Towards Achieving Integer and Load-balancing User Association in Wireless Networks with a Reparameterized Attention-based GNN

Qing Lyu
qing.lyu@tufts.edu

Mai Vu
mai.vu@tufts.edu

Department of Electrical and Computer Engineering
Tufts University
Medford, MA 02155

Abstract

Machine learning (ML) is a promising method for user association in dense wireless networks, where each user (UE) must connect to a unique base station (BS) to balance the loads and maximize network capacity. Two key challenges that hinder direct ML use are producing integer-valued associations while still maintaining gradients, and satisfying BS load constraints. Most existing approaches relax the integer association requirement by using *softmax*, leading to suboptimal results. To address this problem, we propose an attention-based Graph Neural Network with Gumbel-Softmax reparameterization for near-integer outputs, together with Sinkhorn-Knopp normalization and loss regularization for load balancing satisfactions. Numerical results show that our method outperforms all existing ML and non-ML solutions, approaches optimal exhaustive search in small networks, and generalizes well to larger and more dynamic networks without retraining.

1 Introduction

Millimeter wave communication for modern 5G/6G wireless networks provides a promising solution to the spectrum shortage and ever-increasing capacity demand. Methods such as beamforming and user association are imperative for achieving high network capacity when operating in a dense network at high frequency bands such as millimeter wave [1].

Two main challenges in determining user association are the unique (or integer) association requirement and the load balancing constraint. Load balancing in user association is critical in dense wireless networks, where uneven UE distribution and dynamic traffic can overload certain BSs while others are underutilized. Traditional methods formulate this problem as a mixed-integer nonlinear program, solved by dual decomposition [1] or heuristics [2], but the results usually scale poorly with network size and require re-optimization for each new channel, incurring high computation costs.

Graph neural networks (GNNs) can facilitate real-time computation and allow robust scaling to larger networks without retraining [3]. However, GNN design for user association also faces the two challenges of integer values and load balancing constraint. The standard *softmax* operation applied at a model's output leads to poor integer approximation and has no load balancing. Furthermore, attention has been used as a mechanism to aggregate information in a GNN. Architectures such as Graph Attention Networks (GATs) [3] capture local attention by adaptively weighting neighbors during message passing. In contrast, in Transformers [4], the attention mechanism is applied globally over all input sequences and is typically implemented using scaled dot-product attention, which can potentially improve representation learning performance.

We propose a dot-product attention-based GNN architecture integrating Gumbel–Softmax (GS) reparameterization for near-integer associations and Sinkhorn–Knopp (SK) normalization together, called AttSKGS-GNN, with loss regularization for enforcing per-BS maximum load. This GNN design preserves end-to-end differentiability and achieves high communication spectral efficiency, while closely achieving integer association and network load balancing.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI and ML for Next-Generation Wireless Communications and Networking (AI4NextG).

2 System model and problem formulation

2.1 System and signal models

We consider a downlink multicell system where M BSs, each equipped N antennas, serve K single-antenna UEs. Denote the user association matrix as $\boldsymbol{A} \in \mathbb{R}^{K\times M}$, where the association vector between BS m and K UEs is represented by column as $\boldsymbol{a}_m$ with integer elements $a_{k,m} \in \{0,1\}$, if $a_{k,m} = 1$, UE k is associated with BS m. Assume unique association where each UE can only connect to one BS, each row of $\boldsymbol{A}$ sums to 1.

Denote the beamforming matrix at BS m as $\mathbf{V}_m \in \mathbb{C}^{N\times K}$ where each column $\mathbf{v}_{m,k}$ is the beamforming vector for UE k. The transmitted signal from BS m can be written as $\mathbf{x}_m = \boldsymbol{a}_m^T \mathbf{V}_m \mathbf{s} = \sum_{k=1}^{K} a_{k,m}\mathbf{v}_{m,k}s_k$, where $\mathbf{s} \in \mathbb{C}^{K\times 1}$ is the transmitted symbols for the K UEs with $E[\mathbf{s}\mathbf{s}^*] = \mathbf{I}$. The received signal at UE k is $y_k = \sum_{m=1}^{M} a_{k,m}\mathbf{h}_{m,k}^T\mathbf{v}_{m,k}s_k + \sum_{m=1}^{M}\sum_{l=1,l\neq k}^{K} a_{l,m}\mathbf{h}_{m,k}^T\mathbf{v}_{m,l}s_l + n_k$, where $\mathbf{h}_{m,k} \in \mathbb{C}^{N\times 1}$ represents the channel vector from BS m to UE k, and n_k is the noise at UE k following $\mathcal{CN}(0,\sigma_k^2)$. The signal-to-interference-noise ratio (SINR) at UE k is

