Residual Diffusion Models for Joint Source Channel Coding of MIMO CSI

Sravan Kumar Ankireddy [1] **Heasung Kim** [1] **Hyeji Kim** [1]

Abstract

Despite significant advancements in deep learning-based CSI compression, current approaches primarily view it as a source coding problem, neglecting transmission errors. Separate source and channel coding proves suboptimal in finite block length regimes, while autoencoder-based compression schemes struggle with complex channel distributions. We propose Residual-Diffusion Joint Source-Channel Coding (RD-JSCC), leveraging diffusion models to learn robust CSI representations. Our architecture combines a lightweight autoencoder with a residual diffusion module for iterative CSI reconstruction, enabling graceful performance degradation across variable SNR conditions and robust estimation under multipath fading in the uplink feedback channel. Our flexible decoding strategy dynamically selects between autoencoder decoding and diffusion-based refinement based on channel conditions, minimizing the overall computational complexity. Simulations demonstrate RD-JSCC significantly outperforms existing approaches in challenging wireless environments, without adding substantial decoding latency via a two-step inference, offering an efficient solution for next-generation wireless systems.

1. Introduction

As data transmission volumes continue to increase, massive multiple-input multiple-output (MIMO) has emerged as a fundamental technology for scaling next-generation wireless networks. By employing a large array of antennas at the base station (BS), multiple user equipment (UE) can achieve high-throughput communication, even under suboptimal channel conditions. However, achieving this requires accurate channel state information (CSI) to enable effective precoding for downlink transmission. In frequency division duplexing (FDD) systems, the uplink CSI is obtained via channel estimation, while the downlink CSI must be fed back from the user equipment (UE) in an efficient manner (Sim et al., 2016).

Deep learning (DL) has significantly advanced various areas within physical layer communication, including nonlinear channel code design (Kim et al., 2018; Makkuva et al., 2021; Jamali et al., 2022; Ankireddy et al., 2024; 2025), neural channel decoding (Nachmani et al., 2016; Shlezinger et al., 2020; Choukroun & Wolf, 2022; Hebbar et al., 2022; Ankireddy & Kim, 2023; Hebbar et al., 2024), and MIMO channel estimation (Wen et al., 2018; Chun et al., 2019; Soltani et al., 2019). This work focuses specifically on the challenge of *lossy compression* of CSI at the physical layer. Traditional compression techniques, such as compressed sensing (Kuo et al., 2012), do not work well CSI compression due to the lack of inherent sparsity in CSI structures, making deep learning a better alternative. The field of lossy compression using neural networks, commonly referred to as *neural lossy compression*, has gained considerable attention in applications such as image compression (Ballé et al., 2016; 2018; Li et al., 2023b) and video compression (Li et al., 2023a). More recently, similar methodologies have been leveraged to significantly enhance the efficiency of CSI compression (Guo et al., 2022), starting with CSINet (Wen et al., 2018), which achieved substantial improvements over the then state-of-the-art compressed sensing methods by leveraging convolutional neural networks (CNNs). This breakthrough led to a series of subsequent studies that further refined CNN-based compression techniques (Wang et al., 2019; Li et al., 2020; Liu & Simeone, 2021; Lu et al., 2020; Kim et al., 2022). Further, similar approaches have been adapted for joint source channel coding (JSCC) of CSI (Xu et al., 2022).

Recently, neural image compression has witnessed a significant breakthrough in both compression efficiency and reconstruction quality with the adoption of diffusion models (Ho et al., 2020; Song et al., 2020). Originally designed to generate novel images based on various conditioning variables, these models were rapidly adapted for image compression. In (Yang & Mandt, 2023), the authors introduced a diffusion-based compression framework that reconstructs images through a reverse diffusion process

[1]Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA. Correspondence to: Sravan Kumar Ankireddy <sravan.ankireddy@utexas.edu>.

conditioned on contextual information, outperforming certain GAN-based methods. Further, in (Careil et al., 2023), an extremely low rate compression scheme was developed by leveraging the strong image priors of pretrained diffusion models.

While diffusion models have recently garnered considerable attention in image compression, their potential for CSI compression remains comparatively underexplored. A notable advancement in this direction is the generative diffusion-based CSI compression framework proposed in (Kim et al., 2025), demonstrating significant improvements over traditional autoencoder-based methods. Building upon this promising foundation, our work aims to further enhance diffusion-based CSI compression in two key dimensions. First, we prioritize reconstruction fidelity by balancing output diversity, shifting the model's objective from generating novel samples towards maximizing reconstruction accuracy. Next, instead of assuming an ideal noiseless feedback, we simulate a realistic multi-path fading channel to learn robust representations, optimizing the performance end-to-end.

In this work, we propose a novel residual diffusion-based CSI compression framework tailored to compress CSI measurements in massive MIMO systems efficiently. The proposed framework features an encoder leveraging a low-complexity convolutional neural network (CNN) architecture and a two-stage decoding process at the receiver. Specifically, the decoder comprises an initial CNN-based reconstruction stage followed by a U-Net-based diffusion refinement model, progressively enhancing CSI reconstruction quality. The main contributions of this paper are summarized as follows:

- We propose a residual-diffusion-based JSCC scheme that enhances the CSI reconstruction by initializing the reverse diffusion with a coarse CSI estimate, rather than conventional random Gaussian noise.
- We propose a flexible two-stage decoding framework that adaptively switches between low-complexity autoencoder and high-fidelity diffusion-based decoders, based on channel conditions (Sec. 4).
- We validate the efficacy of our proposed method by conducting performance evaluations against state-of-the-art deep learning-based JSCC schemes for CSI compression, using the widely recognized COST2100 outdoor dataset (Wen et al., 2018) and 3GPP indoor dataset using QuaDRiGa (Jaeckel et al., 2017) (Sec. 5).
- We perform a systematic comparative study across multiple neural architectures and diverse channel scenarios, revealing that diffusion-based models significantly outperform autoencoder-based methods only when channel complexity is sufficiently high (Sec. 6).

2. System Model and Problem Formulation

In this work, we consider a massive MIMO system operating in frequency division duplex (FDD) mode, where a base station (BS) with N_t antennas communicates with a user equipment (UE) equipped with N_r antennas. Considering the large-scale nature of MIMO, we assume $N_t \gg 1$ and set $N_r = 1$ for simplicity. The downlink CSI is $H_d \in \mathcal{C}^{N_c \times N_t}$ and the uplink CSI is denoted by $H_u \in \mathcal{C}^{N_c \times N_t}$ in the spatial-frequency domain, where N_c is the number of subcarriers.

The encoder f_{enc} at the UE is designed to efficiently compress the high-dimensional channel measurement H_d into a fixed-length representation $\mathbf{s} \in \mathcal{C}^k$. The compression rate is thus given by

$$r = \frac{k}{N_t \times N_c}.$$

The compressed representation can be transmitted on a noisy uplink channel using k subcarriers, while imposing an unit power constraint for the subcarriers $\frac{1}{k}\mathbb{E}\left[\mathbf{s}\,\mathbf{s}^*\right] = 1$. The received signal at the BS is processed using maximal ratio combining (MRC).

