Learning Compact Representations with Graph Neural Networks for CSI Feedback

Manoj Kumar CM and Lakshmi N. Theagarajan
Department of Data Science and AI, Indian Institute of Technology Madras
Department of Electrical Engineering, Indian Institute of Technology Madras

***Abstract*—This paper presents a graph neural network architecture for compressing and reconstructing channel state information (CSI) in cellular systems, aiming to alleviate the significant overhead that high-dimensional CSI feedback poses in massive MIMO and broadband OFDM environments. We observe that the CSI, in the delay-angle domain, is sparser and possesses locally correlated peaks. We propose a graph neural network (GNN) based autoencoder with attention and adaptive quantizer to exploit these structures for CSI compression. The novel GNN-based approach models relational structures among CSI elements using dynamically constructed graphs and attention mechanisms, capturing both local and global dependencies in the delay-angular domain. GNNs can dynamically define relationships (graph edges) based on the similarity or spatial closeness of features (graph nodes, which represent patches or peaks in delay-angle CSI data). Experimental results, based on data generated using 3GPP-compliant models and real-life CSI measurements, demonstrate that the GNN-based architecture consistently outperforms all other neural architectures, offering up to 30x improvement in performance.**

Keywords: CSI compression, CSI feedback, graph neural networks, autoencoder, attention model.

I. Introduction

The increasing deployment of massive MIMO and broadband OFDM systems in modern wireless networks introduces substantial overhead in acquiring and feeding back high-dimensional channel state information (CSI), which is essential for beamforming and power control. To mitigate this overhead, efficient CSI compression has become critical. Transforming CSI into the delay–angular domain via a two-dimensional discrete Fourier transform (2D-DFT) reveals that most channel energy is concentrated in a few coefficients, enabling compact representations. Further, deep learning–based autoencoders have shown remarkable potential for this task by leveraging the sparsity of CSI in transformed domains. CsiNet [1] and CsiNet-LSTM [2] utilized convolutional and recurrent autoencoders for efficient CSI compression and recovery, while CRNet [3] improved performance through a multi-resolution design that captures residual and multi-scale spatial features.

Recently, researchers have begun exploring graph neural network (GNN)–based architectures for CSI modeling and prediction. For instance, STEM-GNN [4] captures both spectral and temporal correlations in massive MIMO channels using a spectral–temporal graph framework, while ERAN-GNN [5] dynamically adapts graph structures to account for channel aging and temporal evolution. Graph neural networks can capture spatial correlations among disconnected CSI components through a compact representation using graphs. In this paper, we propose a graph attention–based neural architecture for CSI compression. By explicitly modeling relational dependencies among antennas, the proposed GNN captures both local and global structures, leading to more expressive but compact CSI representations. Experimental results demonstrate that our GNN-based approach achieves superior compression performance compared to existing deep learning models such as CsiNet and CRNet. To the best of our knowledge, this is the first time an attention based GNN architecture with adaptive quantization has been developed for CSI compression.

II. Channel Model

The downlink channel of an FDD (frequency division duplex) system is considered. The mobile user equipment (UE) estimates the CSI using the pilot signals from the base station (BS) with multiple antennas (say M). Orthogonal frequency division modulated (OFDM) data symbols are transmitted over multiple frequency domain subcarriers (say, N) at a given instant of time. Thus, the baseband signal received at the receiver in the frequency domain is given by

$$y[f] = \mathbf{H}[f, s]^T \mathbf{x}[f, s] + z[f], \tag{1}$$

where $\mathbf{H}(f, s)$ is the wireless channel gain for the fth subcarrier and sth spatial antenna, $\mathbf{x}$ is the transmitted signal and z is the additive noise. This channel state information (CSI) matrix $\mathbf{H}$ is required for precoding at the transmitter. In current 4G/5G cellular systems, known reference signals are transmitted from which CSI is estimated, and then communicated to the BS via the feedback control channel. Here, the CSI remains approximately constant for a certain time duration (a.k.a. coherence time). To reduce the feedback overhead, CSI data is compressed before communication.

