Generative Diffusion Model-based Compression of MIMO CSI

Heasung Kim[1], Taekyun Lee[1], Hyeji Kim[1], Gustavo De Veciana[1],
Mohamed Amine Arfaoui[2], Asil Koc[2], Phil Pietraski[2], Guodong Zhang[2], and John Kaewell[2].
[1]The University of Texas at Austin [2]InterDigital Communications
Emails: heasung.kim@utexas.edu, taekyun@utexas.edu, hyeji.kim@austin.utexas.edu, deveciana@utexas.edu
{MohamedAmine.Arfaoui, asil.koc, Philip.Pietraski, guodong.zhang, john.kaewell}@interdigital.com

***Abstract*—While neural lossy compression techniques have markedly advanced the efficiency of Channel State Information (CSI) compression and reconstruction for feedback in MIMO communications, efficient algorithms for more challenging and practical tasks—such as CSI compression for future channel prediction and reconstruction with relevant side information—remain underexplored, often resulting in suboptimal performance when existing methods are extended to these scenarios. To that end, we propose a novel framework for compression with side information, featuring an encoding process with fixed-rate compression using a trainable codebook for codeword quantization, and a decoding procedure modeled as a backward diffusion process conditioned on both the codeword and the side information. Experimental results show that our method significantly outperforms existing CSI compression algorithms, often yielding over twofold performance improvement by achieving comparable distortion at less than half the data rate of competing methods in certain scenarios. These findings underscore the potential of diffusion-based compression for practical deployment in communication systems.**

***Keywords*—Compression, computing, CSI, generative models, diffusion models, prediction, side information.**

I. Introduction

In Multiple-Input Multiple-Output (MIMO) communications, data transmission efficiency is closely tied to timely and accurate Channel State Information (CSI) feedback from a User Equipment (UE) to a Base Station (BS) [1]. However, with CSI often comprising hundreds to thousands of elements [2], [3] in modern communications, managing the feedback load becomes challenging. To address this, *lossy compression* techniques have been explore to compress CSI and reduce feedback overhead while retaining critical signal information.

Recent advancements in neural network-based compression approaches [4], [5], [6], commonly referred to as *neural lossy compression* [7], have demonstrated substantial improvements in compression performance over conventional techniques, such as JPEG [8] and Vector Quantization [9] for various compression tasks including CSI compression [1]. The success of neural compression methods is largely attributed to their data-driven nature and ability to learn more effective transformation modules [10].

A pioneering application of neural lossy compression for CSI compression [11] employs convolutional neural networks to achieve significant gains over conventional compressive sensing methods. Following this, several enhanced neural network architectures have been proposed for the task, including [12], [13], [14], [15]. Recent research has focused on advanced coding schemes such as entropy coding or multi-rate coding for CSI compression [16], [17], [18], [19]. Despite these advancements, achieving further improvements in CSI compression remains a challenge. The need for enhanced performance is critical to advancing the reliability and efficiency of data transmission in next-generation communication systems, a topic actively discussed in standardization bodies [20].

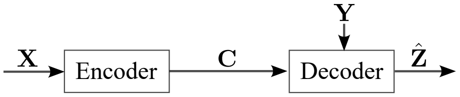

Fig. 1: System Model (Coding for computing with side information)

Recently, further advancement in neural lossy compression has been achieved through the application of diffusion models [21], particularly those employing U-Net architectures [22], which have gained recognition for their exceptional performance in image generation and restoration tasks [23]. The representational and reconstructive capabilities of these models have encouraged researchers to explore their potential in a variety of compression tasks. Notable efforts have focused on integrating diffusion models into compression pipelines, such as transmitting corrupted versions of the input source to the receiver for decoding with a diffusion model [24], or designing codeword-conditioned diffusion-based decoding processes [25].

Existing work on diffusion models for compression has predominantly concentrated on natural image compression, with the primary objective of generating perceptually realistic reconstructions. While this has addressed important challenges in visual data processing, it does not capture a broader set of industrial applications where the objective extends beyond image reconstruction to computing specific target functions from compressed data.