The compressed representation at the receiver is processed in two stages. First, the decoder f_{dec} reconstructs an estimate $\hat{H}_d$ from the compressed representation $\mathbf{s}$, aiming to minimize distortion relative to the original input H_d. Next, to further refine the reconstruction, the noisy estimate $\hat{H}_d$ is processed by a denoising diffusion model f_{den}, iteratively reducing the distortion using the reverse diffusion. Specifically, we use the residual diffusion formulation (Liu et al., 2024). Following standard practice in CSI compression literature, we adopt mean squared error (MSE) as the distortion metric.

The encoder function is defined as $f_{\text{enc}} : H_d \mapsto \mathbf{s}$, while the decoder function is given by $f_{\text{dec}} : \mathbf{s} \mapsto \hat{H}_d$, and the denoising model operates as $f_{\text{den}} : \hat{H}_d \mapsto H'_d$. Given the encoder parameters θ_{enc}, the compressed representation $\mathbf{s}$ is obtained as $\mathbf{s} = f_{\text{enc}}(H_d; \theta_{\text{enc}})$. Similarly, the decoder, parameterized by θ_{dec}, reconstructs an estimate of the target as $\hat{H}_d = f_{\text{dec}}(\mathbf{s}; \theta_{\text{dec}})$. Finally, the denoising model, governed by θ_{den}, further enhances the reconstruction, producing $H'_d = f_{\text{den}}(\hat{H}_d; \theta_{\text{den}})$. The complete set of model parameters is thus given by $\theta = (\theta_{\text{enc}}, \theta_{\text{dec}}, \theta_{\text{den}})$.

The learning process is designed to minimize the following objective function:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{p(H_d, H'_d)}\left[d(H'_d, f_{\text{den}}(f_{\text{dec}}(f_{\text{enc}}(H_d; \theta_{\text{enc}}); \theta_{\text{dec}})))\right] \tag{1}$$

where $d(\cdot, \cdot)$ represents a distortion metric that quantifies reconstruction quality. The parameters θ =

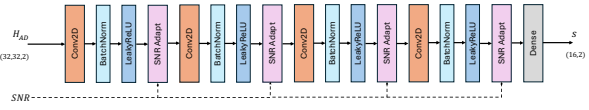
$(\theta_{\text{enc}}, \theta_{\text{dec}}, \theta_{\text{den}})$ are optimized using a two-stage training process, detailed in Sec. 4.

3. Deep Learning for CSI compression

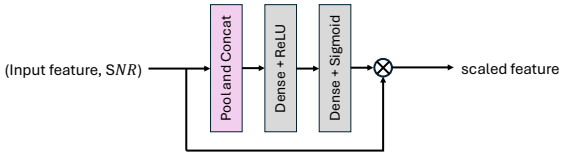
In this section, we introduce a low-complexity autoencoder-based solution for CSI compression. By considering the task of CSI compression as an image compression problem, several works based on an autoencoder structure were proposed (Wen et al., 2018; Lu et al., 2018; Guo et al., 2020; Lu et al., 2020; Hu et al., 2021; Chen et al., 2021; Cao et al., 2021; Ji & Li, 2021). Notable works among them include CSINet (Wen et al., 2018), which was one of the first deep learning approaches proposed and outperformed compressed sensing baselines such as LASSO and BM3D-AMP. CRNet (Lu et al., 2020) proposed a multi-resolution deep learning framework for CSI feedback in massive MIMO systems, enabling scalable compression across different feedback overheads. Recently, a transformer-based architecture that utilizes stripe-wise spatial features was introduced in (Hu et al., 2023) to enhance the efficiency of CSI compression in massive MIMO systems.

3.1. Autoencoder based CSI compression

Given the three-dimensional spatially correlated structure of the CSI matrix, using a convolution-based architecture is an efficient choice for compressing the input. We chose a low-complexity design for the auto-encoder. The encoder is implemented as lightweight CNN layers followed by a fully connected layer. To maintain robustness under changing SNR, we use an SNR-adaptation module that dynamically scales the feature activations based on SNR. The complete architecture of both the encoder and the SNR-adaptation block is shown in Fig. 1.



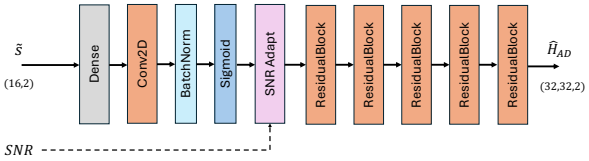
(a) CNN encoder



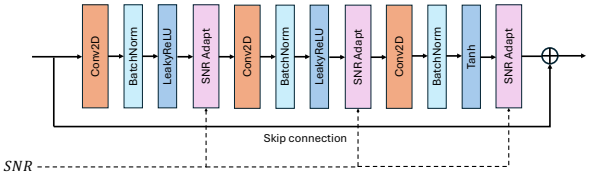
(b) Feature scaling for SNR adaptation

Figure 1: Low-complexity CSI encoder.

The decoder employs a moderately deeper architecture to extract richer representations. It starts with a fully connected layer, followed by an initial convolutional layer and a stack of five residual blocks that enhance feature propagation and stabilize training. An identical SNR-adaptation module is employed in the decoder, mirroring the encoder design. A schematic of the decoder, including the residual block layout, is provided in Fig. 2.



(a) CNN decoder



(b) Residual connection

Figure 2: Low-complexity CSI decoder.

We train the autoencoder in a supervised manner to learn an effective low-complexity compression scheme. The encoder compresses the continuous channel measurement H_d into a compact latent vector $\mathbf{s} \in \mathbb{C}^k$. The decoder reconstructs the approximate channel measurement $\hat{H}_d$. The encoder and decoder are jointly optimized in an end-to-end fashion to minimize the reconstruction loss for CSI. We use the MSE loss to enforce reconstruction fidelity by ensuring that the reconstructed output closely matches the input, given by:

$$\mathcal{L}_{\text{MSE}}(H_d, \hat{H}_d) = \|H_d - \hat{H}_d\|^2, \tag{2}$$

where $\|\cdot\|$ denotes the Frobenius norm.

Several works have demonstrated that CNN–based autoencoders can effectively minimize the reconstruction NMSE for both standalone CSI compression (Wen et al., 2018; Hu et al., 2023) and joint source–channel coding of CSI (Xu et al., 2022). Although this supervised formulation suffices for relatively simple channel distributions, it struggles when the underlying channel distribution becomes more complex. Recent work on CSI compression over the COST2100 outdoor channel (Kim et al., 2025) demonstrated that generative diffusion models significantly outperform conventional CNN autoencoders in such challenging settings. Motivated by these findings, we augment our autoencoder with a diffusion refinement stage at the base station. This hybrid design invokes the diffusion module only when the channel complexity warrants it, thereby delivering superior reconstruction quality without incurring unnecessary computational overhead in simpler channel conditions.

