CSI-StripeFormer: Exploiting Stripe Features for CSI Compression in Massive MIMO System

Qingyong Hu*, Hua Kang*, Huangxun Chen†, Qianyi Huang‡, Qian Zhang*, Min Cheng§
*Hong Kong University of Science and Technology, †Huawei, ‡Sun Yat-Sen University, §Noah's Ark Lab, Huawei
Email: {qhuag, hkangae}@cse.ust.hk, chen.huangxun@huawei.com,
huangqy89@mail.sysu.edu.cn, qianzh@cse.ust.hk, min.cheng@huawei.com

***Abstract*—The massive MIMO gain for wireless communication has been greatly hindered by the feedback overhead of channel state information (CSI) growing linearly with the number of antennas. Recent efforts leverage the DNN-based encoder-decoder framework to exploit correlations within the CSI matrix for better CSI compression. However, existing works have not fully exploited the unique features of CSI, resulting in an unsatisfactory performance under high compression ratios and sensitivity to multipath effects. Instead of treating CSI as common 2D matrices like images, we reveal the intrinsic stripe-based correlation across the CSI matrix. Driven by this insight, we propose CSI-StripeFormer, a stripe-aware encoder-decoder framework to exploit the unique stripe feature for better CSI compression. We design a lightweight encoder with asymmetric convolution kernels to capture various shape features. We further incorporate novel designs tailored for stripe features, including a novel hierarchical Transformer backbone in the decoder and a hybrid attention mechanism to extract and fuse correlations in angular and delay domains. Our evaluation results show that our system achieves an over 7dB channel reconstruction gain under a high compression ratio of 64 in multipath-rich scenarios, significantly superior to current state-of-the-art approaches. This gain can be further improved to 17dB given the extended embedded dimension of our backbone.**

I. Introduction

Massive multiple-input multiple-output (mMIMO) technology exploits spatial diversity gain brought about by massive antennas at the base station (BS) to greatly improve spectral efficiency. However, the mMIMO gain for wireless communication systems can be severely compromised by the bandwidth overhead of feeding back downlink (DL) channel state information (CSI) measured by user equipment (UE). The number of DL CSI parameters grows proportionally to the increase in antennas of BS. Furthermore, as most cellular systems operate in frequency-division duplexing (FDD) mode, *i.e.*, uplink (UL) and DL occur at different frequency bands concurrently, it is hard to eliminate this overhead simply based on channel reciprocity, *i.e.*, DL and UL CSI are equivalent.

To address this issue, people seek to compress the huge DL CSI feedback on UE side and reconstruct the original DL CSI matrix on BS side. Recently, many deep neural network (DNN) powered approaches [1–6] have made notable progress on the CSI compression task. They leverage the DNN-based encoder-decoder framework to exploit the correlation within the DL CSI matrix in an end-to-end manner. Thus, they implicitly relax the strict sparsity assumption and have empirically achieved better channel reconstruction performance than the compressed-sensing-based counterparts [1].

However, there is still much room for improvement towards practical CSI compression for mMIMO. Firstly, existing deep CSI compression systems experience a dramatic increase in channel reconstruction error with a higher compression ratio (CR), as summarized in Fig. 1. Unfortunately, the high CR cases are more critical for the practical scenarios owing to the limited time budget for CSI feedback constrained by the channel coherence time, *i.e.*, the interval during which the channel does not change much. It is significant to improve the performance under high CR for better scalability in the practical FDD mMIMO systems. Secondly, existing systems, *e.g.*, CSINet [1], CRNet [6], SRNet [3] and so on, have unbalanced performances across various scenarios, *i.e.*, consistently much worse on the outdoor dataset than the indoor one generated from the recognized channel model COST2100 [7] shown in Fig. 1. It is found that the outdoor dataset features much richer multipaths than the indoor one as shown in Fig. 2. Further in-depth investigation (Fig. 3) on the existing works across different scenarios validates that the richness of multipath effects can significantly influence the model's performance. It is important to ensure the channel reconstruction performance is robust to multipath issues for stable quality of services.

To further advance this field, we argue that it is crucial to deeply understand the unique features of our compression target—CSI matrix, rather than simply treat it as an ordinary 2D matrice like image. Therefore, we delve into the underlying formation mechanism of the CSI matrix. Unlike images with apparent patch-based locality, the measured CSI matrix in the practical setting has intrinsic correlations across channel components in the same row and column. Through our analysis, it is found that the windowing effect due to limited antennas and sub-carriers diffuses the energy of one signal path in both horizontal and vertical directions of the CSI matrix. Thus, the CSI matrix presents stripe features as the sample in Fig. 4.

This observation inspires us to design CSI-StripeFormer, a stripe-aware encoder-decoder framework to enable better CSI compression. Since the UE is resource-sensitive, we design a lightweight encoder, adopting the asymmetric convolution kernels [8] as the key components to well capture various shaped features. Then, we further incorporate novel designs tailored for stripe features of the CSI matrix in the decoder at the BS side. Firstly, the stripe-based correlation requires the model with global receptive field, *i.e.*, small convolution

kernels with limited receptive field may not fit. Thus, we leverage a hierarchical Transformer-based architecture as the backbone to enable a global receptive field. It includes several layers, each containing several basic StripeFormer blocks specialized for the CSI matrix. Secondly, the model is desired to capture complex stripe-based correlations, where one signal contributes energy to components across stripes, and components in stripes may get superimposed by multiple signals. Besides, stripes in the angular and delay domains of CSI have different physical characteristics and a pair of crossed stripes jointly determine a CSI element. To explicitly incorporate them into the model, we propose StripeFormer, a Transformer-based architecture enhanced with a hybrid attention mechanism with both self-attention and cross-attention operations. The self-attention performs attention on stripes in horizontal and vertical domains respectively to model the stripe-based correlations. Meanwhile, the cross-attention dynamically fuses the representations from both horizontal and vertical attention to combine the distinct impacts of angular and delay domains.

We conduct extensive experiments on two representative datasets of indoor and outdoor scenarios generated from the COST2100 channel model [7]. The evaluation results show that our model achieves the best performance in both scenarios with high CRs. Moreover, for the multipath-rich dataset, we significantly reduce normalized mean squared error (NMSE) at CR=64 over 7 dB compared with the state-of-the-art (SOTA) models [3]. This gain can be further improved to 17 dB by extending the embedded dimension of our backbone. Our model's scalability is also validated since our NMSE at CR=64 is even 10 dB smaller than SOTA models at CR=4 in the multipath-rich scenario. In addition, our model can practically compress CSI matrix from 64 Kbits to 192 bits, *i.e.*, effective CR=341 with a low NMSE of -13.65 dB, even with a simple post-training uniform quantization.

Highlights of our contributions are as follows:

1) We identify the gap towards practical CSI compression: channel reconstruction performance is degraded under high CR and not robust to mutlipath issues, and further investigate the underlying formation mechanism of CSI matrix to reveal its unique stripe features.
2) We propose CSI-StripeFormer, a stripe-aware encoder-decoder framework to explicitly and fully exploit the stripe features. Our model features a novel hierarchical Transformer-based architecture and a hybrid attention mechanism to enable better CSI compression.
3) We conduct comprehensive evaluations to verify the effectiveness of our proposed system. Our proposed system can reduce the NMSE to -14.89 dB for multipath-rich scenarios even under high CR=64, much superior to the SOTA baseline performance of -7.8 dB, *i.e.*, 7 dB gain. This gain can be further improved to 17 dB given the extended embedded dimension of our backbone.

