Adaptive GNN-based Proportional-Fair Scheduling in MIMO Networks for Non-stationary Channels

Yirong Cheng
ECE Department
Rice University
Houston, TX 77005
`yc118@rice.edu`

Divyanshu Pandey
ECE Department
Rice University
Houston, TX 77005
`dp76@rice.edu`

Ashutosh Sabharwal
ECE Department
Rice University
Houston, TX 77005
`ashu@rice.edu`

Abstract

We study adaptive machine learning (ML)-based Proportional Fair (PF) scheduling for MIMO networks under non-stationary channel conditions, which are increasingly prevalent in dense MIMO deployments with complex scattering environments. Traditional PF schedulers face the trade-off limitations between utility and latency, whereas conventional ML-based schedulers, although balancing utility and latency better, degrade under changing channel distributions. We propose a Node-and-Edge Attention Graph Neural Network (NEA-GNN) that exploits both MIMO network structure and PF scheduling criteria. NEA-GNN employs attention-based message passing to jointly capture inter-user interference patterns and per-user metrics, enabling high transmission fairness and throughput while achieving low adaptation overhead, improving long-term utility under dynamic channels. Experiments over simulated and real-world channel measurements demonstrate that NEA-GNN is more sample-efficient than conventional ML models and consistently outperforms existing PF schedulers in non-stationary MIMO environments.

1 Introduction

Motivations in MIMO Scheduler Design With the advancement of next-generation wireless Multi-Input Multi-Output (MIMO) networks, a wider range of user applications can be supported. The emerging applications like AR/VR, 4K/8K Videos, and Remote Healthcare impose stricter requirements on throughput, latency, and fairness to ensure high Quality of Service (QoS). At the same time, MIMO networks must operate under limited communication resources, which makes it challenging to serve all users simultaneously. Therefore, an intelligent MIMO scheduler is needed to allocate wireless resources efficiently and coordinate transmissions across users, thereby enhancing QoS in next-generation MIMO networks.

When designing a MIMO scheduler, several key performance objectives must be considered. The two traditional goals are: **high utility**—balancing spectral efficiency to maximize overall throughput while ensuring fairness to prevent user starvation; and **low latency**—supporting millisecond-level delays required by real-time applications. Beyond these objectives, a less explored but increasingly important challenge is **adapting scheduling policies to non-stationary channel processes**. As MIMO cell sizes continue to shrink with denser deployments Cho et al. [2023], the scattering environment becomes more complex and dynamic, making channel conditions highly sensitive to user mobility. Consequently, channel statistics evolve over time, and since scheduling policies often depend on channel statistics, the scheduler must adapt to time-varying channel distributions.

Prior Works and Advantages of Learning-based Approach Proportional Fair (PF) scheduling Liu et al. [2010] is widely used in MIMO networks to balance throughput and fairness for good utility, but its optimal solution in multi-user MIMO is NP-hard Lee et al. [2012], Prasad et al. [2013]. Most practical schedulers use approximation methods to achieve PF scheduling, yet suffer from the trade-off limitation between high utility and low latency. Simple PF heuristics Yoo and Goldsmith [2006], Benmimoune et al. [2015], Meng et al. [2018], Yang [2018], Chen et al. [2020] reduce latency but hurt utility, while near-optimal methods Zhu et al. [2021], Chen et al. [2021] incur high latency at scale.

More recently, Machine Learning (ML)-based schedulers Cui et al. [2019], An et al. [2023], Comșa et al. [2019] have shown promise by directly learning mappings from channel/user states to scheduling decisions. ML-based PF schedulers achieve both high utility and GPU-parallelized low-latency runtime at the same time. However, most existing works assume fixed channel statistics and degrade significantly under non-stationary distribution shifts Quiñonero-Candela et al. [2022]. Since ML models already strike a better balance between utility and latency than classical methods, we adopt an ML-based scheduler solution and focus on enhancing their adaptivity to dynamic environments to achieve high utility, low latency, and good channel adaptivity.

