TCNet: A Unified Framework for CSI Feedback Compression Leveraging Language Model as Lossless Compressor

Zijiu Yang [1] **Qianqian Yang** [1]

Abstract

Transformer-based architectures have demonstrated strong capability in capturing global dependencies for CSI feedback, yet their high computational overhead limits practical deployment. In contrast, CNNs are more efficient and excel at extracting local features, but struggle with long-range modeling. To leverage the complementary strengths of both, we propose TCNet, a hybrid framework combining CNNs and Swin Transformers to achieve accurate reconstruction with reduced complexity. Beyond lossy compression, we further introduce a language model-based lossless coding scheme that significantly improves bit-level efficiency. Unlike conventional fixed-length or entropy-based encoding methods, our approach employs a lightweight language model as a universal probability estimator for variable-length arithmetic coding. To ensure compatibility with communication data, we design an alignment mechanism that maps CSI representations into a token structure suitable for language modeling. This alignment enables our method to generalize to other compression tasks in wireless communications. Experimental results on COST2100 demonstrate that our framework achieves the best bit rate-NMSE trade-offs, highlighting the potential of integrating language modeling with compression task in wireless communications.

1. Introduction

Massive multiple-input multiple-output (MIMO) has emerged as a cornerstone technology for 5G and future 6G wireless systems, enabling simultaneous service to a large number of users and devices by equipping base stations with hundreds or even thousands of antennas (Dong et al., 2020). Central to its performance is the acquisition of accurate channel state information (CSI), which captures signal propagation characteristics such as scattering, fading, and path loss (Shafin & Liu, 2018). CSI enables precise beamforming, optimized spatial resource allocation, and robust interference management. However, obtaining accurate CSI in dynamic wireless environments remains challenging and necessitates efficient channel estimation and feedback mechanisms.

Massive MIMO supports both time division duplexing (TDD) and frequency division duplexing (FDD) (Chan et al., 2006), as shown in Figure 1. While TDD systems benefit from channel reciprocity to infer downlink CSI from uplink measurements, FDD systems operate over separate frequency bands, breaking this reciprocity. As a result, downlink CSI must be estimated at the user equipment and fed back to the base station. With increasing antenna counts, the CSI feedback overhead in FDD systems grows linearly, posing significant challenges to scalability and spectral efficiency. Addressing the CSI acquisition and feedback bottleneck in FDD massive MIMO is thus critical for unlocking its full potential (Wen et al., 2018).

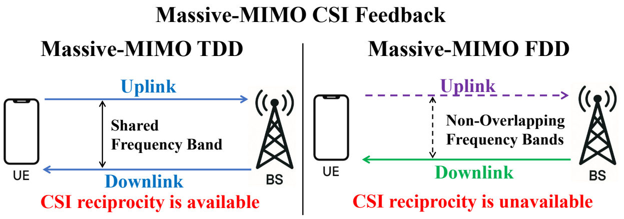

Figure 1. CSI feedback in Massive MIMO systems.

Traditional CSI compression methods such as compressed sensing, are fundamentally limited by their reliance on predefined signal models. In practice, especially under complex noise conditions or non-ideal sampling scenarios, these methods often suffer from degraded performance due to incomplete prior knowledge. To overcome such limitations, data-driven approaches rooted in deep learning have gained increasing attention for their ability to learn signal structures

[1] Zhejiang University. Correspondence to: Qianqian Yang <qianqianyang20@zju.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

directly from data without explicit analytical models.

A milestone in this direction was set by Wen et al. in 2018 with the introduction of CsiNet (Wen et al., 2018), which marked the first application of deep learning to CSI feedback. By adopting an end-to-end encoder–decoder architecture, CsiNet (Wen et al., 2018) significantly improved the efficiency of CSI compression and reconstruction. Following this breakthrough, numerous extensions were proposed, including CsiNet-LSTM (Wang et al., 2018) with temporal modeling, Attention-CSI (Cai et al., 2019) incorporating attention mechanisms, CsiNet+ (Guo et al., 2020) using higher-resolution convolutional kernels, and DCRNet (Tang et al., 2022) employing dilated convolution. For scenarios requiring low computational complexity, CRNet (Lu et al., 2020) achieved notable reductions in complexity while preserving reconstruction quality. CLNet (Ji & Li, 2021) further enhanced accuracy in the complex domain with minimal computational overhead. Recognizing the limitations of convolutional neural networks (CNN) in capturing long-range dependencies, researchers began exploring Transformer-based architectures for CSI feedback. TransNet (Cui et al., 2022), for example, replaced CNN-based encoders with dual-layer Transformer modules, enabling more effective modeling of global CSI features across compression rates.

Transformer-based architectures (Cui et al., 2022) have demonstrated remarkable capability in modeling long-range dependencies and capturing global contextual information, making them attractive for CSI feedback. However, their high computational complexity and extensive memory requirements pose significant challenges, especially in scenarios with limited hardware resources or strict latency constraints. In contrast, CNN-based methods are computationally more efficient and particularly effective at extracting local spatial features, but they struggle to capture global relationships inherent in CSI. This complementary nature of CNNs and Transformers motivates the design of a hybrid architecture that balances global modeling capacity with computational efficiency, which forms a central research objective of this work.

