

GNN-Aided Online Calibration of Phase Shifters

Idan Roth and Lutz Lampe

The University of British Columbia, Vancouver, BC, Canada, {idanroth,Lampe}@ece.ubc.ca

Abstract—While 5G has seen only limited use of large-scale antenna arrays and millimeter-wave bands, 6G is expected to support their broader adoption. A key enabler of the required precise beam steering and spatial multiplexing in these systems is hybrid beamforming with phase-shifter networks (PSNs). However, phase deviations arising from manufacturing tolerances, hardware aging, and temperature fluctuations can severely degrade beamforming performance, especially for the precise beamsteering required in 6G systems, making regular PSN calibration essential. To overcome the limitations of conventional model-based iterative algorithms, we introduce the first machine learning-driven online calibration method for PSNs in hybrid beamforming systems. Our approach leverages graph neural networks (GNNs) to learn the mapping from received pilot signals to PSN phase deviation estimates. Simulation results show that the proposed GNN architecture outperforms traditional optimization-based calibration methods and maintains robustness in dynamically changing environments.

I. INTRODUCTION

Multiple-input multiple-output (MIMO) communications have undergone a dramatic transformation, from their initial deployment in 4G, to beamforming-centric designs in 5G, and now toward native MIMO support and emphasis on MIMO in frequency-division duplex (FDD) bands for 6G [1]. The advent of millimeter-wave (mmWave) technology in 5G has driven extensive research into massive MIMO systems with high-gain directional beamforming. Although mmWave uptake in 5G has lagged behind early projections, the continuing demand for spectrum above 6 GHz, coupled with rapid advances in mmWave and multi-antenna technologies, indicates that 6G will make more extensive use of mmWave bands [1], [2]. Hybrid beamforming (HBF) remains the cornerstone architecture for realizing the benefits of massive MIMO [3]. While 6G is expected to introduce more advanced HBF architectures and integrate machine-learning techniques, these systems will continue to rely on phase shifter networks (PSNs) to steer and shape radiation patterns. Given the increasing demand for precise beam alignment and the high sensitivity of beamforming performance to phase deviations [1], accurate PSN calibration [4] will be essential for effective HBF operation in 6G.

Conventional PSN calibration methods, referred to as *offline* calibration, are conducted in an anechoic chamber prior to an array's deployment. They aim to compensate for static errors primarily arising from manufacturing failures and hardware impairments [5], [6]. However, offline calibration has become overly time-consuming, costly, and increasingly impractical for large antenna systems used in mmWave communications [5], [7]. A more suitable approach for PSN calibration is *on-line* calibration, which utilizes over-the-air pilot measurements

obtained in-situ during the operation of an antenna array. By eliminating the need for a separate, environmentally controlled calibration procedure, this approach not only reduces time and cost but also enables periodic calibration which further address dynamic phase errors caused by transistor aging, electrical stress, and temperature fluctuations. Consequently, it ensures optimal system performance is maintained [6], [7]. State-of-the-art online calibration methods for hybrid beamforming systems [8]–[10] rely on optimization techniques to solve highly non-convex programs. This results in computationally demanding algorithms, with high complexity especially for large antenna systems.

Deep learning-based approaches offer a powerful alternative for solving complex wireless communication problems. Already integrated into 5G systems, they are poised to play an even more prominent role in 6G by tackling high-dimensional challenges that are intractable for classical methods [1], [2]. In this context, graph neural network (GNN) architectures have been shown to outperform traditional optimization approaches, by effectively exploiting graph representations of wireless systems. Such architectures exhibit robust generalization capabilities across unseen environmental parameters and inherently integrate domain-specific prior knowledge into the NN's architecture, thereby enhancing efficiency, performance and scalability. These advantages are exemplified in [11], where GNNs are applied to power control and hybrid precoding in wireless systems. Furthermore, by modeling a reconfigurable intelligent surface and users as interconnected graph nodes, [12] leverages a GNN to directly map received pilots to phase shifts and beamforming matrices, bypassing explicit channel estimation. For beamforming design in a multi-user multiple-input single-output (MISO) downlink system, [13] employs a bipartite GNN over a weighted graph with antennas and users as disjoint node types. [14] introduces a framework with parallel GNNs, each satisfying desired permutation properties, that map uplink pilots directly to analog and digital precoders. This design avoids explicit channel prediction and achieves high performance, low overhead, and strong generalization.

Building on recent advancements in GNN applications, we summarize our main contributions and key findings as follows:

- We introduce a GNN architecture for online calibration of PSNs, marking the first data-driven framework dedicated for this task. Our design employ a heterogeneous bipartite graph that represent antenna elements and users as nodes, thereby embedding domain-specific prior knowledge.
- The proposed GNN design directly maps received pilots and applied PSN phase settings to estimated phase deviations, effectively bypassing the need for explicit channel estimation. In addition, it facilitates independent, parallel estimation of PSN deviations across RF chains.