It is known that the channel is sparser in the delay-angular domain compared to the frequency-space domain representation [6]. The frequency domain representation of the wireless channel can be converted to the delay domain by inverse Fourier transform and the space

domain can be converted to angular domain through Fourier transform. Therefore, the relationship between the space-frequency CSI ($H[n,m]$) and delay-angle CSI ($G[n,m]$) is given by

$$G[n,m] = \frac{1}{\sqrt{MN}} \sum_{n=1}^{N} \sum_{m=1}^{M} H[n,m] e^{j2\pi\left(\frac{ml}{M} - \frac{nk}{N}\right)}. \quad (2)$$

An illustration of the CSI in these two domains is

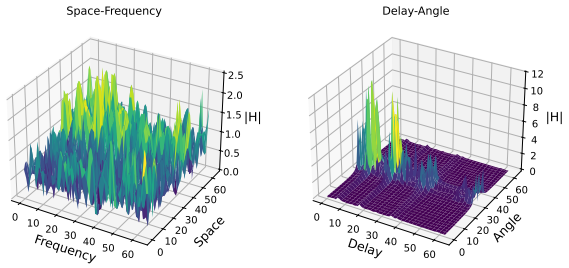

Fig. 1: A CSI sample generated using NVIDIA Sionna.

presented in Fig. 1. It can be observed that in the delay-angle domain the CSI is not only sparser but also is more localized in pockets of a few distributed delay-angle bins. Thus, any compression technique that could take advantage of this 2D localizations and global spatial correlations can perform well. In this paper, we propose to exploit this 2D correlation structure using GNN that models each localized pockets as a node using k-nearest neighbors method and compresses the representation of similar nodes to provide a very high compression ratio while maintaining a low reconstruction error. The local and global CSI/graph-nodes dependencies are discovered through message passing mechanisms in the GNN.

III. Deep Learning Architectures for CSI Compression: From CNNs to GNNs

This section describes the state-of-the-art deep learning architectures CsiNet [1], CRNet [3], and our proposed GNN-based architecture for CSI compression.

A. CNN based CSI Compression

The state-of-the-art deep learning models employ convolutional neural networks (CNN) for compression. CsiNet [1] treats the CSI matrix as a 2D image and employs a CNN autoencoder for compression and reconstruction. The encoder maps high-dimensional CSI to a compact codeword, while the decoder uses fully connected and RefineNet [1] layers to recover the original CSI, effectively leveraging local spatial correlations. CRNet [3] extends this design with a multi-resolution architecture that combines multi-scale convolutions and channel attention. This yields a more expressive yet lightweight model, improving reconstruction accuracy and efficiency, especially at low compression ratios.

While both CsiNet and CRNet rely on convolutions to capture spatial dependencies, they do not explicitly model the relational structure of CSI. To address this, we construct GNN architectures, which can represent CSI as structured data and effectively capture local and global correlations through graph message passing.

B. Graph Neural Networks (GNN)

The above neural architectures inherently assume a rigid grid structure for CSI. However, CSI data can exhibit more complicated, irregular spatial relationships that are not easily captured by regular grids or simple local filters. There could be a subset of patches with an irregular shape in CSI data that are locally correlated. GNNs can naturally model such non Euclidean structures and arbitrary relationships among CSI patches, capturing both local and global dependencies beyond fixed grid patterns. Instead of fixed neighboring patterns (as in CNNs), GNNs dynamically define relationships (edges) based on feature (patches) similarity or spatial closeness. These relations are dynamically discovered using message passing on a graph that represent the input CSI.

The proposed GNN based CSI compressor (GCC) consists of two components: (1) graph attention based autoencoder, and (2) hyperprior generator. The autoencoder architecture incorporates graph attention mechanisms within its encoder and decoder to capture relational dependencies among the elements of the CSI matrix. The hyperprior generator is a learned model that estimates the parameters of the distribution of the latent representations.