II. Preliminaries

In this section, we first introduce our system model and then elaborate on the physical model of wireless channels.

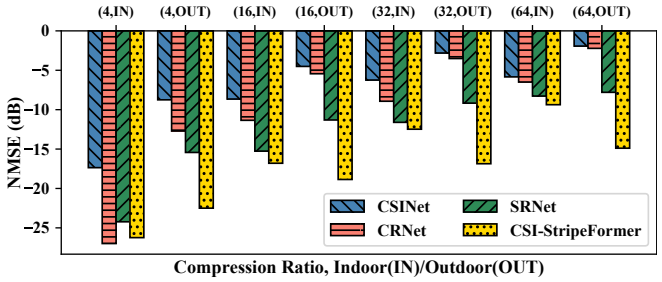


Fig. 1: SOTA performance: normalized mean square error (NMSE) of channel reconstruction for various combinations of compression ratio (CR) and dataset, *i.e.*, (CR, dataset). A smaller NMSE means a better reconstruction.

A. System Model

We consider the downlink of an FDD mMIMO system with $N_t \gg 1$ transmitting antennas at the BS and a single receiving antenna at the UE for brevity. The system adopts Orthogonal Frequency Division Multiplexing (OFDM) with N_c subcarriers. Given the transmitted signal $\boldsymbol{x} \in \mathbb{C}^{N_c \times 1}$, the received signal $\boldsymbol{y}$ is presented as:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{P}\boldsymbol{x} + \boldsymbol{z}, \tag{1}$$

where $\boldsymbol{H} \in \mathbb{C}^{N_c \times N_t}$ denotes the downlink wireless channel matrix, *i.e.*, DL CSI; $\boldsymbol{P} \in \mathbb{C}^{N_t \times N_c}$ is the precoding matrix enforced by the BS for beamforming or eliminating user interference, and $\boldsymbol{z} \in \mathbb{C}^{N_c \times 1}$ is the additive noise. It requires obtaining the DL CSI matrix $\boldsymbol{H}$ to design the corresponding precoding matrix $\boldsymbol{P}$ to get the mMIMO gain. However, it is unaffordable to directly feed back $\boldsymbol{H} \in \mathbb{C}^{N_c \times N_t}$. Thus, we follow the previous efforts to exploit the sparsity of the angular-delay CSI matrix [1]. Specifically, we convert $\boldsymbol{H} \in \mathbb{C}^{N_c \times N_t}$ from the spatial-frequency domain to the angular-delay one by applying discrete Fourier transform (DFT):

$$\tilde{\boldsymbol{H}} = \boldsymbol{F_d}\boldsymbol{H}\boldsymbol{F_a^H}, \tag{2}$$

where $\boldsymbol{F_d}$ and $\boldsymbol{F_a}$ are DFT matrices, $\boldsymbol{F_a^H}$ represents the conjugate transpose of $\boldsymbol{F_a}$. Then, we can select the first N_a rows of $\tilde{\boldsymbol{H}}$ as $\boldsymbol{H_a}$ for initial compression, because multipaths arrive at limited delay intervals and occupy a limited range in the delay domain [1]. This reduces the size of the channel matrix from $N_c \times N_t$ to $N_a \times N_t$ ($N_a < N_c$).

To further decrease the feedback overhead of DL CSI and enable accurate CSI recovery at the BS, we apply the typical DNN-based encoder-decoder framework for CSI compression and reconstruction. The encoder $\boldsymbol{E_\phi}$ compresses the channel matrix $\boldsymbol{H_a}$ into its compact representation, *i.e.*, codewords $\boldsymbol{v}$ based on the desired compression ratio:

$$\boldsymbol{v} = \boldsymbol{E}_\phi(\boldsymbol{H_a}). \tag{3}$$

Once the BS receives the codewords $\boldsymbol{v}$ sent from the UE, a decoder $\boldsymbol{G_\theta}$ tries to reconstruct a channel matrix $\hat{\boldsymbol{H}}_{\boldsymbol{a}}$ from $\boldsymbol{v}$:

$$\hat{\boldsymbol{H}}_{\boldsymbol{a}} = \boldsymbol{G}_\theta(\boldsymbol{v}). \tag{4}$$

Note that ϕ and θ denote the transformation functions of the encoder and decoder. The complete procedure can be expressed as follows:

$$\hat{\boldsymbol{H}}_{\boldsymbol{a}} = \boldsymbol{G}_\theta(\boldsymbol{E}_\phi(\boldsymbol{H_a})). \tag{5}$$

Our goal is to find a pair of encoding and decoding functions ϕ and θ to minimize the difference between the original matrix $\boldsymbol{H_a}$ and the reconstructed one $\hat{\boldsymbol{H}}_{\boldsymbol{a}}$:

$$(\theta, \phi) = \underset{\theta,\phi}{\operatorname{argmin}} \parallel \boldsymbol{H_a} - \boldsymbol{G}_\theta(\boldsymbol{E}_\phi(\boldsymbol{H_a})) \parallel . \quad (6)$$

B. Physical Model of Wireless Channels

Wireless channels characterize the signal distortion during its propagation in the physical space. If a signal x is transmitted through a wireless channel h, the received signal y can be expressed as $y = hx + z$ where z is the additive noise. The specific distortion depends on the physical attributes of both the propagation paths and the transmitted signal. Specifically, the wireless channel of a narrow band signal from a transmitter to a receiver can be expressed as [9]:

$$h(f) = \sum_{i=1}^{K} a(f, d_i) e^{-j2\pi \frac{d_i f}{c} + j\phi(f, d_i)} \quad (7)$$

where K denotes the number of propagation paths, f denotes the signal frequency, d_i denotes the length of the i-th path, c denotes the light speed, $a(f, d_i)$ denotes the amplitude attenuation, and $\phi(f, d_i)$ denotes an additive phase due to the scattering or reflection during the propagation.

Given a BS with an array of N_t antennas, the channel of the n-th antenna can be expressed as [10]:

$$h_n(f) = \sum_{i=1}^{K} (a(f, d_i) e^{-j2\pi \frac{d_i f}{c} + j\phi(f, d_i)}) e^{-j2\pi \frac{nl\cos\theta_i}{c/f}} \quad (8)$$

where θ_i denotes the angle-of-departure (AoD) of the i-th propagation path, d_i denotes the propagation distance of the i-th path from the first antenna, and l denotes the antenna separation between antennas as depicted in Fig. 4(b), *e.g.*, usually from a quarter of a wavelength to half a wavelength.

III. Key Observations

In this section, we illustrate our key observations to inspire a better design for CSI compression and reconstruction.

A. Multipath Effects on CSI Compression

Recent works [1–6] leverage DNN-based encoder-decoder framework to compress CSI at the UE and then recover it at the BS. Unfortunately, it is found that SOTA systems encounter performance degradation in the outdoor scenario as in Fig. 1.

To investigate the reason for consistently worse performance on the outdoor dataset, we analyze the multipath distributions of two public datasets utilizing the well-known MUSIC algorithm [11]. Specifically, we calculate the covariance matrix of CSI matrix, then calculate the eigenvalues of the covariance matrix. We split the signal and noise subspace by selecting p largest eigenvalues based on the SNR ratio. Assuming that the noise part takes up around 2% energy of CSI matrix, *i.e.*, 17dB SNR, the number of multipaths is then calculated as the number of eigenvalues contributing 98% energy overall. It turns out that the outdoor dataset features much richer multipaths (*i.e.*, 'multipath-rich' scenario) than the indoor one (*i.e.*, 'multipath-simple' scenario) as shown in Fig. 2.