In this paper, we explore the application of diffusion models to the *Wyner-Ziv coding* type problems, i.e., compression with side information [26], where an encoder compresses an input source $\mathbf{X}$ into a codeword $\mathbf{C}$ and a corresponding decoder aims to estimate a target by $\hat{\mathbf{Z}}$, which may differ from the input source $\mathbf{X}$, particularly in scenarios with side information $\mathbf{Y}$ (see Fig. 1). Our primary focus is on the challenge of compressing CSI where a UE encodes observed downlink (DL) CSI into a fixed-length codeword. This compressed representation is then transmitted to a BS, which aims to predict future CSI by leveraging the received codeword along with

Part of this work was done during an internship at InterDigital Communications.

uplink (UL) CSI as side information. The correlation between UL and DL CSI, stemming from their frequency-invariant characteristics [27], [28], makes UL CSI a valuable infromation for improving compression efficiency. Additionally, we address the scenario of CSI compression without side information. Across both settings, it is observed that the proposed diffusion model-based compression method significantly outperforms existing approaches in terms of compression effectiveness. These problems align with ongoing, intensive studies within wireless communication standards [29], [20]. Our contributions can be summarized as follows:

- We propose a new fixed-rate coding for a computing scheme with side information, leveraging conditional diffusion models. Our method combines efficient vector quantization with trainable codebook-based encoding, where the input source is compressed using neural modules and quantized via the trained codebook. Decoding is achieved through a deterministic backward diffusion process [25], conditioned on the codeword and side information.
- We demonstrate the effectiveness of our diffusion-based coding scheme in various DL CSI compression scenarios in MIMO communications with Clustered Delay Line (CDL) models in NVIDIA Sionna [30] and COST2100 [11]. The DL CSI compression is an area of significant interest and active discussion within telecommunications standardization [20] due to its key role in improving communication efficiency.
- The simulation results demonstrate that the proposed method significantly surpasses existing CSI compression techniques. For instance, the proposed scheme requires less than half the data rate of competing methods to achieve the comparable distortion and demonstrates a more than *twofold improvement* in some cases. These findings suggest there is a room for improvement in the development in the CSI compression techniques. But there may be challenges in the complexity of this approach. We further discuss how to address these challenges.

II. System Model and Problem Formulation

We consider the system model depicted in Fig. 1, where the encoder processes an input source $\mathbf{X} \in \mathcal{X}$ and compresses it into a codeword $\mathbf{C} \in \mathcal{C}$, with the codeword space defined as $\mathcal{C} = \{0,1\}^B$, representing a B-bit fixed-rate compression. The decoder receives the codeword $\mathbf{C}$ along with side information $\mathbf{Y} \in \mathcal{Y}$. The objective of the decoder is to minimize the distortion between its output $\hat{\mathbf{Z}} \in \mathcal{Z}$ and a target $\mathbf{Z} \in \mathcal{Z}$. The distortion is measured using a function $d : \mathcal{Z} \times \mathcal{Z} \mapsto \mathbb{R}_+$, where $\mathbb{R}_+$ represents the space of non-negative real numbers.

We employ parameterized models for both the encoder and the decoder, denoted by parameter sets θ_{enc} and θ_{dec}, respectively. The encoder function is represented as $f_{\text{enc}} : \mathcal{X} \mapsto \mathcal{C}$, and the decoder function as $f_{\text{dec}} : \mathcal{C} \times \mathcal{Y} \mapsto \mathcal{Z}$. Given the parameter set θ_{enc}, the encoder generates a codeword $\mathbf{C} = f_{\text{enc}}(\mathbf{X}; \theta_{\text{enc}})$. Similarly, the decoder, with parameters θ_{dec}, estimates the target as $\hat{\mathbf{Z}} = f_{\text{dec}}(\mathbf{C}, \mathbf{Y}; \theta_{\text{dec}})$, and $\theta = (\theta_{\text{enc}}, \theta_{\text{dec}})$.

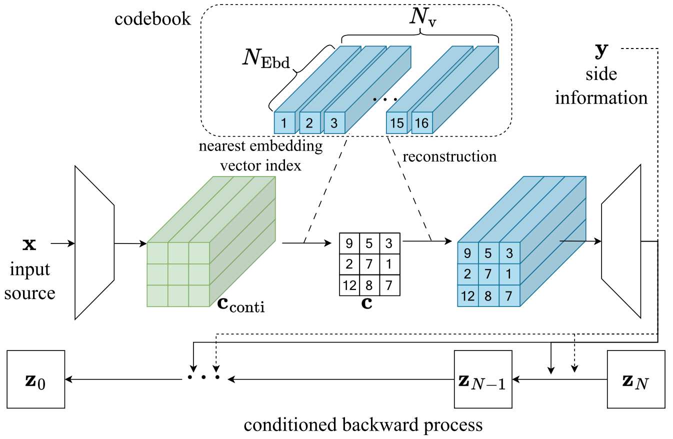

Fig. 2: Proposed compression framework.

The system's optimization objective is to minimize the expected distortion as follows.

$$\underset{\theta}{\text{minimize}} \qquad \mathbb{E}_{p_{\mathbf{X},\mathbf{Y},\mathbf{Z}}}[d(f_{\text{dec}}(f_{\text{enc}}(\mathbf{X};\theta_{\text{enc}}),\mathbf{Y};\theta_{\text{dec}}),\mathbf{Z})], \quad (1)$$

where $p_{\mathbf{X},\mathbf{Y},\mathbf{Z}}$ denotes the joint probability distribution of the input source, the side information, and the target. For the distortion measure d, we consider the element-wise squared error.