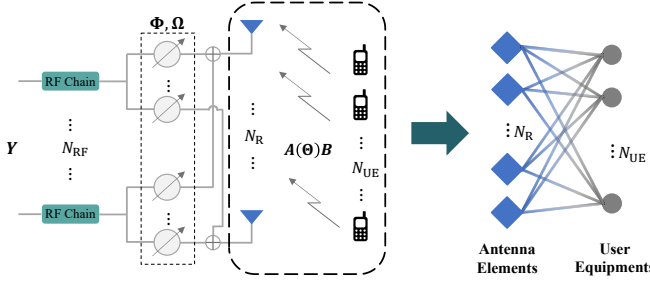


Fig. 1. Left: Multi-user system with hybrid beamforming at the BS. Right: Representation through a heterogeneous bipartite graph.

- Scalability and robustness are further enhanced by sharing layer-wise model parameters among nodes of the same type and exploiting the inherent permutation equivariance (PE) and invariance (PI) properties of our design, features that are challenging to uphold using conventional NNs. This allows our model to adapt to diverse system configurations without the need for retraining.
- Performance evaluations for mmWave multi-user systems and comparisons with the state-of-the-art method [8] confirm the effectiveness of the proposed GNN.

II. SYSTEM MODEL

Although our GNN-aided PSN calibration approach is agnostic to the operating frequency band, we adopt the system model from the closely related work [8], which considers a mmWave multiuser scenario in which single-antenna users transmit uplink pilot signals to calibrate a uniform linear array (ULA) at the base station (BS). As shown on the left side of Fig. 1, we consider N_{UE} users and a BS equipped with N_{R} antenna elements and N_{RF} RF chains. For simplicity, we assume that a fully-connected $N_{\text{RF}} \times N_{\text{R}}$ PSN is deployed to enable hybrid combining. Nevertheless, our proposed method is directly applicable to HBF systems with a partially-connected architecture.

1) *Uplink Pilot Transmission*: The PSN calibration procedure involves a pilot transmission stage, in which each user transmits a training sequence on a dedicated resource to prevent signal interference [8]. During pilot transmission, the BS configures the PSN with Q randomly selected training combiners $\Phi_q \in \mathbb{C}^{N_{\text{RF}} \times N_{\text{R}}}$, $q = 1, \dots, Q$, with unit-modulus elements, each applied to the pilot signals from all users. Let $x_k \in \mathbb{C}$ be a pilot symbol transmitted from the k -th user, where $k = 1, \dots, N_{\text{UE}}$. Then, the sampled received signal $\mathbf{y}_{k,q} \in \mathbb{C}^{N_{\text{RF}} \times 1}$ at the BS combiner output can be written as

$$\mathbf{y}_{k,q} = \Phi_q \mathbf{h}_k x_k + \mathbf{n}_k, \quad (1)$$

where the BS selects the q -th training combiner Φ_q , $\mathbf{h}_k \in \mathbb{C}^{N_{\text{R}} \times 1}$ is the complex channel gain vector, and $\mathbf{n}_k \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I})$ is additive white Gaussian noise.

2) *Channel Model*: As stipulated in (1), we model the channel between the k -th user and the BS ULA to be a narrow-band, flat-fading channel. For simplicity, following [8], we assume a line-of-sight (LOS) channel represented by $\mathbf{h}_k = \beta_k \mathbf{a}(\theta_k)$, where β_k and $\theta_k \in [-\pi/2, \pi/2]$ are the

complex fading gain and angle of arrival (AoA) for the k -th user, respectively, and $\mathbf{a}(\theta)$ is the array response vector in direction θ at the BS. We make the common assumption that the channel characteristics remain unchanged during the Q measurements. Since the LOS channel scenario reflects the most likely condition for active in-situ calibration by service providers, we omit the straightforward extension to non-LOS case considering the space constraints for this paper.

3) *Phase Deviation Model*: We consider B -bit phase shifters with $N_{\text{B}} = 2^B$ discrete phase states and a resolution step size of $\Delta = \frac{2\pi}{N_{\text{B}}}$. Therefore, the nominal phases of the elements of combiner matrices Φ_q are from the set $\mathcal{S} = \{-\pi + \Delta, \dots, -\pi + (N_{\text{B}} - 1)\Delta, \pi\}$. The deviations from the nominal phases are collected in $\Omega \in \mathbb{R}^{N_{\text{RF}} \times N_{\text{R}}}$. The entry $[\Omega]_{(j,i)} = \omega_{j,i}$, captures the error for the phase shift applied between the j -th RF chain and i -th antenna element. Common stochastic models assume that these errors are independent and identically distributed (i.i.d.) according to a uniform distribution [8] $\omega_{j,i} \sim \mathcal{U}[-\delta, \delta]$, or a Gaussian distribution [4] $\omega_{j,i} \sim \mathcal{N}(0, \sigma^2)$. Overall, the actual phases of the PSN are governed by $\Phi_q \odot \mathbf{V}(\Omega)$, where the entries of $\mathbf{V}(\Omega) \in \mathbb{C}^{N_{\text{RF}} \times N_{\text{R}}}$ are $v_{j,i} = e^{j\omega_{j,i}}$ for $j = 1, \dots, N_{\text{RF}}$, $i = 1, \dots, N_{\text{R}}$. Accordingly, we modify (1) as