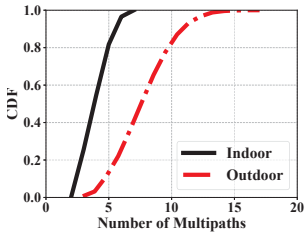


Fig. 2: Multipath distributions across various datasets.

To further validate whether the richness of multipath influences the model's performance, we split the test set of the 'multipath-rich' dataset into four subsets based on the number of multipaths in each CSI sample: [0, 4], [5, 8], [9, 12] and [13, 17]. Then, we run the trained SOTA models including CSINet [1], CRNet [6] and SRNet [3] on all subsets, respectively. As shown in Fig. 3, it is noted that the error of channel reconstruction increases with the number of multipaths in the test samples, though these models are already trained on the 'multipath-rich' dataset. Our investigation shows that the difficulty level of deep CSI compression is highly correlated with the richness of multipaths.

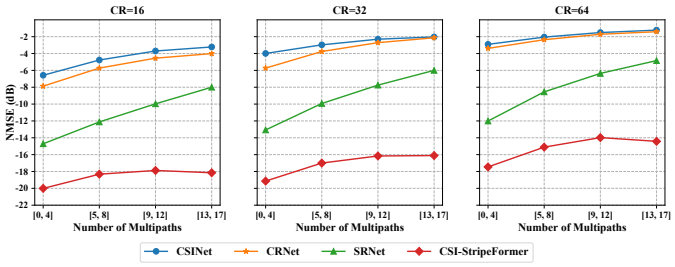
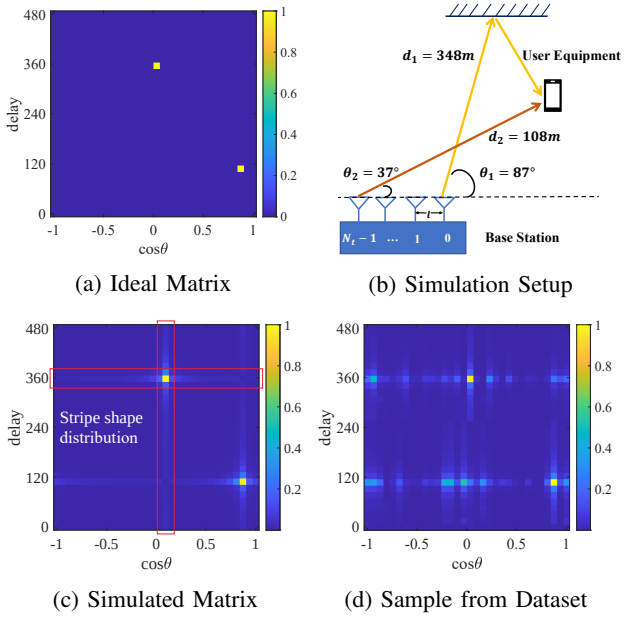


Fig. 3: The channel reconstruction performance of SOTA models degrades with richer multipath effects.

B. CSI Unique Feature: Stripe-based Correlation

Multipath effects on CSI compression encourage us to think about the formation mechanism of our compression target—CSI matrix. Many existing works regard them as ordinary 2D matrice like images and borrow many image-relevant techniques such as convolution operations to build their systems.

However, we argue that it is essential to exploit the unique CSI matrix features for better CSI compression. The BS in mMIMO system samples the signal spatially from antennas and with different frequencies from subcarriers. Assuming the measured CSI matrix $\boldsymbol{H} \in \mathbb{C}^{N_c \times N_t}$ in the spatial-frequency domain carries a signal path at a certain AoD and propagation delay, we can ideally apply DFT to obtain the angular-delay version $\boldsymbol{H_a} \in \mathbb{C}^{N_a \times N_t}$ with a corresponding pixel element. However, the spatial and temporal resolutions are limited by the window size, *i.e.*, the number of antennas and subcarriers. This windowing effect will diffuse the energy of an element in both horizontal and vertical directions of the CSI matrix. Specifically, the signal would be convolved with sinc functions due to the windowing effect in DFT. This sinc function leads to spectral leakage and transforms the peak at a certain AoD



(a) Ideal Matrix (b) Simulation Setup

(c) Simulated Matrix (d) Sample from Dataset

Fig. 4: Illustration of stripe features in angular-delay CSI.

to a stripe across the whole angular domain and the peak at a certain delay to a stripe across the whole delay domain.

To intuitively present the formation of the stripe features of CSI matrix, we simulate a toy setting with two signal paths shown in Fig. 4(b). There are two DL paths, denoted as (AoD, propagation delay) from the BS to the UE. One path is (87°, 348 m), while the other is (37°, 108 m). They have the same signal strength for simplicity. Ideally, two paths correspond to two pixels of the CSI matrix in the angular-delay domain as in Fig. 4(a). However, due to limited antennas and subcarriers, the spectral leakage may occur when the AoD and delay are not exactly the integral multiple of the angular resolution $\frac{c/f}{N_t l cos(\theta)}$ and the delay resolution $\frac{c}{B}$, where B is the bandwidth of subcarriers. This spectral leakage spreads the energy from the peak across the whole stripes as shown in Fig. 4(c), denoted as stripe features of the angular-delay CSI in our work. Though the real sample from the multipath-rich dataset (Fig. 4(d)) is more complex due to the complicated environmental effects like scattering, it also presents apparent stripe features.

To sum up, our compression target, the CSI matrix, differs from images: images have strong correlations in local patch regions, while CSI matrix presents strong correlations across the stripe regions. This observation inspires us to tailor the deep CSI compression system for the stripe-based correlation rather than the patch-based one.

IV. System Design

In this section, we elaborate on our design, CSI-StripeFormer to exploit the stripe features for better CSI compression and reconstruction in mMIMO system.

A. Overview of Key Designs

CSI-StripeFormer features a lightweight encoder on the UE side and a powerful decoder on the BS side, considering their

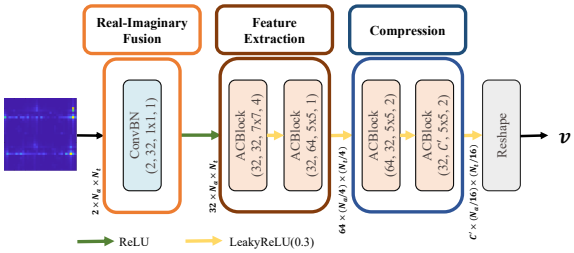


Fig. 5: The architecture of CSI-StripeFormer encoder[1].

asymmetric capacity and resources. We adopt the asymmetric convolution kernels [8] as the key components of the encoder (Fig. 5) to well capture various shaped features. Then, we incorporate designs tailored for stripe features of the CSI matrix in the decoder (Fig. 6).