4. Residual Diffusion for CSI Enhancement

In this section, we introduce a framework for enhancing CSI reconstruction at the receiver using residual diffusion (Liu et al., 2024). Unlike conventional generative diffusion-based approaches, the proposed method can be seamlessly integrated with both learning-based and traditional non-learning-based techniques, as the diffusion module is trained as a standalone denoising model, making it more suitable for practical adaptation.

Preprocessing. To reduce computational complexity, we first transform the complex matrix H_d from the spatial-frequency domain to the angular-delay domain by applying a two-dimensional inverse fast Fourier transform (2D IFFT). This transformation exploits the inherent sparsity in the angular-delay domain, supported by established assumptions (Wang et al., 2018a). Subsequently, we retain only the first 32 elements along the delay dimension, as the remaining coefficients typically approach zero, yielding compact angular-delay domain representations of H_d. The original CSI matrices can then be reconstructed by appending zero matrices of size $32 \times (N_c - 32)$ and performing a 2D FFT. This preprocessing procedure is widely recognized as an efficient CSI representation technique (Wen et al., 2018; Wang et al., 2018b; Lu et al., 2020). Note that for the COST2100 outdoor dataset, the performance evaluations reported in this work are conducted in terms of normalized mean squared error (NMSE) within the cropped angular-delay domain. However, for other datasets where spatial-frequency domain data is available, we convert the estimated channel matrices from the angular-delay domain back to the spatial-frequency domain, and subsequently present the final NMSE results in that domain.

The receiver first obtains a coarse estimate of the channel using the autoencoder described in Sec. 3.1, which we refer to as Stage 1. Unlike certain CSI compression methods that quantize the latent representation into a fixed number of bits using vector quantization, we instead map the latent to a fixed-length continuous vector. This continuous representation allows direct mapping of encoded data to the uplink subcarriers and simulates complex channel impairments such as multi-path within the feedback channel.

In Stage 2, the denoising process leverages the output of the autoencoder to initialize the reverse diffusion procedure with the coarse CSI estimate. Following standard practice in the generative modeling literature, we adopt the U-Net architecture (Ronneberger et al., 2015) as the backbone for the denoising network. Unlike recent generative diffusion-based CSI reconstruction methods (Kim et al., 2025), which initialize reverse diffusion from random Gaussian noise, we employ a residual diffusion approach that iteratively refines the initial CSI estimate, as illustrated in Fig. 3.

While initializing with random noise facilitates the generation of diverse samples from the underlying distribution—an objective well-suited for data synthesis—it is suboptimal for compression and reconstruction tasks, where compressed latent features already provide a strong prior. Residual diffusion addresses this by starting from a coarse channel estimate, leading to more accurate reconstructions. The effectiveness of residual diffusion for reconstruction tasks has been well demonstrated in the context of image compression (Li et al., 2024). We now formally describe the complete CSI compression and reconstruction pipeline.

In Stage 1, the encoder produces a compressed representation $\mathbf{s}$, which is then used to obtain a coarse reconstruction $\hat{H}_d$ of the input channel H_d. During the denoising stage, the objective is to further refine $\hat{H}_d$ by sampling $\mathbf{Z} \sim p(\mathbf{z} \mid \hat{H}_d)$. This leads to the formulation of a residual denoising diffusion process, given by

$$p(\mathbf{z}_{0:T} \mid \hat{H}_d) = p(\mathbf{z}_T) \prod_{t=1}^{T} \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t, \hat{H}_d, t), \beta_t \mathbf{I}), \tag{3}$$

where $\mathbf{z}_{0:T} = (\mathbf{z}_0, \ldots, \mathbf{z}_T)$ denotes a realization of the stochastic process $(\mathbf{Z}_0, \ldots, \mathbf{Z}_T)$, μ_θ is the learnable mean function parameterized by θ, β_t defines the variance schedule at each time step t, and $\mathbf{I}$ is the identity matrix.

In a generative diffusion-based approach, the forward process is defined by progressively adding Gaussian noise with variance $\beta_t \in (0, 1)$ to the clean latent features $\mathbf{z}_0$ according to a predefined schedule, as given by

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_t, \quad t = 1, 2, \ldots, T, \tag{4}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, I)$, $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. As t increases, the noisy latent variable $\mathbf{z}_t$ gradually converges to a standard Gaussian distribution. Typically, T is chosen as a relatively large value (e.g., 20–50), with the reverse diffusion process initialized from pure Gaussian noise. However, in the context of CSI compression, this strategy is suboptimal, as a coarse channel estimate $\hat{H}_d$ can be readily obtained using either conventional compression techniques (Sim et al., 2016) or low-complexity deep learning-based methods (Sun et al., 2024).

To exploit the availability of the coarse estimate $\hat{H}_d$, we adopt a residual diffusion approach as presented in (Li et al., 2024), which modifies the initialization step as

$$\mathbf{z}_N = \sqrt{\bar{\alpha}_N}\hat{H}_d + \sqrt{1 - \bar{\alpha}_N}\boldsymbol{\epsilon}_N, \tag{5}$$

where $N \ll T$. This leads to the following residual diffusion formulation:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\,(\mathbf{z}_0 + \eta_t \mathbf{r}) + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_t, \quad t = 1, 2, \ldots, N, \tag{6}$$

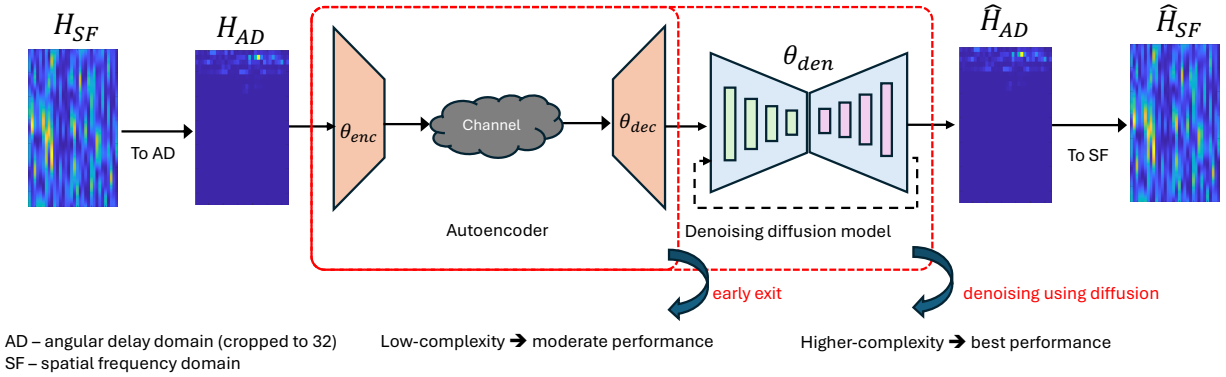


Figure 3: A lightweight autoencoder compresses the MIMO CSI matrix into a low-dimensional latent representation, which is transmitted over a noisy channel. Decoding occurs in two stages: (1) a low-complexity decoder produces a coarse reconstruction, and (2) a diffusion-based denoising model refines the output for enhanced quality.

where $\mathbf{r}$ denotes the residual between the clean channel $\mathbf{z}_0 = H_d$ and the coarse estimate $\hat{H}_d$, i.e., $\mathbf{r} = \hat{H}_d - \mathbf{z}_0$. The weighting sequence $\{\eta_t\}_{t=1}^N$ is designed such that $\eta_1 \to 0$ and $\eta_N = 1$.