$$\gamma_k = \frac{|\sum_{m=1}^{M} a_{k,m}\mathbf{h}_{m,k}^T\mathbf{v}_{m,k}|^2}{\sum_{l=1,l\neq k}^{K}|\sum_{m=1}^{M} a_{l,m}\mathbf{h}_{m,k}^H\mathbf{v}_{m,l}|^2 + \sigma_k^2} \tag{1}$$

The achievable sum-rate (or sum spectral efficiency) is then $R = \sum_{k=1}^{K}\log_2(1+\gamma_k)$ (bps/Hz).

2.2 Problem Formulation

For a given beamforming design, finding the load-balanced integer association for maximizing the sum-rate is formulated as

$$(\mathcal{P}) \quad \max_{\mathbf{A}} \quad \sum_{k=1}^{K}\log_2(1+\gamma_k) \tag{2}$$

$$\text{s.t.} \quad a_{k,m} \in \{0,1\},\ \sum_m a_{k,m} = 1,\ \sum_k a_{k,m} \leq N \tag{2a}$$

where the first two constraints are integer association and unique association (each UE can only associate with one BS). The last constraint is a load constraint which aims to impose load balancing across all BSs (each BS can only serve up to N UEs simultaneously).

$(\mathcal{P})$ is a non-linear discrete optimization problem, whose global optimal solution requires an exhaustive search with exponential complexity, leading to prohibitively high computational cost for large-scale networks. Here we propose to solve $(\mathcal{P})$ by AttSKGS-GNN model described next.

Problem $(\mathcal{P})$ has multiple constraints which are usually challenging to satisfy in ML because of the unconstrained loss minimization and the need to maintain gradients. We introduce a regularized loss function via incorporating a Lagrangian dual multiplier μ_m for each BS load constraint as

$$L(\mathbf{\Omega}) = -\mathbb{E}_{\mathcal{B}}[R] + \sum_m \mu_m \times \text{ReLU}\left(\sum_k a_{k,m} - (N-\delta)\right) \tag{3}$$

where $\mathbf{\Omega}$ represents all trainable parameters that influence the association outputs $a_{k,m}$ and the resulting achievable sum-rate R. The ReLU penalty term becomes active only when a BS exceeds its UE quota N, ensuring differentiability while imposing penalties only for violations. A margin parameter $\delta \in [0, N)$ is introduced to tighten the quota constraint, alleviating cases of violation. The coefficient μ_m serves as a regularization hyperparameter or Lagrange multiplier that adjusts the strength of the penalty—larger μ_m emphasizes stronger constraint satisfaction and encourages balanced loads across BSs, while smaller μ_m relaxes the requirements to satisfy the constraint to favor higher sum-rate performance. This formulation allows the GNN to learn to produce association outcomes that balance sum-rate optimality and load-balancing feasibility through gradient-based optimization, effectively bridging constrained optimization and deep learning via a differentiable Lagrangian dual framework.

3 Attention-based Reparameterized GNN for User Association

In communication networks, the numbers of UEs and BSs can vary over time, making it essential to design ML models that adapt to changes in network size. Traditional fully-connected neural

networks, with fixed input and output dimensions, require resizing and retraining when the number of UEs or BSs changes, making them inefficient and impractical in dynamic environments. In contrast, GNNs enable learning algorithms that leverage recurring optimization structures across various data and network sizes. By reusing per-edge and per-node functions, GNNs inherently support combinatorial generalization, allowing them to operate seamlessly on graphs of different sizes. This adaptability enables GNNs to generalize well to problems with significantly different numbers of nodes and edges from those they were initially trained [3]. As such, we propose a GNN structure to solve for the user association problem in ($\mathcal{P}$).