Deep learning-based methods have greatly reduced the volume of CSI. However, as the compression ratio increases, reconstruction quality tends to degrade significantly. Efficient encoding of the compressed CSI is thus crucial, as it directly impacts the overall compression performance. Current deep learning-based CSI compression approaches typically employ fixed-length encoding schemes, which suffer from limited coding efficiency despite their low computational complexity. In contrast, variable-length coding offers higher encoding efficiency by adapting to the actual data distribution. However, its performance heavily relies on the accuracy of the underlying probability model. Some previous approaches employ entropy models (Yang et al., 2019) trained over extended periods to obtain the probability distribution. While entropy models can capture source distributions relatively accurately, their training is time-consuming and they exhibit limited generalization. Recently, language models have emerged as powerful universal probability estimators. To tackle the challenge of effectively encoding CSI, we Leverage a suitably scaled language model as a distribution predictor for CSI, significantly enhancing the efficiency of the encoding process, offering a promising alternative to traditional entropy models.

To summarize, the main contributions of this paper are as follows:

- We propose a novel CSI feedback framework that integrates a deep learning-based lossy compression network with a language model-based lossless coding module, which not only reduces computational complexity but also improves the overall quality of CSI feedback.
- We are the first to integrate CNN and Transformer architectures for CSI feedback, effectively combining their strengths in local feature extraction and global context modeling, while maintaining compatibility with practical computational complexity constraints.
- To the best of our knowledge, this is the first work to explore the use of language models for lossless CSI encoding. By utilizing the powerful distribution modeling capabilities of language models, our approach achieves higher coding efficiency and lower transmission bit rates.

2. System Model

As shown in Figure 2, We consider a FDD massive MIMO system, where the base station (BS) is equipped with N_t transmit antennas, and each user equipment (UE) is equipped with N_r receive antennas. The system bandwidth is divided into N_c subcarriers in the frequency domain.

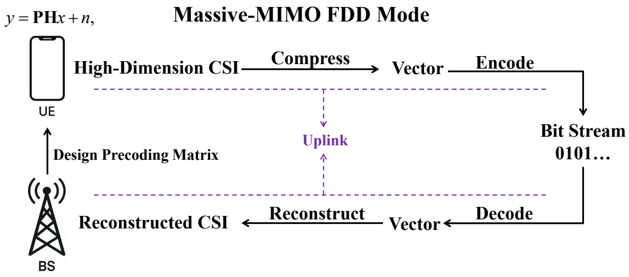

Figure 2. CSI feedback scheme in Massive MIMO FDD systems.

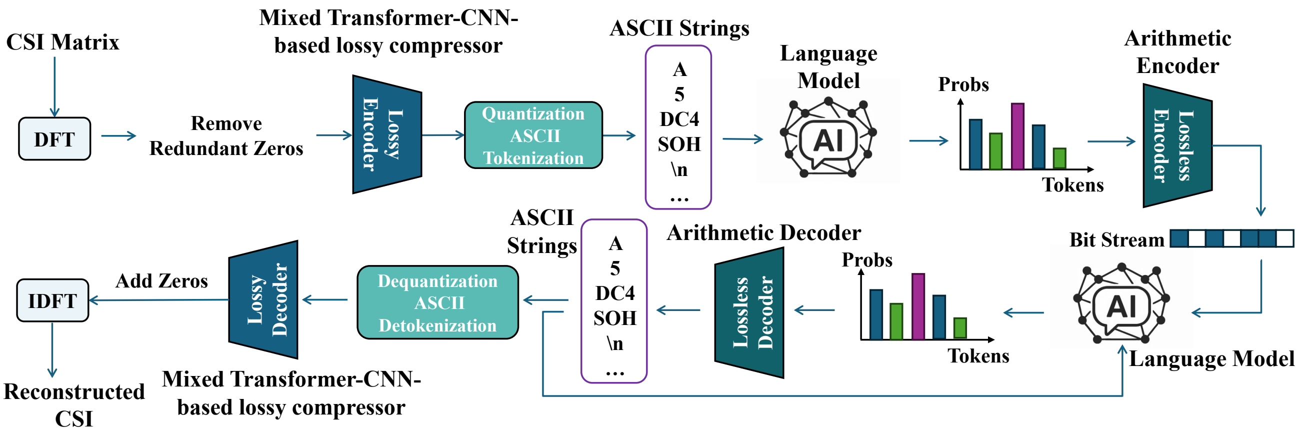

Figure 3. The proposed CSI feedback network TCNet.