III. Proposed Approach

In this section, we introduce a novel fixed-rate coding for computing framework that leverages conditional diffusion models to efficiently compress the input sources and reconstruct target outputs, utilizing available side information. Our encoding scheme employs vector quantization via a trainable codebook for fixed-rate encoding, while the decoding process incorporates a diffusion model conditioned on both the codeword and side information.

A. Fixed-rate encoding with trainable codebook

As depicted in Fig. 2, the encoder takes an input $\mathbf{x}$, a realization of $\mathbf{X}$, and compresses it into a codeword $\mathbf{c}$ in a discrete space. We adopt a discretization module that utilizes a trainable codebook [31]. Specifically, the input $\mathbf{x}$ is processed through neural network layers and transformed into a set of continuous valued vectors $\mathbf{c}_{\text{conti}}$, each with N_{Ebd} dimensions. The codebook consists of N_{v} vectors, each of size N_{Ebd}. Each vector in $\mathbf{c}_{\text{conti}}$ is replaced by the closest codebook vector based on the minimum L^2 distance among the N_{v} vectors, thereby achieving quantization. The codebook, or set of embedding vectors, is trained alongside the model using the following loss function L_{cb}:

$$L_{\text{cb}} = \|\mathtt{sg}[\mathbf{c}_{\text{conti}}] - \mathbf{e}\|^2 + \|\mathbf{c}_{\text{conti}} - \mathtt{sg}[\mathbf{e}]\|^2, \qquad (2)$$

where $\mathtt{sg}$ represents the stop-gradient operation, which treats its input as a constant, preventing gradient backpropagation through it. The variable $\mathbf{e}$ refers to the selected embedding vectors corresponding to $\mathbf{c}_{\text{conti}}$. This loss function ensures that the encoder's output remains close to the selected codebook vectors while simultaneously guiding the codebook vectors to align with the encoder's output.

B. Conditional diffusion model-based decoding with side information

In alignment with existing diffusion-based compression methods, we utilize a diffusion backward process to estimate the target at the decoder. Specifically, we adopt the conditional diffusion model proposed in [25] for the decoding process, applying it to a codeword and side information-conditioned denoising diffusion process.

Given the codeword $\mathbf{C} = \mathbf{c}$ from the encoder and the available side information $\mathbf{Y} = \mathbf{y}$, the ultimate goal of the decoder is to sample $\mathbf{Z} \sim p(\mathbf{z}|\mathbf{c}, \mathbf{y})$ via a conditional denoising diffusion process from a $\mathbf{z}_T$, a realization of $\mathbf{Z}_T$, where the joint distribution of the target $\mathbf{Z} = \mathbf{Z}_0$ and its noise-perturbed states $\{\mathbf{Z}_t\}_{t=1}^{T}$ is modeled as

$$p(\mathbf{z}_{0:T}|\mathbf{c}, \mathbf{y}) = p(\mathbf{z}_T) \prod \mathcal{N}(\mathbf{z}_{t-1}|\mu_\theta(\mathbf{z}_t, \mathbf{c}, \mathbf{y}, t), \beta_t \boldsymbol{I}). \quad (3)$$

Here, $\mathbf{z}_{0:T} = (\mathbf{z}_0, \ldots, \mathbf{z}_T)$, a realization of $(\mathbf{Z}_0, \ldots, \mathbf{Z}_T)$, and μ_θ denotes the parameterized mean function, while β_t represents the variance schedule [32].

Once the mean function μ_θ is learned, an instance $\mathbf{z}$ can be sampled by DDIM sampling [32]. μ_θ can be obtained by minimizing the upper bound of the negative log-likelihood of the target distribution as follows.

$$-\log p(\mathbf{z}_0|\mathbf{c}, \mathbf{y}) \leq \mathbb{E}_{\mathbf{Z}_{1:T} \sim q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \left[\log \frac{q(\mathbf{Z}_{1:T}|\mathbf{z}_0)}{p(\mathbf{Z}_{0:T}|\mathbf{c}, \mathbf{y})} \right] \quad (4a)$$

$$\approx \mathbb{E}_{\mathbf{Z}_0, \mathbf{T}} \frac{\bar{\alpha}_{\mathbf{T}}}{1 - \bar{\alpha}_{\mathbf{T}}} \|\mathbf{Z}_0 - D_\theta(\mathbf{Z}_{\mathbf{t}}, \mathbf{c}, \mathbf{y}, \mathbf{T})\|^2. \quad (4b)$$