3.1 Graph Model and Representation

The wireless network can be modeled as a bipartite graph, where the BSs and UEs are two types of nodes, with no edges between nodes of the same type. Such a graph can be represented as $\mathcal{G} = \{\mathcal{M}, \mathcal{K}, \mathcal{E}\}$, where $\mathcal{M}$ is the set of BS nodes, $\mathcal{K}$ is the set of UE nodes, and $\mathcal{E}$ is the set of edges with $\mathcal{E} = \{(m,k)\}_{m\in\mathcal{M},k\in\mathcal{K}}$.

Let $\mathbf{p} \in \mathbb{R}^M$ and $\mathbf{q} \in \mathbb{R}^K$ denote BS and UE features, respectively. Channel coefficients $\mathbf{H} \in \mathbb{R}^{M\times K\times 2N}$ are represented by concatenating real and imaginary parts. A preprocessing layer maps $(\mathbf{p}, \mathbf{q}, \mathbf{H})$ via three MLPs into initial node and edge embeddings $\mathbf{B}^{(0)} \in \mathbb{R}^{M\times d}$, $\mathbf{C}^{(0)} \in \mathbb{R}^{K\times d}$, and $\mathbf{E}^{(0)} \in \mathbb{R}^{M\times K\times d}$, where d is the embedding size.

3.2 Proposed GNN Architecture

Our proposed AttSKGS-GNN not only updates both nodes and edges, but also uses an attention mechanism assigns adaptive weights to neighbors, highlighting more relevant interactions and improving representation learning. The proposed GNN consists of a preprocessing layer, L update layers, and a postprocessing layer. In each update layer, we perform scaled dot-product attention—using BS embeddings as queries and UE embeddings as keys—to compute attention scores that directly update the edge features. Finally, the postprocessing layer applies Gumbel-Softmax (GS) reparameterization and Sinkhorn-Knopp (SK) normalization to those edge representations, targeting the constraints for unique association and load balancing.

To update the node and edge representations, the l-th updating layer inputs $(\mathbf{B}^{(l-1)}, \mathbf{C}^{(l-1)}, \mathbf{E}^{(l-1)})$, and outputs the updated representations $(\mathbf{B}^{(l)}, \mathbf{C}^{(l)}, \mathbf{E}^{(l)})$ as follows. For the node representation of BS m, the inputs include its representation and the aggregation of all neighboring UEs and edges:

$$\mathbf{b}_m^{(l)} = f_2^{(l)}\left(\mathbf{b}_m^{(l-1)}, \phi_{\text{BS}}^{(l)}(f_1^{(l)}(\mathbf{c}_k^{(l-1)}, \mathbf{e}_{m,k}^{(l-1)}))_{k\in\mathcal{N}_m^{\text{UE}}}\right) \tag{4}$$

where $\mathbf{b}_m^{(l-1)} \triangleq \mathbf{B}_{(m,:)}^{(l-1)}$, $\mathbf{c}_k^{(l-1)} \triangleq \mathbf{C}_{(k,:)}^{(l-1)}$, $\mathbf{e}_{m,k}^{(l-1)} \triangleq \mathbf{E}_{(m,k,:)}^{(l-1)}$ and $\mathcal{N}_m^{\text{UE}}$ denotes the set of neighboring UEs of BS m. f_1 and f_2 are two MLPs, and $\phi_{\text{BS}}^{(l)}$ is an aggregation function.

Similarly, the node representation update for UE k in the l-th updating layer is as follows.

$$\mathbf{c}_k^{(l)} = f_4^{(l)}\left(\mathbf{c}_k^{(l-1)}, \phi_{\text{UE}}^{(l)}(f_3^{(l)}(\mathbf{b}_m^{(l-1)}, \mathbf{e}_{m,k}^{(l-1)}))_{m\in\mathcal{N}_k^{\text{BS}}}\right) \tag{5}$$

where f_3 and f_4 are two new MLPs and $\mathcal{N}_k^{\text{BS}}$ is the set of neighboring BSs of UE k.