1) Graph attention based autoencoder: The encoder (f_e) acts as an inference model, that maps/infers a latent representation $\tilde{\mathbf{Y}}$ (with quantization) for input CSI $\mathbf{X}$. The statistics of this process are represented by the conditional distribution $q(\tilde{\mathbf{Y}} \mid \mathbf{X}; f_e)$. The decoder ($f_d$) acts as a generative model reconstructing the CSI $\hat{\mathbf{X}}$ from the latent representation $\tilde{\mathbf{Y}}$. The statistic used in reconstruction is the conditional distribution $p(\mathbf{X} \mid \tilde{\mathbf{Y}}; f_d)$. The parameters of the autoencoder are obtained by minimizing the Kullback–Leibler divergence between the variational density $q(\tilde{\mathbf{Y}} \mid \mathbf{X})$ and the true (but intractable) posterior $p(\tilde{\mathbf{Y}} \mid \mathbf{X})$ given by

$$\begin{aligned}\mathbb{E}_{\mathbf{X}\sim p(\mathbf{X})}\left[\mathrm{D}_{\mathrm{KL}}(q(\tilde{\mathbf{Y}} \mid \mathbf{X}) \,\|\, p(\tilde{\mathbf{Y}} \mid \mathbf{X}))\right] = \\ \mathbb{E}_{\mathbf{X}\sim p(\mathbf{X})}\Big[\mathbb{E}_{\tilde{\mathbf{Y}}\sim q(\tilde{\mathbf{Y}} \mid \mathbf{X})}\big[\log q(\tilde{\mathbf{Y}} \mid \mathbf{X}) \\ - \log p(\mathbf{X} \mid \tilde{\mathbf{Y}}) - \log p(\tilde{\mathbf{Y}})\big]\Big] + \text{const.}\end{aligned}$$

By conditioning the latent representation $\tilde{\mathbf{Y}}$ on another latent/prior variable $\tilde{\mathbf{z}}$ that captures the statistics of $\tilde{\mathbf{Y}}$, the above loss can be simplified to the rate distortion loss which can be written as:

$$\begin{aligned}\mathcal{L} = \mathbb{E}_{\mathbf{X}\sim p_{\mathbf{X}}}\left[-\log_2 p_{\tilde{\mathbf{Y}}|\tilde{\mathbf{z}}}(\tilde{\mathbf{Y}}|\tilde{\mathbf{z}}) - \log_2 p_{\tilde{\mathbf{z}}}(\tilde{\mathbf{z}})\right] \\ + \lambda \mathbb{E}_{\mathbf{X}\sim p_{\mathbf{X}}}[d(\mathbf{X}, \hat{\mathbf{X}})],\end{aligned} \quad (3)$$

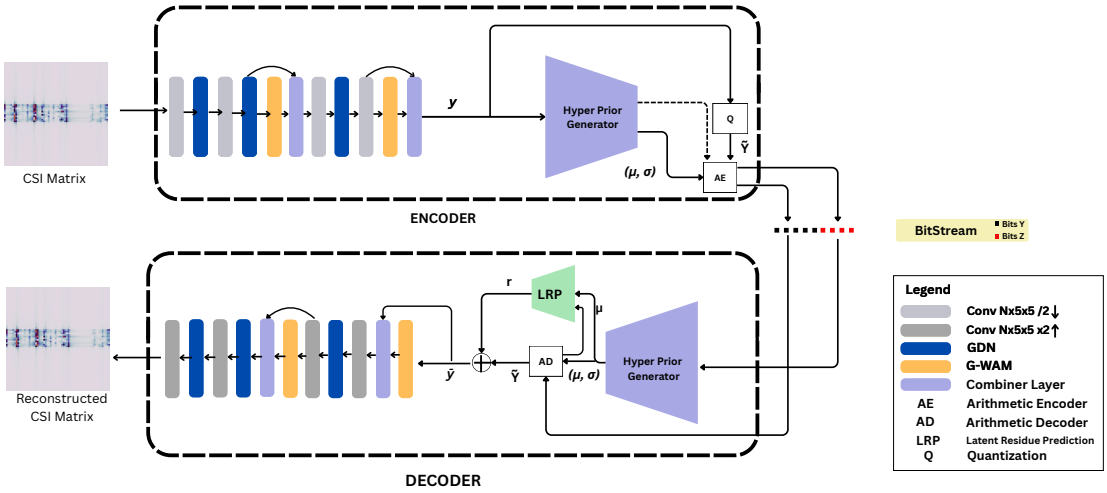

Fig. 2: Architecture of the GNN based CSI Compressor(GCC).

where the first two terms correspond to the rates(entropy) of $\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{z}}$, respectively, the last term is the MSE distortion, $\tilde{\mathbf{Y}}$ is the output of the encoder, $\tilde{\mathbf{z}}$ is the output of the hyperprior generator and $\hat{\mathbf{X}}$ is the reconstructed CSI at the decoder. The conditional distribution $p_{\tilde{\mathbf{Y}}|\tilde{\mathbf{z}}}$ is learned from the input data and this learning is carried out by the hyperprior generator.