Since the residual $\mathbf{r}$ is not available during inference, residual diffusion assumes a linear relationship among $\mathbf{z}_{t-1}$, $\mathbf{z}_t$, and $\mathbf{z}_0$, analogous to the DDIM framework (Song et al., 2020), given by

$$\mathbf{z}_{t-1} = k_t \mathbf{z}_0 + m_t \mathbf{z}_t + \sigma_t \boldsymbol{\epsilon}, \tag{7}$$

where $\sigma_t = 0$ for simplicity and k_t and m_t are weighing coefficients from (Li et al., 2024) .Combining (6) and (7) yields

$$\frac{\eta_t}{\eta_{t-1}} = \frac{\sqrt{1-\bar{\alpha}_t}/\sqrt{\alpha_t}}{\sqrt{1-\bar{\alpha}_{t-1}}/\sqrt{\alpha_{t-1}}} \quad \Rightarrow \quad \eta_t = \lambda \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\alpha_t}}, \tag{8}$$

where the scaling factor λ is set to $\frac{\sqrt{\bar{\alpha}_N}}{\sqrt{1-\bar{\alpha}_N}}$ to satisfy the condition $\eta_N = 1$.

Substituting (8) into (6), the forward diffusion process can be expressed as

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\left(\mathbf{z}_0 + \lambda \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\mathbf{r}\right) + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, \tag{9}$$

which simplifies to

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}(\lambda \mathbf{r} + \boldsymbol{\epsilon}_t). \tag{10}$$

Ultimately, the denoising network is trained to recover the clean channel $\mathbf{z}_0$ from noisy observations at various noise levels encountered during the forward diffusion process. The corresponding diffusion loss is formulated as

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{Z}_0, \mathbf{T}}\left[\frac{\bar{\alpha}_T}{1-\bar{\alpha}_T}\left\|\mathbf{Z}_0 - f_{\text{den}}(\mathbf{Z}_T, \hat{H}_d, T; \theta_{\text{den}})\right\|^2\right], \tag{11}$$

Algorithm 1 Training the autoencoder model

Input: Initial model $(\theta_{\text{enc}}, \theta_{\text{dec}})$, Adam optimizer
Output: Updated model $(\theta_{\text{enc}}, \theta_{\text{dec}})$

for $i = 0$ to N_{train} **do**
  Sample H_d from the channel distribution
  $\mathbf{s} = f_{\text{enc}}(H_d; \theta_{\text{enc}})$
  $\hat{H}_d = f_{\text{dec}}(\mathbf{s}; \theta_{\text{dec}})$
  $\mathcal{L}_{\text{MSE}} = \|H_d - \hat{H}_d\|^2$
  $\text{Adam}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{L}_{\text{MSE}})$
end for

where $f_{\text{den}}(\cdot)$ is the denoising network parameterized by θ_{den}.

Diffusion with χ-prediction. In Equation (11), the loss is computed between the clean image and the model's prediction, compelling the network to directly predict the image rather than the conventional approach of predicting the added noise. This technique, known as χ-*prediction*, proves particularly advantageous when operating with a limited number of denoising steps (e.g., $T < 50$). As our experimental results in subsequent sections demonstrate, this approach facilitates low-latency inference by enabling a transition to one-step decoding during inference with minimal or no degradation in reconstruction quality.

4.1. Training

We propose a two-stage training strategy to optimize performance while maintaining computational efficiency. In the first stage, the autoencoder model is trained using the MSE loss (2) between the input CSI matrix and the estimated CSI matrix. Due to the relatively small number of parameters and faster convergence, the computational overhead for Stage 1 is minimal. The detailed training procedure for this stage is outlined in Alg. 1.

Algorithm 2 Training the denoising U-Net

Input: Pretrained $(\theta_{\text{enc}}, \theta_{\text{dec}})$. Initial parameters for denoising network θ_{den}, variance schedule $\{\bar{\alpha}_t\}_{t=0}^{T}$, Adam optimizer

Output: Updated denoising network θ_{den}

for $i = 0$ to N_{train} **do**
  Sample H_d form the channel distribution
  $\mathbf{s} = f_{\text{enc}}(H_d; \theta_{\text{enc}})$
  $\hat{H}_d = f_{\text{dec}}(\mathbf{s}; \theta_{\text{dec}})$
  $\mathcal{L}_{\text{diff}} = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t} \|\mathbf{z}_0 - f_{\text{den}}(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}(\lambda \mathbf{r} + \boldsymbol{\epsilon}_t))\|^2$
  Adam$(\theta_{\text{den}}, \mathcal{L}_{\text{diff}})$
end for

In the second stage, the weights of the autoencoder model are frozen, while a residual conditional denoising diffusion model, based on the U-Net architecture, is trained. This training process optimizes the diffusion loss defined in (11), essentially training a denoising network. The detailed training procedure for this stage is outlined in Alg. 2.

5. Experimental Setup and Results

5.1. Baselines and comparison.

To evaluate our diffusion-based approach, we benchmark against ADJSCC (Xu et al., 2022), a recent state-of-the-art non-linear transform method. ADJSCC employs a neural network to transform CSI information from the spatial frequency domain to a low-dimensional representation—bypassing the traditional IFFT approach—before further compression via a secondary network. We also include the deep JSCC variant of CSINet+ (Guo et al., 2020), another widely recognized benchmark in CSI compression literature. To maintain fairness in comparison, we utilize identical datasets and training protocols across all evaluated models.

We implement two variants of the ADJSCC baseline: (1) the original architecture as presented in (Xu et al., 2022), and (2) a parameter-scaled version that matches the complexity of our proposed RD-JSCC model, by increasing the layers and channel dimensions. Additionally, we examine a supervised variant of RD-JSCC, where both the autoencoder and U-Net components are trained end-to-end using an MSE loss function. Our primary contribution, RD-JSCC—which integrates an autoencoder with diffusion-based U-Net training—demonstrates performance improvements of an order of magnitude compared to (Xu et al., 2022) at equivalent NMSE targets.