The inequality arises from the variational upper bound [33], with the approximation derived from [34], [25]. In this expression, $\mathbf{T}$ is a scalar random variable such that $\mathbf{T} \sim \mathcal{U}(0, \ldots, T)$, $\bar{\alpha}_t = \prod_{j=1}^{t} \alpha_j$ with $\alpha_t = 1 - \beta_t$, and D_θ represents the denoising function that takes as input the t-step perturbed target $\mathbf{z}_t$, codeword $\mathbf{c}$, side information $\mathbf{y}$, and the step number t, and outputs the predicted target. From this, we have

$$\mu_\theta(\mathbf{z}_t, \mathbf{c}, \mathbf{y}, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \beta_t \left(\frac{\mathbf{z}_t - \sqrt{\bar{\alpha}_t} D_\theta(\mathbf{z}_t, \mathbf{c}, \mathbf{y}, t)}{1 - \bar{\alpha}_t} \right) \right). \quad (5)$$

Based on this, the decoding operation $f_{\text{dec}}(\mathbf{c}, \mathbf{y}; \theta_{\text{dec}})$ follows an iterative process as expressed in (6), starting from $t = T$ proceeding sequentially down to $t = 1$. We initialize the process by setting $\mathbf{z}_T = \mathbf{0}$.

$$\mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} D_\theta(\mathbf{z}_t, \mathbf{c}, \mathbf{y}, t) + \sqrt{1 - \bar{\alpha}_{t-1}} \left(\frac{\mathbf{z}_t - \sqrt{\bar{\alpha}_t} D_\theta(\mathbf{z}_t, \mathbf{c}, \mathbf{y}, t)}{\sqrt{1 - \bar{\alpha}_t}} \right). \quad (6)$$

C. Training

By combining the codebook loss L_{cb} with the approximated likelihood loss in (4), we optimize the likelihood function during the training of the codebook. The total loss function to be minimized is defined as

$$L = \mathbb{E}_{\mathbf{Z}_0, \mathbf{T}} \frac{\bar{\alpha}_{\mathbf{T}}}{1 - \bar{\alpha}_{\mathbf{T}}} \|\mathbf{Z}_0 - D_\theta(\mathbf{Z}_{\mathbf{T}}, \mathbf{c}, \mathbf{y}, \mathbf{T})\|^2 + \eta L_{\text{cb}}, \quad (7)$$

where η is a hyperparameter that controls the weighting of the codebook loss. The complete training procedure is detailed in Algorithm 1, where the function GradientDescent(θ, L) represents a single gradient descent update step with respect to the objective function L, parameterized by θ.

Algorithm 1 Training

```
Input: Initial model ($\theta_{\text{enc}}, \theta_{\text{dec}}$), codebook loss weight factor
    $\eta$, $\{\bar{\alpha}_t\}_{t=0}^{T}$, GradientDescent optimizer
Output: Updated $\theta_{\text{enc}}, \theta_{\text{dec}}$
for $i = 0$ to $N_{\text{train}}$ do
    Sample $(\mathbf{x}, \mathbf{y}, \mathbf{z}) \sim p_{\mathbf{X},\mathbf{Y},\mathbf{Z}}$, $t \sim \mathcal{U}(0, \ldots, T)$, $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$
    $\mathbf{c} = f_{\text{enc}}(\mathbf{x}; \theta_{\text{enc}})$
    $L_{\text{cb}} = \|\texttt{sg}[\mathbf{c}_{\text{conti}}] - \mathbf{e}\|^2 + \|\mathbf{c}_{\text{conti}} - \texttt{sg}[\mathbf{e}]\|^2$
    $L = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t} \|\mathbf{z}_0 - D_\theta(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \mathbf{c}, \mathbf{y}, t)\|^2 + \eta L_{\text{cb}}$
    GradientDescent($\theta$, $L$)
```

D. Model Architecture

For the encoder and decoder design, we adopt the neural layers described in [25], including Downsampling Units (DU), Upsampling Units (UU), a ResNet Blocks (RNB), a Linear Attention Layers (Attn), a Convolutional Layers (Conv), and a Transposed Convolution (ConvT). The following model description is based on an input source $\mathbf{X}$ that has been preprocessed into a $32 \times 32 \times 2$ tensor (see Section IV-B1 for preprocessing details).

Encoder. The encoder comprises two RNB + DU blocks, which increase the channel dimensions while reducing the spatial dimensions of the input tensor. These blocks progressively raise the channel dimensions to 64 and then to 128, producing an output tensor of $8 \times 8 \times 128$ for quantization.