Accordingly, the downlink CSI for a user can be modeled as

$$\mathbf{H} = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,N_t-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_c-1,0} & h_{N_c-1,1} & \cdots & h_{N_c-1,N_t-1} \end{bmatrix}, \quad (1)$$

where the i-th row corresponds to the channel coefficients of the i-th subcarrier, and the j-th column represents the channel response associated with the j-th transmit antenna. We follow (Wen et al., 2018) to employ Discrete Fourier Transform to process CSI, as expressed as

$$h_{k,l} = \frac{1}{N_c N_t} \sum_{m=0}^{N_c-1} \sum_{n=0}^{N_t-1} h_{m,n} e^{-j2\pi\left(\frac{km}{N_c}+\frac{nl}{N_t}\right)}, \quad (2)$$

where $k = 0, 1, ..., N_c - 1$ and $l = 0, 1, ..., N_t - 1$ represent the channel coefficient corresponding to the k-th delay and the l-th angular direction in the angle-delay domain.

Let the transformed CSI matrix be denoted as $\mathbf{H}'$. the majority of the energy in $\mathbf{H}'$ is concentrated within the first N_a rows ($N_a \ll N_c$), while the remaining entries are approximately zero. Therefore, it is reasonable to approximate $\mathbf{H}'$ by retaining only its first N_a rows, constructing a low-dimensional representative matrix $\mathbf{H}_a$, as expressed by

$$\mathbf{H}_a = \text{Truncate}_{row}(\mathbf{H}'), \text{rows} = 0, 1, 2, ..., N_a - 1. \quad (3)$$

Although the dimension of $\mathbf{H}_a$ is reduced, there is still need to design compressors $f_c(.)$ and encoders $f_e(.)$ to further compress and encode $\mathbf{H}_a$ for CSI feedback following equation (4).

$$\begin{aligned} v &= f_c(\mathbf{H}_a), \\ b &= f_e(v), \end{aligned} \quad (4)$$

where v is the CSI vector compressed by deep learning-based compressor and b is the bit stream. At the receiver, a decoder $f_d(.)$ and a reconstructor $f_{dc}(.)$ are employed to recover the CSI from the received b, as expressed by

$$\begin{aligned} v' &= f_d(b), \\ \mathbf{H}'_a &= f_{dc}(v'), \end{aligned} \quad (5)$$

where v' and $\mathbf{H}'_a$ respectively represent the decoded vector and reconstructed CSI.

3. Design of TCNet

3.1. General Framework

Figure 3 illustrates the whole feedback network TCNet we propose. TCNet incorporates a hybrid Transformer–CNN architecture for feature encoding and decoding, along with a uniform quantization module, an ASCII-level discretization component, a probabilistic language modeling unit, and an adaptive arithmetic coder. This integrated design enables efficient CSI compression while preserving high reconstruction fidelity.

At the encoder side, the original CSI matrix is first transformed into a sparse frequency-domain representation via 2D DFT. After removing zero-valued elements, the sparsified CSI is passed through the Transformer–CNN-based compressor to extract compact feature representations. These feature vectors are then activated using a sigmoid function as expressed by

$$\text{Sigmoid} = \frac{1}{1 + e^{-x}}. \quad (6)$$

Subsequently, the CSI feature vectors are uniformly quantized using a n-bit quantizer. The quantized data are then converted into symbol sequences through an ASCII-based tokenization scheme. A language model is employed to learn the global dependencies among tokens and to construct the corresponding probability distribution. Based on