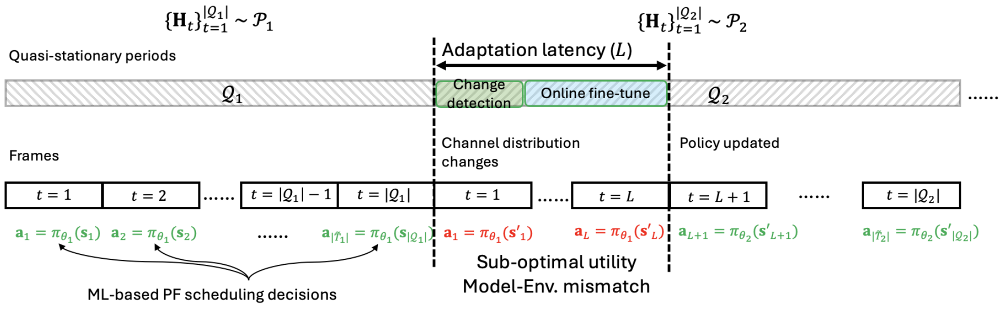
GNN for Online Adaptive ML-based PF Scheduler In non-stationary environments, ML model–environment mismatch reduces scheduling utility. To mitigate the harm, an adaptive ML-based PF scheduler must fine-tune online to new channel distributions while minimizing fine-tuning overhead, thereby reducing mismatch duration and improving long-term utility. Graph Neural Networks (GNNs) are natural candidates, as their architecture explicitly encodes network topology, leading to higher sample efficiency and better generalization in large-scale MIMO Shen et al. [2023]. Hence, in this work, we design a **Node-and-Edge Attention GNN (NEA-GNN)** that leverages both PF scheduling criteria and MIMO topology structure. NEA-GNN achieves significantly shorter online fine-tuning compared to conventional ML models and thereby has better average utility over time. To the best of our knowledge, we are the first to exploit the high sample efficiency of GNNs for adaptive PF scheduling in non-stationary MIMO networks.

2 System Model

Network Setup Consider an uplink MIMO network where a base station with M antennas serves N single-antenna users in a single-carrier system. Scheduling occurs per frame t with channel matrix $\mathbf{H}_t \in \mathbb{C}^{M\times N}$. Due to user mobility and change of scattering environments, the channel matrix $\mathbf{H}_t$ may have different statistical distributions over time. In this work, we assume users with slow mobility (e.g., pedestrian users), where the channel can be modeled as block-wise stationary Meinilä et al. [2009]. Mathematically, the overall time horizon $\mathcal{T}$ is partitioned into multiple quasi-stationary periods, denoted as $\{\mathcal{Q}_i\}_{i=1,2,3,\ldots}$. Within each quasi-stationary period, the channel state process remains stationary with a fixed distribution $\mathbf{H}_t \sim \mathcal{P}_i$ and a correlation matrix $\mathbf{R}_i = \mathbb{E}[\mathbf{H}_t^H \mathbf{H}_t]$.

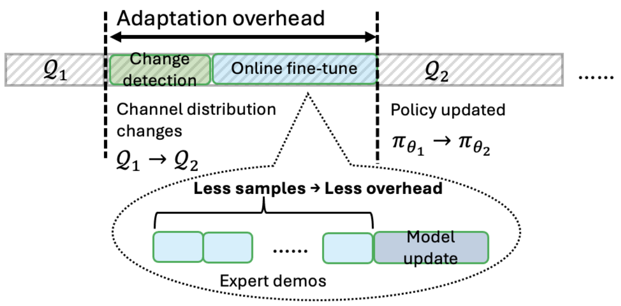
Online Adaptation Framework Let π_{θ_i} be the ML scheduler for $\mathcal{Q}_i$. Upon entering a new period, we detect distribution shifts using a change detection scheme that combines CUSUM-based self-monitoring with selective expert invocation. Concretely, the scheduler tracks variation of short-term utility via CUSUM statistics to flag potential shifts Page [1954], Basseville et al. [1993]; suspicious cases are verified by consulting an expert scheduler of a near-optimal greedy heuristic Zhu et al. [2021], invoked both reactively (when CUSUM signals degrade) and proactively (periodic refresh to avoid missed detections). Once a shift is confirmed, we fine-tune $\pi_{\theta_{i-1}}$ to π_{θ_i} via Behavior Cloning (BC) Pomerleau [1988], using expert-generated data $\mathcal{D} = \{(\mathbf{s}_t, \hat{\mathbf{a}}_t)\}_{t\in\mathcal{Q}_i}$, where $\mathbf{s}_t = \{\mathbf{H}_t, \mathbf{c}(t)\}$ denotes the channel state (channel matrix and average rate vector) and $\hat{\mathbf{a}}_t \in \{0,1\}^N$ is the expert's scheduling decision.