5.2. Dataset

We evaluate our method on the COST2100 outdoor dataset (Liu et al., 2012), which is widely adopted due to its realistic and complex channel characteristics. The dataset provides 10^5 training samples and 2×10^4 test samples. It is important to note that the original spatial-frequency domain representations are not publicly available; instead, we utilize the provided 32×32 cropped complex channel matrices in the angular-delay domain. Consequently, we omit the non-linear transformation module used in ADJSCC and retain only the inner encoder-decoder components that operate directly on the 32×32 angular-delay domain complex inputs. Since the input dimensions of the inner module match those of the COST2100 cropped channels, this setup enables a fair and consistent comparison. All evaluations are reported in terms of NMSE measured in the angular-delay domain.

To simulate a realistic feedback channel, we generate uplink channel realizations using QuaDRiGa (Jaeckel et al., 2017), adhering to the 3GPP TR 38.901 channel specification (TR), with an uplink carrier frequency of 5.4 GHz. We adopt the simulation configuration outlined in (Xu et al., 2022) and assume line-of-sight (LOS). The base station (BS) is positioned at the center of a 20 m $\times$ 20 m area and is equipped with a uniform linear array (ULA) comprising $N_{\text{t}} = 32$ omnidirectional elements spaced at half-wavelength intervals. The user equipment (UE) employs a single omnidirectional antenna ($N_{\text{r}} = 1$). The antenna heights are set to 3 m at the BS and 1.5 m at the UE. The simulation includes $N_c = 32$ subcarriers in the uplink transmission.

5.3. Model configuration

Encoder. To ensure a low-complexity design for the UE, we choose a small number of convolution channels and do not utilize any residual connections in the encoder architecture. The detailed architectural choices are summarized in Tab. 1.

Layer	Channels	Kernel size
Input Conv	2	(11,11)
Conv layer 1	32	(9,9)
Conv layer 2	48	(7,7)
Output Conv	2	(5,5)

Table 1: CNN encoder.

Decoder. The receiver employs a series of residual blocks with varying channel widths and kernel sizes. This deeper architecture, combined with residual connections, enhances reconstruction performance while introducing only a modest increase in decoding latency. The detailed

architectural choices are summarized in Tab. 2.

Layer	Channels	Kernel size
Input Conv	2	(7,7)
Residual Block Conv layer 1	16	(7,7)
Residual Block Conv layer 2	24	(5,5)
Residual Block Conv layer 3	2	(3,3)

Table 2: Residual Block decoder.

Diffusion. The architecture for the diffusion denoising network is based on (Yang & Mandt, 2023) and (Kim et al., 2025). We choose an initial embedding width of 64 channels and dimension multipliers $\{1, 2, 3, 4\}$ for the successive down-sampling and mirrored up-sampling stages. Skip connections link each down-sampling block to its symmetric up-sampling counterpart, preserving high-resolution features throughout the reverse diffusion trajectory. The U-Net outputs a refined estimate of the CSI matrix that is fed back, iteratively refining the coarse estimate, based on the number of steps used in reverse diffusion. Th edetails of Hyperparametrs are provided in Appendix.

5.4. Results

In Fig. 4, we present the NMSE in reconstruction for the angular-delay domain CSI measurements of the COST2100 outdoor dataset. We configured the feedback bandwidth to $k = 16$, compressing each 32×32 CSI measurement (1024 complex coefficients) to a 16-dimensional latent vector, achieving a compression rate of 164.

Our evaluation begins with the exact architectures from ADJSCC (Xu et al., 2022) and CSINet+ (Guo et al., 2020), which demonstrate notably poor performance. When scaling these architectures by incorporating additional intermediate layers and increasing convolutional channels, performance improves marginally, but all schemes still saturate at an NMSE greater than -4 dB. This reveals that conventional autoencoder approaches trained with supervised learning objectives exhibit poor scaling characteristics relative to model size and fail to adequately capture the underlying channel characteristics despite substantial parameter counts.

We then developed a hybrid architecture combining the original autoencoder model from (Xu et al., 2022) with a U-Net backbone for denoising. The U-Net backbone has been widely employed in image processing for enhanced denoising, de-blurring, and more recently, image generation. Our hybrid model, termed U-Net based JSCC, was trained with the same supervised training objective as (Xu et al., 2022), minimizing the end-to-end MSE. Remarkably, despite having a similar parameter count as the larger variants of ADJSCC and CSINet+, the U-Net based JSCC

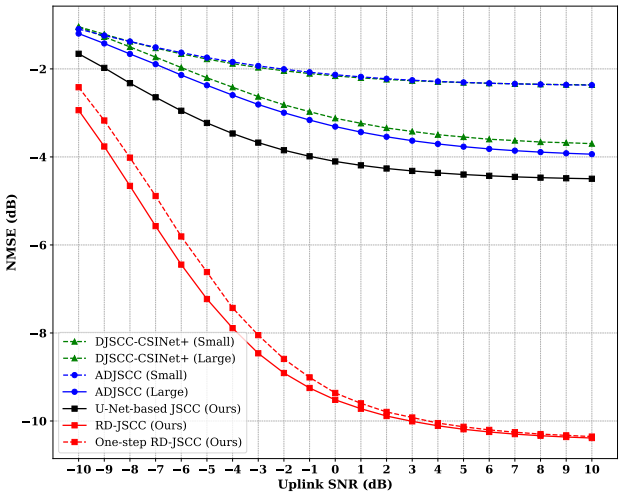


Figure 4: RD-JSCC achieves an order-of-magnitude improvement in performance over state-of-the-art deep JSCC baselines on the COST2100 outdoor dataset for feedback bandwidth of $k = 16$.

shows significant performance improvements, achieving an NMSE lower than -9 dB. This highlights the importance of architectures that scale effectively with increasing model size.

Finally, we trained the U-Net based JSCC model with a diffusion objective instead of the supervised objective. This approach, which we refer to as RD-JSCC, achieves the best performance among all evaluated schemes. During decoding at the receiver, the autoencoder head first produces a coarse estimate of the CSI. The U-Net model then initializes the reverse diffusion with this coarse estimate and refines the CSI at each denoising step, iteratively feeding the improved estimate back into the U-Net until completing the specified number of denoising steps. While our default configuration uses 20 denoising steps, we found that to minimize inference latency, a 2-step reverse diffusion performs remarkably well with only a negligible NMSE penalty, achieving an NMSE below -12 dB. These findings underscore the fundamental advantage of diffusion-based modeling for complex channel distributions where traditional supervised approaches reach their representational limits.