3.3 Attention-based Edge Update

The update of edge representation is of special importance in our model since the edges will be converted to learned association matrix at the end. For the bipartite graph, we propose to use an attention factor that measures the importance of all M BSs to each UE. Since the scaled dot-product attention factor is computed via matrix multiplication [4], we replicate the k-th UE representation $\mathbf{c}_k^{(l-1)}$ M times to form a matrix $\mathbf{C}_k^{(l-1)} \in \mathbb{R}^{M\times d}$. This replication is reasonable because, on the UE side, we focus solely on the k-th UE. Then, $\mathbf{C}_k^{(l-1)}$ serves as the query, matrix $\mathbf{B}^{(l-1)}$ composed of M BS representations serves as the key, and edge matrix $\mathbf{E}_k^{(l-1)} \in \mathbb{R}^{M\times d}$ containing the edge representations from all M BSs to UE k, acts as the value. Accordingly, the attention matrix from all BSs to UE k is computed as

$$\widetilde{\mathbf{G}}_k^{(l)} = \text{softmax}\left(d^{-\frac{1}{2}}\mathbf{C}_k^{(l-1)}\mathbf{B}^{(l-1)^T}\right)\mathbf{E}_k^{(l-1)} \tag{6}$$

where the scaling factor $d^{-\frac{1}{2}}$ is introduced to prevent extremely small gradients in the *softmax* function when the dimensionality d is large [4].

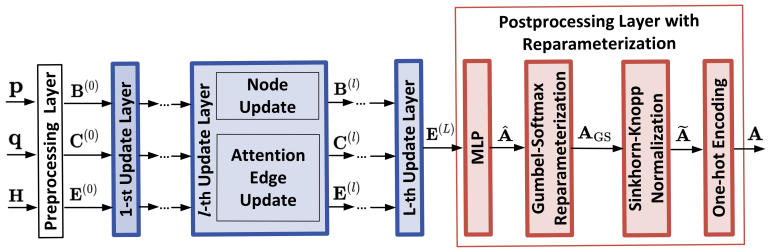


Figure 1: The architecture of the proposed attention-based GNN, where all update layers (in gray) share the same structure and the l-th layer is shown in detail. The post-processing layer (in orange) is shown in detail with the reparameterization steps, where all $\mathbf{A}_*$ matrices are of size $K \times M$, $\mathbf{A}_{\text{GS}}$ is near-integer, $\widetilde{\mathbf{A}}$ satisfies both unique association and load balancing, and the final output $\mathbf{A}$ satisfies integer and unique association.

Then applying layer normalization to $\widetilde{\mathbf{G}}_k^{(l)}$, the output attention matrix for UE k is given as

$$\mathbf{G}_k^{(l)} = \text{LayerNorm}(\widetilde{\mathbf{G}}_k^{(l)}) \tag{7}$$

The attention factors $\mathbf{G}_k^{(l)}$ are subsequently used to update the edge representations. Specifically, each edge representation is updated by its representation from last layer and its corresponding attention factor. This allows the model to selectively emphasize more relevant BS for each UE k, thereby enhancing the expressiveness of the edge representations. The attention-based edge update mechanism is designed as:

$$\mathbf{e}_{m,k}^{(l)} = f_5^{(l)}\left(\mathbf{e}_{m,k}^{(l-1)}, \mathbf{g}_{m,k}^{(l)}\right) \tag{8}$$

where f_5 is an MLP and $\mathbf{g}_{m,k}^{(l)} \in \mathbb{R}^d$ is the m-th row vector of the attention matrix $\mathbf{G}_k^{(l)}$ in Eq. (7). This attention-based edge update efficiently captures all of UE k's interactions with the BSs in one step—no separate aggregation is needed.