Online Fine-tuning Objective In this work, we focus on designing an efficient online fine-tuning mechanism with short overhead for better adaptivity. Behavior Cloning fine-tunes the ML model using a standard supervised learning mechanism, and consists of two steps in our online fine-tuning case: (i) collect a training dataset $\mathcal{D}$ online with expert demonstration; (ii) update the ML model parameter using gradient methods. While the gradient updates are comparatively lightweight, since the ML model must remain compact for real-time latency and hardware constraints, we argue that the dominant overhead in online fine-tuning is the expert-driven sample collection, as it requires invoking the near-optimal expert for every sample, which has very high run-time complexity. Therefore, the key objective in reducing fine-tuning overhead is to reduce the amount of expert samples needed for



$\{\mathbf{H}_t\}_{t=1}^{|\mathcal{Q}_1|} \sim \mathcal{P}_1$
$\{\mathbf{H}_t\}_{t=1}^{|\mathcal{Q}_2|} \sim \mathcal{P}_2$
Quasi-stationary periods
Adaptation latency (L)
$\mathcal{Q}_1$
Change detection
Online fine-tune
$\mathcal{Q}_2$
Channel distribution changes
Policy updated
Frames
$t = 1$
$t = 2$
$t = |\mathcal{Q}_1| - 1$
$t = |\mathcal{Q}_1|$
$t = 1$
$t = L$
$t = L + 1$
$t = |\mathcal{Q}_2|$
$\mathbf{a}_1 = \pi_{\theta_1}(\mathbf{s}_1)$
$\mathbf{a}_2 = \pi_{\theta_1}(\mathbf{s}_2)$
$\mathbf{a}_{|\mathcal{T}_1|} = \pi_{\theta_1}(\mathbf{s}_{|\mathcal{Q}_1|})$
$\mathbf{a}_1 = \pi_{\theta_1}(\mathbf{s}'_1)$
$\mathbf{a}_L = \pi_{\theta_1}(\mathbf{s}'_L)$
$\mathbf{a}_{L+1} = \pi_{\theta_2}(\mathbf{s}'_{L+1})$
$\mathbf{a}_{|\mathcal{T}_2|} = \pi_{\theta_2}(\mathbf{s}'_{|\mathcal{Q}_2|})$
Sub-optimal utility
Model-Env. mismatch
ML-based PF scheduling decisions

Figure 1: The framework of online adaptive ML-based PF scheduler.



Adaptation overhead
$\mathcal{Q}_1$
Change detection
Online fine-tune
$\mathcal{Q}_2$
Channel distribution changes
$\mathcal{Q}_1 \to \mathcal{Q}_2$
Policy updated
$\pi_{\theta_1} \to \pi_{\theta_2}$
Less samples → Less overhead
Model update
Expert demos

Figure 2: The fine-tuning overhead breakdown illustrating the need for sample-efficient ML models.

effective training (i.e., reducing the size of the training dataset $|\mathcal{D}|$). We adopt the NEA-GNN ML model to achieve the objective.

3 Topology Structure Utilized GNN Design

We adopt a Graph Neural Network (GNN) architecture for Behavior Cloning, following its strong performance in wireless scheduling Zhang et al. [2022], Zhao et al. [2022], Li et al. [2024], Liu and Yang [2024]. Unlike MLPs or CNNs, GNNs naturally encode network topology (e.g., inter-user interference) and achieve higher sample efficiency and lower generalization error in large-scale MIMO Shen et al. [2023], enabling rapid adaptation with fewer expert samples. Figure 3 shows the block diagram of our GNN model structure, where we will provide a detailed explanation for each part in the sequel.

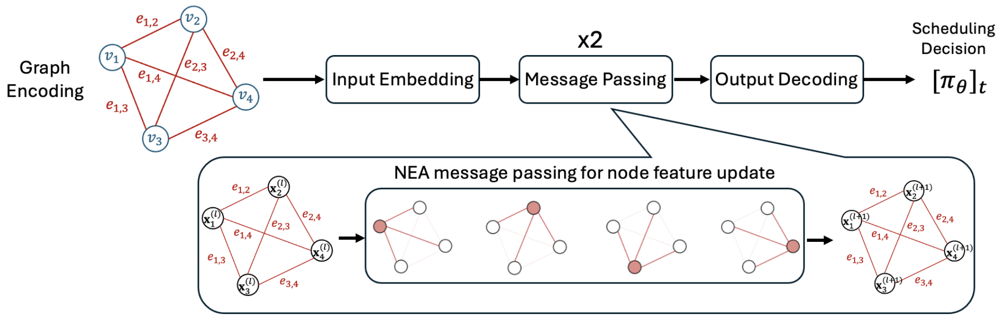


Figure 3: The block diagram of our NEA-GNN model.