6. Channel Distribution vs. Model Complexity

While it is clear that diffusion models have the potential to enhance CSI reconstruction quality at the base station significantly, they also introduce substantial computational complexity. Therefore, it is crucial to justify this added complexity by employing diffusion-based refinement only when truly necessary. In the current state-of-the-art diffusion-based CSI compression method (Kim et al.,

2025), the same network architecture is applied across both the Clustered Delay Line (CDL) and the more challenging COST2100 outdoor datasets, despite the latter being considerably more complex. Moreover, the comparisons to existing baselines do not account for the substantial differences in model size. For example, baseline models used in (Kim et al., 2025) such as CSINet and CRNet typically have around 400K parameters, whereas the generative diffusion-based approach presented utilizes approximately 15M parameters. This significant disparity in parameter count complicates the assessment of whether performance gains arise from the diffusion modeling itself or simply from increased model size.

In this section, we conduct a systematic ablation study to isolate the benefits of diffusion modeling and to understand its advantages relative to the underlying channel complexity.

Architecture Choices. We evaluate three different architectures for this study. First, we consider a simple convolutional autoencoder trained in a supervised manner. Second, we consider an autoencoder followed by a U-Net denoising network, also trained end-to-end with supervised loss. Finally, we evaluate the RD-JSCC model, where the autoencoder and U-Net are trained jointly using the residual diffusion objective. In all three cases, the architectures are appropriately scaled to maintain comparable parameter counts across models and the details are provided in Appendix.

Channel Model Choices. In addition to the COST2100 outdoor scenario (Liu et al., 2012) analyzed in Sec. 5, we now evaluate performance under the indoor open-area channel model specified in 3GPP TR 38.901 (TR). This environment is characterized by significantly lower spatial complexity and information density than COST2100, providing a useful contrast in dataset complexity.

Analysis of results in Figure 5 reveals that the performance gap between our diffusion-based approach and autoencoder baselines narrows considerably for the 3GPP indoor dataset, in stark contrast to the COST2100 outdoor results presented in Figure 4. This observation highlights an important nuance: Despite the general superiority of diffusion-based CSI compression, the complexity-performance trade-off becomes less favorable when modeling simpler channel environments, where conventional low-complexity autoencoder architectures may offer a more efficient solution.

Notably, the flexible formulation of our residual diffusion framework provides an additional operational advantage, allowing for early termination of the decoding process after stage 1, regardless of prevailing channel conditions, thereby offering adaptive decoding complexity scaling on application requirements.

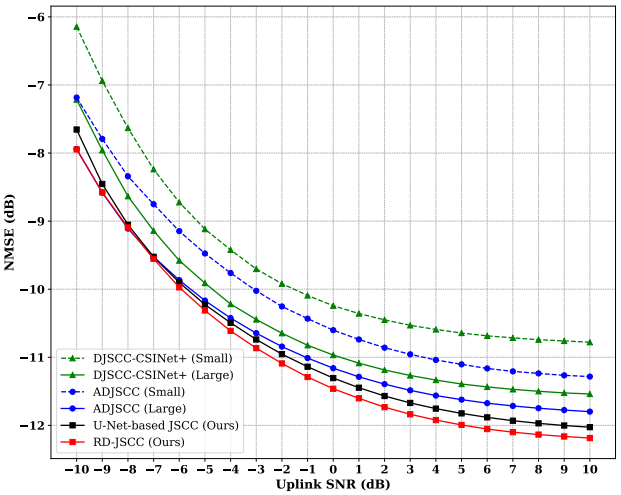


Figure 5: Under low-complexity channel conditions, such as the 3GPP indoor scenario, the performance gap between diffusion-based and autoencoder-based approaches narrows significantly.

7. Conclusion and Remarks

In this work, we introduce a hybrid JSCC scheme for MIMO CSI compression combining autoencoder-based initial estimation with diffusion-based refinement. Our residual diffusion approach initializes reverse diffusion with a coarse autoencoder estimate, tailoring the reverse diffusion specifically for reconstruction. This two-stage framework can be exited after either stage based on channel conditions and performance requirements, balancing computational complexity against fidelity. Further, using *x-prediction*, we enable single-step diffusion inference with minimal performance loss, substantially reducing latency. Our experiments across varying channel complexities show diffusion-based refinement delivers optimal value for complex channel distributions, while autoencoder solutions efficiently serve simpler channel distributions. These findings highlight diffusion models' potential for enhancing CSI reconstruction at base stations under challenging uplink conditions. Future work includes developing multi-rate compression within a single model, investigating quantization effects, exploring weight quantization techniques, and optimizing the U-Net backbone for improved computational efficiency while preserving performance.

References

Ankireddy, S. K. and Kim, H. Interpreting Neural Min-Sum Decoders. In *ICC 2023-IEEE International Conference on Communications*, pp. 6645–6651. IEEE, 2023.

Ankireddy, S. K., Hebbar, S. A., Wan, H., Cho, J., and Zhang, C. Nested construction of polar codes via transformers. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pp. 1409–1414. IEEE, 2024.

Ankireddy, S. K., Narayanan, K., and Kim, H. LightCode: Light analytical and neural codes for channels with feedback. *IEEE Journal on Selected Areas in Communications*, 2025.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Cao, Z., Shih, W.-T., Guo, J., Wen, C.-K., and Jin, S. Lightweight convolutional neural networks for csi feedback in massive mimo. *IEEE Communications Letters*, 25(8):2624–2628, 2021.

Careil, M., Muckley, M. J., Verbeek, J., and Lathuilière, S. Towards image compression with perfect realism at ultra-low bitrates. In *The Twelfth International Conference on Learning Representations*, 2023.

Chen, X., Deng, C., Zhou, B., Zhang, H., Yang, G., and Ma, S. High-accuracy csi feedback with super-resolution network for massive mimo systems. *IEEE Wireless Communications Letters*, 11(1):141–145, 2021.

Choukroun, Y. and Wolf, L. Error correction code transformer. *Advances in Neural Information Processing Systems*, 35:38695–38705, 2022.

Chun, C.-J., Kang, J.-M., and Kim, I.-M. Deep learning-based channel estimation for massive MIMO systems. *IEEE Wireless Communications Letters*, 8(4):1228–1231, 2019.

Guo, J., Wen, C.-K., Jin, S., and Li, G. Y. Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis. *IEEE Transactions on Wireless Communications*, 19(4):2827–2840, 2020.

Guo, J., Wen, C.-K., Jin, S., and Li, G. Y. Overview of deep learning-based CSI feedback in massive MIMO systems. *IEEE Transactions on Communications*, 70(12):8017–8045, 2022.

Hebbar, S. A., Mishra, R. K., Ankireddy, S. K., Makkuva, A. V., Kim, H., and Viswanath, P. TinyTurbo: Efficient Turbo Decoders on Edge. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2797–2802. IEEE, 2022.

Hebbar, S. A., Ankireddy, S. K., Kim, H., Oh, S., and Viswanath, P. DeepPolar: Inventing Nonlinear Large-Kernel Polar Codes via Deep Learning. *arXiv preprint arXiv:2402.08864*, 2024.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hu, Q., Kang, H., Chen, H., Huang, Q., Zhang, Q., and Cheng, M. Csi-stripeformer: Exploiting stripe features for csi compression in massive mimo system. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pp. 1–10. IEEE, 2023.