The encoder first transforms the input CSI matrix $\mathbf{X}$ into a latent representation $\mathbf{Y} = f_e(\mathbf{X}; \phi)$, where ϕ represents the parameters of the encoder network. This network has a series of convolutional layers with kernel size 5 x 5 and stride 2, interleaved with generalized divisive normalization (GDN) [7] and graph based window attention mechanism (G-WAM) [8]. GDN is a parametric and invertible nonlinearity that converts the input to approximately follow Gaussian distribution by normalizing each activation with a learned and pooled activity measure.

G-WAM computes attention over non-overlapping sub-matrices of the feature maps, each of which are treated as nodes of a dynamically constructed graph. The graphs are constructed as follows: (1) the output features of the GDN block are segregated into four different sets; these sets form the vertex set of four graphs, (2) for each vertex/node, a local neighborhood is determined using k-nearest neighbors search in the feature space. Attention weights are computed based on feature similarity, and each node's embedding is updated by graph message passing using information from its neighbors weighted by the learned attention coefficients. Each G-WAM block is followed by a combiner layer which merges the attention enhanced features with the original features that were input to the attention block - acting as a skip connection. This dynamic graph formulation of the GNN based on the input CSI enables capturing non-Euclidean correlations and patterns for efficient compression.

The generated latent representation $\mathbf{Y}$ is quantized using a learned mean $\boldsymbol{\mu}$ as $\tilde{\mathbf{Y}} = Q(\mathbf{Y} - \boldsymbol{\mu}) + \boldsymbol{\mu}$, where $\boldsymbol{\mu}$ is obtained from the hyperprior generator using $\mathbf{Y}$. The compressed encoder output is the bit sequence $\mathbf{b}$ generated by lossless arithmetic encoding of $\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{z}}$.

At the decoder, the quantized side information $\tilde{\mathbf{z}}$ is obtained from arithmetic decoding, using which the statistics of the latent representations $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\sigma}}$ are estimated from the hyperprior generator. A latent residue prediction (LRP) network is employed at the decoder to estimate the quantization error in latent representations, i.e., $\mathbf{r} = \mathbf{Y} - \tilde{\mathbf{Y}}$. This residual is modeled as a function of the hyperpriors and the set of previously arithmetic decoded $\tilde{\mathbf{Y}}$ elements. This predicted residual is then added back to the quantized latents as $\hat{\mathbf{Y}} = \tilde{\mathbf{Y}} + \hat{\mathbf{r}}$ at the decoder. This reduces the distortion in the reconstruction. Finally, $\hat{\mathbf{Y}}$ is passed through the decoder to reconstruct the image: $\hat{\mathbf{X}} = f_d(\hat{\mathbf{Y}}; \theta)$, where θ denotes the parameters of the decoder. The decoder network comprises of series of convolutional transpose layers with kernel size 5 x 5 and stride 2, interleaved with GDN and G-WAM. The overall architecture of the proposed GCC is shown in Fig. 2. After LRP, the decoder applies layers as in the encoder but in reverse order. Therefore, it should be noted that the proposed GCC does not have a conventional autoencoder architecture.

Note: Due to the non-differentiable nature of the quantization operation $Q()$, optimizing ϕ and θ using gradient descent becomes intractable. This is overcome by approximating the quantization operation as adding uniform noise (i.e., the quantization noise) to the latent variables $\mathbf{Y}$. This approximation makes the process differentiable, allowing for the use of gradient descent.

2) Hyperprior Generator: The hyperprior generator produces ($\mathbf{z}$) a latent representation for the prior distribution $p(\tilde{\mathbf{Y}})$. This is given by $\mathbf{z} = h_e(\mathbf{Y}, \phi_h)$, where ϕ_h

are the parameters of the hyperprior generator and h_e is the hyperprior generator network. This is quantized to obtain $\tilde{\mathbf{z}} = Q(\mathbf{z})$. The prior distribution is modeled as a fully factorized density, i.e.,

$$p_{\tilde{\mathbf{Y}}|\tilde{\mathbf{z}}}(\tilde{\mathbf{Y}}|\tilde{\mathbf{z}};\theta_h) = \prod_i \left(\mathcal{N}(0,{\sigma_i}^2) * \mathcal{U}\left(-\tfrac{1}{2},\tfrac{1}{2}\right)\right)(\tilde{Y}_i),$$
$$\boldsymbol{\mu},\boldsymbol{\sigma} = h_e(\mathbf{z};\theta_h), \tag{4}$$

where each Y_i is modeled as Gaussian with standard deviation σ_i which is not fixed but learned from the input. This model captures the spatial dependencies between the elements in the latent representation. This is a suitable model due to the presence of the GDN layer.