Structure-utilized Graph Encoding We represent the MIMO network with N users as a fully connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where node v_k denotes user k and edge $e_{k,j}$ models pairwise relationships. For notational convenience, we reuse the notation to denote node features as v_k and edge features as $e_{k,j}$ throughout. For proportional fair (PF) scheduling Liu et al. [2010], at frame t in a quasi-stationary period $\mathcal{Q}$:

$$\mathbf{a}_t^{\mathsf{PF}} = \arg\max_{\mathbf{a}_t} \sum_{k=1}^{N} \frac{r_{k,\mathbf{a}_t}(t)}{c_k(t)} \cdot a_{k,t}, \tag{1}$$

where the scheduling decision depends on average rates $c_k(t)$ and the instantaneous rates $r_{k,\mathbf{a}_t}(t)$ of user k at t^{th} frame, which is further determined by the inter-user channel correlation embedded in the Gram matrix $\mathbf{G}_t = \mathbf{H}_t^H \mathbf{H}_t$. Hence we decode our node and edge features to illustrate both per-user and inter-user network characteristics: $v_k = c_k(t) \in \mathbb{R}$ and $e_{k,j} = \{\mathbf{G}_t\}_{k,j} \in \mathbb{C}$. Edge features are split into real/imaginary parts: $\mathbf{e}_{k,j} \in \mathbb{R}^2$.

Input Embedding We first embed the scalar node feature v_k into a d_e-dimensional vector $\mathbf{x}_k^{(0)}$ via a shared embedding layer Φ_{in} as

$$\mathbf{x}_k^{(0)} = \Phi_{\mathsf{in}}(v_k) = \mathrm{BN} \circ \mathrm{ReLU}\big(\mathbf{W}_{\mathsf{in}} \cdot v_k + \mathbf{b}_{\mathsf{in}}\big), \quad \forall k. \tag{2}$$

Here $\mathbf{x}_k^{(0)} \in \mathbb{R}^{d_e}$, $\mathbf{W}_{\mathsf{in}}, \mathbf{b}_{\mathsf{in}} \in \mathbb{R}^{d_e \times 1}$, BN denotes Batch Normalization, and ReLU is the activation function. Embedding into a higher-dimensional space enhances feature expressiveness, allowing the GNN to capture richer patterns in the subsequent message passing stage.

Message Passing After embedding, node features are refined via message passing, where each node aggregates information from neighbors to capture both per-user metrics (e.g., average rates) and inter-user relations (e.g., interference patterns). Node features encode user-specific performance, while edge features capture pairwise interactions—both essential for PF scheduling's throughput–fairness trade-off. We adopt a Node-and-Edge Attention GNN (NEA-GNN), which incorporates both node and edge features into the aggregation process. Unlike fixed-weight aggregation in conventional GNNs, NEA-GNN learns attention scores as functions of node states and edge attributes, improving expressivity, sample efficiency, and scalability for dynamic MIMO interference patterns. For a message passing layer l, the message from neighbor j to node k is expressed as

$$\mathbf{m}_{k,j}^{(l)} = \Phi_{\mathsf{MP}}(\mathbf{x}_k^{(l)}, \mathbf{x}_j^{(l)}, \mathbf{e}_{k,j}) = \hat{\alpha}_{k,j}^{(l)} \cdot \Big(\phi_V\big(\mathbf{x}_j^{(l)}\big) + \phi_E\big(\mathbf{e}_{k,j}\big)\Big), \quad \forall j \neq k, \tag{3}$$

where ϕ_V and ϕ_E project the node and edge features defined as

$$\phi_V(\mathbf{x}) = \mathbf{W}_V \cdot \mathbf{x} + \mathbf{b}_V, \quad \mathbf{W}_V \in \mathbb{R}^{d_e \times d_e}, \mathbf{b}_V \in \mathbb{R}^{d_e}, \tag{4}$$

$$\phi_E(\mathbf{e}) = \mathbf{W}_E \cdot \mathbf{e} + \mathbf{b}_E, \quad \mathbf{W}_E \in \mathbb{R}^{2 \times d_e}, \mathbf{b}_E \in \mathbb{R}^{d_e}. \tag{5}$$

The attention scores are computed as

$$\hat{\alpha}_{k,j}^{(l)} = \mathsf{SoftMax}\left(\frac{\Big\langle \phi_Q(\mathbf{x}_k^{(l)}),\ \phi_K(\mathbf{x}_j^{(l)}) + \phi_E(\mathbf{e}_{k,j}) \Big\rangle}{\sqrt{d_e}}\right), \tag{6}$$

with projections:

$$\phi_Q(\mathbf{x}) = \mathbf{W}_Q \cdot \mathbf{x} + \mathbf{b}_Q, \quad \phi_K(\mathbf{x}) = \mathbf{W}_K \cdot \mathbf{x} + \mathbf{b}_K, \quad \mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d_e \times d_e},\ \mathbf{b}_Q, \mathbf{b}_K \in \mathbb{R}^{d_e}. \tag{7}$$