Quantization. The encoder output undergoes quantization via a codebook-based vector quantization scheme. The output tensor, structured as 64 vectors (from the 8×8 layout), each with dimensionality 128, is quantized by replacing each vector with the closest of N_{v} embedding vectors in the codebook (each of size $N_{\text{Ebd}} = 128$), based on the minimum L^2 distance. The selected codebook indices are then concatenated to form the codeword, which is transmitted to the decoder.

Decoder. The decoding process consists of two primary components: (1) reconstructing the quantized latent space from the codeword and (2) executing the diffusion backward process, conditioned on the reconstructed codeword and side information. The decoder begins by receiving the codeword, which contains the vector indices. Using the codebook, it reconstructs the tensor of size $8 \times 8 \times 128$ by selecting embedding vectors corresponding to the codeword indices. This tensor is then processed by two sets of RNB + UU, which progressively upsample and reduce the channel dimensions. After the first RNB + UU, the output size becomes $16 \times 16 \times 64$. A second set of RNB + UU further upsamples the data, resulting in a final tensor of size $32 \times 32 \times 8$. These outputs are

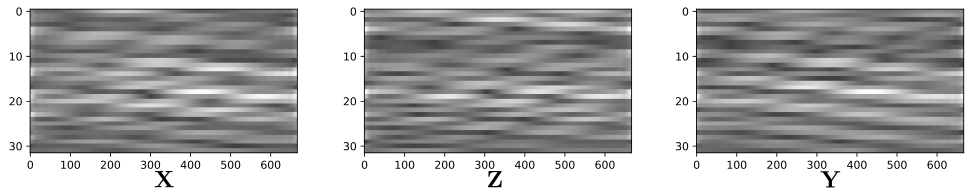

Fig. 4: Magnitude visualization of CSI samples: The task is to predict $\mathbf{Z}$ (future DL CSI) from compressed $\mathbf{X}$ (current DL CSI), leveraging $\mathbf{Y}$ (UL CSI) as side information.

used as conditioning information for the diffusion backward process, implemented through a U-Net architecture. The U-Net architecture leverages the output of the decoder and additional side information (if available) as conditioning inputs. The side information is concatenated with the decoder's last output with size $32 \times 32 \times 8$, and the time step information is also provided as input to the U-Net. For the U-Net layers, we use an embedding dimension of 64 and dimension multipliers of 1, 2, 3, and 4 for the downsampling and upsampling stages, following the architecture outlined in [25]. If no side information is provided, the concatenation step is omitted, and the U-Net processes only the decoder output.

IV. Numerical Results

A. Hyperparameters and Performance Metric

We set $\eta = 4.5 \times 10^{-4}$ as the weight in the codebook loss in (7). The model is trained for a total of $N_{\text{train}} = 3 \times 10^5$ steps, using a batch size of 100. We utilize the Adam optimizer [35] for `GradientDescent` training, with a learning rate of 10^{-3}. For the diffusion backward process, we set the number of denoising steps $T = 4$ and apply a cosine variance schedule, with corresponding values for the noise parameters: $\alpha_1 = 8.47 \times 10^{-1}$, $\alpha_2 = 4.93 \times 10^{-1}$, $\alpha_3 = 1.44 \times 10^{-1}$, and $\alpha_4 = 1.44 \times 10^{-4}$.

The performance is evaluated using the Normalized Mean Squared Error (NMSE), defined as $\mathbb{E}[\|\mathbf{z} - \hat{\mathbf{z}}\|_{\text{F}}^2 / \|\mathbf{z}\|_{\text{F}}^2]$, where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm of the matrix.

B. CSI Compression for channel prediction

1) Simulation Settings: We utilize NVIDIA Sionna [30] for real-time CSI data generation. We simulate the CSI of a MIMO system in a downlink configuration, where the BS is equipped with 32 antennas and the UE is equipped with a single antenna. The system operates in the frequency domain with 667 Orthogonal Frequency Division Multiplexing (OFDM) subcarriers, each spaced 15 kHz apart. The CDL-C profile simulates the CSI with a delay spread of 300 ns. The carrier frequency is set to 2.11 GHz for the downlink and 1.91 GHz for the uplink. Channel variations over time are simulated at an interval of 5 ms to generate future CSI instances. To model realistic mobility, the UE is simulated at a speed of 5 m/s, reflecting typical vehicular scenarios. The CDL model is parameterized using an omnidirectional antenna pattern at the UE and the 3rd Generation Partnership Project (3GPP) technical specification (TS) 38.901 antenna pattern at the BS [36]. Both the BS and UE antennas are vertically polarized.

