in \mathbb{C}^k$ can be defined as $z_{c,i} = z_i + jz_{i+k}$ for $i = 1, \dots, k$. We define the bandwidth compression ratio (BCR) as $R = k/m$, where m represents the source bandwidth and k is the channel bandwidth. To satisfy an average power constraint P before

transmission over the physical channel, the transmitted signal vector z_c is subject to $\frac{1}{k}\mathbb{E}[z_c^* z_c] \leq P$, which can be obtained according to the following normalization

$$\bar{z}_c = \sqrt{kP} \frac{z_c}{\sqrt{z_c^* z_c}}, \quad (2)$$

where z_c^* is the conjugate transpose of z_c and P is a unit power constraint.

B. Physical Channel Layer

The complex-valued channel input symbols $\bar{z}_c$ are then transmitted over the physical channel. In this work, we assume that each image is transmitted independently and the channel state remains constant during each image transmission. As a result, AWGN channel and Rayleigh slow fading channel are considered to introduce random corruption to the transmitted symbols. The received symbol at the receiver side, in general, can be expressed as

$$\hat{z}_{c,i} = h_{c,i} \bar{z}_{c,i} + n_{c,i}, \quad (3)$$

where $h_{c,i} \sim \mathcal{CN}(0, 1)$ represents the physical channel's coefficients sampled from an independent and identical distribution (i.i.d.), $n_{c,i}$ is additive noise sampled from an i.i.d. Gaussian distribution, i.e., $n \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$, and σ^2 is the average noise power. We assume the channel's coefficients are estimated from pilot symbols at the receiver side for equalization.

The complex-valued vector output channel $\hat{z}_c$ is then transformed into a real-valued vector $\hat{z} \in \mathbb{R}^{2k}$, which is subsequently processed by neural networks at the receiver.

C. Adaptive Denoising Diffusion Model

At the receiver, the obtained noisy vector $\hat{z}$ is fed into the ADDM module, where the wireless channel noise is eliminated progressively by a designated denoising scheduler to output the refined signal $\tilde{z}$. To enhance the channel denoising capability, CSI y is integrated through the attention layers of the U-Net model, enabling the system to dynamically adjust the diffusion process to diverse noise levels. In this work, we utilize channel SNR information as CSI, which is consistent with prior literature. The channel denoising process of ADDM is simplified as

$$\tilde{z} = \text{ADDM}(\hat{z}, y, \theta), \quad (4)$$

where θ denotes the ADDM's parameters.

The detailed implementation of the ADDM module is provided in Section III.

D. Joint Source-Channel Decoder

After receiving a refined version of $\tilde{z}$, a JSC decoder $\mathcal{D}_\phi$ then reconstructs $\tilde{z}$ into $\hat{x}$ through a series of transformations from the encoded latent representation to the original image resolution to minimize the average distortion $d(x, \hat{x})$. The reconstruction process can be formulated as

$$\hat{x} = \mathcal{D}_\phi(\tilde{z}) = \mathcal{D}_\phi(\mathcal{E}_\varphi(x)), \quad (5)$$

where ϕ defines the decoder's parameters.

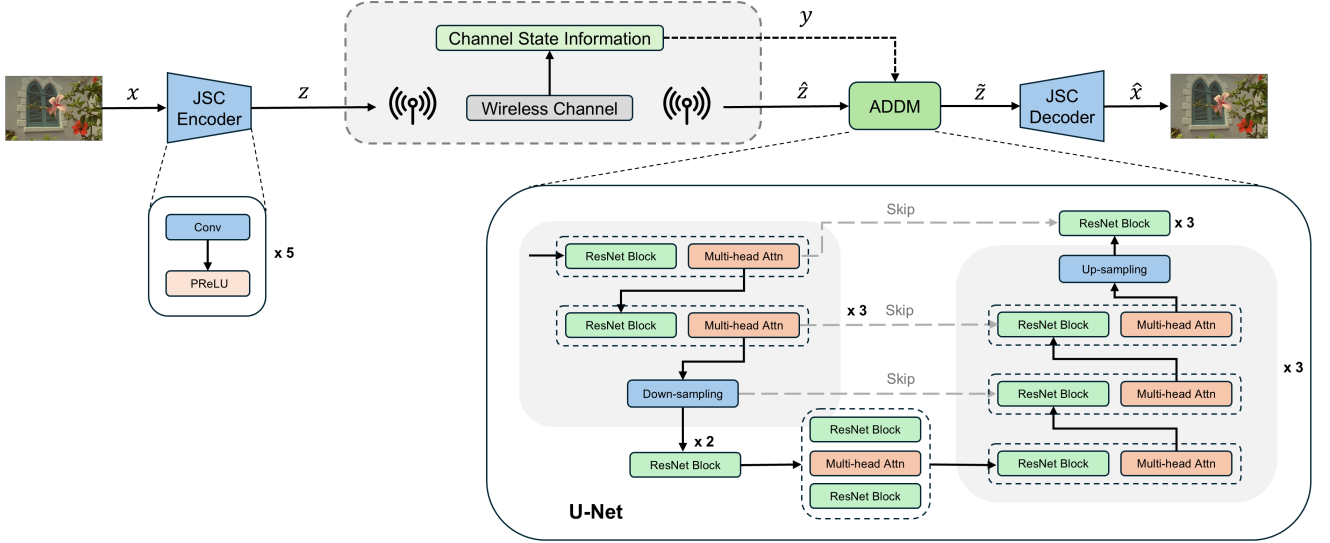


Fig. 2: Architecture of the ADDM-JSCC model. By modifying both diffusion processes and integrating CSI into the ADDM module, we can condition the denoising process to further eliminate the channel impairments at a specific noise level for image reconstruction fidelity enhancement.

We optimize the resulting JSC encoder and decoder in an end-to-end fashion during the training process according to the mean squared error (MSE) loss function between the original signal x and the reconstructed signal $\hat{x}$:

$$L_{JSCC}(\varphi, \phi) = d(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2, \quad (6)$$

where x_i and $\hat{x}_i$ represent the color intensity of each pixel component corresponding to x and $\hat{x}$, respectively. By training a model to minimize this loss function, we can obtain the optimal parameters φ^* and ϕ^* for the JSCC encoder and decoder.

III. DIFFUSION-BASED ADAPTIVE CHANNEL DENOISER FOR DEEPIJSCC

This section presents the underlying theoretical principles of the proposed ADDM for DeepJSCC. We begin by reviewing the foundational concepts of the denoising diffusion probabilistic model to establish the necessary background. Subsequently, we introduce the ADDM, detailing its mathematical formulation and highlighting the key innovations that enable adaptive channel denoising.

A. Denoising Diffusion Probabilistic Model

1) *Forward Diffusion Process*: In the forward process, also known as the diffusion process, a complex data distribution is gradually transformed into a simple, tractable one, typically an isotropic Gaussian distribution. This is achieved through the controlled, incremental injection of Gaussian noise over a finite number of timesteps. Specifically, the forward process is mathematically defined as a fixed Markov chain that systematically degrades an initial data sample. The latent vector z_0 , by progressively adding Gaussian noise over T discrete

timesteps, can be formulated by the transition kernel as [9]

$$q(z_t|z_{t-1}) \sim \mathcal{N}(z_t; \sqrt{1 - \beta_t}z_{t-1}, \beta_t \mathbf{I}), \quad (7)$$

where z_t at any given timestep t is dependent only on the immediately preceding state z_{t-1} and is conditionally independent of all other previous states. β_t is a small positive constant at a timestep t from a predefined variance schedule ($0 < \beta_1 < \beta_2 < \dots < \beta_T < 1$). This ensures that, as t approaches the final timestep T , the cumulative effect of the added noise completely overwhelms the original signal, causing the distribution of z_T to converge to a standard isotropic Gaussian distribution $\mathcal{N}(0, \mathbf{I})$.

The joint probability of the entire sequence of noisy latents $z_{1:T}$ given the initial state z_0 is the product of these individual transitions

$$q(z_{1:T}|z_0) = \prod_{t=1}^T q(z_t|z_{t-1}). \quad (8)$$

By defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, we can sample z_t directly from the original latent vector z_0 at any arbitrary timestep t without needing to iterate through the entire Markov chain. The recursive substitution of the transition kernel leads to the following closed-form expression

$$q(z_t|z_0) \sim \mathcal{N}(z_t; \sqrt{\bar{\alpha}_t}z_0, (1 - \bar{\alpha}_t)\mathbf{I}). \quad (9)$$

This can be rewritten as a direct sampling equation by applying the reparameterization [37]

$$z_t = \sqrt{\bar{\alpha}_t}z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad (10)$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is a random standard Gaussian noise. As $t \rightarrow T$, z_T approaches a pure isotropic Gaussian distribution.

Eq. (10) is fundamental for efficient model training, as it allows for the random sampling of training pairs (z_0, t) and

the direct, one-shot computation of the corresponding noisy latent z_t .