3.4 Reparameterization Process

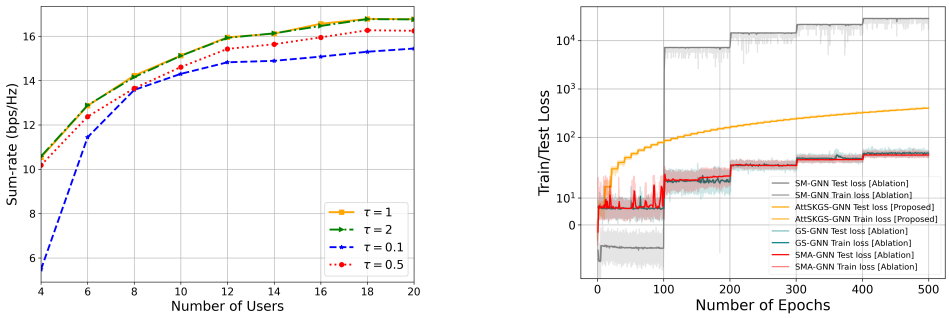
After updating the representations for L layers in the GNN, the resulting edge representation $\mathbf{E}^{(L)}$ is projected into the user association matrix $\mathbf{A}$. Before this projection, an MLP is applied to $\mathbf{E}^{(L)}$ to reshape it into $\hat{\mathbf{A}} \in \mathbb{R}^{K \times M}$ to match the dimensions of $\mathbf{A}$. Next, for each UE k, we apply the GS reparameterization [8] to the k-th row of $\hat{\mathbf{A}}$ with a small temperature value τ, yielding near-integer association factors. Repeating this process for all K UEs yields a near integer association matrix $\mathbf{A}_{\text{GS}}$ which encodes the probabilistic association between K UEs and M BSs.

To achieve a load-balanced output, we apply the SK normalization to scale $\mathbf{A}_{\text{GS}}$. Denote the initial matrix for SK normalization as $\widetilde{\mathbf{A}}^{(0)} \triangleq \mathbf{A}_{\text{GS}}$. After i iterations of SK normalization, the resulting matrix $\widetilde{\mathbf{A}} \triangleq \widetilde{\mathbf{A}}^{(i)}$ will satisfies both constraints for unique association and load balancing, that is, the second and third constraints in (2a). Nevertheless, $\widetilde{\mathbf{A}}$ remains real-valued and non-integer, hence does not yet satisfy the first constraint in (2a). We note that both GS reparameterization and SK normalization drive entries in $\widetilde{\mathbf{A}}$ quite close to binary, effectively approximating integer assignment. To obtain a binary matrix, we apply the one-hot coding to $\widetilde{\mathbf{A}}$ as described next.

To enforce integer-valued associations while preserving differentiability, each row of $\widetilde{\mathbf{A}}$ is one-hot encoded after training. The resulting $\mathbf{A}_{\text{1hot}}$ assigns each UE to the BS with the highest probability, ensuring unique integer association. $\mathbf{A}_{\text{1hot}}$ is then used for sum-rate evaluation, while the loss in Eq. (3) and backpropagation rely on the continuous $\widetilde{\mathbf{A}}$ during training.

We note that the column sums of $\mathbf{A}_{\text{1hot}}$ may violate the BS load constraint, since the one-hot operation only guarantees that each UE is assigned to a single BS without considering the maximum number of UEs a BS can serve. Thus, although $\widetilde{\mathbf{A}}$ satisfies the load-balancing constraint after SK normalization, its one-hot version $\mathbf{A}_{\text{1hot}}$ may still overload some BSs.

Nevertheless, the reparameterization process produces an integer $\mathbf{A}_{\text{1hot}}$ that closely satisfies load balancing while achieving the best sum-rate, as shown in the ablation study. During training, gradients are preserved by using $\widetilde{\mathbf{A}}$ instead of $\mathbf{A}_{\text{1hot}}$, facilitating effective representation learning. The overall process is illustrated in Fig. 1.



(a) (b)

Figure 2: Performance tests: (a) Evaluation of the sensitivity for temperature factor the GS τ. (b) Training and test behaviors of the proposed GNN for association, showing the regularized loss (Eq. (3)); test loss (lighter shade) nearly overlaps training. These results use the BFGNN in [6] for design the beamforming in Eq. (2).

3.5 Regularization Updates and Training

During training, the GNN inputs samples consisting of $(\mathbf{p}, \mathbf{q}, \mathbf{H})$ and outputs the real-valued association matrix $\widetilde{\mathbf{A}}$. The loss function in Eq. (3)–both the negative rate term and the load-violation penalty–is defined on the real-valued association matrix $\widetilde{\mathbf{A}}$, so that all gradients are retained.