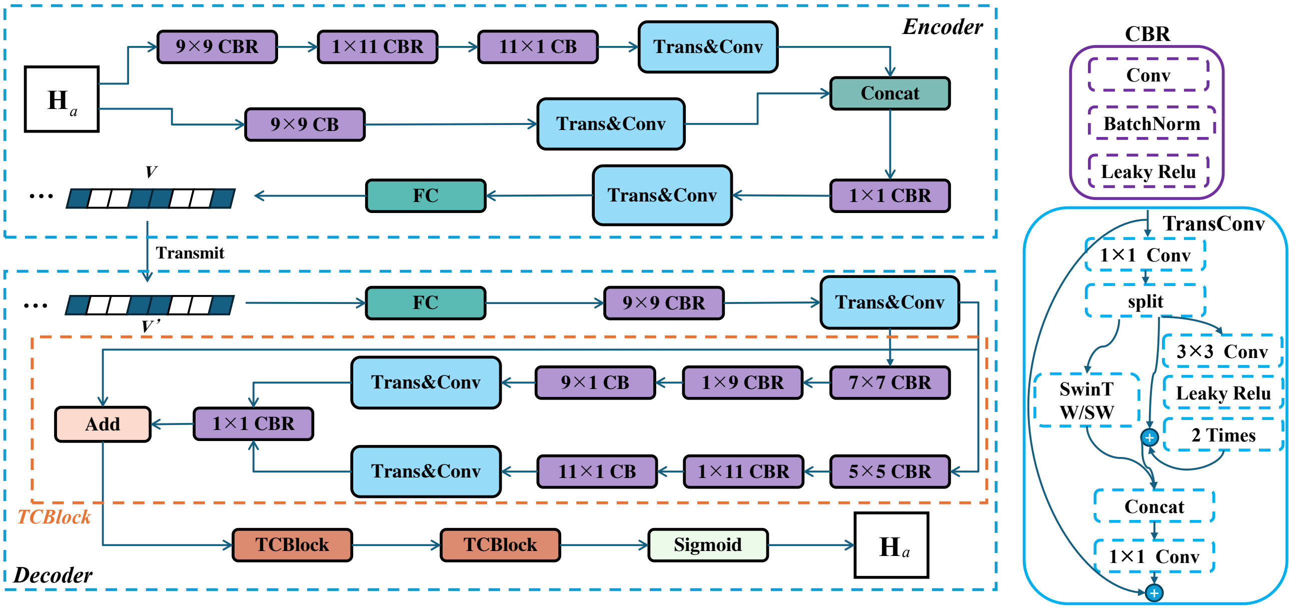

Figure 4. The proposed Mixed Transformer-CNN-based Lossy Compressor.

this distribution, an adaptive arithmetic coder performs lossless encoding on the token sequence, generating a compact bitstream for transmission.

At the decoder side, the received bitstream is decoded using arithmetic decoding guided by the same language model's probability distribution, thereby recovering the original token sequence. The ASCII detokenization module then maps the tokens back to quantized values, which are subsequently dequantized. Finally, a logit function is applied to reconstruct the activated feature values, completing the decoding process, as expressed by

$$\text{Logit} = \ln \frac{y}{1-y}. \tag{7}$$

Subsequently, the CSI is reconstructed using the decoder module based on the Transformer–CNN architecture. Zero-padding is then applied to restore the frequency-domain representation to its original full matrix shape. Finally, an inverse discrete Fourier transform (IDFT) is performed to convert the CSI from the frequency domain back to the time domain, thereby completing the reconstruction of the original CSI.

In TCNet, both tokenization and detokenization are entirely lossless. However, the quantization and dequantization steps inherently introduce quantization errors, which directly affect the fidelity of the reconstructed data.

3.2. Mixed Transformer-CNN-based Lossy Compressor

Existing lossy CSI compression methods are typically built upon either CNNs or Transformer architectures. While CNNs exhibit strong capabilities in capturing local features, they are inherently limited in modeling long-range dependencies. On the other hand, Transformers excel at learning global contextual relationships but often struggle with preserving fine-grained local spatial details, which are critical for accurately reconstructing complex CSI distributions.

To address these limitations and motivated by TCM network (Liu et al., 2023), this paper proposes a hybrid network architecture that integrates the strengths of both CNNs and Transformers, as shown in Figure 4. The proposed design aims to achieve an effective balance between local feature extraction and global semantic modeling, while also reducing computational complexity.

To begin with, the processed CSI $\mathbf{H}_a$ is fed into two parallel convolutional branches. The first branch consists of three sequential convolutional blocks for hierarchical feature extraction. It begins with a convolutional block using a 9×9 kernel with batch normalization and ReLU activation (CBR), followed by a 1×11 CBR, and then a 11×1 convolutional block with only convolution and batch normalization. The output is then passed into a TransConv module, a hybrid Transformer–CNN unit designed to capture both global and local features.

The second branch contains a 9×9 CBR, followed by a TransConv module. Outputs from both branches are concatenated and passed through a 1×1 CBR to extract element-wise features, followed by another TransConv module for further representation learning. Finally, a fully connected layer compresses the output into a feature vector sequence according to the desired compression ratio.

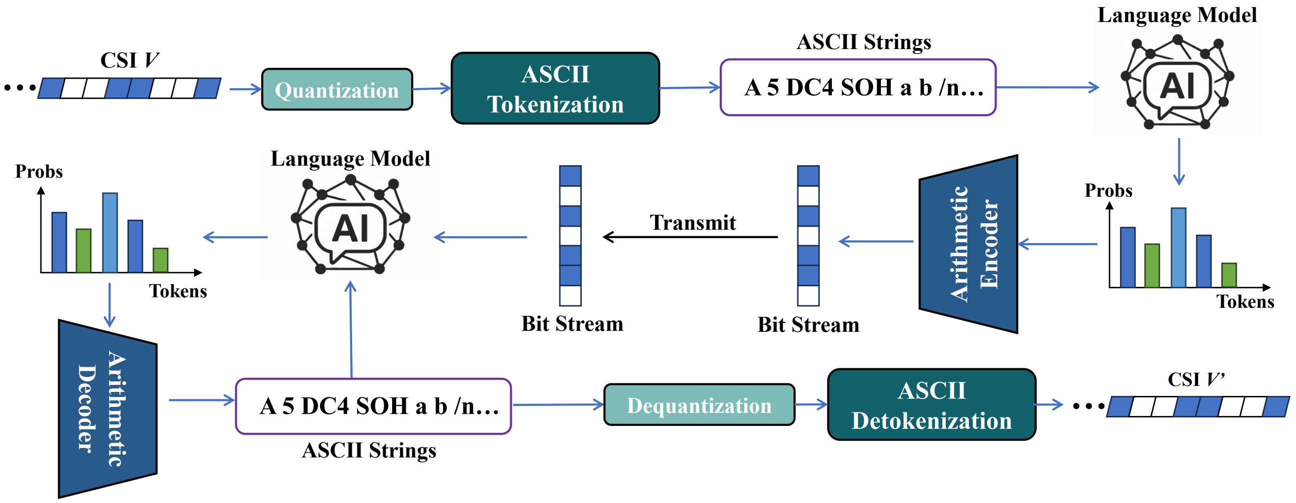

Figure 5. The proposed language model-based coding scheme.