$$\mathbf{y}_{k,q} = (\Phi_q \odot \mathbf{V}(\Omega)) \mathbf{h}_k x_k + \mathbf{n}_k, \quad (2)$$

where $\odot$ denote the Hadamard product. Using (2), and assuming without loss of generality that $x_k = 1$ for all pilots, we can write the matrix $\mathbf{Y}_q = [\mathbf{y}_{1,q}, \dots, \mathbf{y}_{N_{\text{UE}},q}] \in \mathbb{C}^{N_{\text{RF}} \times N_{\text{UE}}}$ of received pilots from all users at the q -th measurement as

$$\mathbf{Y}_q = (\Phi_q \odot \mathbf{V}(\Omega)) \mathbf{A}(\Theta) \mathbf{B} + \mathbf{N}, \quad (3)$$

where $\mathbf{A}(\Theta) = [\mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_{N_{\text{UE}}})]$, $\mathbf{B} = \text{diag}([\beta_1, \dots, \beta_{N_{\text{UE}}}]^T)$, and $\mathbf{N} = [\mathbf{n}_1, \dots, \mathbf{n}_{N_{\text{UE}}}]$. $\text{diag}(\cdot)$ denotes the operator that maps a vector to a diagonal matrix.

We note that a more comprehensive error model, such as the one considered in [9], would make phase deviations depend on the nominal phase states. For ease of exposition, we retain the simpler phase deviation model. However, the solution we present in the next section can readily be extended to accommodate this more complex model.

III. GNN-BASED CALIBRATION

The online calibration task of the PSN involves estimating the $N_{\text{R}} N_{\text{RF}}$ elements of Ω to compensate for non-ideal system components. Traditional optimization approaches rely on (3) and a mean-squared error (MSE) formulation as the objective function. This leads to iteratively solving a highly non-convex optimization problem of joint channel and phase deviation estimation [8]. In the following, we first address the phase ambiguities present in traditional optimization-based online calibration, and then propose an alternative approach based on learning from data using GNNs.

A. Ambiguities in Online Calibration

In the case of joint calibration and channel estimation as in [8], PSN calibration is insensitive to an absolute

signal delay and the absolute orientation of the antenna array. Considering (3), delay ambiguity can be seen from the fact that infinitely many equivalent solutions exist in the form $\{(\mathbf{e}^{j\tau}\mathbf{V}(\boldsymbol{\Omega}), \mathbf{e}^{-j\tau}\mathbf{B}), \tau \in [-\pi, \pi)\}$. Similarly, by defining $\mathbf{R}(\rho) = \text{diag}([1, \mathbf{e}^{j\rho}, \dots, \mathbf{e}^{j(N_R-1)\rho}]^T)$, all solutions in $\{(\mathbf{V}(\boldsymbol{\Omega})\mathbf{R}(\rho), (\mathbf{R}(\rho))^{-1}\mathbf{A}(\boldsymbol{\Theta}), \rho \in [-\pi, \pi)\}$ are indistinguishable, indicating an angle ambiguity.

In [8], where the MSE between $\mathbf{Y}_q$ and the reconstructed pilots is applied as objective function, ambiguities are managed by subtracting the first components $\hat{\omega}_{1,1}$ and $\hat{\theta}_{1,1}$ from all phase deviation and channel AoA estimations, respectively, when comparing the estimates with the correspondingly adjusted true vectors. However, this is not applicable for our learned solution, which does not estimate the channel directly. Instead, we adopt a supervised approach, which renders the loss function free from such ambiguities.

B. Data-driven Approach

Motivated by [12], our design directly maps received pilot signals, acquired based on the employed combiner settings, to phase deviations, i.e.,

$$\hat{\boldsymbol{\Omega}} = \mathcal{F}(\mathbf{Y}, \boldsymbol{\Phi}|\mathcal{W}), \quad (4)$$

where $\mathcal{F}$ denotes the mapping realized by a GNN with the set $\mathcal{W}$ of learnable parameters, $\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \dots, \mathbf{Y}_Q^T]^T \in \mathbb{C}^{QN_{\text{RF}} \times N_{\text{UE}}}$, and $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1^T, \boldsymbol{\Phi}_2^T, \dots, \boldsymbol{\Phi}_Q^T]^T \in \mathbb{C}^{QN_{\text{RF}} \times N_R}$. We note that the mapping (4) bypasses the need for explicit channel estimation. The GNN parameters are trained using the loss function

$$\text{MSE} = \frac{1}{N_R N_{\text{RF}}} \mathbb{E} \left[\left\| \hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega} \right\|_F^2 \right]. \quad (5)$$

This supervised learning approach could be based on, for example, synthesized data, a choice justified by the GNN's strong generalization capabilities as we will discuss in Section IV-C. However, we note that in-situ fine-tuning of the GNN model under distribution shifts cannot rely on supervised learning. To address this challenge, an unsupervised fine-tuning solution will be discussed in Section IV-C.