To gradually penalize load-balancing violations during training, we update the regularization coefficient μ_m in Eq. (3) every 100 epochs as

$$\mu_m = \mu_m^{\text{last}} + \mathbb{E}_{\text{epoch}}[\text{ReLU}(\sum_k \widetilde{a}_{k,m} - (N - \delta))] \tag{9}$$

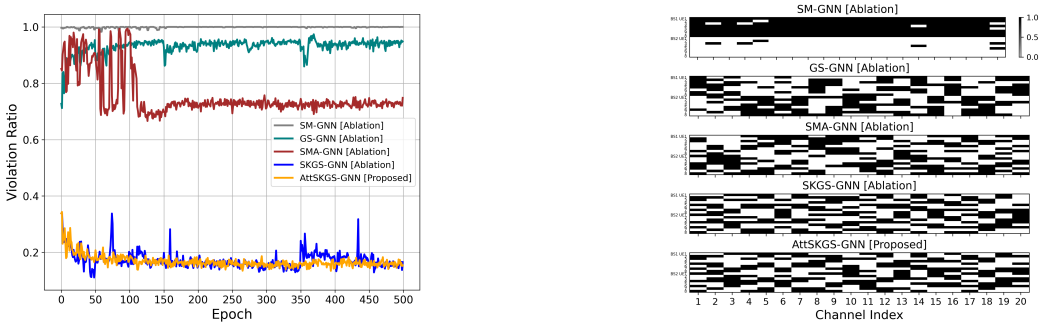
where μ_m^{last} is the previous value and the second term is the average violation penalty across all samples in the epoch. This dynamic adjustment gradually emphasizes the load-balancing penalty, stabilizing training and allowing progress in primal optimization before the penalty is increased. During training, $\widetilde{\mathbf{A}}$ is used for backpropagation in the loss, while performance evaluation is based on the instantaneous sum-rate computed from $\mathbf{A}_{\text{1hot}}$.

4 Numerical Results

We evaluate our GNN in a multi-user multi-cell MIMO beamforming network at 28 GHz carrier frequency and 1 GHz bandwidth, with noise of -174 dBm/Hz. The network consists of 2 BSs and a varying number of UEs uniformly distributed over a 200m $\times$ 200m area. Each BS has 4 ULA antennas and 30 dBm transmit power, with maximum ratio transmission (MRT) fixed for training. The channel is modeled as $\mathbf{h} = \frac{1}{\sqrt{I}} \sum_{i=1}^{I} \alpha_i e^{j\phi_i} \mathbf{a}_{tx}(\theta_i^{tx})$, where α_i is the i-th path gain and ϕ_i is the uniformly distributed phase [7] and $\mathbf{a}_{tx}(\cdot) \in \mathbb{C}^N$ is the array response vector. We independently generate $990,000$ training samples and $10,000$ test samples independently without overlap.

All GNN variants use MLPs with two hidden layers of 1024 nodes, trained with Adam and 400 mini-batches per epoch. The GNN has $L = 2$ layers, $d = 512$ features, and is trained on 2 BSs and 8 UEs, with inference tested on larger networks. Performance is averaged over $3,000$ unseen samples. We adopt mean aggregation, a warm-restart scheduler (initial period 50 epochs, multiplier 2) with learning rate cycling between 10^{-8} and 5×10^{-5}, GS temperature $\tau = 1$, and regularization factor $\delta = 2$. All experiments are implemented in PyTorch on an NVIDIA A100 GPU.

We evaluate our association schemes against several benchmarks. **ES_BFGNN** is exhaustive search for association and using BFGNN [6] for beamforming. **JointGNN** jointly optimizes association and beamforming, using *softmax* for association [8]. **SMA-GNN**, **GS-GNN** and **SKGS-GNN** all adopt the GNN structure in [6] but use the edge representations for association only, with different reparameterization methods applied accordingly. **SMA-GNN** uses *softmax*, and **GS-GNN** and **SKGS-GNN** use GS and SKGS, respectively. **RandInt_BFGNN** generates random load-balanced integer associations and using BFGNN [6] for beamforming. **DA_MRT** applies the traditional dual optimization for load-balanced association. **MaxSINR_BFGNN** applies the maximum SINR for association and BFGNN [6] for beamforming. The ablation study is shown in Fig. 2, 3 and Table 1.