In the decoding stage, the compressed feature vector is first projected back to its original size using a fully connected layer. This is followed by a 9×9 convolutional block and a TransConv module to integrate both local and global information. The reconstructed CSI is then progressively refined through three TCBlock modules. Each TCBlock is based on a residual design inspired by ResNet (He et al., 2016), and includes a TransConv module whose configuration differs from the encoder in terms of convolutional and attention dimensions. This adjustment allows the decoder to better reconstruct global features from higher-dimensional representations. A 1×1 convolutional block is then used to enhance detail-level accuracy. After passing through all three TCBlocks, the final CSI matrix is reconstructed using a sigmoid activation function.

3.3. Language Model-Based Arithmetic Coding

Pretrained large language models (LLMs) have demonstrated strong potential for lossless compression across various data modalities, including text, images, and audio (Delétang et al., 2023). This success is largely attributed to the extensive and diverse nature of the training corpora used to build such models. However, data of communications such as CSI exhibits statistical characteristics that differ significantly from those of natural language or visual data. As a result, directly applying pretrained LLMs for modeling the distribution of CSI tends to yield suboptimal compression performance. To address this issue, we propose retraining a language model specifically tailored to the statistical properties of CSI. Furthermore, to meet the real-time requirements of CSI feedback, we adopt a lightweight single-layer Transformer (Vaswani et al., 2017) as our probabilistic predictor.

Figure 5 illustrates the proposed language model-based coding scheme. At the encoder, the CSI feature vector generated by the lossy compressor is first processed through n-bit uniform quantization, producing a sequence of 2^n discrete symbols $S = \{0, 1, 2, ..., 2^n - 1\}$. These symbols are then tokenized into a sequence of ASCII characters A. These characters are input to the trained language model $LM(.)$, predicting the conditional probability distribution of each character based on its contextual history, as expressed by

$$\rho = LM(A). \tag{8}$$

These predicted probabilities are then used by an adaptive arithmetic encoder to produce a compact, lossless bitstream for transmission. We apply infinite precision arithmetic coders and refer to (Witten et al., 1987) for the finite-precision implementation. To be precise, arithmetic encoding represents a sequence of symbols as the binary representation of a real number λ within the interval $[0, 1)$. The encoding process progressively narrows this interval to isolate a unique subinterval corresponding to the entire symbol sequence. Initially, the full interval is defined as $[0, 1)$. At each encoding step, the current interval $I_{k-1} = [l_{k-1}, u_k - 1)$ is partitioned into multiple sub-intervals $\tilde{I}_k(x_1), \tilde{I}_k(x_2), ...,$ each associated with a distinct symbol from a predefined alphabet $X = x_1, x_2, ..., x_N$. These sub-intervals are sized in proportion to the conditional probability of each symbol given the preceding context. We employ

$$\tilde{I}_k(x) = \Bigg[l_{k-1} + (u_{k-1} - l_{k-1}) \times \sum_{y<x} \rho\left(y \mid x_{<k}\right) \\ + (u_{k-1} - l_{k-1}) \times \sum_{y \leq x} \rho\left(y \mid x_{<k}\right) \Bigg]. \tag{9}$$

To encode the symbol at position k, the interval from the previous step is subdivided based on the symbol probabilities conditioned on all prior symbols. The sub-interval assigned to a given symbol is calculated by accumulating the probabilities of all symbols that precede it in the predefined ordering, thus determining the lower and upper bounds of the current interval. The encoder then selects the sub-interval corresponding to the current symbol as the updated interval for the next step.

At the decoder, the process is reversed. Starting with a predefined placeholder token, the language model estimates the probability distribution of the first character. This distribution, combined with the received bitstream, is passed to the arithmetic decoder to recover the first ASCII character. The recovered character and the placeholder token form the context for predicting the next character, and this process continues iteratively until the entire ASCII string is reconstructed. The reconstructed character sequence is then mapped back to quantized values via ASCII detokenization and subsequently dequantized to restore the CSI feature vector.

In fact, the precise source entropy ρ is unknown and is estimated with a parametric probabilistic model $\hat{\rho}$. Therefore, the expected suboptimal number of bits is the cross-entropy

$$H(\rho, \hat{\rho}) = E_{x \sim p}\left[\sum_{i=1}^{n} -\log_2 \hat{\rho}(x_i|x_{<i})\right]. \qquad (10)$$

Therefore, reducing the log-loss corresponds to lowering the compression rate when the model is utilized as a lossless compressor through arithmetic coding. In essence, contemporary language model training frameworks inherently pursue a maximum compression goal.