2) *Reverse Diffusion Process*: While the forward process is a fixed procedure for destroying data, the reverse process is a learned, generative procedure for creating data from noise. It is the core of the diffusion model's generative capability. The reverse process aims to generate a clean data sample z_0 by starting from the pure noise $z_T \sim \mathcal{N}(0, \mathbf{I})$, and iteratively denoising it over T timesteps. This process is modeled as another Markov chain $p_\theta(z_{0:T-1}|z_T)$ with learned transition probabilities $p_\theta(z_{t-1}|z_t)$:

$$p_\theta(z_0, \dots, z_{T-1}|z_T) = \prod_{t=1}^T p_\theta(z_{t-1}|z_t), \quad (11)$$

where each transition $p_\theta(\cdot)$ is designed to approximate a single step of the true, but unknown, reverse process. This formulation originates from the challenge that the true reverse posterior distribution $q(z_{t-1}|z_t)$ is intractable to compute directly. This intractability arises because evaluating this distribution would require marginalizing over all possible initial data points z_0 , which involves integrating over the entire, unknown data distribution. Since the data distribution $q(z_0)$ is precisely what the model is trying to learn, this path is computationally infeasible. This fundamental challenge necessitates the use of a powerful function approximator, a neural network with learnable parameters θ , to learn an approximation of this posterior, denoted by $p_\theta(z_{t-1}|z_t)$ as mentioned above.

The key insight of the DDPM framework [9] is that the reverse posterior becomes tractable when it is additionally conditioned on the starting point z_0 . Applying Bayes' theorem, the distribution $q(z_{t-1}|z_t, z_0)$ can be shown to be a Gaussian given by

$$q(z_{t-1}|z_t, z_0) \sim \mathcal{N}\left(z_{t-1}; \tilde{\mu}(z_t, z_0, t), \tilde{\beta}_t \mathbf{I}\right), \quad (12)$$

where

$$\tilde{\mu}(z_t, z_0, t) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} z_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} z_0, \quad (13)$$

$$\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t. \quad (14)$$

By substituting z_0 obtained from (10) in (13), we can achieve

$$\tilde{\mu}(z_t, z_0, t) = \frac{1}{\sqrt{\alpha_t}} \left(z_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_t \right). \quad (15)$$

To approximate the target distribution $q(\cdot)$, the neural network $p_\theta(z_{t-1}|z_t)$ therefore only needs to approximate its mean $\tilde{\mu}(z_t, z_0, t)$. We simply set the approximated mean μ_θ in the same form as the target mean with a learnable neural network $\epsilon_\theta(z_t, t)$

$$\mu_\theta(z_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(z_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(z_t, t) \right), \quad (16)$$

where $\epsilon_\theta(z_t, t)$ is a trained neural network to predict the noise component added to z_t . The variance is fixed to a constant $\tilde{\beta}_t$ without learnable parameters.

The model parameters θ are optimized by maximizing

the evidence lower bound (ELBO) of the log-likelihood $\log p_\theta(z_0)$. The ELBO provides a tractable lower bound on this intractable quantity and can be decomposed into a series of Kullback-Leibler (KL) divergence terms. Each term measures the discrepancy between the learned reverse transition $p_\theta(z_{t-1}|z_t)$ and the true but tractable forward process posterior $q(z_{t-1}|z_t, z_0)$ at each timestep t . After significant manipulation and a crucial reparameterization of the mean $\mu_\theta(z_t, t)$ to predict the noise component ϵ that was added to create z_t , this complex objective can be greatly simplified. The final objective function for the training process reduces to minimizing the simplified MSE between the true noise ϵ and the noise predicted by the network $\epsilon_\theta(z_t, t)$ as given by

$$L_{ELBO}(\theta) = \mathbb{E}_{t, z_0, \epsilon \sim \mathcal{N}(0, \mathbf{I})} \left\| \epsilon - \epsilon_\theta(\sqrt{\alpha_t} z_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \right\|_2^2, \quad (17)$$

where ϵ denotes the random noise for a given timestep t .

B. Adaptive Denoising Diffusion Model

Standard conditional diffusion models typically utilize external information, such as class labels [38], text embeddings or semantic maps [20], to guide a fixed, predefined denoising process. In these architectures, the conditioning information influences the reverse path but does not alter the forward noising process. However, in a wireless communication context, the received signal $\hat{z}$ is not merely an external conditioning variable but a corrupted version of the target clean signal z caused by wireless channel noise. We design the ADDM framework to leverage this intrinsic relationship by fundamentally altering both the diffusion paths based on $\hat{z}$. This creates a more direct and efficient reconstruction process by allowing the model to be aware of the channel corruption from the very beginning.

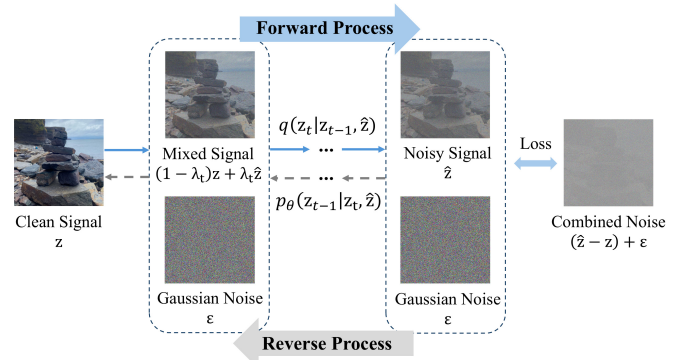


Fig. 3: A visualization of the diffusion process (solid blue arrows) and the reverse process (dashed gray arrows) of the proposed denoising diffusion model. $\hat{z}$ is the noisy version of the signal z obtained after transmitting through the wireless channel. Note that z is a latent representation vector instead of an image as illustrated.

To incorporate the noisy signal $\hat{z}$ into the diffusion process, an interpolation parameter λ_t is introduced to combine it with the original signal z . We can then define the forward diffusion

process as follows:

$$q_{\text{addm}}(z_t|z_0, \hat{z}) \sim \mathcal{N}(z_t; (1 - \lambda_t)\sqrt{\alpha_t}z_0 + \lambda_t\sqrt{\alpha_t}\hat{z}, \delta_t\mathbf{I}), \quad (18)$$

where δ_t represents the variance. Different from the original diffusion process $q(z_t|z_0)$ in (9), the Gaussian mean in (18) is a linear interpolation between the clean data z_0 and the noisy data $\hat{z}$, controlled by the ratio λ_t . The mean of the forward transition at step t is no longer solely dependent on the previous state but is now a weighted average with a channel-corrupted signal $\hat{z}$. The interpolation schedule is set such that λ_t gradually increases from 0 to $\lambda_T \approx 1$. At the beginning of the forward process (when t is small), λ_t is close to zero, anchoring the mean of z_t to the clean signal z_0 . As the process evolves and t increases, the weight λ_t grows, turning the mean to progressively shift away from the clean signal and towards the noisy received signal $\hat{z}$. This mechanism has a profound and intuitive effect on the diffusion trajectory, as visualized in Fig. 3. Note that the original signal z is literally z_0 in the diffusion process at the initial time step $t = 0$.

By marginalizing out $\hat{z}$ in the product of $q(z_t|z_0, \hat{z})$ and $p_{\hat{z}}(\hat{z}|z_0)$, we can achieve

$$q_{\text{addm}}(z_t|z_0) = \int q(z_t|z_0, \hat{z})p_{\hat{z}}(\hat{z}|z_0)d\hat{z}, \quad (19)$$

and the resulting $q_{\text{addm}}(z_t|z_0)$ becomes equivalent to the original diffusion process $q(z_t|z_0)$, provided that the variance δ_t is defined as

$$\delta_t = (1 - \bar{\alpha}_t) - \lambda_t^2 \bar{\alpha}_t. \quad (20)$$

Here, δ_t precisely accounts for the additional variance introduced into the forward process by the noisy signal $\hat{z}$ via the interpolation term λ_t . Hence, (18) can be considered as a generalized version of the original diffusion probabilistic model.