Firstly, the stripe-based correlation requires the model to have a global receptive field, *i.e.*, small convolution kernels with a limited receptive field extensively used by previous works may not be suitable. Thus, we leverage a hierarchical Transformer-based architecture as the backbone to enable a stripe-aware global receptive field. Secondly, due to the stripe-based correlation, a signal contributes energy to other components in the same horizontal and vertical stripes, *i.e.*, one element is the superimposition of multiple signals. Besides, one signal in CSI matrix is jointly determined by elements of the angular and delay domains. However, the two domains have different physical characteristics like window size, resolution, and energy distributions. Thus, the model should be equipped with the capability to extract correlations of the components in the stripes as well as combine information from both the horizontal and vertical directions. Therefore, we propose to design StripeFormer, a Transformer-based architecture enhanced with a hybrid attention mechanism (Fig. 7) in the decoder.

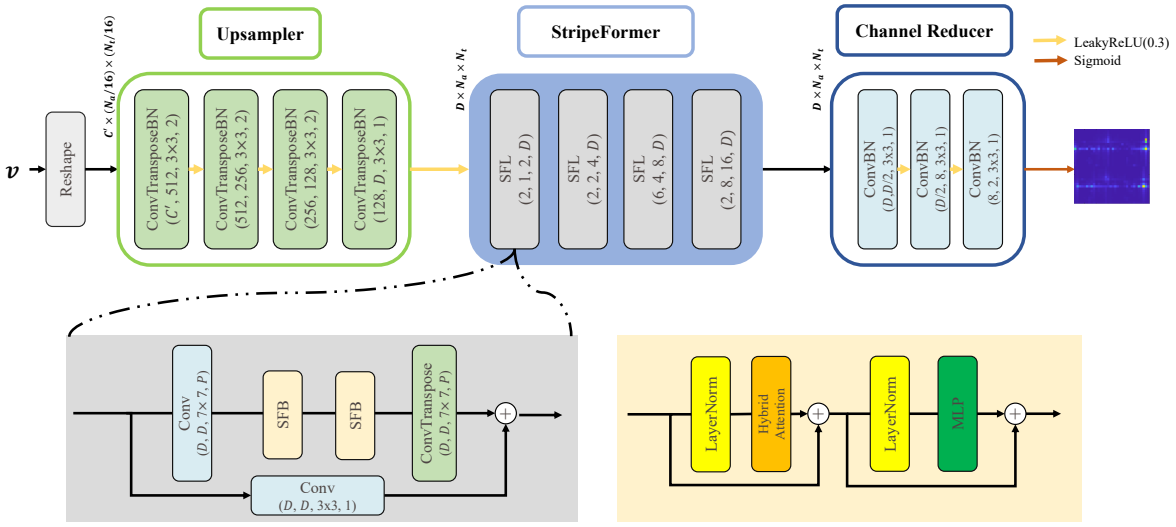
Note that in Fig. 5 and Fig. 6, the configuration of convolution and transposed convolution kernels is denoted as quad-tuple (input channel, output channel, kernel size, stride). The configuration of StripeFormer Layer (SFL) in Fig. 6 is denoted as a quad-tuple (number of StripeFormer Blocks (SFBs), split size, number of heads, embedded dimension).

B. CSI-StripeFormer Encoder

The encoder acts as the compressor on the UE side. Since the UE is resource-sensitive, our major design consideration on the encoder is to balance the complexity and the performance. Our resultant design is illustrated in Fig. 5, including three main functional components: real-imaginary fusion block, feature extraction block, and compression block.

We first utilize the real-imaginary channel fusion block [2] to handle the complex values of the CSI matrix. Basically, an element in CSI matrix has both real and imaginary parts, which determine the signal's phase and amplitude jointly. This block takes the real and imaginary input channels as inputs ($\mathbb{R}^{2\times N_a\times N_t}$), and fuses them with a point-wise convolution. It enlarges the input channels from 2 to a higher dimension (*e.g.*, 32 in our settings) representation $\boldsymbol{f_c} \in \mathbb{R}^{32\times N_a\times N_t}$.

[1]The performance is similar if we replace ReLU as LeakyReLU(0.3).



(a) StripeFormer Layer (SFL) (b) StripeFormer Block (SFB)

Fig. 6: The architecture of CSI-StripeFormer decoder.

Inspired by the stripe observation, regions with high correlations may not be regular rectangles or squares. Therefore, we adopt asymmetric convolution blocks (ACBlock) [8] instead of conventional convolutional kernels in the subsequent blocks to well capture various shape features. ACBlock can preserve features from multiple shapes without introducing extra computing burdens during the deployment, fulfilling the lightweight requirement of UE. For example, a 3×3 asymmetric kernel is made up of one 3×3 kernel, one 1×3 kernel for horizontal features, and one 3×1 kernel for vertical features during training. Upon deployment, the weights of the three kernels are fused to form a normal 3×3 kernel. For more details, we refer readers to ACNet [8].

Given the fused feature matrix $\boldsymbol{f_c} \in \mathbb{R}^{32 \times N_a \times N_t}$, the feature extraction block with two stacked ACBlocks transforms it into the informative embeddings. The first ACBlock is a 7×7 kernel with a stride of 4 to split the spatial size (N_a, N_t) as $(\frac{N_a}{4}, \frac{N_t}{4})$ patches. The second one is a 5×5 kernel to expand feature dimension from 32 to 64 to generate informative representations $\boldsymbol{p_c} \in \mathbb{R}^{64 \times \frac{N_a}{4} \times \frac{N_t}{4}}$. Finally, we design the compression block to compress the high dimension feature maps $\boldsymbol{p_c} \in \mathbb{R}^{64 \times \frac{N_a}{4} \times \frac{N_t}{4}}$. It reduces the spatial size using two 5×5 ACBlocks with stride $= 2$ to get $\boldsymbol{v_c} \in \mathbb{R}^{C' \times \frac{N_a}{16} \times \frac{N_t}{16}}$, where C' is configurable according to the required CR. The final output $\boldsymbol{v} \in \mathbb{R}^{C' \cdot \frac{N_a}{16} \cdot \frac{N_t}{16}}$ is the reshaped vector of $\boldsymbol{v_c}$.

Based on Equation 3, the compression ratio of the encoder can then be calculated as:

$$\text{CR} = \frac{\text{Size}(\boldsymbol{H_a})}{\text{Size}(\boldsymbol{v})} = \frac{2 * N_a * N_t}{C' * \frac{N_a}{16} * \frac{N_t}{16}} = \frac{512}{C'} \tag{9}$$

C. CSI-StripeFormer Decoder

The decoder aims to recover the original CSI matrix ($\mathbb{R}^{2 \times N_a \times N_t}$) from the compressed codewords $v \in \mathbb{R}^{C' \cdot \frac{N_a}{16} \cdot \frac{N_t}{16}}$. Compared with resource-sensitive UE, the decoder on the BS side has few computing constraints, leaving us with more room to design powerful models. The core component of CSI-StripeFormer decoder is a novel stripe-attention Transformer block to exploit the stripe-based correlations in CSI matrix in an end-to-end manner. There are three main components in the decoder: Upsampler, StripeFormer and Channel Reducer, as shown in Fig. 6.

1) Upsampler

We firstly reshape the compressed vector $\boldsymbol{v} \in \mathbb{R}^{C' \cdot \frac{N_a}{16} \cdot \frac{N_t}{16}}$ back to $\mathbb{R}^{C' \times \frac{N_a}{16} \times \frac{N_t}{16}}$, and then feed it to the Upsampler block. The Upsampler block contains four transposed convolution kernels, each followed by batch normalization [12] and LeakyReLU [13]. As a coarse recovery, it performs upsampling to generate a matrix $\boldsymbol{m_d} \in \mathbb{R}^{D \times N_a \times N_t}$ with the same spatial size as the original CSI matrix, where D denotes the number of feature maps as the embedded dimension.