Hu, Z., Guo, J., Liu, G., Zheng, H., and Xue, J. Mrfnet: A deep learning-based csi feedback approach of massive mimo systems. *IEEE Communications Letters*, 25(10): 3310–3314, 2021.

Jaeckel, S., Raschkowski, L., Börner, K., Thiele, L., Burkhardt, F., and Eberlein, E. Quadriga-quasi deterministic radio channel generator, user manual and documentation. *Fraunhofer Heinrich Hertz Institute, Tech. Rep. v2. 0.0*, 2017.

Jamali, M. V., Saber, H., Hatami, H., and Bae, J. H. Productae: Toward training larger channel codes based on neural product codes. In *ICC 2022-IEEE International Conference on Communications*, pp. 3898–3903. IEEE, 2022.

Ji, S. and Li, M. Clnet: Complex input lightweight neural network designed for massive mimo csi feedback. *IEEE Wireless Communications Letters*, 10(10):2318–2322, 2021.

Kim, H., Jiang, Y., Rana, R., Kannan, S., Oh, S., and Viswanath, P. Communication algorithms via deep learning. *arXiv preprint arXiv:1805.09317*, 2018.

Kim, H., Kim, H., and de Veciana, G. Learning variable-rate codes for CSI feedback. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1435–1441, 2022. doi: 10.1109/GLOBECOM48099.2022.10000954.

Kim, H., Lee, T., Kim, H., De Veciana, G., Arfaoui, M. A., Koc, A., Pietraski, P., Zhang, G., and Kaewell, J. Generative Diffusion Model-based Compression of MIMO CSI. *arXiv preprint arXiv:2503.03753*, 2025.

Kuo, P.-H., Kung, H., and Ting, P.-A. Compressive sensing based channel feedback protocols for spatially-correlated massive antenna arrays. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 492–497. IEEE, 2012.

Li, J., Li, B., and Lu, Y. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22616–22626, 2023a.

Li, P.-h., Ankireddy, S. K., Zhao, R. P., Nourkhiz Mahjoub, H., Moradi Pari, E., Topcu, U., Chinchali, S., and Kim, H. Task-aware distributed source coding under dynamic bandwidth. *Advances in Neural Information Processing Systems*, 36:406–417, 2023b.

Li, Q., Zhang, A., Liu, P., Li, J., and Li, C. A Novel CSI Feedback Approach for Massive MIMO Using LSTM-Attention CNN. *IEEE Access*, 8:7295–7302, 2020. doi: 10.1109/ACCESS.2020.2963896. URL https://ieeexplore.ieee.org/document/8949444.

Li, Z., Zhou, Y., Wei, H., Ge, C., and Mian, A. Diffusion-based Extreme Image Compression with Compressed Feature Initialization. *arXiv preprint arXiv:2410.02640*, 2024.

Liu, J., Wang, Q., Fan, H., Wang, Y., Tang, Y., and Qu, L. Residual denoising diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2773–2783, 2024.

Liu, L., Oestges, C., Poutanen, J., Haneda, K., Vainikainen, P., Quitin, F., Tufvesson, F., and De Doncker, P. The cost 2100 mimo channel model. *IEEE Wireless Communications*, 19(6):92–99, 2012.

Liu, Y. and Simeone, O. HyperRNN: Deep Learning-Aided Downlink CSI Acquisition via Partial Channel Reciprocity for FDD Massive MIMO. In *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 31–35, 2021. doi: 10.1109/SPAWC51858.2021.9593122. URL https://ieeexplore.ieee.org/document/9593122.

Lu, C., Xu, W., Shen, H., Zhu, J., and Wang, K. Mimo channel information feedback using deep recurrent network. *IEEE Communications Letters*, 23(1):188–191, 2018.

Lu, Z., Wang, J., and Song, J. Multi-resolution CSI feedback with deep learning in massive MIMO system. In *ICC 2020-2020 IEEE international conference on communications (ICC)*, pp. 1–6. IEEE, 2020.

Makkuva, A. V., Liu, X., Jamali, M. V., Mahdavifar, H., Oh, S., and Viswanath, P. Ko codes: inventing nonlinear encoding and decoding for reliable wireless communication via deep-learning. In *International Conference on Machine Learning*, pp. 7368–7378. PMLR, 2021.

Nachmani, E., Be'ery, Y., and Burshtein, D. Learning to decode linear codes using deep learning. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 341–346. IEEE, 2016.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Shlezinger, N., Farsad, N., Eldar, Y. C., and Goldsmith, A. J. ViterbiNet: A deep learning based Viterbi algorithm for symbol detection. *IEEE Transactions on Wireless Communications*, 19(5):3319–3331, 2020.

Sim, M. S., Park, J., Chae, C.-B., and Heath, R. W. Compressed channel feedback for correlated massive MIMO systems. *Journal of Communications and Networks*, 18 (1):95–104, 2016.

Soltani, M., Pourahmadi, V., Mirzaei, A., and Sheikhzadeh, H. Deep learning-based channel estimation. *IEEE Communications Letters*, 23(4):652–655, 2019.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Sun, X., Zhang, Z., and Yang, L. An Efficient Network with Novel Quantization Designed for Massive MIMO CSI Feedback. *arXiv preprint arXiv:2405.20068*, 2024.

TR, E. 5g; study on channel model for frequencies from 0.5 to 100 ghz. *3GPP TR 38.901 version 16.1. 0 Release 16*.

Wang, B., Gao, F., Jin, S., Lin, H., and Li, G. Y. Spatial-and frequency-wideband effects in millimeter-wave massive mimo systems. *IEEE Transactions on Signal Processing*, 66(13):3393–3406, 2018a.

Wang, T., Wen, C.-K., Jin, S., and Li, G. Y. Deep learning-based csi feedback approach for time-varying massive mimo channels. *IEEE Wireless Communications Letters*, 8(2):416–419, 2018b.

Wang, T., Wen, C.-K., Jin, S., and Li, G. Y. Deep Learning-Based CSI Feedback Approach for Time-Varying Massive MIMO Channels. *IEEE Wireless Communications Letters*, 8(2):416–419, 2019. doi: 10.1109/

LWC.2018.2886113. URL https://ieeexplore.ieee.org/document/8576629.

Wen, C.-K., Shih, W.-T., and Jin, S. Deep learning for massive mimo csi feedback. *IEEE Wireless Communications Letters*, 7(5):748–751, 2018.

Xu, J., Ai, B., Wang, N., and Chen, W. Deep joint source-channel coding for csi feedback: An end-to-end approach. *IEEE Journal on Selected Areas in Communications*, 41(1):260–273, 2022.

Yang, R. and Mandt, S. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36:64971–64995, 2023.