Messages are aggregated as $\mathbf{m}_k^{(l)} = \sum_{j \neq k} \mathbf{m}_{k,j}^{(l)}$, and node features are updated with a residual connection:

$$\mathbf{x}_k^{(l+1)} = \mathrm{ReLU}\big(\phi_R(\mathbf{x}_k^{(l)}) + \mathbf{m}_k^{(l)}\big), \quad \phi_R(\mathbf{x}) = \mathbf{W}_R \cdot \mathbf{x} + \mathbf{b}_R, \quad \mathbf{W}_R \in \mathbb{R}^{d_e \times d_e},\ \mathbf{b}_R \in \mathbb{R}^{d_e}. \tag{8}$$

Output Decoding Lastly, we decode the output from message passing into PF scheduling decisions. Specifically, we project each node's final representation into a scalar scheduling score via a 2-layer MLP Φ_{out}:

$$[\pi_\theta]_{k,t} = \Phi_{\mathsf{out}}\big(\mathbf{x}_k^{(l)}\big) = \mathsf{Sigmoid}\Big(\mathbf{W}_2 \cdot \mathrm{BN} \circ \mathrm{ReLU}\big(\mathbf{W}_1 \cdot \mathbf{x}_k^{(l)} + \mathbf{b}_1\big) + \mathbf{b}_2\Big) \tag{9}$$

where $[\pi_\theta]_{k,t} \in (0,1)$, the final output of our GNN π_θ over node v_k at t^{th} frame, is the likelihood of the user k being scheduled at t^{th} frame. At each frame, the GNN would output a scalar of $[\pi_\theta]_{k,t} \in (0,1)$ for all the nodes within $\mathcal{G}$ to form a soft scheduling assignment vector $[\pi_\theta]_t \in (0,1)^N$. We derive our final binary scheduling decision $\tilde{a}_{k,t} \in \{0,1\}^N$ by picking the K users with the largest scheduling likelihood in $[\pi_\theta]_t$.

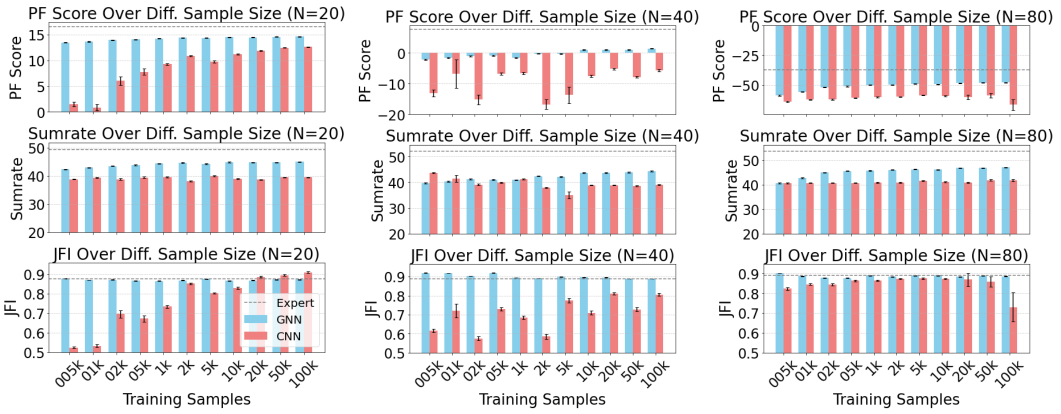


Figure 4: Utilities for both NEA-GNN and CNN-based PF scheduler with same training samples under different network sizes.

4 Experiments and Results

In this section, we aim to demonstrate that our proposed NEA-GNN model exhibits better sample efficiency than traditional ML models by achieving a higher PF scheduling utility when trained with the same number of samples. We first compare the utility performance of our NEA-GNN model with that of a conventional CNN model under an ablation test with varying MIMO network sizes and training sample sizes, using a simulated Kronecker channel model Oestges [2006]. Then, we test both the NEA-GNN and CNN-based PF schedulers against more classical PF schedulers under a real-life non-stationary channel process to validate that our GNN-based PF scheduler can achieve a better operational point that jointly considers utility, latency, and adaptivity.

Sample Efficiency Ablation Test Using the simulated Kronecker channel model (details in Appendix), we compare our proposed NEA-GNN model with a dual-branch CNN baseline. The CNN processes the channel matrix via convolution and average pooling to obtain spatial interference features, and processes the user average rate vector using fully connected (FC) layers to extract fairness-related features. The spatial interference and fairness-related feature vectors are concatenated and passed through an FC network to produce per-user scheduling scores, with the top-K users selected for transmission.