With the mean of the forward process as a function of $\hat{z}$, the reverse process' goal is no longer a fixed path back to z_0 , but a dynamic trajectory that takes the received noisy signal into account. This constitutes a much deeper form of conditioning, as the model learns to reverse a process that has been "pre-guided" by the received noisy signal $\hat{z}$ from the very beginning. This mechanism makes the denoising task more direct and potentially more efficient, as the model is not trying to learn a standard generative path with external information but rather reversing a path that already contains the interested information. For the reverse process, we start at the step T with the noisy signal $\hat{z}$, where the variance is δ_T , corresponding to (18) with $\lambda_T = 1$

$$p_{\text{addm}}(z_T|\hat{z}) \sim \mathcal{N}(z_T; \sqrt{\alpha_T}\hat{z}, \delta_T\mathbf{I}). \quad (21)$$

Following a similar approach as in (12), the reverse process denoted by the dashed gray arrows in Fig. 3 is designed to predict z_{t-1} based on the current signal z_t and the noisy signal $\hat{z}$

$$p_{\text{addm}}(z_{t-1}|z_t, \hat{z}) \sim \mathcal{N}(z_{t-1}; \mu_{\theta}(z_t, \hat{z}, t), \tilde{\delta}_t\mathbf{I}), \quad (22)$$

where $\mu_{\theta}(z_t, \hat{z}, t)$ is the estimated mean of the reverse process. Similar to (16), this term can be expressed as a linear combination

of z_t , $\hat{z}$, and an estimated noise term ϵ_{θ} , weighted by coefficients c_z , $c_{\hat{z}}$, and c_{ϵ} , respectively, as

$$\mu_{\theta}(z_t, \hat{z}, t) = c_z z_t + c_{\hat{z}} \hat{z} - c_{\epsilon} \epsilon_{\theta}(z_t, \hat{z}, t), \quad (23)$$

where the model $\epsilon_{\theta}(z_t, \hat{z}, t)$ is trained to estimate the combined Gaussian and physical channel noise. The coefficients c_z , $c_{\hat{z}}$, and c_{ϵ} are derived from solving the ELBO optimization criterion as in [22]

$$c_z = \frac{(1 - \lambda_t)\delta_{t-1}}{(1 - \lambda_{t-1})\delta_t} \sqrt{\alpha_t} + (1 - \lambda_{t-1}) \frac{\delta_{t|t-1}}{\delta_t} \frac{1}{\sqrt{\alpha_t}}, \quad (24)$$

$$c_{\hat{z}} = (\lambda_{t-1}\delta_t - \frac{\lambda_t(1 - \lambda_t)}{1 - \lambda_{t-1}} \alpha_t \delta_{t-1}) \frac{\sqrt{\alpha_{t-1}}}{\delta_t}, \quad (25)$$

$$c_{\epsilon} = (1 - \lambda_{t-1}) \frac{\delta_{t|t-1}}{\delta_t} \frac{\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\alpha_t}}. \quad (26)$$

The term $\delta_{t|t-1}$ is calculated by ensuring δ_t satisfies (20), resulting in

$$\delta_{t|t-1} = \delta_t - \left(\frac{1 - \lambda_t}{1 - \lambda_{t-1}}\right)^2 \alpha_t \delta_{t-1}. \quad (27)$$

Additionally, the variance term in (22) is derived as

$$\tilde{\delta}_t = \frac{\delta_{t|t-1} * \delta_t}{\delta_{t-1}}. \quad (28)$$

With all the derived factors, the loss function $L_{\text{addm}}(\theta)$ is obtained accordingly as

$$\mathbb{E}_{t, z_0, \epsilon, \hat{z}} \left\| \left(\frac{\lambda_t \sqrt{\alpha_t}}{\sqrt{1 - \bar{\alpha}_t}} (\hat{z} - z_0) + \frac{\sqrt{\delta_t}}{\sqrt{1 - \bar{\alpha}_t}} \epsilon \right) - \epsilon_{\theta}(z_t, \hat{z}, t) \right\|_2^2, \quad (29)$$

where ϵ is the Gaussian noise in z_t . This objective function fundamentally changes the task of the denoising network. In contrast to (17) where ϵ_0 is trained to predict only the Gaussian noise ϵ , the model $\epsilon_{\theta}(z_t, \hat{z}, t)$ in the ADDM module estimates both the Gaussian noise ϵ and the physical channel noise component $(\hat{z} - z)$ present in z_t . Therefore, the coefficients of $(\hat{z} - z)$ and ϵ are selected the same as the coefficients for $\hat{z}$ and the standard deviations in (18), dynamically controlling the relative importance of estimating the channel noise versus the Gaussian noise at each time step t . When the ratio λ_t is set to 0, the reverse process in (22) and the objective function in (29) become (12) and (17) as in the original diffusion probabilistic model.

C. Channel Adaptivity Through CSI-Conditioned U-Net

To enhance further the adaptivity and resilience of the proposed approach to varying channel conditions, we integrate additional CSI into the U-Net architecture. Specifically, CSI denoted by y is fed into the ResNet block, which is then broadcast to the subsequent U-Net layers by multi-head attention (MHA) mechanism, as described in the system architecture in Fig. 2 and Fig. 4. The reverse diffusion process in (22) therefore includes y as another condition, which can be formulated as

$$p_{\text{addm}}(z_{t-1}|z_t, \hat{z}, y) \sim \mathcal{N}(z_{t-1}; \mu_{\theta}(z_t, \hat{z}, y, t), \tilde{\delta}_t\mathbf{I}), \quad (30)$$

where $\mu_\theta(z_t, \hat{z}, y, t)$ is modified as

$$\mu_\theta(z_t, \hat{z}, y, t) = c_z z_t + c_{\hat{z}} \hat{z} - c_\epsilon \epsilon_\theta(z_t, \hat{z}, y, t). \quad (31)$$

The CSI y is measured in the channel SNR value, typically projected into a high-dimensional embedding vector. This vector is then used to modulate the U-Net's internal feature representations. In this study, we concatenate this conditioning embedding with the timestep embedding along the channel dimension before they are broadcast throughout the network. These integrations into the ResNet block are demonstrated as given in Fig. 4.

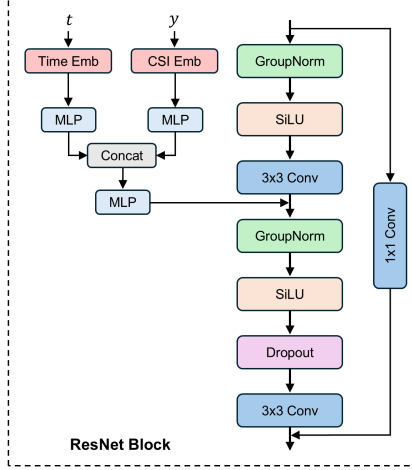


Fig. 4: The CSI y vector is concatenated with the time embedding along the channel dimension to feed into the ResNet block inside the U-Net.

By making the noise prediction function explicitly dependent on the CSI, i.e., $\epsilon_\theta(z_t, \hat{z}, y, t)$, the network learns to produce different noise estimates for the exact same inputs z_t and $\hat{z}$ but under different channel conditions. This allows a single model, with a single set of weights, to learn a variety of specialized denoising strategies, one for each potential SNR level. Hence, the objective function in (29) is altered accordingly, taking into account the channel conditions y .

D. ADDM Training and Sampling Algorithm

The training algorithm's goal is to optimize the denoising network ϵ_θ . This is achieved by repeatedly sampling a clean/noisy pair $(z_0, \hat{z})$, a random time step t , and Gaussian noise ϵ . A noisy latent z_t is then constructed using the modified forward process, which interpolates between the clean and noisy signals. The model's parameters are updated via a gradient descent step on the loss function modified from (29), which trains the network to predict a composite noise including both the injected Gaussian noise and the physical channel noise. The sampling algorithm outlines the reverse process used at the receiver to reconstruct the clean signal z_0 from the received noisy signal $\hat{z}$. The process is initialized at the time step T with a distribution centered around the received signal $\hat{z}$ rather than pure noise. It then iterates backward from $t = T$ to 1. In each step, it estimates the less noisy state z_{t-1} using the trained model's noise prediction ϵ_θ and the

Algorithm 1 : Training Algorithm

Input: $z_0, \hat{z}, y$.
Output: trained model $\epsilon_\theta(z_t, \hat{z}, y, t)$.
1: **Initialize** hyperparameters $T, \beta_t, \alpha_t, \bar{\alpha}_t$, and λ_t .
2: **for** $i = 1, 2, \dots, I_{\text{iter}}$ **do**
3: Randomly sample $(z_0, \hat{z}), t \sim \text{Uniform}[T]$, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
4: $z_t = ((1 - \lambda_t)\sqrt{\alpha_t}z_0 + \lambda_t\sqrt{\bar{\alpha}_t}\hat{z}) + \sqrt{\delta_t}\epsilon$
5: Take a gradient descent step on
 $\nabla_\theta \left\| \frac{1}{\sqrt{1-\alpha_t}} (\lambda_t\sqrt{\bar{\alpha}_t}(\hat{z} - z_0) + \sqrt{\delta_t}\epsilon) - \epsilon_\theta(z_t, \hat{z}, y, t) \right\|_2^2$
6: **end for**

Algorithm 2 : Sampling Algorithm

Input: $\hat{z}, y$.
Output: clean signal $\tilde{z}$.
1: Randomly sample $z_T \sim \mathcal{N}(z_T; \sqrt{\alpha_T}\hat{z}, \delta_T \mathbf{I})$
2: **for** $t = T, T-1, \dots, 1$ **do**
3: $z_{t-1} = c_z z_t + c_{\hat{z}} \hat{z} - c_\epsilon \epsilon_\theta(z_t, \hat{z}, y, t) + \sqrt{\delta_t}\epsilon$ with $c_z, c_{\hat{z}}, c_\epsilon$ computed in (24), (25), and (26); $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.
4: **end for**

derived coefficients $(c_z, c_{\hat{z}}, c_\epsilon)$ that define the mean of the reverse transition. The final output of this iterative denoising procedure is the refined signal $\tilde{z}$.