C. GNN Architecture

As opposed to conventional NNs, GNNs naturally retain several beneficial features. They intrinsically capture the interactions between users and antenna elements by embedding their relational structure as prior knowledge. Furthermore, as introduced later in this section, an important desired aspect of the proposed GNN is its design to exhibit PI with respect to the user nodes and PE with respect to the antenna nodes. In other words, the learned model is constrained so that the estimated phase deviations $\hat{\boldsymbol{\Omega}}$ remain unaffected by users' ordering and are permuted accordingly with any rearrangement of the antenna nodes. This can be expressed as: $\hat{\boldsymbol{\Omega}}\boldsymbol{\Pi}_R = \mathcal{F}(\mathbf{Y}\boldsymbol{\Pi}_{\text{UE}}, \boldsymbol{\Phi}\boldsymbol{\Pi}_R|\mathcal{W})$, where $\boldsymbol{\Pi}_R$ and $\boldsymbol{\Pi}_{\text{UE}}$ are permutation matrices that permutes the antenna elements and users indices, respectively. These matrices are individually applied to each sub-matrix $\mathbf{Y}_q$ and $\boldsymbol{\Phi}_q$. While these properties are naturally

embedded in GNNs, they are difficult to uphold using feed-forward NNs. Finally, as we will see, further sharing the model parameters across user nodes at each layer enables our GNN to adapt to fluctuations in user count without the need for retraining, a flexibility that feed-forward NNs lack, as they necessitate input layer adjustments and thus retraining.

To address the PSN online calibration task formulated in (4), we propose a heterogeneous GNN architecture. Our design leverages the spatial-based graph convolutional network (GCN) framework of message passing neural networks (MPNN) [15]. In MPNN, each node in the graph is represented by a hidden state vector. For brevity, our complete architecture is referred to as a GNN.

1) *Graph Representation*: For the GNN model to incorporate structural knowledge about the problem formulation, we represent our system as a heterogeneous undirected fully-connected bipartite graph as shown on the right side of Fig. 1. The graph comprises N_R antenna nodes and N_{UE} user nodes. We associate a hidden state vector with each node: $\mathbf{d}_i \in \mathbb{R}^{D_h}$, $i = 1, 2, \dots, N_R$, and $\mathbf{b}_k \in \mathbb{R}^{D_h}$, $k = 1, 2, \dots, N_{\text{UE}}$ for the antenna nodes and user nodes, respectively, with the hyper-parameter D_h denoting their dimension. These vectors encode the essential node-specific information. To further capture and manage the unique characteristics and relationships among system components, we incorporate NN layer-wise parameter sharing across nodes of the same type.

During operation, the GNN iteratively updates the hidden states layer by layer, enabling the extraction of richer knowledge. This updating process follows the message-passing paradigm, where each node modifies its own hidden state based on the aggregated information of hidden states from its neighboring nodes at the preceding layer. After applying multiple layers, the hidden state vectors of the antenna nodes are used for estimating the PSN phase deviation.

In the proposed graph representation, we intentionally omit the possible relation between pilot signals received at different RF chains for the estimation task. The GNN is therefore trained for a single RF chain and subsequently replicated across all RF chains to estimate the phase deviations respective to each chain's phase shifters. This design choice not only renders our architecture independent of the number of RF chains and thus scalable with respect to this parameter, but also simplifies the training phase and enables efficient parallel calibration during inference.

2) *GNN Design*: The details of our proposed GNN architecture are shown in Fig. 2. Since we consider one RF chain, our estimation goal becomes the vector $\boldsymbol{\Omega} \in \mathbb{R}^{(N_{\text{RF}}=1) \times N_R}$. Furthermore, the inputs to the GNN are the combiner phase settings for each of the N_R phase shifters across Q measurements, which we denote by $\phi_i \in \mathbb{C}^{Q \times 1}$, $i = 1, \dots, N_R$, and the pilots received over Q measurements from the N_{UE} users, which we denote by $\mathbf{y}_k \in \mathbb{C}^{Q \times 1}$, $k = 1, \dots, N_{\text{UE}}$. The GNN consists of an initialization module, followed by T MPNN update layers, each comprising aggregation and combination stages, and a readout function as a final module to estimate the PSN phase deviations.

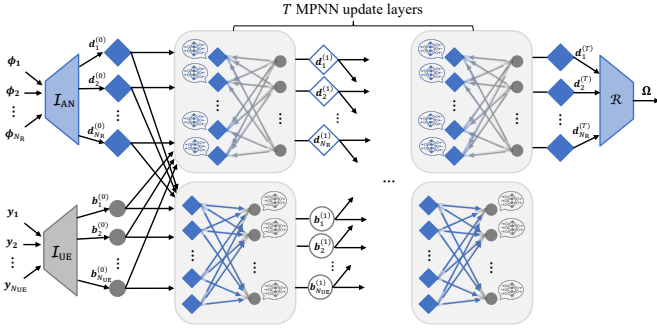


Fig. 2. The proposed GNN architecture, consisting of an initialization module, T update layers, and a final readout function.