2) StripeFormer

To fully exploit the stripe-based correlation, we propose StripeFormer as the backbone of our decoder. Taking an overview perspective from Fig. 6, StripeFormer consists of four StripeFormer Layers (SFLs) shown in Fig. 6(a). The key component of each SFL is the StripeFormer Block (SFB) shown in Fig. 6(b), while the key component of each SFB is the hybrid attention block shown in Fig. 7. Next, we will illustrate the technical details of StripeFormerin a bottom-to-up manner, *i.e.*, we first elaborate on the mechanism of hybrid attention, then introduce the SFB design, and finally describe the design of SFL and the overall StripeFormer.

Hybrid Attention: In SFB, we design a hybrid-attention mechanism as shown in Fig. 7 to improve the CSI reconstruction performance via explicitly embedding the unique stripe-based correlation of the CSI matrix. It extracts stripe-based correlations in the CSI matrix via two steps: Angular-Delay Self-Attention and Angular-Delay Cross-Attention.

1. Angular-Delay Self-Attention: The stripes in two domains,

i.e., angular and delay, have different physical characteristics like window size, resolution, energy/noise distributions *etc*. Thus, we adopt CSWin Attention [14] to model stripe-based correlations separately within two domains. In a nutshell, we project the $D-$dimension input feature $X \in \mathbb{R}^{D\times N_a\times N_t}$ linearly into K heads based on multi-head self-attention mechanism [15]. The first $K/2$ heads conduct self-attention for the horizontal stripes, while the remaining $K/2$ computes self-attention for the vertical ones.

For the self-attention of the horizontal (angular) domain, X is evenly split into non-overlapping horizontal stripes $[H_1, H_2, ..., H_M]$ with stripe width w in the vertical (delay) domain. Stripe widths denote the size of the area under consideration. Supposing the dimensions of the projected queries, keys, and values of the k-th head are d_k, the self-attention output of the horizontal domain is calculated as:

$$\begin{aligned}
\big[H_1, H_2, ..., H_M\big] &= \text{Split}(X),\\
\big[Q_i^k, K_i^k, V_i^k\big] &= \big[H_i W_Q^k, H_i W_K^k, H_i W_V^k\big],\\
A_i^k &= \text{Softmax}\big[\frac{Q_i^k (K_i^k)^T}{\sqrt{d_k}}\big],\\
\text{LePE}(V_i^k) &= \text{Conv}(V_i^k),\\
O_i^k &= A_i^k V_i^k + \text{LePE}(V_i^k),\\
\text{H-Atten}^k(X) &= [O_1^k, O_2^k, ..., O_M^k],\\
\text{H-Atten}(X) &= [\text{H-Atten}^1(X), ..., \text{H-Atten}^N(X)].
\end{aligned} \tag{10}$$

Here, the query Q_i^k, key K_i^k and value V_i^k are learnable linear embeddings of H_i with the projection matrices $W_Q^k \in \mathbb{R}^{C\times d_k}, W_K^k \in \mathbb{R}^{C\times d_k}, W_V^k \in \mathbb{R}^{C\times d_k}$, respectively. A_i^k is the attention map calculated from the correlations of the query and key with a softmax function. Then we can calculate the features O_i^k from the production of attention maps A_i^K and values V_i^k. Note that the attention mechanism is permutation-invariant, it may ignore important positional information within the CSI matrix. Thus, we add a local positional encoding computed by a convolutional kernel [14] to compensate for this positional information. The attention outputs of horizontal (angular) stripes $\text{H-Atten}^k(X)$ are a concatenation of $[O_1^k, O_2^k, ..., O_M^k]$, representing the k-th head results. The final result of $\text{H-Atten}(X)$ is the concatenation of M heads where $N = K/2$. Self-attention for the vertical (delay) domain can be derived similarly. We denote the self-attention output of the vertical stripe as $\text{V-Atten}(X)$.

2. Angular-Delay Cross-Attention: Based on Equation (8) and the stripe-based correlation illustrated in Section III-B, an element in the angular-delay CSI matrix is determined by other angular and delay components in the same stripes. Considering the varying wireless channels, we utilize a residual cross-attention module [16] to dynamically capture the correlation features between two single-domain attention outputs, $\text{H-Atten}(X)$ and $\text{V-Atten}(X)$.

The major difference between self-attention and cross-attention is: the former calculates the query Q, key K, and value V from the same domain, while the latter derives the query, key and value from different domains. Formally, the cross-attention output $\text{CrossAtten}(X_1, X_2)$ is defined as:

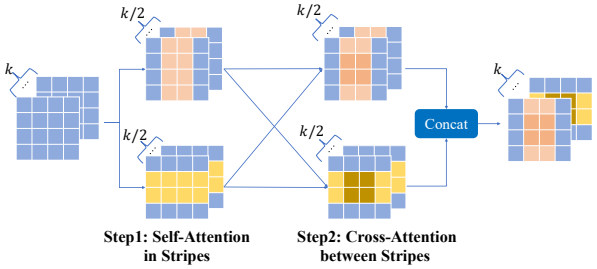


Fig. 7: Hybrid Attention in StripeFormer Block.

$$\begin{aligned}
\big[Q_{X_2}, K_{X_2}, V_{X_1}\big] &= \big[X_2 W_Q, X_2 W_K, X_1 W_V\big],\\
\text{CrxAtten}(X1, X2) &= [\text{Softmax}(Q_{X_2} K_{X_2}^T)] V_{X_1} + X1.
\end{aligned} \tag{11}$$

In our design, we fuse the correlations of angular and delay domains by using one single domain as (query, key), and the other domain as value. The final output Y of hybrid attention block is the concatenation of the fused attention features:

$$\begin{aligned}
\text{H-V Atten} &= \text{CrxAtten}(\text{H-Atten(X), V-Atten(X)}),\\
\text{V-H Atten} &= \text{CrxAtten}(\text{V-Atten(X), H-Atten(X)}),\\
Y &= \text{Concat}\big[\text{H-V Atten, V-H Atten}\big] W^O,
\end{aligned} \tag{12}$$

where $W^O \in \mathbb{R}^{D\times D}$ is the commonly used projection matrix to project the attention results to the target dimension.

StripeFormer Block (SFB): As shown in Fig. 6(b), SFB consists of hybrid attention block, layer normalization (LN) block [17] and multilayer perceptron (MLP) block. The formal specification of SFB is as follows:

$$\begin{aligned}
\hat{X}^l &= \text{Hybrid-Attention}(\text{LN}(X^{l-1})) + X^{l-1},\\
X^l &= \text{MLP}(\text{LN}(\hat{X}^l)) + \hat{X}^l.
\end{aligned} \tag{13}$$

Here, X^l is the output of the l-th StripeFormer block or the output of the convolutional embeddings.