4. Experimental Results

4.1. Experimental Setup

In this work, the model performance was systematically evaluated under an indoor environment at 5.3 GHz and an outdoor environment at 300 MHz. All channel data were generated based on the standard parameters of the COST2100 channel model, and the experimental settings were kept consistent with those used by the baseline methods for a fair comparison.

The simulation parameters for the communication system are as follows: the base station is equipped with a uniform linear array consisting of 32 antennas. Operating in FDD mode, the frequency domain is divided into 1024 subcarriers, with an angular resolution set to 32. The COST2100 dataset comprises 150,000 channel samples, partitioned into 100,000 samples for training, 30,000 for validation, and 20,000 for testing.

The model was implemented using the PyTorch framework. Network parameters were initialized using the Xavier initialization method. The learning rate scheduler followed a cosine annealing schedule, with an initial learning rate of 0.002 and a minimum learning rate of 0.00005. Training was conducted over 500 epochs, with the first 20 epochs designated as a warm-up phase.

For the Transformer language model, the vocabulary size was set to 256 and the embedding dimension to 256. The model architecture consists of 4 stacked layers of multi-head self-attention, each with 8 attention heads. The feed-forward network within each layer has a hidden dimension four times the embedding size, i.e., 1024. The batch size was set to 16.

4.2. Experimental Metrics

To evaluate the performance of the feedback network, we adopt the normalized mean square error (NMSE) to compare the reconstruction accuracy of different methods. In addition, the floating point operations (FLOPs) are employed to measure the computational complexity of each method. Specifically, the NMSE is calculated between the DFT-processed CSI matrix and the reconstructed CSI matrix., as expressed by

$$\text{NMSE} = E\left\{\frac{\|\mathbf{H}_a - \hat{\mathbf{H}}_a\|_2^2}{\|\mathbf{H}_a\|_2^2}\right\}. \qquad (11)$$

To evaluate the effectiveness of the network in terms of transmission efficiency, we also adopt the number of bits required per CSI value as a metric to quantify the compression performance. Assuming the original CSI data consists of N elements ($\mathbf{H}_a$, we follow (Wen et al., 2018)) and the total number of transmitted bits is B. Therefore, the bit rate per CSI element is defined as

$$\text{Bit Rate} = \frac{B}{N}. \qquad (12)$$

4.3. Quantitative Results

We compare the proposed method, TCNet, with several representative baselines, including CLNet (Ji & Li, 2021), CRNet (Lu et al., 2020), CsiNet+ (Guo et al., 2020), and TransNet (Cui et al., 2022), in terms of NMSE and bit rate. Among these methods, only CsiNet+ adopts mu-law quantization, while the others employ uniform quantization.

Table 1 presents the average bit rates and the corresponding NMSE of TCNet, compared with CLNet, CRNet, and TransNet under 7-bit quantization on the COST2100 dataset. When the bit rate reaches 0.3275 bits and 0.176 bits, TCNet achieves NMSEs of -14.68 dB and -10.17 dB, respectively, surpassing CLNet and CRNet by approximately 2 to 3 dB. Even under extremely low bit rate conditions, TCNet maintains a reconstruction accuracy of -7.80 dB, whereas CLNet and CRNet only reach -6.34 dB and -6.49 dB at an average bit rate of 0.1094 bits, clearly demonstrating TCNet's superior error resilience and robustness within the evaluated compression range. TransNet achieves a lower NMSE at slightly higher bit rates; however, its bit rate-NMSE tradeoff remains slightly inferior to that of TCNet.

Table 1. Experimental results of 7-bit quantized TCNet and other methods' NMSE on the COST2100 Dataset.

Method	FLOPs	Bit Rate	Entropy	NMSE
CLNet	4.05M	1.75	NA	-28.58
	3.01M	0.875	NA	-15.16
	2.48M	0.4375	NA	-11.13
	2.22M	0.2188	NA	-8.94
	2.09M	0.1094	NA	-6.34
CRNet	5.12M	1.75	NA	-25.61
	4.07M	0.875	NA	-15.57
	3.55M	0.4375	NA	-11.33
	3.29M	0.2188	NA	-8.92
	3.16M	0.1094	NA	-6.49
TransNet	35.72M	1.75	NA	-30.01
	34.70M	0.875	NA	-22.89
	34.14M	0.4375	NA	-15.06
	33.88M	0.2188	NA	-10.18
	33.75M	0.1094	NA	-5.777
TCNet	**31.47M**	**1.258**	**1.256**	**-28.39**
	30.42M	**0.675**	**0.671**	**-19.39**
	29.89M	**0.3275**	**0.3250**	**-14.68**
	29.64M	**0.176**	**0.1726**	**-10.17**
	29.50M	**0.0983**	**0.0845**	**-7.80**

Regarding the entropy of the symbol sequence, the bit rate achieved by TCNet approaches the theoretical lower bound, indicating a near-optimal coding efficiency. Moreover, as the length of the symbol sequence increases, TCNet exhibits enhanced compression capability. This is attributed to the fact that longer sequences contain richer temporal and spatial correlations, making them more predictable. The language model is able to effectively capture and exploit these statistical patterns for more efficient entropy coding. In contrast, shorter sequences tend to exhibit weaker memory and predictability, resulting in more dispersed probability distributions that are harder to model.