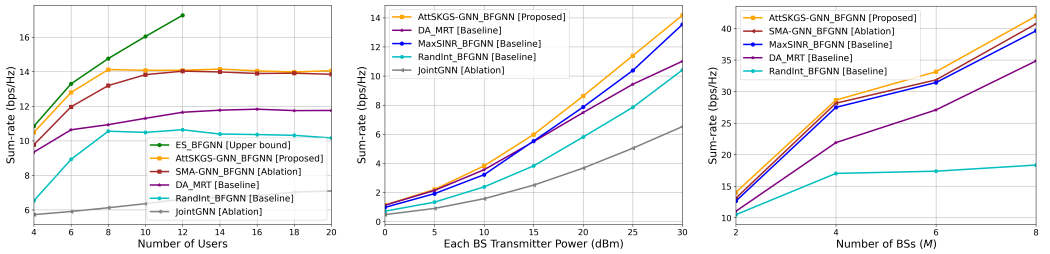


(a) (b)

Figure 3: Performance tests: (a) Load-balancing violation ratio vs. training epoch. (b) One-hot coded integer association output (black=1, white=0).

	AttSKGS	SM [8]	SMA	GS	SKGS
Violation on load balancing (%)	**16.2%**	99.2%	72.3%	94.2%	18.5%
Sum-rate with MRT (with violations)	6.54	-	5.92	**9.32**	6.49
Load-balanced sum-rate with MRT	**6.24**	-	6.12	5.92	6.01
Load-balanced sum-rate with BFGNN in [6]	**14.1**	6.0	13.2	13.2	13.5
Training time (hour)	36.98 (h)	47.78	36.41	**28.92**	31.79
Inference time (ms)	**70** (ms)	85	84	71	72
Number of trainable parameters	**45.67**M	54.60M	53.01M	53.01M	53.01M

Table 1: Ablation study of 5 schemes: All models are trained with 2 BSs (4 antennas each) and 8 UEs. The top two rows show training performance after one-hot coding for the output, which can violate the load balancing constraint. The third and fourth rows are test performance when the overloaded users are dropped. All sum rates are reported in bps/Hz.



(a) (b) (c)

Figure 4: Generalization ability tests: GNNs trained on 8 UEs and 2 BSs with $P_t = 30$dBm, then applied to networks with (a) varying UEs or (b) P_t values, without retraining. (c) shows generalization to both more BSs and more UEs, where the number of UEs is $M \times N$.

The temperature parameter τ in the Gumbel-softmax controls the discreteness of the output: smaller τ values make it more near-integer but reduce gradient smoothness, hindering stable learning and convergence, and may degrade end performance. Based on extensive testing as shown in Fig. 2(a), $\tau = 1$ and $\tau = 2$ yield the best and nearly identical results; thus, we fix $\tau = 1$ in the paper. Fig. 2(b) shows the negative of the regularized loss in Eq. (3). In SM-GNN, omitting the regularization term yields a pure sum-rate objective; adding it later causes a sharp rise. Baseline methods that update μ_m every 100 epochs exhibit periodic spikes from abrupt penalty jumps. In contrast, AttSKGS-GNN applies SK normalization and GS reparameterization continuously, letting μ_m adapt smoothly as shown in Eq. (9) and the loss decrease steadily.

Fig. 3(a) reports the load balancing violation ratio. For each GNN model, its load violation ratio δ for an epoch is defined as the fraction of samples in which at least one BS m experiences an overload, denoted as $\delta = \frac{1}{S}\sum_{s=1}^{S} \mathbf{1}\left(\max_m \left(\sum_k a_{k,m}^{(s)}\right) > N\right)$, where S is the number of sam-

ples in an epoch, $\mathbf{1}(\cdot)$ is the indicator function and $\sum_k a_{k,m}^{(s)}$ is the one-hot encoded association factor in $\mathbf{A}_{\text{1hot}}$. SK-based methods (SKGS, AttSKGS) clearly outperform those without SK, while AttSKGS achieves the smoothest, lowest violation curve, confirming its stronger and more stable load balancing. Fig. 3(b) illustrates the one-hot coded associations. AttSKGS-GNN achieves the most load-balanced assignments with the highest probability.