a) Initialization: The initialization module assigns an initial hidden state to each node for the subsequent iterative hidden state update process. It employs input embedding to increase the dimensions of the node features. Type-specific fully-connected NNs are utilized to generate initial hidden states for each node type. The initial hidden states of the user nodes $\mathbf{b}_k^{(0)}$, $k = 1, \dots, N_{\text{UE}}$, are generated based on the received pilot signals as input:

$$\mathbf{b}_k^{(0)} = \mathcal{I}_{\text{UE}}(\text{Re}\{\mathbf{y}_k\}, \text{Im}\{\mathbf{y}_k\}), \quad (6)$$

where $(\cdot, \cdot)$ denotes concatenation, and $\text{Re}(\cdot)$, and $\text{Im}(\cdot)$ extract the element-wise real and imaginary parts, respectively. The hidden states of the antenna nodes $\mathbf{d}_i^{(0)}$, $i = 1, \dots, N_{\text{R}}$, are initialized based on the phase shifters' settings as input:

$$\mathbf{d}_i^{(0)} = \mathcal{I}_{\text{AN}}(\text{Re}\{\phi_i\}, \text{Im}\{\phi_i\}). \quad (7)$$

Both $\mathcal{I}_{\text{UE}}$ and $\mathcal{I}_{\text{AN}}$ are implemented using a two-layer fully-connected NN, with each layer comprising a linear layer followed by batch-normalization (BN), and a leaky rectified linear unit (LeakyReLU) activation function. The users' hidden states vectors $\mathbf{b}_1^{(0)}, \dots, \mathbf{b}_{N_{\text{UE}}}^{(0)}$ now contain features related to the channels and the PSN deviation, while the antennas hidden states vectors $\mathbf{d}_1^{(0)}, \dots, \mathbf{d}_{N_{\text{R}}}^{(0)}$ hold information about the PSN's desired assigned phases.

b) MPNN Update Layers: The update of the nodes' hidden states in our GNN is based on the message passing paradigm, where each update layer consists of *aggregation* and *combination* stages [11]. Accordingly, the update of the user nodes' hidden states can be written as

$$\mathbf{a}_k^{(t)} = \mathcal{P}_{i \in \mathcal{N}_{\text{UE}}(k)}(\mu_{i,k}^{(t)}) = \mathcal{P}_{i \in \mathcal{N}_{\text{UE}}(k)}(\mathcal{M}_{\text{UE}}^{(t)}(\mathbf{d}_i^{(t-1)})), \quad (8a)$$

$$\mathbf{b}_k^{(t)} = \mathcal{U}_{\text{UE}}^{(t)}((\mathcal{D}_{\text{UE}}^{(t)}(\mathbf{b}_k^{(t-1)}), \mathbf{a}_k^{(t)})), \quad (8b)$$

and analogously for the update of the antenna nodes' hidden states, we write

$$\mathbf{c}_i^{(t)} = \mathcal{P}_{k \in \mathcal{N}_{\text{AN}}(i)}(\nu_{k,i}^{(t)}) = \mathcal{P}_{k \in \mathcal{N}_{\text{AN}}(i)}(\mathcal{M}_{\text{AN}}^{(t)}(\mathbf{b}_k^{(t-1)})), \quad (9a)$$

$$\mathbf{d}_i^{(t)} = \mathcal{U}_{\text{AN}}^{(t)}((\mathcal{D}_{\text{AN}}^{(t)}(\mathbf{d}_i^{(t-1)}), \mathbf{c}_i^{(t)})). \quad (9b)$$

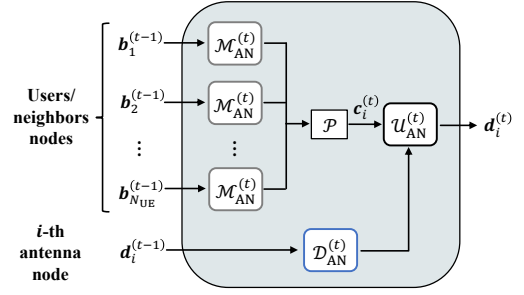


Fig. 3. Block diagram for hidden state update for the i -th antenna node.

In (8a) and (9a), $\mu_{i,k}^{(t)}$ and $\nu_{k,i}^{(t)}$ represent the messages computed from the i -th antenna node to the k -th user node and vice versa at layer t . The sets $\mathcal{N}_{\text{UE}}(k)$ and $\mathcal{N}_{\text{AN}}(i)$ denote the neighboring nodes of the k -th user and i -th antenna, respectively, and $\mathcal{P}$ is a pooling function that operates over these sets to aggregate incoming messages. The message generation functions, $\mathcal{M}_{\text{UE}}^{(t)}$ and $\mathcal{M}_{\text{AN}}^{(t)}$, produce the individual messages, while the combination functions, $\mathcal{U}_{\text{UE}}^{(t)}$ and $\mathcal{U}_{\text{AN}}^{(t)}$, integrate the aggregated messages with each node's previous hidden state. Additionally, the functions $\mathcal{D}_{\text{UE}}^{(t)}$ and $\mathcal{D}_{\text{AN}}^{(t)}$ modify the hidden state before combination. These functions are node-specific, corresponding to either users (UE) or antennas (AN), and are implemented using single-layer fully connected NNs, each consisting of a linear layer followed by BN, a LeakyReLU activation function, and a dropout layer for regularization.