Table 2 and 3 presents a comparison of TCNet with CLNet, CRNet, CsiNet+, and TransNet under 6-bit and 5-bit quantization on the COST2100 dataset. The results show that TCNet consistently demonstrates superior performance across different bit rate conditions. Moreover, the average bit rate of TCNet encoding is still nearly equal to the source entropy, further demonstrating the accuracy of the language model's probability distribution prediction and the efficiency of the adaptive arithmetic coding scheme.

Table 2. Experimental results of 6-bit quantized TCNet and other methods' NMSE on the COST2100 Dataset.

Method	FLOPs	Bit Rate	Entropy	NMSE
CLNet	4.05M	1.5	NA	-27.16
	3.01M	0.75	NA	-14.07
	2.48M	0.375	NA	-11.09
	2.22M	0.1875	NA	-8.92
	2.09M	0.0938	NA	-6.33
CRNet	5.12M	1.5	NA	-22.97
	4.07M	0.75	NA	-14.38
	3.55M	0.375	NA	-11.27
	3.29M	0.1875	NA	-8.90
	3.16M	0.0938	NA	-6.48
CsiNet+[1]	24.57M	1.5	NA	-24.97
	23.52M	0.75	NA	-18.03
	23.00M	0.375	NA	-14.02
	22.74M	0.1875	NA	-10.35
TransNet	35.72M	1.5	NA	-28.84
	34.70M	0.75	NA	-22.55
	34.14M	0.375	NA	-14.91
	33.88M	0.1875	NA	-10.15
	33.75M	0.0938	NA	-5.708
TCNet	**31.47M**	**1.058**	**1.018**	**-26.78**
	30.42M	**0.535**	**0.533**	**-19.22**
	29.89M	**0.265**	**0.2649**	**-14.54**
	29.64M	**0.1445**	**0.1408**	**-10.13**
	29.50M	**0.0831**	**0.0806**	**-7.79**

[1] CsiNet+ is implemented with μ-law quantization.

4.4. Qualitative Results

To clearly illustrate the effectiveness of TCNet, the reconstruction NMSE of various methods at the corresponding bit rate is plotted together in an Bit Rate-NMSE curve, as shown in Figure 6. At low bit rates, TCNet achieves better performance than CsiNet+ and TransNet, all outperforming CLNet and CRNet. As the bit rate increases, the reconstruction advantage of TCNet becomes pronounced, which stems from the higher bit rate enabling the language model to capture stronger sequential correlations, thereby facilitating more accurate predictions and enhancing compression efficiency. Furthermore, we evaluate the performance of various methods under low bit-rate constraints in the COST2100 outdoor scenario, as shown in Figure 7. As illustrated, TCNet is capable of achieving relatively high-fidelity CSI feedback with low transmission cost.

4.5. Discussion

Although TCNet has demonstrated promising performance, several aspects remain to be discussed when considering its deployment in practical scenarios.

Table 3. Experimental results of 5-bit quantized TCNet and other methods' NMSE on the COST2100 Dataset.

METHOD	FLOPS	BIT RATE	ENTROPY	NMSE
CLNET	4.05M	1.25	NA	-23.82
	3.01M	0.625	NA	-11.13
	2.48M	0.3125	NA	-10.94
	2.22M	0.1563	NA	-8.82
	2.09M	0.0781	NA	-6.01
CRNET	5.12M	1.25	NA	-18.29
	4.07M	0.625	NA	-11.11
	3.55M	0.3125	NA	-11.04
	3.29M	0.1563	NA	-8.79
	3.16M	0.0781	NA	-6.46
CSINET+[1]	24.57M	1.25	NA	-22.13
	23.52M	0.625	NA	-17.23
	23.00M	0.3125	NA	-13.64
	22.74M	0.1536	NA	-10.10
TRANSNET	35.72M	1.25	NA	-25.85
	34.70M	0.625	NA	-21.40
	34.14M	0.3125	NA	-14.34
	33.88M	0.1536	NA	-10.04
	33.75M	0.0781	NA	-5.443
TCNET	**31.47M**	**0.746**	**0.7425**	**-22.85**
	30.42M	**0.411**	**0.408**	**-18.55**
	29.89M	**0.2025**	**0.2013**	**-13.98**
	29.64M	**0.1138**	**0.1125**	**-9.95**
	29.50M	**0.068**	**0.0655**	**-7.71**

[1] CSINET+ IS IMPLEMENTED WITH μ-LAW QUANTIZATION.