The setup results in a 32×667 complex matrix for the DL CSI, UL CSI, and the future DL CSI (target). We simulate the channel's time evolution by generating 71 consecutive time slots (14 OFDM symbols per 5 slots, plus one), with the first slot serving as the input source $\mathbf{X}$ and the last slot's UL and DL information serving as the side information $\mathbf{Y}$ and the target $\mathbf{Z}$ for future DL CSI prediction, respectively. This UL CSI is correlated with the DL CSI through frequency-invariant characteristics [27], [28]. We assume perfect UL CSI acquisition, and the prediction is performed within the same time slot. For simplicity, each CSI instance contains only a single time slot information instead of the full 14 time slots. A sample of the input, target output, and side information is provided in Figure 4.

To reduce computational overhead, we first apply the 2D Inverse Fast Fourier Transform (IFFT) to the complex matrices $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$, converting the data from the spatial-frequency domain to the angular-delay domain. This transformation induces sparsity in the data, as supported by certain assumptions [37]. We then retain only the first 32 elements in the delay domain, as the remaining values tend to zero, yielding cropped angular-delay domain representations for $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{Y}$. The original CSI can be reconstructed by appending zero matrices of size 32×635 and performing a 2D FFT. The preprocessing is a widely adopted technique for efficient CSI representation [11], [12], [15] and we evaluate the NMSE performance within the cropped angular-delay domain.

2) Baselines: To evaluate our proposed method, we benchmark it against the CsiNet [11] and CRNet [15] models. Initially designed for CSI compression without accounting for side information or bit-level quantization, these baselines were minimally adapted to enable a fair comparison.

(a) CsiNet with Uniform Quantization. We employ the encoder and decoder architectures from [11], training the model with Mean Squared Error (MSE) loss between the network output and the target representation, $\mathbf{Z}$. In the original CsiNet, the latent representation is a continuous vector, denoted $\mathbf{c}_{\text{conti}}$ whose dimension is $N_{\text{cl,f}}$. To achieve discrete codeword, we apply 6-bit uniform quantization to each element of the latent vector $\mathbf{c}_{\text{conti}}$. Specifically, the encoder's output is constrained within [-1, 1] by a hyperbolic tangent activation (`tanh`). Each element of the latent vector is quantized with 6 bits, resulting in a $(N_{\text{cl,f}} \times 6)$-bit length codeword. To ensure gradient-based optimization remains effective, we use a stop-gradient approach, updating encoder parameters based on gradients from the pre-quantized values as $\mathbf{c} = \mathbf{c}_{\text{conti}} + \mathtt{sg}[\mathbf{c} - \mathbf{c}_{\text{conti}}]$ where $\mathbf{c}$ is the discretized input to the decoder. This design

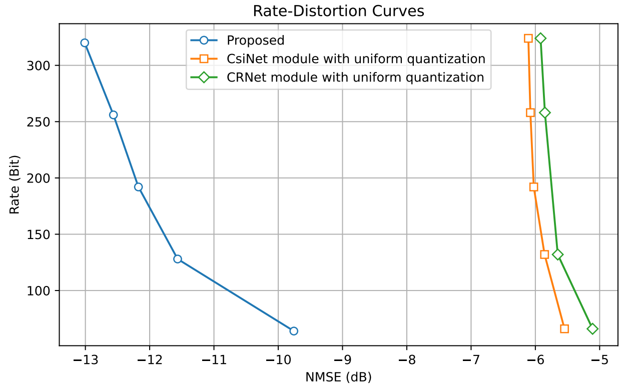

Fig. 5: Rate-distortion curves for experiments in Sec. IV-B.

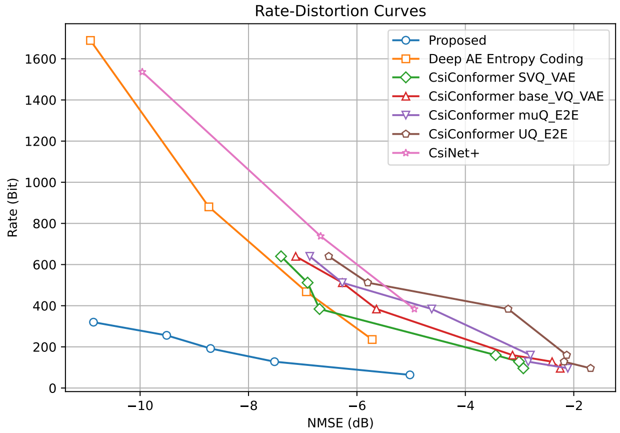

Fig. 6: Rate-distortion curves for experiments in Sec. IV-C.

supports end-to-end training of the encoder and decoder. Side information is integrated into decoding by modifying the original `ResidualBlock` in [11], where the side information is concatenated with the input being refined. For consistency, input data is scaled to lie within [-1, 1], and we apply `tanh` to the decoder output.