The training and sampling algorithms are summarized in Algorithms 1 and 2, respectively.

IV. SIMULATION RESULTS AND DISCUSSIONS

In this section, we describe the simulation settings and present different simulation results to illustrate the promising performance of our proposed ADDM-JSCC approach under various channel conditions and SNR values.

A. Simulation Settings

In the simulation, the ADDM architecture is mainly based on the U-Net architecture improved from [9], as shown in Fig. 2. The U-Net model features a symmetric encoder-decoder structure with skip connections that link corresponding layers between a series of down-sampling and up-sampling paths. This design allows the model to combine high-level semantic information from the deeper layers with fine-grained, high-resolution features from the shallower layers, which is crucial for effective denoising. The architecture consists of:

- **Residual Blocks** [39]: Both the down-sampling and up-sampling paths consist of wide residual blocks (ResNet blocks), which help in training deep networks by mitigating the vanishing gradient problem. The architecture of a ResNet block is illustrated as shown in Fig. 4.
- **Attention Mechanisms** [40]: To facilitate the adaptive denoising process, MHA layers are integrated into the U-Net, particularly at lower-resolution feature maps (e.g., 16×16). This allows the model to focus on the most relevant parts of the latent representation. In this work, we configure the MHA layers with 8 attention heads and 256 embedding dimensions.
- **Downsampling and Upsampling Blocks:** These blocks consist of convolutional layers with the proper numbers

of input and output channels, along with defined stride and padding values.

- **Conditional Input:** The model is conditioned on both the timesteps t of the diffusion process and the CSI y , which is provided as SNR values. These scalar values are transformed into high-dimensional vector embeddings using sinusoidal positional encodings and then processed by multi-layer perceptron (MLP) networks to feed into each residual block as given in Fig. 4. This conditioning enables the ADDM to tailor its denoising strategy to the specific noise level of the channel and the current stage of the reverse diffusion process.

For DeepJSCC, the encoder and decoder modules are based on the architecture from [3]. This architecture is a fully convolutional neural network (CNN) designed as an autoencoder. The encoder uses a series of convolutional layers with progressively smaller spatial dimensions and more feature channels to compress the input image into a compact latent representation, z . The decoder mirrors this structure, using transposed convolutional layers to upsample the latent representation back to the original image dimensions. The final layer of the decoder is a convolutional layer with a sigmoid activation function to produce the output image pixels, which are normalized between 0 and 1. The entire autoencoder is trained end-to-end to minimize the distortion between the input and reconstructed images. In both ADDM-JSCC and the baseline systems, we utilize the exact same fully convolutional neural network architecture for JSCC. Maintaining this identical autoencoder structure facilitates a direct and fair comparison of our proposed method and the baselines.

Datasets: The models are trained and evaluated on two large-scale, diverse image datasets to assess performance and generalization across different image resolutions and content complexities:

- **ADE20K** [41]: This is a comprehensive scene-parsing dataset containing over 27,000 images with dense annotations for objects and object parts across 365 different scenes. For the simulation, images from this dataset are randomly cropped to patches of size 128×128 pixels.
- **ImageNet** [42]: A foundational large-scale dataset for object recognition, ImageNet contains over 14 million images organized according to the WordNet hierarchy, spanning more than 20,000 categories. We only utilize its subset, which consists of 50,000 images across 1,000 object classes. Images from this dataset are randomly cropped to 256×256 pixels during training.

textbfBenchmarks: We compare our proposed method with the original baseline DeepJSCC. Each of these baseline models is optimized for a specific, fixed SNR value. Typically, a specialized model optimized for a specific SNR establishes a theoretical performance bound for a single model trained over a range of SNRs (multi-SNR training) at the same test point [12]. A comparison with these specialized models can highlight the superior performance of the proposed model ADDM-JSCC, if it consistently outperforms this upper bound across the entire SNR range by significant margins. As a result, it inherently outperforms any single multi-SNR trained version

Algorithm 3 : The Joint Training Algorithm for the ADDM and JSCC System

The first training stage

Input: Training dataset D .

- 1: **Initialize** batch size N , learning rate γ .
- 2: **repeat**
- 3: Sample a data batch $\mathbf{x} = [x_1, \dots, x_N]$ from D
- 4: **for** each data batch $\mathbf{x} \in D$ **do**
- 5: The encoder $\mathcal{E}_\varphi$ encodes an image batch $\mathbf{x}$ into a latent representation z , $z = \mathcal{E}_\varphi(\mathbf{x})$
- 6: The decoder $\mathcal{D}_\phi$ reconstructs $\hat{\mathbf{x}}$ from the latent z , $\hat{\mathbf{x}} = \mathcal{D}_\phi(z) = \mathcal{D}_\phi(\mathcal{E}_\varphi(\mathbf{x}))$
- 7: **end for**
- 8: Compute $L_{JSCC}(\varphi, \phi)$ in (6) and update φ, ϕ .
- 9: **until** Converged

The second training stage

- 1: **repeat**
- 2: Sample a batch of latent representations z_0 from the first stage, the corresponding noisy version $\hat{z}$, and y
- 3: Train the ADDM with Algorithm 1 to update θ .
- 4: **until** Converged

The third training stage

Input: Training dataset D , y .

- 1: **Initialize** hyperparameters T , β_t , α_t , $\bar{\alpha}_t$, and λ_t .
 - 2: **repeat**
 - 3: Sample a data batch $\mathbf{x} = [x_1, \dots, x_N]$ from D
 - 4: $t \sim \text{Uniform}[T]$
 - 5: $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
 - 6: Perform forward propagation through the noisy channel and the trained ADDM.
 - 7: Compute $L_{JSCC}(\varphi, \phi)$ in (6) and update φ, ϕ .
 - 8: **until** Converged
-

of the same baseline, which would necessarily fall below that bound. Therefore, we omit the multi-SNR DeepJSCC in the simulation.

Training Details: We implement the system with 200 denoising or time steps T and the Adam optimizer [43] with a learning rate of 10^{-4} . The interpolation parameter $0 < \lambda_t < 1$ is set in a linear schedule over the corresponding number of T , which gradually shifts the focus from the clean signal to the noisy received signal during the forward process. For the variance scheduler, we also consider a linear schedule in a range $\beta_t \in [0.0001, 0.0095]$. This gradual increase in noise variance ensures that the forward diffusion process smoothly transforms the data into an isotropic Gaussian distribution. The batch sizes of 64 and 32 are adopted for the ADE20K dataset and the ImageNet dataset, respectively. We set the bandwidth compression ratio (BCR) as $R = \frac{1}{6}$ during the simulation. The simulation parameters are summarized in Table II.

TABLE II: Simulation Parameters

Parameter	Value
Denoising steps	200
Learning rate	10^{-4} with Adam optimizer
Interpolation weight	$0 < \lambda_t < 1$
Variance scheduler	$\beta_t \in [0.0001, 0.0095]$
Batch size	$N = 32$ or 64
Training SNR	$[-4, 1, 4, 7, 13, 19]$ dB
BCR	$R = 1/6$

The entire training process of the ADDM-JSCC system consists of three stages. In the first training stage, the DeepJSCC encoder $\mathcal{E}_\varphi$ and decoder $\mathcal{D}_\phi$ are trained jointly without the ADDM module to learn the distribution of the semantic latent space through the loss function in (6). In the second training stage, we fix the DeepJSCC's parameters such that the ADDM module can focus on learning the denoising diffusion process given the latent representation z obtained from the first stage, the corresponding noisy version $\hat{z}$, and CSI y . The ADDM is trained under various channel conditions with a uniform distribution of SNR values from [-4, 1, 4, 7, 13, 19] dB, allowing it to learn a robust mapping for a wide variety of channel variations. In the third training stage, the DeepJSCC encoder and decoder are retrained jointly with forward data propagation through the physical channel and the trained ADDM to continue minimizing the distortion $d(x, \hat{x})$ and we update only the learning parameters φ and ϕ . We train the models using an early-stopping scheme until the change in the loss function becomes negligible and the model performance saturates. The joint training process of the ADDM and DeepJSCC system is summarized in Algorithm 3.

B. MSE/PSNR Performance and Visualization Results

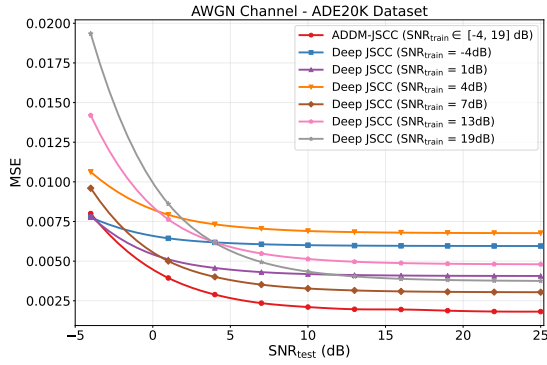


Fig. 5: MSE comparison under the AWGN channel versus SNRs on ADE20K dataset.