StripeFormer Layer (SFL) and StripeFormer: Besides the major stripe-based correlations, the elements in the CSI matrix with adjacent delays or AoDs may have correlations. Thus, we design the SFL as a two-branch structure (Fig. 6(a)) to jointly consider stripe features and potential patch features. One branch consists of a series of SFBs to capture stripe features, while the other utilizes a convolution kernel for local features. Since the computational cost of attention mechanism grows as $O(n^2)$ with the input size n. In the stripe branch, we first adopt a convolution kernel to embed the input into patches for computation reduction and a transposed convolution kernel to recover the output of SFBs back to the original input size. The final output of SFL is the sum of the two branches. Several stacked SFLs with different split sizes construct the final hierarchical architecture of StripeFormer. We denote the output of StripeFormer as $\boldsymbol{s_d} \in \mathbb{R}^{D\times N_a\times N_t}$, where D is the embedded dimension of StripeFormer.

3) Channel Reducer

We adopt three convolution kernels as Channel Reducer to reduce the high dimensional output of StripeFormer. It gradually reduces $\boldsymbol{s_d}$ from $\mathbb{R}^{D\times N_a\times N_t}$ to $\mathbb{R}^{2\times N_a\times N_t}$, *i.e.*, the size of the original angular-delay CSI matrix.

V. Evaluation

A. Evaluation Methodology

1) Dataset and Metric

To ensure a fair comparison, we adopt a public benchmark dataset [18] used by many channel compression works [1–6]. The CSI samples are generated by the widely-recognized COST2100 channel model [7]. The BS has $N_t = 32$ uniform linear array antennas, and each UE has $N_r = 1$ antennas. There are $N_c = 1024$ subcarriers with 20 MHz bandwidth. The dataset contains two typical scenarios. One is the **indoor picocellular scenario** at the 5.3 GHz band, while the other is the **outdoor rural scenario** at the 300 MHz band. BS is positioned at the center of 20 m square area in the indoor case and 400 m square area in the outdoor case. UEs are randomly positioned in the area. The initial 'cut-off' compression is set as $N_a = 32$ to keep the first 32 rows of the original 1024-subcarriers CSI matrix. DFT is applied to transform the spatial-frequency CSI into the angular-delay one $\boldsymbol{H_a} \in \mathbb{R}^{2\times 32\times 32}$. Each scenario contains 150K CSI samples, 100K for training, 30K for validation, and 20K for testing.

To evaluate model effectiveness, we measure the accuracy of the CSI reconstruction via evaluating Normalized Mean Square Error (NMSE) as the quantitative metric.

$$\text{NMSE} = \mathbb{E}\left[\frac{\| \boldsymbol{H_a} - \hat{\boldsymbol{H}}_{\boldsymbol{a}} \|^2}{\| \boldsymbol{H_a} \|^2}\right]. \tag{14}$$

2) Training Scheme and Model Hyper-parameters

We implement our system with PyTorch on a server with one NVIDIA GPU 3090 card. We use the Mean Squared Error as the loss function to optimize the model towards the objective in Equation 6. We train the model with Adam Optimizer [19] for 1000 epochs with a batch size of 200 for the outdoor dataset and a batch size of 400 for the indoor dataset. The learning rate is warmed up to $1e^{-4}$ in 30 epochs and reduced to $5e^{-6}$ with the cosine decay scheme used by CRNet [6] for the outdoor dataset, while warmed up to $2e^{-4}$ for the indoor dataset. For multipath-simple indoor dataset, we adopt data augmentation including adding noise, flipping over vertical and horizontal axes, mixing up with other samples [20] and phase rotation [21] to mitigate the potential overfitting issues during training without extra burdens in deployment.

We denote the hyper-parameter configuration of the convolution kernel and transposed convolution kernel as tuple $[c_{in}, c_{out}, k, s, p]$, indicating the input channels size, output channels size, kernel size, stride, and patch size. All transposed convolution kernels are set with output_padding $= 1$.

Encoder: The configuration of the real-imaginary fusion, feature extraction, and compression blocks are $\{[2, 32, 1, 1, 0]\}$, $\{[32, 32, 7, 4, 3]$, $[32, 64, 5, 1, 2]\}$ and $\{[64, 32, 5, 2, 2]$, $[32, C', 5, 2, 2]\}$. C'=$\frac{512}{\text{CR}}$ is set according to the required compression ratio CR (Equation 9).

Decoder: The Upsampler consists of four transposed convolution kernels: $\{[C', 512, 3, 2, 1]$, $[512, 256, 3, 2, 1]$, $[256, 128, 3, 2, 1]$ and $[128, D, 3, 2, 1]\}$, where D=128 is the default embedded dimension of StripeFormer. StripeFormer splits the input with a (2, 2) patch size for the outdoor dataset and (4, 4) for the indoor dataset. StripeFormer contains 4 SFLs. The four layers are equipped with 2, 2, 6, and 2 SFBs respectively. The numbers of heads and the split widths for computing attention in four SFLs are (2, 4, 8, 16) and (1, 2, 4, 8), respectively. The Channel Reducer has three convolution kernels with parameters of $\{[D, D/2, 3, 1, 1]$, $[D/2, 8, 3, 1, 1]$, and $[8, 2, 3, 1, 1]\}$.

3) Baselines

We compare our model against the following baselines.

(a) **Deep Neural Network (DNN) Based:** The first channel compression model CSINet [1] has proved that DNN-based methods outperform compressed-sensing based solutions. CSI-StripeFormer also belongs to this category. We mainly compare our model against several SOTA DNN-based models, including CSINet [1], CRNet [6], TransNet [22] and ACCsiNet [23].

(b) **Hybrid Model (HM) Based:** They first extract important components from sparse CSI, then use DNN models to further compress the extracted parts. Thus, they require extra position information to guide the DNN models. We compare our model with two SOTA hybrid models: SRNet[3] and IdasNet [4].

B. Overall Performance

As shown in Table I, CSI-StripeFormer achieves SOTA performance under high compression ratios. Our model can significantly reduce NMSE by over 7dB compared with the best baseline SRNet [3]. We further find that our model's performance under CR=64 is even comparable with that of SRNet under CR=4 in the multipath-rich outdoor scenario. This improvement proves that our design pushes the limits of channel reconstruction under multipath-rich scenarios with high CRs. It is noticed that the gain on the indoor dataset is not as significant as the outdoor, probably because the outdoor case features richer multipaths (Fig. 2) and thus benefits more from our model.

C. Multipath Effects on CSI-StripeFormer

To evaluate the robustness of our model against multipath effects, we evaluate CSI-StripeFormer with different multipath conditions in the 'multipath-rich' outdoor dataset. As shown in Fig. 3, our model is more robust to various conditions than the best baseline SRNet [3].