(b) *CRNet with Uniform Quantization.* Similarly, we adapt the encoder and decoder architectures from [15], which also use a floating-point latent vector. The quantization follows the same procedure as for CsiNet, applying 6-bit quantization after constraining values to [-1, 1] via a `tanh` activation function. To incorporate side information, we concatenate it with the decoder's input. Specifically, the CRNet decoder first takes the quantized codeword $\mathbf{c}$ of size $N_{\text{cl,f}}$, applies a dense layer to match the target dimensionality, and reshapes it to the target form of $32\times32\times2$. The side information is then concatenated, forming a tensor of size $32\times32\times4$ for further decoding. The parameters in this neural network are trained using MSE loss to minimize reconstruction distortion with respect to the target.

3) Results: The rate-distortion curves comparing the proposed method with the baseline models are presented in Fig. 5. The results demonstrate that the proposed method outperforms the baseline methods, as the proposed method achieves lower distortion for a given rate. In contrast, the baseline methods exhibit performance saturation around -6 dB NMSE, where increasing the bit rate provides only marginal improvements. For instance, increasing the bit rate by over 100 bits at 132-bit rate yields less than a 0.5 dB gain, likely due to the limited representational capacity of the neural modules or inefficient encoding schemes. In contrast, the proposed method demonstrates a gain exceeding 1 dB with an increase of just 64 bits (from 64-bit to 128-bit compression) and achieves an additional gain of over 0.5 dB when progressing from 128-bit to 192-bit compression. These results suggest that straightforward extensions of existing neural lossy compression methods may be suboptimal for CSI compression tasks, particularly in the context of future CSI prediction.

C. CSI Compression for channel reconstruction

1) Simulation Settings: To further assess the proposed compression approach, we conduct experiments focused on CSI compression aiming to reconstruct $\mathbf{X}$. This task serves as a simplified version of the prediction scenario described in Section IV-B, with the objective limited to reconstructing the input source $\mathbf{X}$ (i.e., $\mathbf{Z} = \mathbf{X}$) without incorporating side information. For simulation, we use the widely adopted COST2100 outdoor dataset [11], comprising 10^5 training samples and 2×10^4 test samples.

2) Baselines: We compare the performance of our fixed-rate coding scheme against the following state-of-the-art CSI compression algorithms: (a) CsiConformer [38], which integrates convolutional operations with self-attention mechanisms to enhance CSI feedback accuracy. In Fig. 6, we plot the performance of this approach with the relevant quantization methods reported in the work [38], (b) CsiNet+ [39], which improves performance by fine-tuning the decoder parameters based on quantization bits. The authors introduce an offset network to mitigate the impact of quantization distortion, thereby improving overall compression performance, (c) Deep AE Entropy Coding [17], where the authors propose entropy-based coding for CSI quantization. Note that entropy coding allows variable-length codewords depending on the input instance, and the rate-distortion curve of entropy coding may act as a lower bound for fixed-rate coding in ideal scenarios.

3) Results: In Fig. 6, the rate-distortion curves for the baseline methods and the proposed method are presented. The baseline performance data are sourced directly from [38], [39], [17], where the performance of the baseline CsiConformer model is reported with the relevant quantization methods. The results show that the proposed scheme significantly outperforms the existing approaches, as its rate-distortion curve consistently forms the lower bound compared to the other baselines. For instance, to achieve a distortion of approximately -7 dB, the proposed method requires fewer than 150 bits, whereas all baseline methods require more than 400 bits, demonstrating inferior performance. Notably, the proposed fixed-rate coding scheme also outperforms deep autoencoder-based entropy coding, which allows variable-length codewords based on the input instance. This highlights the superior efficiency of the proposed method, showing that it can significantly reduce the number of bits required for a given distortion level, thus offering substantial savings in radio resources for communication.

V. Discussion

We presented a new fixed-rate coding scheme with side information for DL CSI compression, which uses a vector quantization method using a trainable codebook and the diffusion-based backward process for decoding, conditioned on both the codeword and side information. Experimental results demonstrated that the proposed method significantly outperformed state-of-the-art CSI compression techniques, effectively reducing the required bit rate for a given distortion across diverse scenarios. These findings highlight the strong potential of our scheme for future network applications, where minimizing transmission rates is crucial. Further research could focus on enhancing computational efficiency by exploring more lightweight architectures [40] or fast sampling process [41], increasing the practicality of the scheme for real-world deployment.

References

[1] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Overview of deep learning-based CSI feedback in massive MIMO systems," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 8017–8045, 2022.

[2] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*. Cambridge, MA, USA: Academic Press, 2013.