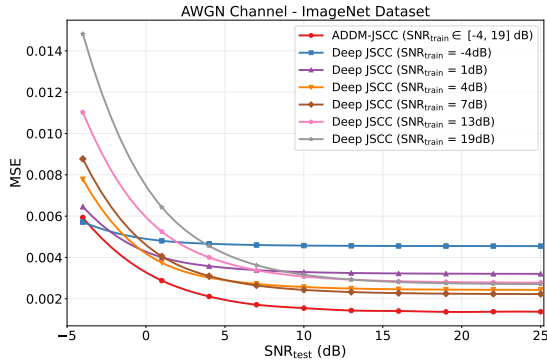


Fig. 6: MSE comparison under the AWGN channel versus SNRs on ImageNet dataset.

1) Impact of the ADDM Under Different Channel Types:

The MSE provides a measure of the average squared difference between the pixels of the reconstructed and original images.

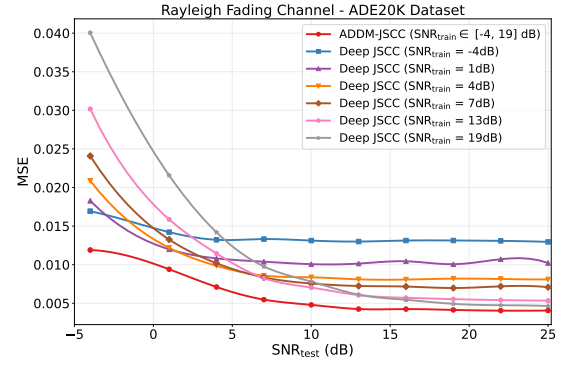


Fig. 7: MSE comparison under the Rayleigh fading channel versus SNRs on ADE20K dataset.

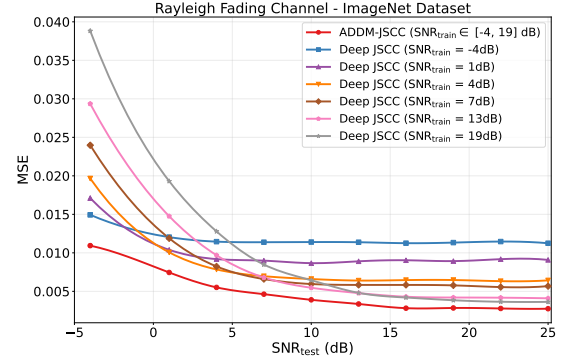


Fig. 8: MSE comparison under the Rayleigh fading channel versus SNRs on ImageNet dataset.

It is a fundamental metric that is particularly sensitive to large deviations, meaning a lower MSE indicates the model can avoid significant reconstruction failures. Under the AWGN channel, the MSE results presented in Fig. 5 (ADE20K dataset) and Fig. 6 (ImageNet dataset) already establish a clear and consistent advantage for the ADDM-JSCC (red curve). Across the entire range of tested SNRs, the proposed model achieves a lower MSE than the best-performing specialized DeepJSCC baseline. The performance gap is particularly noticeable in the low SNR regime (e.g., below 5 dB), where the noise power is comparable to or greater than the signal power, and the received signal is heavily corrupted under the severe channel conditions. The baseline models struggle to recover the original signal, leading to a sharp increase in MSE. In contrast, the ADDM-JSCC exhibits a more graceful degradation. This is because, as the channel quality worsens, the diffusion model learns to rely more heavily on its powerful capability to learn prior signal distributions, generating a plausible reconstruction rather than trying to invert a heavily distorted signal directly.

A similar tendency is witnessed in Figs. 7 and 8 under the Rayleigh fading channel. However, this channel model is more challenging, resulting in higher MSE errors compared to the AWGN channel. The ADDM-JSCC's superior performance in this scenario consistently suggests that its generative nature and learning of channel noise distribution for effective denoising are critical to its robustness.

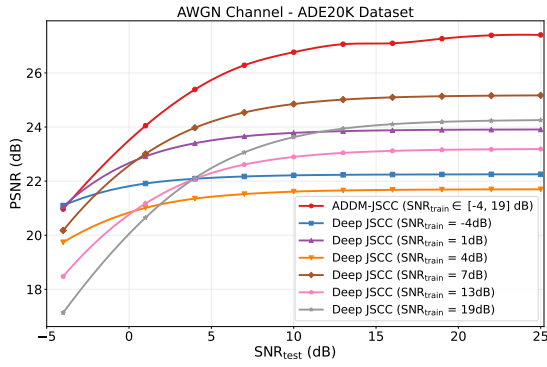


Fig. 9: PSNR performance versus SNRs over the AWGN channel on ADE20K dataset.

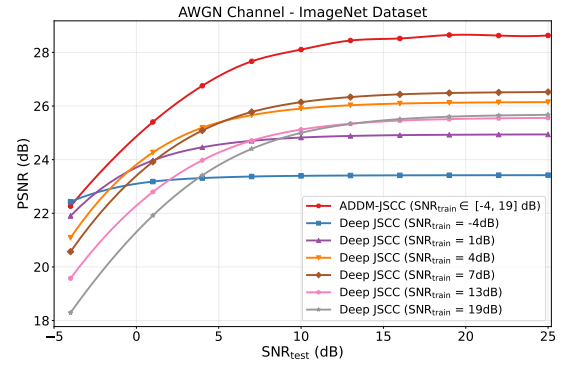


Fig. 11: PSNR performance versus SNRs over the AWGN channel on ImageNet dataset.

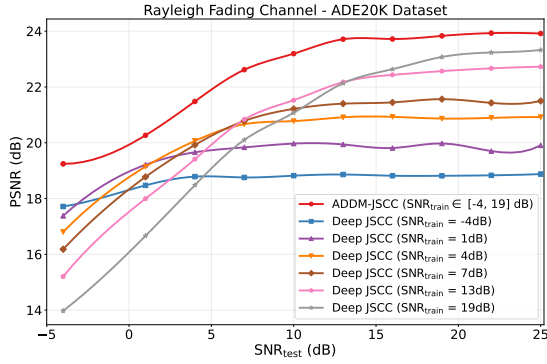


Fig. 10: PSNR performance versus SNRs over the Rayleigh fading channel on ADE20K dataset.

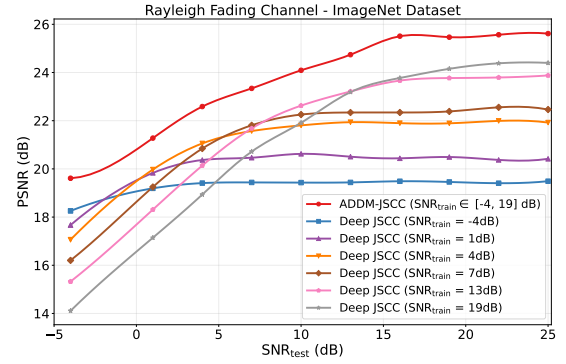


Fig. 12: PSNR performance versus SNRs over the Rayleigh fading channel on ImageNet dataset.

In terms of PSNR, a logarithmic metric that compares the maximum possible power of a signal to the power of the corrupting noise, the core advantage of our approach is evident in the performance plots. The ADDM-JSCC model consistently establishes the upper performance envelope across all test SNRs with various channel conditions. Under the AWGN channel, at a test SNR of 5 dB as shown in Fig. 11 for example, our model achieves a PSNR of approximately 26.5 dB, decisively outperforming the specialized DeepJSCC models trained at 4 dB (PSNR $\approx$ 25 dB) and 7 dB (PSNR $\approx$ 25.5 dB). As channel quality improves, the performance scales significantly, peaking at approximately 28.7 dB around a 20 dB SNR, a level unmatched by any single fixed-SNR baseline. This demonstrates that the adaptive diffusion mechanism is not only necessary for robustness but also highly efficient at leveraging cleaner signals to achieve high-fidelity reconstructions. The superiority of the ADDM module is even more pronounced in the challenging Rayleigh fading channel environment. In Fig. 10 and Fig. 12, DeepJSCC models exhibit a relative fluctuation and plateaus, failing to exploit better channel conditions at medium train SNR. Meanwhile, the ADDM-JSCC model adapts to this dynamic environment with superior grace. At a low SNR of 0 dB on ImageNet dataset, it achieves a PSNR of 21.5 dB, surpassing the corresponding DeepJSCC model by around 2 dB. Similar results are observed in the high SNR regime. This represents a significant performance enhancement, averaging 1.5-3 dB over the best-

performing fixed baseline for a given test SNR. In general, the performance comparisons indicate the remarkable efficiency of the ADDM module, which transforms the DeepJSCC architecture into a more intelligent, channel-aware system. Rather than being confined to a single noise level of DeepJSCC, our model leverages CSI to dynamically tailor its generative denoising process. This results in a single model that not only matches but also significantly outperforms the baselines.