D. Validation of Hybrid Attention

We compare the SFB with two other Transformer blocks widely adopted in computer vision to validate the effectiveness of the hybrid attention. We evaluate both Transformer blocks on the multipath-rich outdoor dataset with CR=64. The first is CSWin Transformer [14], computing the attention from horizontal and vertical stripes separately without fusing. The second is Swin Transformer [24], computing the local window attention with shifted windows to extract local features. For a fair comparison, we keep the remaining parts the same and only replace SFB with these two blocks. We set the parameters of CSWin Transformer the same as SFB and set the shifted

Category	Model	CR=4		CR=8		CR=16		CR=32		CR=64	
		Indoor	Outdoor	Indoor	Outdoor	Indoor	Outdoor	Indoor	Outdoor	Indoor	Outdoor
HM	SRNet	-24.23	-15.43	-19.26	-13.47	-15.26	-11.31	-11.61	-9.17	-8.27	-7.80
	IdasNet	/	/	-18.87	-10.34	-13.51	-6.15	-10.13	-5.03	-9.34	-3.63
DNN	CSINet	-17.36	-8.75	-12.70	-7.61	-8.65	-4.51	-6.24	-2.81	-5.84	-1.93
	CRNet	-26.99	-12.71	-16.01	-8.04	-11.35	-5.44	-8.93	-3.51	-6.49	-2.22
	ACCsiNet	/	/	/	/	-14.81	-11.76	-11.00	-9.14	-7.46	-7.11
	TransNet	**-32.38**	-14.86	**-22.91**	-9.99	-15.00	-7.82	-10.49	-4.13	-6.08	-2.62
	Ours	-26.24	**-22.50**	-22.29	**-20.35**	**-16.80**	**-18.86**	**-12.48**	**-16.86**	**-9.37**	**-14.89**

TABLE I: NMSE(dB) of channel reconstruction across various compression ratios (CR) and datasets.
("**Bold**" represents the best performance, "Underline" represents the second best performance, and "/" means no reported performance.)

window size to 4 for Swin block. As shown in Table II, the SFB with hybrid attention achieves the best performance, validating the effectiveness of the proposed hybrid attention mechanism. It is also noticed that CSWin performs better than Swin. This implicitly supports our insights on exploiting the stripe-based correlation to tailor the model design for learning a better representation of the CSI matrix.

Transformer Block Type	NMSE↓ (dB)
None	-4.73
CSWin [14]	-12.19
Swin [24]	-11.66
Our SFB	**-14.89**

TABLE II: Comparison with other Transformer blocks.
"↓" indicates the lower NMSE the better performance.

E. Quantization Influence on CSI-StripeFormer

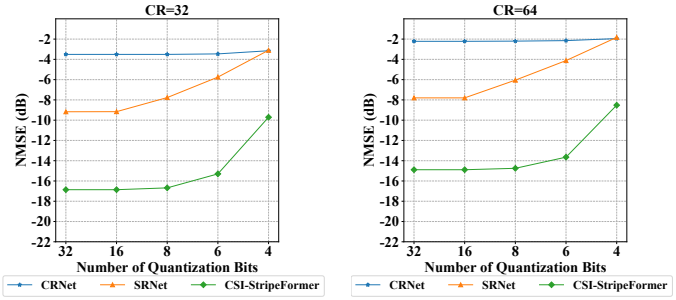
In practical FDD mMIMO systems, we can transmit quantized values with fewer bits instead of 32-bit floats. This quantization brings extra compression gain but may degrade the channel reconstruction performance. We adopt uniform quantization on the trained model to evaluate the quantization effect. We compare our model mainly with the best baseline, SRNet [3]. It is worth mentioning that our model is trained without quantization for a fair comparison. As shown in Fig. 8, our model shows little degradation when there are only 6 quantization bits. This evaluation validates that our model can tolerate the quantization error, and still outperforms the baselines even when they adopt 32-bit floats. The NMSE of our model at 6 quantization bits is -13.65 dB, much lower than SRNet's -7.80 dB at 32 quantization bits. Thus, our model can practically compress the CSI matrix from 64 Kbits to 192 bits with a low reconstruction error.

F. Ablation Study

We evaluate the effects of key hyper-parameters of our model. The ablation study is conducted under the 'multipath-rich' outdoor scenario with a compression ratio of 64.

1) Impact of Embedded Dimension

The embedded dimension determines the feature space of the model. We vary the dimension D from 32 to 256 to evaluate its effect on our model. As shown in Table III, our model has good scalability, *e.g.*, NMSE can be further reduced to -25.11 dB for CR=64. Given extended embedded dimension D=256, our model's reconstruction error at CR=64 is much lower (around 10 dB) than the baselines at CR=4.



(a) CR=32 (b) CR=64

Fig. 8: Quantization effects on the multipath-rich dataset.

D	32	64	128	256
NMSE↓ (dB)	-6.07	-7.62	-14.89	**-25.11**

TABLE III: Impact of embedded dimension D

2) Impact of StripeFormer Layer Configuration

The default layer setting is four layers equipped with [2, 2, 6, 2] blocks separately. To evaluate the influence of layers, we compare the default setting with four other configurations (CFG) under the outdoor scenario with CR=64. Each configuration is trained for 1000 epochs. Table IV presents the comparison results. Comparing CFG 1 and 3, it is found that adding more Transformer blocks with the same layers can improve the performance. Comparing CFG 1 and 2, different block distributions for the same block and layer numbers have little influence. A comparison between CFG 3 and CFG 4 indicates that adding more layers with the same block number can reduce the NMSE. This is because adding more layers can extract different resolution features, while blocks in the same layer only focus on the same resolution. For CFG 4 and 5, a deeper model with more layers and blocks can perform better.

G. Network Complexity

In this part, we present the number of FLOPs (floating-point operations per second) and parameters of our encoder

CFG	Layers	Split Width	NMSE↓ (dB)
1	[2, 2, 6, 2]	[1, 2, 4, 8]	-14.89
2	[3, 3, 3, 3]	[1, 2, 4, 8]	**-14.99**
3	[1, 1, 3, 1]	[1, 2, 4, 8]	-13.65
4	[2, 2, 2]	[1, 2, 4]	-12.12
5	[2, 2]	[1, 2]	-9.85

TABLE IV: Impact of SFL configurations

and decoder separately. We select one Transformer-based baseline TransNet [22], one high-performance baseline SRNet [3] and one lightweight baseline CRNet [6] for comparison. The results are shown in Table V. Basically, we offload the computing burden from UE to BS since BS has sufficient computing resources. From Table V, our encoder is relatively lightweight compared with TransNet [22] but has a much better performance. Our decoder is heavier in exchange for significantly better channel reconstruction performance.

Model	CR	UE		BS		NMSE↓ (dB)
		Params	FLOPs	Params	FLOPs	
CRNet	32	131K	383K	**136K**	**3.23M**	-3.51
TransNet	32	271K	16.91M	276K	16.97M	-4.13
SRNet	32	**58K**	**238K**	2.07M	658M	-9.17
Ours	32	165K	7.50M	11.38M	5.76G	**-16.86**

TABLE V: Model parameters and FLOPs

VI. Discussion and Future Work

1) Model Compression

We currently exchange model complexity for significantly better channel reconstruction performance via a heavier but more powerful decoder in the resource-rich BS. Nevertheless, it is desirable to have the best of both worlds: a lightweight yet powerful design. It is expected that the promising progress on techniques including model compression [25], pruning [26] and distillation [27] could be integrated to make the design more lightweight while preserving much of its capability.

2) Scenario Adaptation

As the indoor/outdoor distribution shift shown in Fig. 2, current deep CSI compression systems including ours train a dedicated model for each scenario, which may cause performance differences. It is envisioned that they can be deployed to indoor microcells and outdoor BSs respectively. However, it is worth pursuing a universal model across various scenarios (domains). We think this problem falls in the scope of domain adaption widely discussed in the ML community [28]. In our future work, we plan to integrate relevant domain generalization techniques [29–31] to design unified CSI compression models easily adapted to various scenarios. This will also reduce the data collection burden and speed up practical deployment.

3) Real-World Deployment

Our work is currently evaluated on the public dataset used by most existing works to facilitate comparison. Towards real-world deployment, we plan to build a prototype platform and conduct further evaluation on the CSI collected in real environments like [32] to investigate the model robustness against practical factors, *e.g.*, the impact of hardware imperfections on CSI in our future work.