Note that the update stage for the nodes can be executed in parallel. Furthermore, the proposed GNN framework updates the hidden states at each layer using NNs with weights shared across nodes of the same type. In particular, using the same weights for all user nodes facilitates generalization across varying number of users without requiring retraining. Indeed, the learned aggregation and combination operations, as detailed in (9a) and (9b), can accommodate changes in user count since such changes only alter the size of the input graph. A block diagram for the update of antenna nodes as described in (9) is shown in Fig. 3.

c) Final Module: After T update steps, the hidden state vectors of the antenna nodes, $\mathbf{d}_i^{(T)}$, $i = 1, \dots, N_{\text{R}}$, produced by the GNN in the final iteration, are fed into a readout function $\mathcal{R}$ to estimate the PSN phase deviations. For this purpose, a single-layer fully-connected NN is utilized:

$$[\hat{\Omega}]_{(1,i)} = \mathcal{R}(\mathbf{d}_i^{(T)}). \quad (10)$$

A sigmoid activation function is included to map the outputs onto the range of $[-\kappa, \kappa]$ via $2\kappa \text{sigm}(\cdot) - \kappa$, with $\kappa = \delta$ when $\omega_{j,i} \sim \mathcal{U}[-\delta, \delta]$, and $\kappa = 3\sigma$ when $\omega_{j,i} \sim \mathcal{N}(0, \sigma^2)$.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our GNN-based online calibration method. First, we introduce the system parameters and the specifics of GNN training and testing, followed by the presentation of our quantitative results.

A. Simulation Setup

1) *Antenna Array and PSN*: We consider a BS equipped with an N_R -element ULA, with element spacing $d = \lambda/2$. As the proposed calibration operates independently for each RF chain, without loss of generality, we assume $N_{RF} = 1$. Similar to [8], we analyze scenarios with $N_{UE} = 4$, and $N_R = 16, 32$. We note that this would be the numbers of antennas per RF chain in a partially connected hybrid beamforming system and thus represent meaningful massive MIMO scenarios. The ULA employs $(B = 5)$ -bit phase shifters, commonly available commercially, and we set the number of the randomly selected training combiners to $Q = 16$ for $N_R = 16$, and $Q = 28$ for $N_R = 32$. The PSN phase deviations follow an i.i.d. Gaussian distribution $\mathcal{N}(0, \sigma^2)$, with $\sigma \approx 5.8^\circ$, representing the standard deviation of a uniform distribution over $[-10^\circ, 10^\circ]$.

2) *Channel*: The AoAs for all users are uniformly distributed as $\mathcal{U}[-\frac{\pi}{2}, \frac{\pi}{2}]$ [8]. Unless stated otherwise, we follow [8] and set the complex fading gains to be unit modulus with random phases from $\mathcal{U}[-\pi, \pi]$.

3) *Performance and SNR*: For performance evaluation, we measure the empirical Monte-Carlo approximation of the root-MSE (RMSE) per PSN element, i.e., $\text{RMSE} = \sqrt{\text{MSE}}$ with the MSE from (5). To assess the estimator performance under a given noise condition, we define the signal-to-noise ratio (SNR) as the ratio of the received signal power, averaged over PSN phase deviations and channel realizations, to the noise power.

B. GNN Training and Testing

The GNN was implemented using the Deep Graph Library (DGL) with PyTorch. It was trained over 200 epochs, utilizing 387,072 samples per epoch, with validation on additional 34,816 samples conducted every three epochs. Hyperparameter were tuned via random search, uniformly selecting batch sizes, learning rates, and weight decay values from the sets $\{512, 1024\}$, $\{0.5, 1, 5, 10\} \times 10^{-4}$ and $\{0, 0.1, 1, 10\} \times 10^{-7}$, respectively. Dropout values were uniformly distributed across the range $[0, 0.5]$, and the hidden state size, D_h , within $[128, 1400]$ for $N_R = 16$ antennas, and between $[256, 2300]$ for $N_R = 32$ antennas. Our GNN architecture utilized $T = 2$ update layers, and employ the element-wise mean as the pooling function for (8a) and (9a) based on its empirical performance. For optimization, we adopted the Adam optimizer paired with PyTorch ReduceLROnPlateau learning rate scheduler, which reduced the learning rate by a factor of 0.5 after five consecutive epochs without validation loss improvement. Training was terminated if no validation improvement was observed over 13 consecutive validation epochs. The empirical RMSE values reported in our performance evaluation were derived from a testing dataset consisting of 17,408 samples.