Both the NEA-GNN and CNN models are initially pre-trained under a channel environment where the channel realizations are drawn from a distribution characterized by parameters (details in Appendix) with $\alpha_1 = 1$, $\alpha_2 = 1$, $\beta_1 = 0.8$, and $\beta_2 = 0$. The pretraining dataset consists of $N_p = 200{,}000$ samples, all generated based on scheduling decisions provided by the expert demonstration policy based on Zhu et al. [2021].

Following pretraining, we fine-tune both models in a new channel environment where $\alpha_1 = 0.5$, representing a scenario in which half of the clustered users are moving away from the base station, resulting in reduced channel gains after the distributional shift.

To evaluate the robustness and scalability of the models, we conduct experiments across varying network sizes with total user counts of $N = 20$, 40, and 80. In all scenarios, the base station is equipped with $M = 10$ antennas, and the scheduler is constrained to select a maximum of $K = 8$ users for transmission at each time frame. We use metrics of PF Score, Sum of rates, and Jain Fairness Index (JFI) for utility evaluation:

$$\text{PF Score} = \sum_k \log(c_k(t)), \quad \text{Sumrate} = \mathbb{E}_t[\sum_k r_{k,\pi}(t)],$$

$$\text{Jain Fairness Index (JFI)} = \left(\sum_{k=1}^{N} c_k(t)\right)^2 \Big/ \left(N \cdot \sum_{k=1}^{N} (c_k(t))^2\right).$$

where JFI ranges from $\frac{1}{N}$ to 1 and is maximized when all users receive the same allocation.

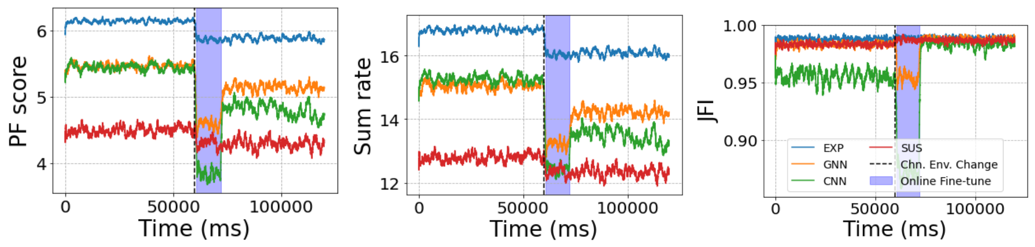


Figure 5: The scheduling performance for common MIMO schedulers over time under the non-stationary MIMO channel state process from the RENEW dataset.

For each setup of network training sample size, we run both GNN- and CNN-based schedulers over 1,000 consecutive time frames and repeat 200 independent trials in total to measure the mean and standard deviation of all metrics above. Figure 4 shows that the GNN-based PF scheduler consistently outperforms the CNN baseline across all network sizes and training budgets. GNN achieves $\geq 85\%$ spectral efficiency and $\leq 2\%$ fairness loss relative to the expert when trained with >10k samples, while the CNN lags behind with the same samples, especially for larger networks ($N = 40, 80$) where its utility gap grows, achieving only $\sim 75\%$ of expert's spectral efficiency, and with no guarantee over the JFI metric converging towards the expert. Even for the smaller network with $N = 20$, CNN shows slower gains and lower final performance: even though CNN and GNN could both have their fairness metrics converge close to the expert, CNN has only 81% of expert spectral efficiency under 100k training samples, whereas GNN reaches $\sim 85\%$ of expert's spectral efficiency with only 500 training samples, and $\sim 94\%$ expert's spectral efficiency with 100k training samples.

By requiring fewer data samples to reach a satisfactory level of utility, the NEA-GNN enables faster fine-tuning and reduces the overall overhead involved in adapting the PF scheduler to changing channel conditions.

Scheduling under Real-World Channel Next, we validate that the sample efficiency of the GNN model can benefit the long-term scheduling utility over time, using real-world channel state process measurements that model two channel distribution environments under physical user movements (details in Appendix). We compare the PF utility scores as long as the sum rate and JFI metrics of our GNN-based scheduler with the conventional CNN-based scheduler, the near-optimal expert (EXP) Zhu et al. [2021], and a simple heuristic-based PF scheduler called Semi-orthogonal User Selection (SUS) Yoo and Goldsmith [2006].