Further, $\tilde{\mathbf{z}}$ is the quantized version of $\mathbf{z}$ and the additive quantization noise is given by the uniform distribution. Since no prior knowledge exists about $\tilde{\mathbf{z}}$, its distribution is modeled using a non-parametric density model as

$$p_{\tilde{\mathbf{z}}|\boldsymbol{\psi}}(\tilde{\mathbf{z}} \mid \boldsymbol{\psi}) = \prod_i \left(p_{z_i|\psi^{(i)}}\left(\psi^{(i)}\right) * \mathcal{U}\left(-\tfrac{1}{2},\tfrac{1}{2}\right)\right)(\tilde{z}_i), \tag{5}$$

where the vectors $\psi^{(i)}$ encapsulate the parameters of each univariate distribution $p_{z_i|\psi^{(i)}}$. The hyperprior generator network h_e consists of series of 3 x 3 convolutional layers with Gelu activation, the hyperprior decoder network h_d consists of a similar architecture but in reverse order, h_d predicts $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ at from $\tilde{\mathbf{z}}$.

IV. Experimental Results

To evaluate the proposed GCC, we conducted experiments on both synthetic and real-world CSI datasets. The compression rate is measured as the ratio of the number of bits reduced due to compression to the number of bits in the input. We evaluated the performance for different compression rates from 40% (lower compression) to 95+ % (higher compression) in steps of 10%. The neural networks were trained using a batch size of 64 and for a maximum of 25 epochs.

A. CSI Datasets

For training, CSI matrices were generated using the clustered delay line (CDL) channel models implemented in NVIDIA Sionna [9]. This CSI dataset follows 3GPP specifications, with varying CDL profiles, delay spreads, angular spreads, path loss, and are simulated for a 64 antenna BS. An example of the generated CSI is presented in Fig. 1.

To test with real-world CSI data, we used the reconfigurable eco-system for next-generation end-to-end wireless (RENEW) dataset [10]. This dataset was collected using the Argos massive MIMO platform with 64 antenna BS operating at the 2.4 GHz ISM band. The measurement environment includes 4 line-of-sight and 5 non-line-of-sight clusters in near stable outdoor environments, covering over 225 user locations.

B. Performance Analysis

We choose the following metrics to evaluate the performance of the CSI compression techniques: (i) cosine similarity: measures the magnitude of normalized inner product between two vectors, and (ii) normalized mean square error (NMSE): measures the cumulative error in all the elements of the CSI matrix after reconstruction, reported in dB scale. Lower values of NMSE indicate higher performance, while higher values of cosine similarity denote better performance. When the CSI is represented in the delay-angle domain, the CSI matrix is converted from space-frequency domain before the compression operation and converted back to the space-frequency domain after the reconstruction. Thus, all performance metrics are evaluated for the CSI in space-frequency domain only, while the compression can be either in space-frequency or delay-angle domain.

C. Observations

Figures 3 and 4 compare the performance of CsiNet, CRNET and the proposed GCC at different compression rates in 3GPP channel models generated using Sionna and RENEW dataset, respectively. The results are presented for both compressing the CSI directly in the frequency-space domain and compressing the CSI after delay-angle transformed domain. As expected, the compression in the delay-angle domain is higher due to the compact CSI representation in this domain. Further, it can be clearly seen that the GNN based CSI compression outperforms the state-of-the-art deep learning models by a significant margin at all compression rates.