C. Performance Results

1) *Comparison With Benchmark*: We consider the state-of-the-art optimization-based method developed in [8] as the benchmark. Since their calibration method achieves the Cramér-Rao bound (CRB) as shown in [8, Fig. 3], we directly

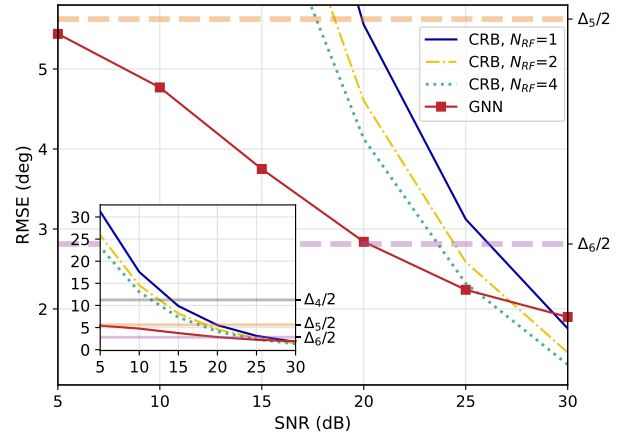


Fig. 4. Performance of the proposed GNN vs. SNR with $N_R = 32$. CRB curves as benchmarks, and $\Delta_B/2 := \pi/N_B$ values, corresponding to $B = 4, 5, 6$ bits, as target values for phase shifter error correction.

compare against the CRB performance predictions in the following. Fig. 4 shows the RMSE in degrees as a function of SNR for $N_R = 32$ antennas. The figure includes the CRBs for $N_{RF} = 1, 2, 4$ and the calibration curve for the GNN. Our first observation is the significant gap between the GNN curve and CRB values as the SNR decreases. This is due to the fact that the learning-based method is able to incorporate statistical information about phase deviations through the use of the supervised MSE loss (5). Such a seamless integration of prior information is not possible in the traditional optimization methods, which treat the unknown parameters as deterministic. Consequently, their performance is confined by the CRB. Second, although our method performs slightly above the CRB at SNRs beyond 26 dB, the accuracy achieved at lower SNRs is more practically significant since such conditions are common in online calibration. We note that RMSE values around the range of $2^\circ - 5^\circ$ are meaningful calibration accuracies considering typical 4 to 6 bits phase shifters (i.e., within $\Delta/2 = \frac{\pi}{N_B}$, which are also shown in the figure). Hence, the performance gains displayed in Fig. 4 are substantial. Third, the CRB curves for $N_{RF} > 1$ indicate that joint estimation across multiple RF chains is beneficial for calibration. This is because calibration includes the explicit (as in [8]) or implicit (as in our GNN) task of channel estimation, which is common for all RF chains. However, the corresponding gains are small. This supports our design choice to treat each RF chain separately, resulting in a smaller, more efficient GNN model that is independent of the number of RF chains and can be duplicated to run in parallel for each chain.

2) *Complexity Comparison*: Besides the performance advantage, the GNN-based approach also enjoys a lower computational complexity than the optimization-based method. To make this more concrete, we consider the number of real-valued multiplications required for inference. Consulting [8], we obtain that the computational complexity of their algorithm is $\mathcal{O}(CN_{RF}[C_1N_{UE}(QN_R \log(N_R) + Q^2N_RN_{RF}) + C_2N_R^2])$, where C, C_1 and C_2 are the number of iterations for the outer loop, and the two inner loops, respectively. For the

GNN, considering that all nodes participate during initialization and updating phases, while only antenna element nodes are involved in the readout phase, the overall computational complexity per RF chain is $\mathcal{O}((N_R + N_{UE})(D_h Q + T D_h^2))$. Different from the traditional method, the GNN-based approach scales linearly in each of N_R , N_{RF} and Q , offering a clear complexity advantage as system dimensions increase.

3) *Robustness*: Next, we assess the generalization capabilities of the trained GNN, i.e., its ability to perform under system conditions that differ from those encountered during training. For this analysis, we consider $N_R = 16$, and train the GNN model under evaluation using the optimized hyperparameters at an SNR of 15 dB.

a) *Channel Model*: We first examine the case of changes in the channel model. For this we dispense with the unit-modulus assumption for path gains but generate their amplitudes uniformly distributed within $[0.6, 1.0]$, i.e., accounting for gain variations across paths. This resulted in a negligible increase in RMSE from 3.94° in the matched case to 4.14° in the mismatched case.

b) *Number of Users*: Next, we vary the number of users N_{UE} between training and testing. Fig. 5 shows the RMSE as a function of N_{UE}^{test} when (i) $N_{UE}^{\text{test}} = N_{UE}^{\text{train}}$ and (ii) training is fixed with $N_{UE}^{\text{train}} = 4$ and N_{UE}^{test} varies. We observe that the proposed GNN exhibits strong generalization capabilities by closely following the benchmark, experiencing only a subtle performance loss of less than 0.11° . This highlights the versatility of our GNN model, which effectively handles dynamically varying input sizes without the need for retraining, through weight sharing and PI design.

c) *Phase Deviation Model*: Finally, we test the GNN's resilience to shifts in the distribution of PSN phase deviations from Gaussian distribution during training to uniform distribution during inference, maintaining the same standard deviation of $\sigma = 5.8^\circ$. The RMSE is only slightly increased from 3.62° for the matched case to 3.73° for the mismatched case.