From Figure 5, we observe that our GNN scheduler achieves higher PF utility than both SUS and CNN schedulers, balancing spectral efficiency and fairness without sacrificing either metric across different channel environments. The ML-based scheduler outperforms traditional heuristics like SUS by learning from near-optimal experts. Compared to conventional ML models such as CNN, GNN achieves higher sample efficiency, leading to even better utility with the same online training sample sizes. While the expert PF scheduler (EXP) attains the highest utility, it suffers from high runtime complexity, making it impractical under real-time latency constraints (see Appendix). Overall, our GNN-based scheduler delivers strong long-term utility under non-stationary channels with low scheduling latency.

5 Conclusion

In this work, we consider an adaptive ML-based Proportional-Fair scheduler for a MIMO network with non-stationary channels, which have become more common with the trend of denser MIMO deployment, thereby enabling the scheduling operation to yield high utility, low latency, and good channel adaptivity simultaneously. Specifically, we consider an efficient online fine-tuning mechanism to reduce the adaptation overhead. We design a NEA-GNN model that leverages the structure of MIMO network topology, ensuring its computational flow aligns with the PF scheduling criteria. The NEA-GNN model employs an attention-based message passing mechanism that incorporates both the spatial inter-user interference pattern and the user's average rate to achieve high throughput and transmission fairness at the same time. Both an ablation test over simulated channels and an online scheduler implementation over real-world channel measurements have been conducted, demonstrating that the NEA-GNN model achieves better sample efficiency than conventional ML models. Furthermore, it provides the scheduler with higher long-term utilities over non-stationary channels than most existing practical PF schedulers.

A Adaptation vs. Generalization-based ML scheduler

ML adaptation strategies include: (i) *Generalization*—pretraining over diverse channel distributions with minimal adaptation; and (ii) *Adaptation*—pretraining on limited distributions and fine-tuning online for new ones. Generalization methods (e.g., meta-learning Finn et al. [2017], hypernetworks Chauhan et al. [2023], Rezaei-Shoshtari et al. [2023]) require large datasets over high-dimensional channel embeddings, making them costly and less effective, with so far no existing PF scheduling work applying generalization methods to our best knowledge. We adopt the adaptation principle for manageable data requirements and low runtime complexity.

B Simulated Channel Model

We evaluate schedulers under a quasi-stationary wide-sense stationary (WSS) MIMO channel modeled by a Kronecker structure with transmit-side correlation. For each quasi-stationary period $\mathcal{Q}_i$, we draw channel at t^{th} frame as:

$$\mathbf{H}_t = \tilde{\mathbf{H}}_t \Lambda^{1/2} \mathbf{U}^H, \tag{10}$$

where $\tilde{\mathbf{H}}_t \sim \mathcal{CN}(0,1)$, i.i.d.; and $\mathbf{R} = \mathbf{U}\Lambda\mathbf{U}^H$ is the inter-user correlation (Gram) matrix. By construction,

$$\mathbb{E}[\mathbf{H}_t^H \mathbf{H}_t] = \mathbf{R}. \tag{11}$$

To generate diverse user channel distributions with user mobilities over time, we start from a normalized correlation $\tilde{\mathbf{R}}$ with unit diagonal and two uncorrelated user clusters of sizes N_1 and N_2 ($N_1 + N_2 = N$).

$$\tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 \end{bmatrix},\ \mathbf{X}_1 = (1-\beta_1)\cdot \mathbf{I}_{N_1} + \beta_1 \cdot \mathbf{J}_{N_1},\ \mathbf{X}_2 = (1-\beta_2)\cdot \mathbf{I}_{N_2} + \beta_2 \cdot \mathbf{J}_{N_2}. \tag{12}$$

where $\beta_1, \beta_2 \in [0,1]$ control intra-cluster correlation, $\mathbf{I}_n$ is the $n \times n$ identity, and $\mathbf{J}_n$ is the all-ones matrix. Per-cluster average gains are set with:

$$\mathbf{D} = \begin{bmatrix} \alpha_1 \cdot \mathbf{I}_{N_1} & \mathbf{0} \\ \mathbf{0} & \alpha_2 \cdot \mathbf{I}_{N_2} \end{bmatrix}, \quad \mathbf{R} = \mathbf{D}^H \tilde{\mathbf{R}} \mathbf{D}. \tag{13}$$

where $\alpha_1, \alpha_2 \in [0,1]$ scale cluster-wise channel strengths.

Varying $(\alpha_1, \alpha_2, \beta_1, \beta_2)$, along with MIMO size parameter N, M and the scheduling budget K, yields families of $\{\mathbf{R}_i\}$that represent different user clusters, spatial separations, and channel gain imbalances across quasi-stationary periods.