[3] A. Zaidi, F. Athley, J. Medbo, U. Gustavsson, G. Durisi, and X. Chen, *5G Physical Layer: Principles, Models and Technology Components*. Cambridge, MA, USA: Academic Press, 2018.

[4] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[5] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–23.

[6] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–19.

[7] Y. Yang, S. Mandt, L. Theis *et al.*, "An introduction to neural data compression," *Found. Trends Comput. Graph. Vis.*, vol. 15, no. 2, pp. 113–200, 2023.

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

[9] J. Ziv, "On universal quantization," *IEEE Trans. Inf. Theory*, vol. 31, no. 3, pp. 344–347, 1985.

[10] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[11] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, 2018.

[12] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416–419, 2018.

[13] Q. Li, A. Zhang, P. Liu, J. Li, and C. Li, "A novel CSI feedback approach for massive MIMO using LSTM-attention CNN," *IEEE Access*, vol. 8, pp. 7295–7302, 2020.

[14] Y. Liu and O. Simeone, "HyperRNN: Deep learning-aided downlink CSI acquisition via partial channel reciprocity for FDD massive MIMO," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2021, pp. 31–35.

[15] Z. Lu, J. Wang, and J. Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[16] B. Park, H. Do, and N. Lee, "Multi-rate variable-length CSI compression for FDD massive MIMO," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2024, pp. 7715–7719.

[17] S. Ravula and S. Jain, "Deep autoencoder-based massive MIMO CSI feedback with quantization and entropy coding," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.

[18] M. Nerini, V. Rizzello, M. Joham, W. Utschick, and B. Clerckx, "Machine learning-based CSI feedback with variable length in FDD massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 2886–2900, 2022.

[19] H. Kim, H. Kim, and G. De Veciana, "Learning variable-rate codes for CSI feedback," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2022, pp. 1435–1441.

[20] X. Lin, "An overview of 5G advanced evolution in 3GPP release 18," *IEEE Commun. Stand. Mag.*, vol. 6, no. 3, pp. 77–83, 2022.

[21] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *ACM Comput. Surv.*, vol. 56, no. 4, pp. 1–39, 2023.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*. Springer, 2015.

[23] 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. 10 850–10 869, 2023.

[24] L. Theis, T. Salimans, M. D. Hoffman, and F. Mentzer, "Lossy compression with gaussian diffusion," *arXiv preprint arXiv:2206.08889*, 2022, [Online]. Available: https://arxiv.org/abs/2206.08889.

[25] R. Yang and S. Mandt, "Lossy image compression with conditional diffusion models," in *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

[26] D. Rebollo-Monedero and B. Girod, "Generalization of the rate-distortion function for wyner-ziv coding of noisy sources in the quadratic-gaussian case," in *Data Compression Conference*. IEEE, 2005, pp. 23–32.

[27] D. Vasisht, S. Kumar, H. Rahul, and D. Katabi, "Eliminating channel feedback in next-generation cellular networks," in *Proc. ACM SIGCOMM Conf.*, 2016, pp. 398–411.

[28] D. Han, J. Park, and N. Lee, "FDD massive MIMO without CSI feedback," *IEEE Trans. Wireless Commun.*, vol. 23, no. 5, pp. 4518–4530, 2023.

[29] 3GPP, "NR; physical layer procedures for data (release 15)," 3GPP, Tech. Rep. TR 38.214, 12 2017.

[30] J. Hoydis, S. Cammerer, F. A. Aoudia, A. Vem, N. Binder, G. Marcus, and A. Keller, "Sionna: An open-source library for next-generation physical layer research," *arXiv preprint arXiv:2203.11854*, 2022, [Online]. Available: https://arxiv.org/abs/2203.11854.

[31] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[32] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022, pp. 1–22.

[33] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[34] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014, [Online]. Available: https://arxiv.org/abs/1412.6980.

[36] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," Technical Specification Group Radio Access Network TR 38.901, 2019.

[37] B. Wang, F. Gao, S. Jin, H. Lin, and G. Y. Li, "Spatial- and frequency-wideband effects in millimeter-wave massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3393–3406, 2018.

[38] X. Sun, Z. Zhang, and L. Yang, "An efficient network with novel quantization designed for massive MIMO CSI feedback," *arXiv preprint arXiv:2405.20068*, 2024, [Online]. Available: https://arxiv.org/abs/2405.20068.

[39] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2827–2840, 2020.

[40] J. Ruan, S. Xiang, M. Xie, T. Liu, and Y. Fu, "Malunet: A multi-attention and light-weight unet for skin lesion segmentation," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2022, pp. 1150–1156.

[41] X. Liu, C. Gong *et al.*, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.