The reconstructed images are visualized as shown in Fig. 13, which provides a qualitative comparison at the SNR of 2 dB under the AWGN channel. The image reconstructed by the DeepJSCC baseline is degraded by noticeable blurring, loss of texture, and visible artifacts, indicating that the system is struggling to deal with channel noise. The ADDM-JSCC reconstruction, however, exhibits a remarkable ability to preserve and even regenerate fine details, textures, and sharp edges, while ensuring overall semantic coherence. For example, DeepJSCC reconstructs the bird image with indistinct feathers and body parts, whereas those of ADDM-JSCC are significantly clearer and recognizable. The man's features are considerably sharper and more natural in the proposed method's output than in the baseline, which exhibits blurring and a loss of skin texture. These qualitative results provide tangible evidence that the ADDM-JSCC is capable of recovering fine graphical details and intricate patterns with a remarkable reduction in blur and noise. This illustration highlights the generative capability of our proposed method

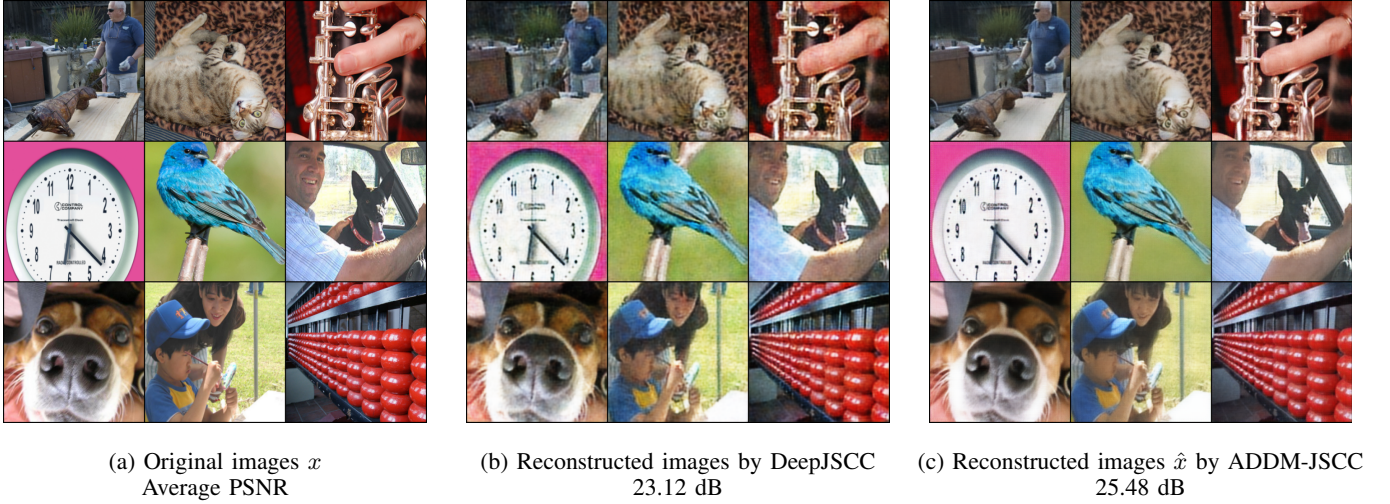


Fig. 13: Visualization of reconstructed images between ADDM-JSCC and the baseline under the AWGN channel from ImageNet dataset at $\text{SNR} = 2$ dB.

to restore semantically important information, such as intact textures or recognizable patterns.

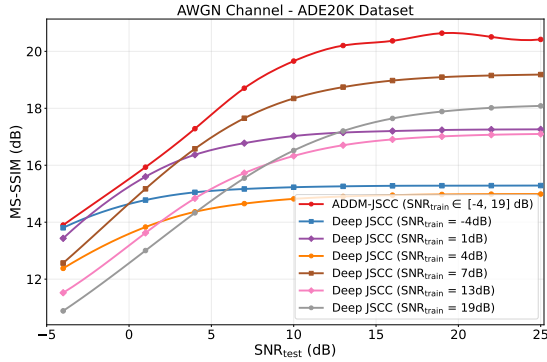


Fig. 14: MS-SSIM performance versus SNRs over the AWGN channel on ADE20K dataset.

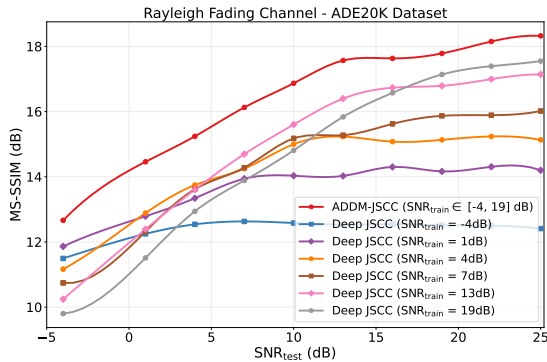


Fig. 15: MS-SSIM performance versus SNRs over the Rayleigh fading channel on ADE20K dataset.

2) Impact of Training Datasets and Image Resolutions:

The validation of the ADDM-JSCC model across diverse data sources and image resolutions demonstrates its generalization capability and scalability. Transitioning from 128×128 to

256×256 resolution exponentially increases the number of pixels and the complexity of the image distribution that the model must learn. ImageNet images contain higher-frequency details, more intricate textures, and a wider variety of semantic objects. The model performance thus varies based on the dataset with different image resolutions. As observed by comparing the results for the ADE20K and ImageNet datasets, the PSNR values are generally higher for the ImageNet dataset across all models and SNR levels. For instance, at a test SNR of 10 dB over the AWGN channel, the ADDM-JSCC model achieves a PSNR of approximately 26.5 dB on ADE20K in Fig. 9, whereas on the higher-resolution ImageNet dataset in Fig. 11, it reaches above 28 dB. This trend is expected, as higher-resolution images contain more intricate details, and the consistent bandwidth compression ratio enables the encoding of more information, resulting in superior reconstruction quality. Additionally, ADDM-JSCC maintains its performance advantage on this more challenging dataset, indicating that the underlying diffusion-based generative process is well-suited to learning and rendering complex, high-dimensional data. If the learned prior of the given signal is rich and detailed enough, it can guide the reconstruction accurately.

C. MS-SSIM Performance

To further verify the effectiveness of our proposed method, we evaluate the system model under the multi-scale structural similarity index (MS-SSIM), which provides a more comprehensive evaluation of image quality by focusing on the preservation of structural information, contrast, and luminance across multiple scales. Our analysis of MS-SSIM performance on ADE20K dataset under both AWGN and Rayleigh fading channels is illustrated in Fig. 14 and Fig. 15, respectively.

As shown in Fig. 14, the ADDM-JSCC (red curve) consistently establishes the upper bound of performance across the entire tested SNR range. While specialized DeepJSCC baselines exhibit plateaus relative to their training SNR, our adaptive approach scales gracefully, achieving MS-SSIM val-

ues exceeding 20 dB at high SNRs. This demonstrates that the integration of the diffusion denoising mechanisms effectively preserves structural coherence that traditional autoencoder-based DeepJSCC models lose under channel stress.

Similarly, for the Rayleigh fading channel in Fig. 15, ADDM-JSCC maintains a significant lead over the baselines. Notably, at a test SNR of 10 dB, the proposed model achieves an MS-SSIM of approximately 17 dB, while the best-performing specialized baseline ($\text{SNR}_{\text{train}} = 13$ dB) remains around 15.5 dB. Although this nonlinear multiscale metric is quite sensitive to variability, especially in the current more complex and dynamic environment, our models still show slight fluctuations due to statistical averaging methods with repeated running times implemented during the simulation. These results confirm that the generative prior of our ADDM module performs well in reconstructing accurate structural textures even when the physical channel introduces more severe distortion.

D. Ablation Study

1) *Impact of CSI as an Additional Condition:* In this experiment, we investigate the impacts of channel state information as an additional condition to the adaptively denoising process. Under the AWGN channel on the ADE20K dataset, as shown in Fig. 16, the CSI-conditioned model consistently outperforms its non-CSI counterpart. However, this margin is negligible, only around a 0.5 dB difference in PSNR values at most. In the Rayleigh fading channel as given in Fig. 17, which is more complex and dynamic, the signal quality can fluctuate rapidly and experience varying levels of fading. The CSI-aware model can leverage the provided SNR to become more resilient to severe channel conditions, while the model without CSI exhibits only a slight decrease of nearly 1 dB in PSNR across a wide range of channel conditions.