4) *Fine Tuning*: Overall, these results are encouraging in that the trained GNN can effectively operate across various environments, which supports the supervised learning approach pursued in this work. Nevertheless, the model could be fine-tuned infrequently during operation, particularly in response to significant channel shifts. This can be accomplished with unsupervised training using the MSE loss function as in [8] aided by an existing channel estimator to evaluate the loss. We note that although the channel is jointly estimated across multiple RF chains, the GNN continues to operate independently, preserving its modular design.

V. CONCLUSION

PSN calibration is a critical enabler for large-scale antenna deployments and precise beam steering in 6G networks. State-of-the-art online PSN calibration methods rely on computationally intensive iterative algorithms. In this paper, we introduce the first learning-based online calibration method for phase shifter networks in massive MIMO systems. Our

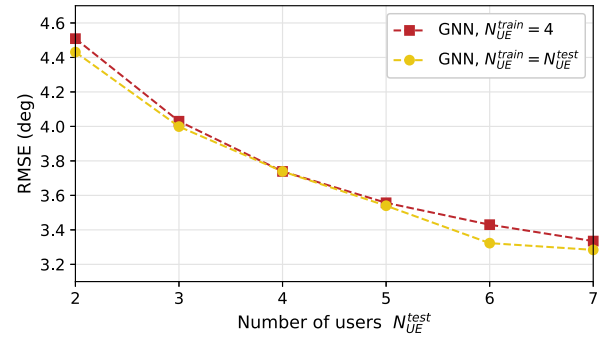


Fig. 5. Robustness performance of the GNN vs. the number of users for $N_R = 16$ and SNR = 15 dB. The GNN was trained with $N_{UE}^{\text{train}} = 4$.

solution employs a computationally efficient and robust heterogeneous GNN architecture that estimates PSN phase deviations directly from received pilot signals, eliminating the need for explicit channel estimation. By training on phase-deviation samples, the network learns real-world deviation patterns and integrates prior knowledge of their distribution. Simulation results demonstrate that our method surpasses the CRB and generalizes effectively to changing environments.

REFERENCES

- [1] J. G. Andrews, T. E. Humphreys, and T. Ji, "6G takes shape," *IEEE BITS Inf. Theory Mag.*, vol. 4, no. 1, pp. 2–24, 2024.
- [2] C.-X. Wang *et al.*, "On the road to 6G: Visions, requirements, key technologies, and testbeds," *IEEE Commun. Surv. Tuts.*, vol. 25, no. 2, pp. 905–974, 2023.
- [3] I. Ahmed *et al.*, "A survey on hybrid beamforming techniques in 5G: Architecture and system model perspectives," *IEEE Commun. Surv. Tuts.*, vol. 20, no. 4, pp. 3060–3097, 2018.
- [4] A. J. van den Biggelaar, U. Johannsen, P. Mattheijssen, and A. B. Smolders, "Improved statistical model on the effect of random errors in the phase and amplitude of element excitations on the array radiation pattern," *IEEE Trans. Antennas Propag.*, no. 5, pp. 2309–2317, 2018.
- [5] Y. Qi *et al.*, "5G over-the-air measurement challenges: Overview," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 6, pp. 1661–1670, 2017.
- [6] M. A. Salas-Natera, R. M. Rodríguez-Osorio, and L. de Haro, "Procedure for measurement, characterization, and calibration of active antenna arrays," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 2, pp. 377–391, 2012.
- [7] T. Moon, J. Gaun, and H. Hassanieh, "Online millimeter wave phased array calibration based on channel estimation," in *VTS*, pp. 1–6, 2019.
- [8] X. Wei, Y. Jiang, Q. Liu, and X. Wang, "Calibration of phase shifter network for hybrid beamforming in mmWave massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 68, pp. 2302–2315, 2020.
- [9] W. Zhang and Y. Jiang, "Over-the-air calibration of phase shifter network for hybrid MIMO systems," *IEEE Trans. Signal Process.*, vol. 70, pp. 3456–3467, 2022.
- [10] M. Wang, J. Chen, J. Tao, and H. Li, "Over-the-air antenna array calibration for mmWave hybrid beamforming systems based on Monte Carlo Markov chain method," *IEEE Trans. Veh. Technol.*, vol. 72, 2022.
- [11] Y. Shen, J. Zhang, S. H. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 3554–3569, 2023.
- [12] T. Jiang, H. V. Cheng, and W. Yu, "Learning to reflect and to beamform for intelligent reflecting surface with implicit channel estimation," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1931–1945, 2021.
- [13] J. Kim, H. Lee, S.-E. Hong, and S.-H. Park, "A bipartite graph neural network approach for scalable beamforming optimization," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 333–347, 2023.
- [14] R. Wang, C. Yang, S. Han, J. Wu, S. Han, and X. Wang, "Learning end-to-end hybrid precoding for multi-user mmWave mobile system with GNNs," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 2, 2024.
- [15] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, 2017.