For both simulated and real-world datasets, GCC provides better or similar cosine similarity and NMSE at 98% compression rate compared to CRNet at 40%. In a system with 64 antennas, 64 subcarriers and 8-bit real-valued resolution, uncompressed CSI data requires 65536 bits of feedback; whereas, CRNet requires 39322 bits with 0.88 cosine similarity and -8 dB NMSE, but GCC requires only 1311 bits with 0.91 cosine similarity and -8 db NMSE. Thus, the proposed GCC can provide up to 30x higher compression than the state-of-the-art deep learning architectures for CSI feedback.

V. Interpretability Analysis

In this section, we analyze the GNN based CSI compressor layer-wise to understand its behavior.

A. Graph Visualization

First, we study the graphs that are dynamically generated by the GNN. For this analysis, we create a synthetic CSI matrix $\mathbf{H}_1$ with a single peak at small values of delay and angle. Another CSI matrix $\mathbf{H}_2$ is obtained with a similar peak at larger values of delay and angle; $\mathbf{H}_1$ and $\mathbf{H}_2$ are matrices that differ only by permutation.

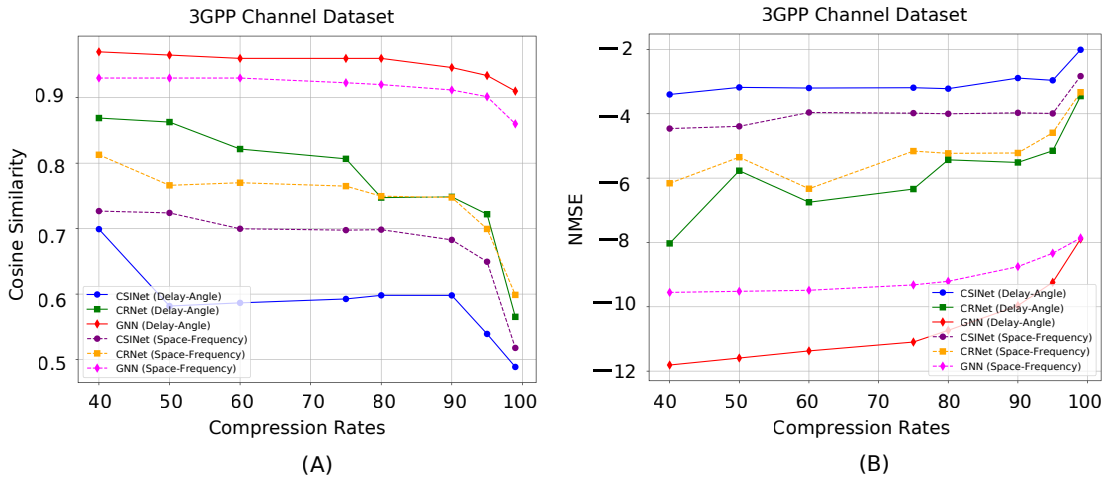

Fig. 3: Performance analysis of CSI compression networks with 3GPP channels generated using Sionna [9].

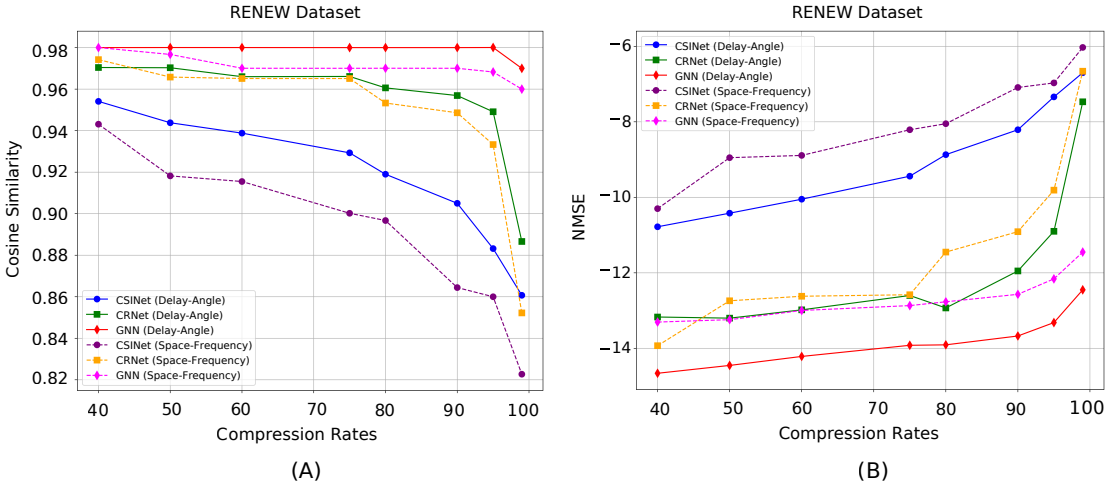

Fig. 4: Performance analysis of CSI compression networks with RENEW dataset [10].