The model’s ability to sustain performance without precise CSI (or even without CSI entirely) stems from the following mechanisms. Firstly, the model is trained on a uniform distribution of SNR values (from -4 dB to 19 dB). This exposure allows the U-Net to implicitly infer the noise level from the structure of the corrupted signal $\hat{z}$ itself and especially from the forward diffusion process. As shown in (18), the forward diffusion process is already anchored to the noisy received signal $\hat{z}$ via the interpolation parameter λ_t , and the channel corruption is already present in the input data. Therefore, the U-Net model can learn the noise level within the data itself. This parameterization enables the model to be aware of channel corruption from the very beginning, leading to suboptimal performance. Secondly, unlike traditional DeepJSCC, which tries to invert a distorted signal directly, ADDM uses its learned prior to refine and reconstruct the original signal. By leveraging the CSI within the data itself from the forward process, the system with an attention mechanism can learn how to dynamically modulate its own internal feature representations during the iterative denoising process. For instance, when the SNR is high, the attention mechanism can learn to reduce the influence of the generative prior and trust the received latent features more, preserving the integrity of

the transmitted signal. Conversely, when the SNR is relatively low, it can learn to amplify the influence of the learned prior, effectively ignoring the highly corrupted input and relying on its generative capabilities to reconstruct semantically plausible data. Thirdly, with joint optimization, the encoder is trained to produce latent vectors specifically suited for this diffusion-based decoder (as illustrated in Stage 3 of Algorithm 3), the resulting representation is inherently more resilient to channel corruption. These justify that our approach is not over-reliant on the CSI, which is an important merit under scenarios where the channel information is unavailable or the CSI estimation is imperfect.

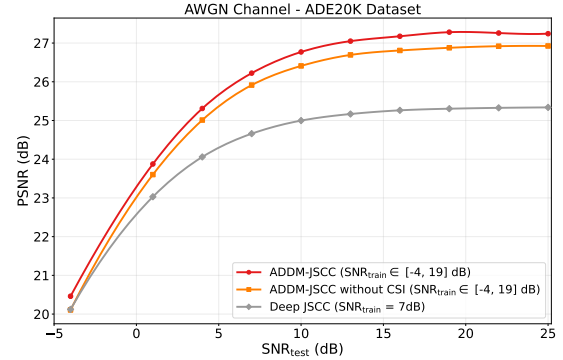


Fig. 16: PSNR performance versus SNRs over the AWGN channel on ADE20K dataset with or without CSI.

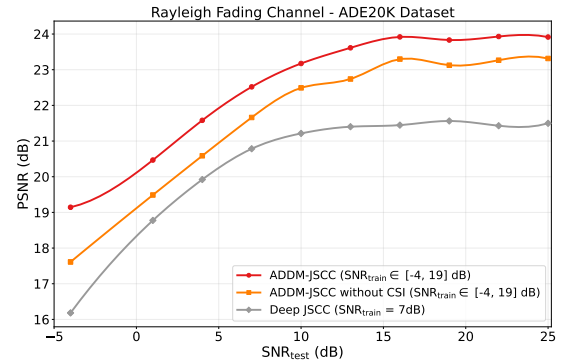


Fig. 17: PSNR performance versus SNRs over the Rayleigh fading channel on ADE20K dataset with or without CSI.

2) *Impact of the Number of Sampling Steps:* This ablation study quantifies how the number of sampling steps, T , influences the reconstruction fidelity. Fig. 18 illustrates the PSNR performance of ADDM-JSCC (trained at 4 dB and 13 dB) versus the number of sampling steps over the AWGN channel.

It can be observed that the model performance increases as the number of steps increases in the first place. This is because more steps allow the reverse diffusion process to better refine the signal from the initial noisy state. Notably, even with only 50 steps, ADDM-JSCC (4 dB model) achieves a PSNR of approximately 25 dB, which already decisively outperforms the fixed DeepJSCC baseline of nearly 23 dB.

However, increasing the sampling steps beyond 200 leads to a sharp degradation in PSNR. This performance drop

suggests that the reverse trajectory begins to diverge from the learned data distribution when the inference schedule exceeds the training steps, potentially introducing artifacts or “over-refining” the signal away from the ground truth. These findings justify our decision to use 200 sampling steps in simulations as mentioned earlier. This proper sampling step selection provides an effective trade-off between computational cost, inference time, and reconstruction quality.

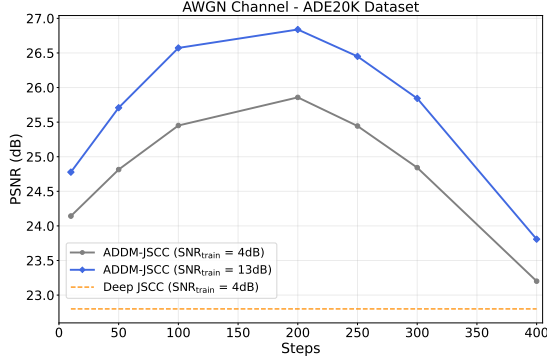


Fig. 18: PSNR performance versus sampling steps T over the AWGN channel on ADE20K dataset.

E. Complexity Analysis

In this section, we provide a brief discussion of the computational complexity involved in our ADDM-JSCC framework, including JSC encoder, ADDM module, and JSC decoder.

We first consider the JSC encoder and decoder, which are fully CNNs with 5 convolutional layers. Let $P = H \times W$ denote the number of spatial pixels of input feature maps, K represent the kernel size, and C be the number of input channels, assuming feature channels remain constant. Therefore, the computational complexity for the JSC encoder/decoder is calculated as

$$\text{FLOPs}_{\text{JSC Enc/Dec}}^{\text{JSC}} \approx \mathcal{O}(PC^2K^2).$$

Our ADDM module utilizes an improved U-Net architecture. Hence, its computational complexity arises mainly from the U-Net’s layers with different blocks. For the ResNet block with 2 convolutional layers, the total complexity for convolution per denoising step becomes $\mathcal{O}(PC^2K^2)$. The complexity of our framework is dominated by the MHA blocks. For a feature map with P pixels and C channels, the multi-head attention operation incurs a computational overhead of $\mathcal{O}(P^2C)$ [40], while the feedforward network contributes $\mathcal{O}(PC^2)$. As a result, the total computational cost for a MHA block for each denoising step is approximately $\mathcal{O}(P^2C + PC^2)$. In addition to ResNet and MHA blocks, the U-Net also includes the up-sample/down-sampling blocks and skip connections. However, their complexity is negligible compared to that of the ResNet blocks, especially the MHA blocks.

Assuming the ADDM framework contains L_1 ResNet blocks and L_2 MHA blocks, the overall complexity of our

proposed method over T timesteps can be expressed as

$$\text{FLOPs}^{\text{ADDM}} \approx \mathcal{O}(T(L_1PC^2K^2 + L_2(PC^2 + P^2C))).$$

As we can see that the encoding stage with only the encoder is parameter-efficient, which can be deployed into resource-constrained applications, such as mobile and IoT devices, where minimal computational overhead is essential. Meanwhile, most computational cost is concentrated in the ADDM framework of the decoding stage, which does not necessarily pose an underlying bottleneck, as decoding is typically performed on more capable hardware with abundant computational resources. Additionally, the diffusion processes in our work occur in the compressed latent space z rather than the high-dimensional pixel space. With an increasing bandwidth compression ratio, the spatial dimension P of the feature maps is significantly smaller than the source image, dramatically reducing the per-step operation count compared to traditional image diffusion models. Nevertheless, we note that the superior performance of the proposed scheme also comes with a trade-off. Diffusion-based models typically incur higher inference latency due to their iterative denoising process. This underscores the importance of studying novel techniques to significantly reduce the computational cost and accelerate diffusion models for practical applications. The recent potential methods can be dimensionality reduction [44], knowledge distillation [45], model quantization [46], and structural pruning [47].

V. CONCLUSION

In this paper, we have introduced ADDM, a novel diffusion model framework integrated into the vanilla Deep JSCC pipeline for adaptive channel denoising to enhance the system performance in wireless image transmission. Unlike existing methods, our proposed approach modifies both the forward diffusion and reverse diffusion processes with a degraded noisy signal. Additionally, by accommodating CSI through multi-head attention layers within a U-Net architecture, our proposed approach intelligently guides the generative denoising process at the receiver. This allows the system to tailor its reconstruction strategy to the specific level of channel impairment, relying more on its powerful generative prior in noisy conditions and preserving fine details when the channel is favorable. Through extensive simulations on ADE20K and ImageNet datasets under both AWGN and challenging Rayleigh fading channels, we have demonstrated the effectiveness of this approach. The ADDM-JSCC model consistently outperforms a range of specialized, fixed-SNR DeepJSCC models and addresses the critical need for a single, robust model capable of adapting to a diverse range of SNRs. Importantly, our model proves that it is not over-reliant on CSI, maintaining high performance even when channel information is unavailable or imperfectly estimated. Future work will focus on extending this adaptive diffusion-based approach to other cases, such as MIMO or multi-user communications, as well as investigating complexity reduction techniques to facilitate deployment in resource-constrained and latency-sensitive applications.