VII. Related Works

There are two lines of research for CSI feedback reduction related to our work: 1) leverage known UL CSI to infer DL CSI; 2) compress the DL CSI at UE and reconstruct at BS.

For the former, existing works [10, 33, 34] transform the UL channel to the DL channel since both signals experience the same physical paths. However, the performance may degrade under increased DL and UL frequency difference [32] due to partial reciprocity [35]. Our work belongs to the second category and is immune to this issue.

For the latter, compressed sensing (CS)-based methods [36–38] utilize the sparsity of the CSI matrix to project it into a low-dimension space. However, CSI matrix is not always low-ranking or sparse under complicated wireless environments. Thus, DNNs have been adopted to relax the sparsity assumption. CSINet [1] is the first to explore CNN for CSI compression, outperforming the CS-based methods under various CRs. Motivated by this performance gain, many follow-up works, including CLNet [2], SRNet [3], IdasNet [4], CSINet-LSTM [5], CRNet [6], ACCsiNet [23] and so on, have been proposed to improve CSI compression. However, most of them treat CSI as images without considering the uniqueness of CSI. TransNet [22] also uses the Transformer [15] as the backbone. However, it just applies the Transformer without considering CSI domain information. Our work reveals the intrinsic stripe-based correlation across channel components of CSI matrix, and design a stripe-aware encoder-decoder framework to enable better CSI compression under high compression ratios.

VIII. Concluding Remarks

In this work, we identify the multipath effects on channel compression and present a unique observation on stripe-based correlation of the angular-delay CSI matrix. We then exploit this insight to design a stripe-aware encoder-decoder framework, CSI-StripeFormer to enable better CSI compression. We propose to explicitly incorporate stripe features into the model design with a novel hybrid attention module and a hierarchical Transformer-based architecture. It is believed that CSI-StripeFormer advances the field of channel compression towards practical usage in massive MIMO systems.

Acknowledgement

The authors would like to thank anonymous reviewers for their valuable comments. This research is supported in part by RGC under Contract CERG 16203719, 16204820, and Contract R8015, L3016, the Key-Area Research and Development Program of Guangdong Province (No.2020B0101390001), as well as National Natural Science Foundation of China (No. 62002150). We also thank the Turing AI Computing Cloud (TACC) [39] and HKUST iSING Lab for providing computation resources on their platform. Qian Zhang is the corresponding author.

REFERENCES

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

[2] S. Ji and M. Li, "Clnet: Complex input lightweight neural network designed for massive MIMO CSI feedback," *IEEE Wirel. Commun. Lett.*, vol. 10, no. 10, pp. 2318–2322, 2021.

[3] X. Chen, C. Deng, B. Zhou, H. Zhang, G. Yang, and S. Ma, "High-accuracy CSI feedback with super-resolution network for massive MIMO systems," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 1, pp. 141–145, 2022.

[4] Z. Yin, W. Xu, R. Xie, S. Zhang, D. W. K. Ng, and X. You, "Deep CSI compression for massive MIMO: A self-information model-driven neural network," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 10, pp. 8872–8886, 2022.

[5] T. Wang, C. Wen, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 2, pp. 416–419, 2019.

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

[7] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. D. Doncker, "The COST 2100 MIMO channel model," *IEEE Wirel. Commun.*, vol. 19, no. 6, pp. 92–99, 2012.

[8] X. Ding, Y. Guo, G. Ding, and J. Han, "Acnet: Strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks," in *ICCV*. IEEE, 2019, pp. 1911–1920.

[9] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

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

[11] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, 1986.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 37. JMLR.org, 2015, pp. 448–456.

[13] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *CoRR*, vol. abs/1505.00853, 2015.

[14] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "Cswin transformer: A general vision transformer backbone with cross-shaped windows," in *CVPR*. IEEE, 2022, pp. 12 114–12 124.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[16] K. Qian, S. Zhu, X. Zhang, and L. E. Li, "Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals," in *CVPR*. Computer Vision Foundation / IEEE, 2021, pp. 444–453.

[17] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 10 524–10 533.

[18] C. Wen and W. Shih, "Python code for CSINet," https://github.com/sydney222/Python_CsiNet.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[20] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR (Poster)*. OpenReview.net, 2018.

[21] T. Zheng, Z. Chen, S. Zhang, C. Cai, and J. Luo, "More-fi: Motion-robust and fine-grained respiration monitoring via deep-learning UWB radar," in *SenSys*. ACM, 2021, pp. 111–124.

[22] Y. Cui, A. Guo, and C. Song, "Transnet: Full attention network for CSI feedback in FDD massive MIMO system," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 5, pp. 903–907, 2022.

[23] B. Cao, Y. Yang, P. Ran, D. He, and G. He, "Accsinet: Asymmetric convolution-based autoencoder framework for massive MIMO CSI feedback," *IEEE Commun. Lett.*, vol. 25, no. 12, pp. 3873–3877, 2021.

[24] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*. IEEE, 2021, pp. 9992–10 002.

[25] Z. Li, E. Wallace, S. Shen, K. Lin, K. Keutzer, D. Klein, and J. Gonzalez, "Train big, then compress: Rethinking model size for efficient training and inference of transformers," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 5958–5968.

[26] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," in *ICLR (Poster)*. OpenReview.net, 2019.

[27] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *ICLR (Poster)*. OpenReview.net, 2018.

[28] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 766–785, 2021.

[29] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *CVPR*. Computer Vision Foundation / IEEE, 2020, pp. 12 553–12 562.

[30] D. Mahajan, S. Tople, and A. Sharma, "Domain generalization using causal matching," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 7313–7324.

[31] S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao, "Domain generalization via entropy regularization," in *NeurIPS*, 2020.

[32] X. Zhang, L. Zhong, and A. Sabharwal, "Directional training for FDD massive MIMO," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 8, pp. 5183–5197, 2018.

[33] A. Bakshi, Y. Mao, K. Srinivasan, and S. Parthasarathy, "Fast and efficient cross band channel prediction using machine learning," in *MobiCom*. ACM, 2019, pp. 37:1–37:16.

[34] Z. Liu, G. Singh, C. Xu, and D. Vasisht, "FIRE: enabling reciprocity for FDD MIMO systems," in *MobiCom*. ACM, 2021, pp. 628–641.

[35] Z. Zhong, L. Fan, and S. Ge, "FDD massive MIMO uplink and downlink channel reciprocity properties: Full or partial reciprocity?" in *GLOBECOM*. IEEE, 2020, pp. 1–5.

[36] L. Lu, G. Y. Li, D. Qiao, and W. Han, "Sparsity-enhancing basis for compressive sensing based channel feedback in massive MIMO systems," in *GLOBECOM*. IEEE, 2015, pp. 1–6.

[37] Z. Gao, L. Dai, S. Han, C. I, Z. Wang, and L. Hanzo, "Compressive sensing techniques for next-generation wireless communications," *IEEE Wirel. Commun.*, vol. 25, no. 3, pp. 144–153, 2018.

[38] X. Rao and V. K. N. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261–3271, 2014.

[39] K. Xu, X. Wan, H. Wang, Z. Ren, X. Liao, D. Sun, C. Zeng, and K. Chen, "Tacc: A full-stack cloud computing infrastructure for machine learning tasks," *arXiv preprint arXiv:2110.01556*, 2021.