Table 1 compares different GNN models. AttSKGS-GNN, which integrates an attention mechanism, has fewer trainable parameters than SKGS-GNN. Unlike SKGS-GNN's large MLP-based edge updates, attention replaces MLPs with dot-product attention, reducing parameters but increasing training cost. Specifically, AttSKGS-GNN must compute gradients not only for linear projections (queries, keys, values) but also for the attention weights and *softmax*, leading to a longer computational chain, higher memory use, and greater cost.

For the sum-rate comparisons in Table 1, all models adopt the same beamforming scheme (MRT) to ensure fair evaluation in both converged and load-balanced cases in the second and third rows. The higher converged sum-rate of GS-GNN during training stems from frequent violations of the load constraint (Fig. 3(a)), which leave only a few UEs connected to one BS and artificially boost the rate. During inference, extra UEs exceeding the BS load limits are dropped. Under these conditions, the proposed AttSKGS-GNN delivers the best overall performance. For fair comparison with SM-GNN, beamforming is further optimized using the GNN in [6], and the corresponding load-balanced sum-rate is reported in row 4 of Table 1.

Fig. 4(a) shows the generalization performance as the number of UEs increases. The GNNs are trained on a fixed network size with MRT and evaluated on larger UE counts or different transmit powers P_t. When a BS exceeds its load limit, excess UEs are dropped before beamforming is applied [6], ensuring that no more than eight UEs are served. Note that ES provides the true optimum but is computationally feasible only for small networks, our proposed AttSKGS-GNN closely matches its performance up to eight UEs and remain near-optimal as the network scales. Fig. 4(b) shows that AttSKGS-GNN consistently outperforms all other baselines across all power levels, demonstrating strong robustness to interference, user density, and transmit-power variations—without retraining. Finally, Fig. 4(c) confirms robust generalization to larger networks with more BSs and more UEs, where the number of UEs is set to $M \times N$ with N being the number of BS antennas.

5 Conclusion

In this paper, we proposed a learning-based approach that combines a novel attention-based GNN architecture with Gumbel-Softmax reparameterization and Sinkhorn-Knopp normalization. To encourage the load constraint, we utilized a regularization term in the loss function. By learning to produce near load-balanced and integer associations, the network can output a higher sum-rate with less interference. Numerical results show that our proposed association method achieves near-optimal performance and generalizes well to larger and more dynamic networks.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Science Foundation under CNS grants 2340284 and 2403285.

References

[1] A. Alizadeh and M. Vu, "Load balancing user association in millimeter wave MIMO networks," IEEE Trans. Wireless Commun., vol. 18, no. 6, pp. 2932–2945, Jun. 2019.
[2] S. Corroy, L. Falconetti, and R. Mathar, "Dynamic cell association for downlink sum rate maximization in multi-cell heterogeneous networks," in Proc. IEEE Int. Conf. Commun. (ICC), 2012.
[3] P. W. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," arXiv preprint arXiv:1806.01261, 2018.
[4] A. Vaswani et al., "Attention is all you need," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017.
[5] P. A. Knight. "The Sinkhorn–Knopp algorithm: convergence and applications." SIAM J. Matrix Anal. Appl., 30(1):261–275, 2008.
[6] Q. Lyu and M. Vu, "Efficient edge-update GNN structure for beamforming and power allocation in wireless networks," in Proc. IEEE Global Commun. Conf. (GLOBECOM), 2025, accepted.
[7] M. R. Akdeniz et al., "Millimeter wave channel modeling and cellular capacity evaluation," IEEE J. Sel. Areas Commun., vol. 32, no. 6, pp. 1164–1179, Jun. 2014.
[8] Q. Lyu and M. Vu, "Joint beamforming and integer user association using a GNN with Gumbel-Softmax reparameterizations," IEEE Trans. Veh. Technol., 2025.