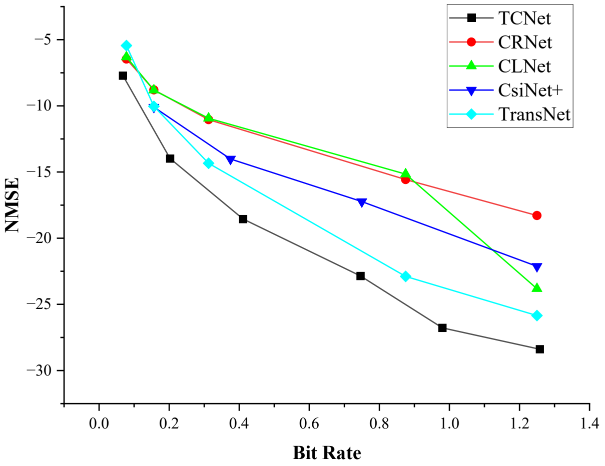

Figure 6. Qualitative results of quantized TCNet and other methods' NMSE versus bit rate plot on the indoor scenario of COST2100 dataset.

- **Autoregressive Decoding**. Autoregressive decoding adopted in TCNet inherently introduces latency due to its sequential nature, where each output token depends on the prediction of the previous one. However, techniques such as knowledge distillation, speculative decoding, and lookahead decoding enable partial parallelism while preserving accuracy. These methods can be effectively employed and the latency bottleneck of autoregressive decoding is not insurmountable.

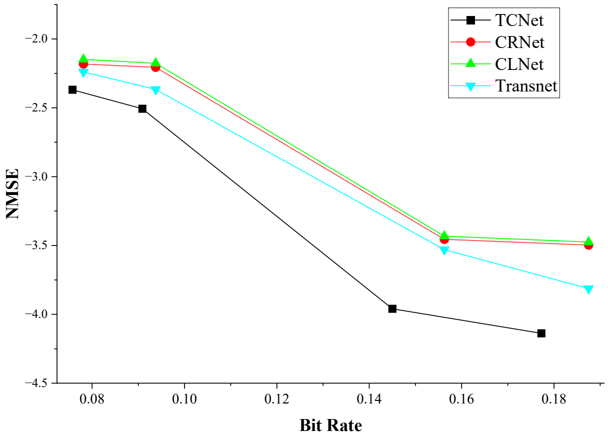

Figure 7. Qualitative results of quantized TCNet and other methods' NMSE versus bit rate plot on the outdoor scenario of COST2100 dataset.

- **Edge Computing**. Deep learning-based methods typically require substantial computational resources. Thanks to the development of mobile CPUs, embedded GPUs, and NPUs, models with computational complexity on the order of tens of MegaFLOPs (MFLOPs) are well-suited for deployment on edge devices, providing greater potential for the advancement of deep learning-based CSI feedback.

5. Conclusion

This paper presents TCNet, a novel CSI feedback framework that efficiently compresses CSI from both lossy and lossless perspectives. Firstly, we integrate CNNs and Swin Transformers to capture local and global features, achieving high reconstruction accuracy with reduced complexity. Experiments on the COST2100 dataset show that our method consistently outperforms prior methods. Meanwhile, we combine language model with adaptive arithmetic coding, offering superior bit rate–NMSE trade-offs without the need to train quantizers. The results highlight the strong potential of integrating language models with statistical coding for efficient lossless compression.

References

Cai, Q., Dong, C., and Niu, K. Attention model for massive mimo csi compression feedback and recovery. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–5. IEEE, 2019.

Chan, P. W., Lo, E. S., Wang, R. R., Au, E. K., Lau, V. K., Cheng, R. S., Mow, W. H., Murch, R. D., and Letaief, K. B. The evolution path of 4g networks: Fdd or tdd? *IEEE Communications Magazine*, 44(12):42–50, 2006.

Cui, Y., Guo, A., and Song, C. Transnet: Full attention network for csi feedback in fdd massive mimo system. *IEEE Wireless Communications Letters*, 11(5):903–907, 2022.

Delétang, G., Ruoss, A., Duquenne, P.-A., Catt, E., Genewein, T., Mattern, C., Grau-Moya, J., Wenliang, L. K., Aitchison, M., Orseau, L., et al. Language modeling is compression. *arXiv preprint arXiv:2309.10668*, 2023.

Dong, Y., Wang, H., and Yao, Y.-D. Channel estimation for one-bit multiuser massive mimo using conditional gan. *IEEE Communications Letters*, 25(3):854–858, 2020.

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

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

Liu, J., Sun, H., and Katto, J. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14388–14397, 2023.

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

Shafin, R. and Liu, L. Multi-cell multi-user massive fd-mimo: Downlink precoding and throughput analysis. *IEEE transactions on wireless communications*, 18(1): 487–502, 2018.

Tang, S., Xia, J., Fan, L., Lei, X., Xu, W., and Nallanathan, A. Dilated convolution based csi feedback compression for massive mimo systems. *IEEE Transactions on Vehicular Technology*, 71(10):11216–11221, 2022. doi: 10.1109/TVT.2022.3183596.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

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

Witten, I. H., Neal, R. M., and Cleary, J. G. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.

Yang, Q., Mashhadi, M. B., and Gündüz, D. Deep convolutional compression for massive mimo csi feedback. In *2019 IEEE 29th international workshop on machine learning for signal processing (MLSP)*, pp. 1–6. IEEE, 2019.