REFERENCES

- [1] E. C. Strinati and S. Barbarossa, "6G networks: Beyond shannon towards semantic and goal-oriented communications," *Comput. Netw.*, vol. 190, no. 8, pp. 1–17, May 2021.
- [2] C. E. Shannon, "A mathematical theory of communication," *The Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
- [3] E. Boursoulatz, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sep. 2019.
- [4] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 210–219, Feb. 2022.
- [5] K. Niu, J. Dai, S. Yao, S. Wang, Z. Si, X. Qin, and P. Zhang, "A paradigm shift toward semantic communications," *IEEE Commun. Mag.*, vol. 60, no. 11, pp. 113–119, Nov. 2022.
- [6] J. Xu, T.-Y. Tung, B. Ai, W. Chen, Y. Sun, and D. Gündüz, "Deep joint source-channel coding for semantic communications," *IEEE Commun. Mag.*, vol. 61, no. 11, pp. 42–48, Nov. 2023.
- [7] L. V. Nguyen, Y. Liu, Q. N. Le, and T. Q. Duong, "Physical layer security for semantic communications: Challenges and GenAI-based strategies," *IEEE Netw.*, Jun. 2026, early access, doi: 10.1109/MNET.2026.3700516.
- [8] 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 Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 213–250, Feb. 2023.
- [9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, Vancouver, BC, Canada, Dec. 2020, pp. 6840–6851.
- [10] M. Yang and H.-S. Kim, "Deep joint source-channel coding for wireless image transmission with adaptive rate control," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Singapore, May 2022.
- [11] 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.
- [12] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, "Wireless image transmission using deep source channel coding with attention modules," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2315–2328, Apr. 2021.
- [13] J. Pan, H. Li, G. Zhang, Y. Cai, and G. Yu, "Deep refinement-based joint source channel coding over time-varying channels," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Dubai, UAE, Apr. 2024, pp. 1–6.
- [14] L. V. Nguyen, T. T. Nguyen, O. A. Dobre, and T. Q. Duong, "Digital semantic communication with neural image compression," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM) Workshops*, London, UK, May 2025.
- [15] Y. Yan, L. Li, X. Zhang, W. Lin, W. Cheng, and Z. Han, "Adaptive semantic generation and NOMA-based interference-aware transmission for 6G networks," *IEEE Trans. Wireless Commun.*, vol. 24, no. 3, pp. 2404–2416, Mar. 2025.
- [16] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10850–10869, Mar. 2023.
- [17] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng, and S. Z. Li, "A survey on generative diffusion models," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 2814–2830, Feb. 2024.
- [18] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4713–4726, Apr. 2023.
- [19] B. Kavar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, New Orleans, LA, USA, Dec. 2022, pp. 23 593–23 606.
- [20] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proc. IEEE/CVF Int. Conf. on Comput. Vision (ICCV)*, Paris, France, Oct. 2023, pp. 3836–3847.
- [21] J.-M. Lemerrier, J. Richter, S. Welker, E. Moliner, V. Välimäki, and T. Gerkman, "Diffusion models for audio restoration: A review," *IEEE Signal Process. Mag.*, vol. 41, no. 6, pp. 72–84, Nov. 2024.
- [22] Y.-J. Lu, Z.-Q. Wang, S. Watanabe, A. Richard, C. Yu, and Y. Tsao, "Conditional diffusion probabilistic model for speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Singapore, May 2022, pp. 7402–7406.
- [23] P. Gonzalez, Z.-H. Tan, J. Østergaard, J. Jensen, T. S. Alstrøm, and T. May, "Investigating the design space of diffusion models for speech enhancement," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 32, pp. 4486–4500, Oct. 2024.
- [24] E. Moliner, J. Lehtinen, and V. Välimäki, "Solving audio inverse problems with a diffusion model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023.
- [25] K. Zhang, L. Li, W. Lin, Y. Yan, R. Li, W. Cheng, and Z. Han, "Semantic successive refinement: A Generative AI-aided semantic communication framework," *IEEE Trans. Cogn. Commun. Netw.*, vol. 11, no. 2, pp. 687–699, Apr. 2025.
- [26] P. Yang, G. Zhang, and Y. Cai, "Rate-adaptive generative semantic communication using conditional diffusion models," *IEEE Wireless Commun. Lett.*, vol. 14, no. 2, pp. 539–543, Feb. 2025.
- [27] M. Letafati, S. Ali, and M. Latva-Aho, "Conditional denoising diffusion probabilistic models for data reconstruction enhancement in wireless communications," *IEEE Trans. Mach. Learning Commun. Netw.*, vol. 3, pp. 133–146, Dec. 2025.
- [28] S. F. Yilmaz, X. Niu, B. Bai, W. Han, L. Deng, and D. Gündüz, "High perceptual quality wireless image delivery with denoising diffusion models," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM) Workshops*, Vancouver, BC, Canada, May 2024, pp. 1–5.
- [29] L. Qiao, M. B. Mashhadi, Z. Gao, C. H. Foh, P. Xiao, and M. Bennis, "Latency-aware generative semantic communications with pre-trained diffusion models," *IEEE Wireless Commun. Lett.*, vol. 13, no. 10, pp. 2652–2656, Oct. 2024.
- [30] B. Fesl, M. B. F. Strasser, M. Joham, and W. Utschick, "Diffusion-based generative prior for low-complexity MIMO channel estimation," *IEEE Wireless Commun. Lett.*, vol. 13, no. 12, pp. 3493–3497, Oct. 2024.
- [31] M. Kim, R. Fritschek, and R. F. Schaefer, "Learning end-to-end channel coding with diffusion models," in *Proc. Int. ITG Workshop Smart Antennas Conf. Syst., Commun., Coding (WSA & SCC)*, Braunschweig, Germany, Feb. 2023, pp. 1–6.
- [32] L. Guo, W. Chen, Y. Sun, B. Ai, N. Pappas, and T. Q. S. Quek, "Diffusion-driven semantic communication for generative models with bandwidth constraints," *IEEE Trans. Wireless Commun.*, vol. 24, no. 8, pp. 6490–6503, Aug. 2025.
- [33] B. Xu, S. Han, X. Xu, W. Li, R. Meng, C. Dong, and P. Zhang, "Semantic prior aided channel-adaptive equalizing and de-noising semantic communication system with latent diffusion model," *IEEE Trans. Wireless Commun.*, vol. 24, no. 6, pp. 4614–4630, Aug. 2025.
- [34] T. Wu, Z. Chen, D. He, L. Qian, Y. Xu, M. Tao, and W. Zhang, "CDDM: Channel denoising diffusion models for wireless semantic communications," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 11 168–11 183, Sep. 2024.
- [35] L. V. Nguyen, T. T. Nguyen, O. A. Dobre, and T. Q. Duong, "Leveraging stable diffusion with context-aware prompts for semantic communication," in *Proc. IEEE Int. Conf. Mobile Ad-Hoc Smart Syst. (MASS) Workshops*, Seoul, South Korea, Aug. 2024, pp. 610–615.
- [36] B. Xu, R. Meng, Y. Chen, X. Xu, C. Dong, and H. Sun, "Latent semantic diffusion-based channel adaptive de-noising semcom for future 6G systems," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Kuala Lumpur, Malaysia, Dec. 2023, pp. 1229–1234.
- [37] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 28, Dec. 2015.
- [38] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, Dec. 2021, pp. 8780–8794.
- [39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recogn. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
- [40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. in Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, Dec. 2017.
- [41] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20k dataset," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recogn. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5122–5130.
- [42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recogn. (CVPR)*, Miami, FL, USA, Jun. 2009, pp. 248–255.
- [43] D. Kinga, J. B. Adam *et al.*, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, vol. 5, no. 6, CA, USA, May 2015.
- [44] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc.*

- IEEE/CVF Conf. Comput. Vision Pattern Recogn. (CVPR)*, New Orleans, Louisiana, USA, Jun. 2022, pp. 10 684–10 695.
- [45] T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and W. T. Freeman, “Improved distribution matching distillation for fast image synthesis,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, Vancouver, BC, Canada, Dec. 2024, pp. 47 455–47 487.
 - [46] X. Li, Y. Liu, L. Lian, H. Yang, Z. Dong, D. Kang, S. Zhang, and K. Keutzer, “Q-Diffusion: Quantizing diffusion models,” in *Proc. IEEE/CVF Int. Conf. on Comput. Vision (ICCV)*, Paris, France, Oct. 2023, pp. 17 535–17 545.
 - [47] T. Castells, H.-K. Song, B.-K. Kim, and S. Choi, “LD-Pruner: Efficient pruning of latent diffusion models using task-agnostic insights,” in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recogn. (CVPR)*, Seattle, Washington, USA, Jun. 2024, pp. 821–830.