This is illustrated in Figs. 5 (A) and (B). The four graphs that are computed by the GNN (as explained in Section III-B1) for these inputs are plotted in Fig. 5. For this visualization, the learned high dimensional node embeddings are mapped on to 2D space through t-SNE projections. The edge weights denote the attention distribution across the nodes, highlighting the model's ability to learn spatial correlations. Different graphs represent different regions of the CSI matrix. Since the CSI matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ have zeroes in most entries, the graphs corresponding to those regions are sparse. This attention guided partitioning enables the GNN to extract more informative and discriminative representations, especially in regions with dense or overlapping signal components. As expected, the graph representations of $\mathbf{H}_2$ is a permuted representation of that of $\mathbf{H}_1$. In the last row of Fig. 5, the graphs generated by the GNN for the input $\mathbf{H}_1 + \mathbf{H}_2$ are presented. We can see that they are an exact union of the graphs of $\mathbf{H}_1$ and $\mathbf{H}_2$.

This analysis shows how the GCC constructs graphs that capture different spatial regions of the input CSI matrix and how only the significant features identified by the attention mechanism are aggregated.

B. Ablation Study

Next, we conduct ablation studies on the proposed GCC to understand the importance of each layer. We remove one of the eight layers (2 graph layers, 4 convolutional layers, hyperprior layer, LRP layer) one-by-one to study their effect on the reconstruction performance at a high (95%) and a low (about 40%) compression rates. The results are provided in Table I. These results indicate that the graph layers and convolutional layers 3 & 4 are critical for preserving reconstruction quality. It is surprising to observe that the LRP and hyperprior layers, that are responsible for adaptive quantization, do not affect the performance significantly. This study helps us to infer the significance of each layer and identify suitable layers for pruning to reduce computational complexity.