8. Hyperparameters for training

To train the diffusion model, we adopt a similar training methodology and set of hyperparameters used in (Kim et al., 2025). The Adam optimizer is employed with a cosine annealing learning rate scheduler that goes from an initial learning rate of 3×10^{-4} to 1×10^{-5} and a batch size of 100 is used. For the diffusion process, we use a cosine beta schedule for determining the noise variance at each step.

As described in Sec. 4, the model is trained in two stages. In the first stage, the autoencoder model is trained for $N_{\text{train}} = 10^5$ iterations using the MSE loss (2). In the second stage, the diffusion-based U-Net is trained for $N_{\text{train}} = 10^6$ iterations. The coarse estimate produced by the autoencoder serves as the initialization point for reverse diffusion, with denoising performed over $T = 20$ steps during training. However, during inference, we can use a 2-step denoising for the reverse diffusion, with a small penalty in performance. This is primarily enabled by training the diffusion model using *x-prediction* instead of Gaussian noise, thus making the inference $10\times$ faster. Performance is evaluated using the NMSE, which is given as $\mathbb{E}\left[\|\mathbf{z} - \hat{\mathbf{z}}\|/\|\mathbf{z}\|\right]$ for the ground truth $\mathbf{z}$ and reconstruction $\hat{\mathbf{z}}$,

9. Computational Complexity

We now evaluate the computational complexity of RD-JSCC by measuring the throughput of each module on both the COST2100 outdoor and 3GPP indoor datasets. Throughput is measured in terms of samples processed per second (samples/s), averaged over multiple runs with a batch size of 10^3. All simulations were conducted on a system equipped with an `AMD Ryzen Threadripper PRO 5975WX 32-Core processor` and an `NVIDIA GeForce RTX 4090 GPU`.

As shown in Tables 5 and 6, the encoder and decoder exhibit high throughput due to their lightweight convolutional architecture. The encoder consistently achieves throughput near 9.5×10^4 samples/s across both datasets, making it suitable for low-power UEs. The decoder is slightly more complex due to residual layers, but still maintains a high throughput of approximately 8.25×10^4 samples/s.

In contrast, the diffusion model, though delivering significant performance improvements under complex channel conditions, introduces higher computational cost. The 2-step residual diffusion refinement achieves throughput of 3.9×10^3 samples/s on COST2100 and 8.2×10^3 samples/s on the 3GPP indoor dataset. For applications requiring higher fidelity, the full 20-step diffusion yields throughput in the 10^2 samples/s range, although the improvements in NMSE are marginal compared to 2-step diffusion. Hence, invoking the diffusion refinement module at the decoder can incur a throughput penalty of $10\times$ to $100\times$, and should therefore be used judiciously, only when necessary based on the underlying channel complexity.

Model	Encoder	Decoder	Denoising
DJSCC-CSINet+ (Small)	152K	118K	–
DJSCC-CSINet+ (Large)	152K	12M	–
ADJSCC (Small)	152K	118K	–
ADJSCC (Large)	152K	12M	–
U-Net-based JSCC	152K	118K	13.7M
RD-JSCC	152K	118K	13.9M

Table 3: Parameter count for each model architecture used for COST2100 dataset experiment.

Model	Encoder	Decoder	Denoising
DJSCC-CSINet+ (Small)	167K	191K	–
DJSCC-CSINet+ (Large)	167K	1.3M	–
ADJSCC (Small)	168K	197K	–
ADJSCC (Large)	168K	1.3M	–
U-Net-based JSCC	152K	118K	1.25M
RD-JSCC	152K	118K	1.28M

Table 4: Parameter count for each model architecture used for 3GPP indoor dataset experiment.

These results reinforce the practicality of RD-JSCC's hybrid decoding strategy, unlike a single diffusion-based solution presented in (Kim et al., 2025). For simple channels, the decoder alone may suffice, while complex scenarios can selectively invoke diffusion-based refinement. Furthermore, early-exit and low-step inference modes offer flexible complexity-performance trade-offs.

Complete details of parameters used in each experiment are provided below, in Tab. 3 and Tab. 4.

10. Ablation: Effect of Residual Diffusion Formulation

To quantify the impact of the residual formulation in our denoising diffusion process, we conduct an ablation study comparing our proposed RD-JSCC scheme against a conventional GD-JSCC baseline. The baseline follows the standard formulation introduced in (Kim et al., 2025), where the reverse diffusion process is initialized from pure Gaussian noise and trained to generate the CSI purely based on the conditioning signal. In contrast, RD-JSCC initializes the reverse diffusion with a coarse reconstruction obtained from a lightweight autoencoder, and performs iterative denoising on the residual between the ground-truth CSI and this initial estimate.

This residual formulation enables a modified diffusion objective that focuses on learning the residual signal, which is often sparser and easier to model. As shown in Fig. 6, RD-JSCC consistently outperforms the GD-JSCC baseline

Module	Throughput (samples/s)
Encoder	9.5×10^4
Decoder	8.2×10^4
Diffusion (2-step)	3.9×10^3
Diffusion (20-step)	3.9×10^2

Table 5: Throughput of each module in RD-JSCC for COST2100.

Module	Throughput (samples/s)
Encoder	9.5×10^4
Decoder	8.2×10^4
Diffusion (2-step)	8.5×10^3
Diffusion (20-step)	8.5×10^2

Table 6: Throughput of each module in RD-JSCC for 3GPP indoor.

across all tested SNR levels. Notably, it achieves up to a 1 dB improvement in effective SNR for a given NMSE target, clearly demonstrating the advantages of the residual formulation in diffusion-based compression.

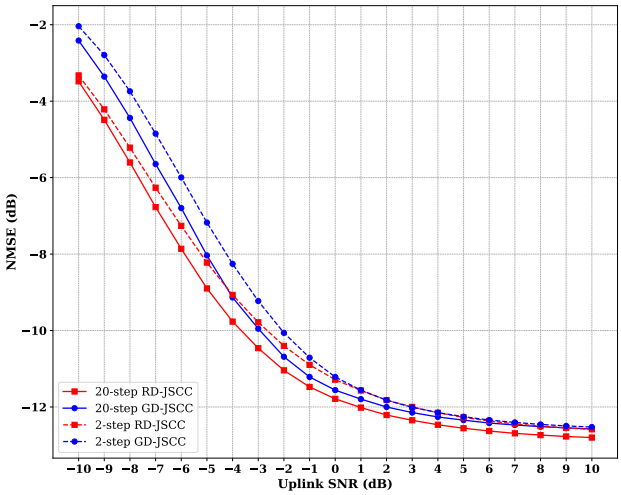


Figure 6: Comparison between standard GD-JSCC and our proposed RD-JSCC. By initializing reverse diffusion with a coarse CSI estimate and modifying the denoising objective to predict residual noise, RD-JSCC achieves up to 1 dB SNR improvement at a fixed NMSE target.