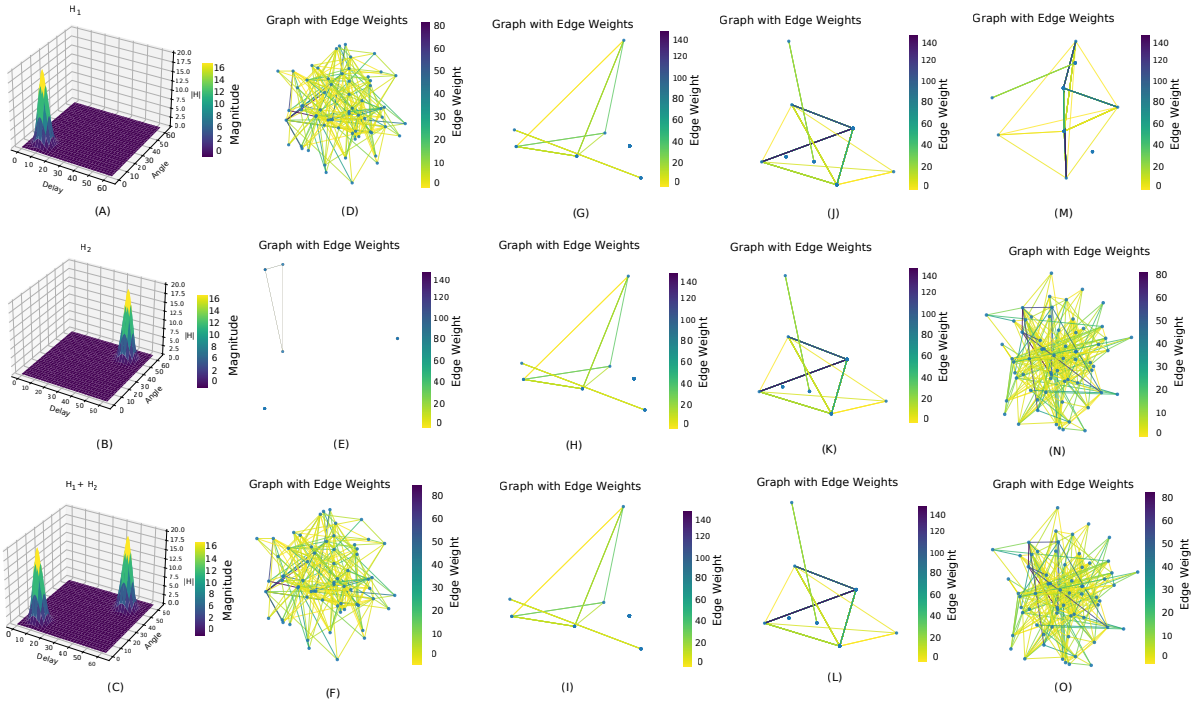

Fig. 5: Visualization of the GNN attention mechanism on different CSI inputs. Rows one to three show the four learned graphs generated by the GCC for input CSI matrices $\mathbf{H}_1$, $\mathbf{H}_2$ and $\mathbf{H}_1 + \mathbf{H}_2$, respectively. The edge color denotes the attention weight, with darker color denoting higher spatial correlation.

Layer Removed	**High (95%)**		**Moderate (35–45%)**	
	Cosine	**NMSE**	**Cosine**	**NMSE**
Graph Block 1	0.96	-10.31	0.97	-12.29
Graph Block 2	0.96	-10.43	0.97	-12.57
LRP Block	0.97	-12.43	0.98	-13.61
Hyper Prior Block	0.97	-11.64	0.98	-14.01
Conv1 Block	0.96	-10.83	0.98	-13.21
Conv2 Block	0.96	-11.01	0.98	-13.32
Conv3 Block	0.96	-10.55	0.97	-12.79
Conv4 Block	0.96	-10.52	0.98	-12.65
No Layer Removed	**0.97**	**-12.45**	**0.98**	**-14.66**

TABLE I: Results of ablation study on GCC.

VI. Conclusions

This work presents a GNN based approach for compressing CSI (GCC) in cellular systems. The proposed model captures both local and global correlations in the delay–angle domain using dynamic graph representations and attention mechanisms. Experiments on 3GPP compliant CSI data and real-life CSI show that compression in the delay angle domain achieves superior reconstruction performance, with up to 30x improvement over the state-of-the-art neural architectures for CSI compression in terms of cosine similarity and NMSE.

References

[1] Chao-Kai Wen, Wan-Ting Shih, and Shi Jin, "Deep learning for massive mimo CSI feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[2] Tianqi Wang, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 416–419, 2018.

[3] Zhilin Lu, Jintao Wang, and Jian Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," in *ICC 2020-2020 IEEE international conference on communications (ICC)*. IEEE, 2020, pp. 1–6.

[4] Sharan Mourya, Pavan Reddy, SaiDhiraj Amuru, and Kiran Kumar Kuchi, "Spectral temporal graph neural network for massive mimo csi prediction," *IEEE Wireless Communications Letters*, vol. 13, no. 5, pp. 1399–1403, 2024.

[5] Ravi Kumar and Manivasakan Rathinam, "An ERAN-based dynamic graph neural network for csi prediction in massive mimo systems," *IEEE Wireless Communications Letters*, pp. 1–1, 2025.

[6] Akbar M Sayeed, "Deconstructing multiantenna fading channels," *IEEE Transactions on Signal processing*, vol. 50, no. 10, pp. 2563–2579, 2002.

[7] Johannes Ballé, Valero Laparra, and Eero P Simoncelli, "Density modeling of images using a generalized normalization transformation," *arXiv preprint arXiv:1511.06281*, 2015.

[8] Gabriele Spadaro, Alberto Presta, Enzo Tartaglione, Jhony H Giraldo, Marco Grangetto, and Attilio Fiandrotti, "Gabic: Graph-based attention block for image compression," in *2024 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2024, pp. 1802–1808.

[9] Jakob Hoydis, Sebastian Cammerer, Fayçal Ait Aoudia, Merlin Nimier-David, Lorenzo Maggi, Guillermo Marcus, Avinash Vem, and Alexander Keller, "Sionna," 2022, https://nvlabs.github.io/sionna/.

[10] Xu Du and Ashutosh Sabharwal, "Massive mimo channels with inter-user angle correlation: Open-access dataset, analysis and measurement-based validation," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2021.