C Real-world Channel Dataset

We evaluate the fine-tuning of our GNN-based scheduler on real-world non-stationary channels using the RENEW dataset Du and Sabharwal [2021], which contains 2.4 GHz OFDM MIMO measurements from multiple user clusters in a campus building, covering both LoS and NLoS conditions. Each cluster spans a 4 m radius and includes measurements at 25 locations over 150 frames. To generate sufficient training data, we approximate the OFDM system as single-carrier by randomly selecting one of the 52 subcarrier frequencies over time, and model each user as randomly moving among the 25 locations within the same cluster. Initially, three users are in one cluster and later move to a different cluster as shown in Figure 6, creating two distinct channel environments. At the environment change (detected by a change detector), the GNN scheduler fine-tunes using $N_f = 5,000$ expert samples. Results show the adaptive GNN-based PF scheduler sustains higher utility over time in this real-world scenario.

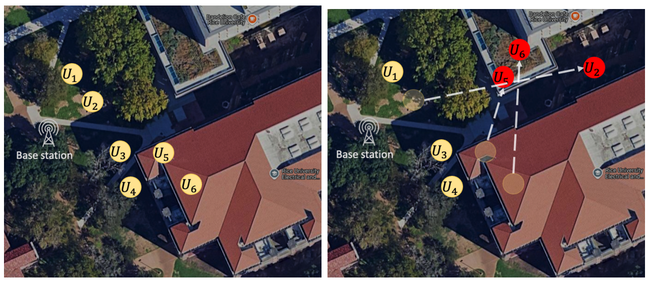


Figure 6: The user locations of the RENEW dataset with realistic MIMO channel measurements.

Latency Comparison between the Expert and ML-based Scheduler Even though the near-optimal expert can achieve good utilities as shown in both Figure 4 and 5, it suffers from the fact that it requires enumerated search in different user selection for the scheduling decision in a single time frame Zhu et al. [2021], which is highly expensive in time. Hence, the expert would incur high scheduling latency, especially with larger MIMO network scales, as shown in Table 1, which makes it impractical to be directly implemented in MIMO systems.

Scheduler	Real Chn. (N=6)	Sim. Chn. (N=20)	Sim. Chn. (N=40)
Expert	0.49ms	4.01ms	11.94ms
GNN-based	0.42ms	0.67ms	0.79ms

Table 1: Average latency of the near-optimal expert compared to GNN-based scheduler.

References

Qing An, Santiago Segarra, Chris Dick, Ashutosh Sabharwal, and Rahman Doost-Mohammady. A deep reinforcement learning-based resource scheduler for massive mimo networks. *IEEE Transactions on Machine Learning in Communications and Networking*, 1:242–257, 2023. doi: 10.1109/TMLCN.2023.3313988.

Michele Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. prentice Hall Englewood Cliffs, 1993.

Mouncef Benmimoune, Elmahdi Driouch, Wessam Ajib, and Daniel Massicotte. Joint transmit antenna selection and user scheduling for massive mimo systems. In *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 381–386, 2015. doi: 10.1109/WCNC.2015.7127500.

Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A Clifton. A brief review of hypernetworks in deep learning. *arXiv preprint arXiv:2306.06955*, 2023.

Cheng-Ming Chen, Qing Wang, Abdo Gaber, Andrea P Guevara, and Sofie Pollin. User scheduling and antenna topology in dense massive mimo networks: An experimental study. *IEEE Transactions on Wireless Communications*, 19(9):6210–6223, 2020.

Yongce Chen, Yubo Wu, Y. Thomas Hou, and Wenjing Lou. mcore: Achieving sub-millisecond scheduling for 5g mu-mimo systems. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021. doi: 10.1109/INFOCOM42981.2021.9488684.

Hewon Cho, Sudarshan Mukherjee, Dongsun Kim, Taegyun Noh, and Jemin Lee. Facing to wireless network densification in 6g: Challenges and opportunities. *ICT Express*, 9(3):517–524, 2023.

Ioan-Sorin Comșa, Sijing Zhang, Mehmet Aydin, Pierre Kuonen, Ramona Trestian, and Gheorghiță Ghinea. A comparison of reinforcement learning algorithms in fairness-oriented ofdma schedulers. *Information*, 10(10): 315, 2019.

Wei Cui, Kaiming Shen, and Wei Yu. Spatial deep learning for wireless scheduling. *ieee journal on selected areas in communications*, 37(6):1248–1261, 2019.

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*, pages 1–1, 2021. doi: 10.1109/TVT.2021.3131606.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Suk-Bok Lee, Ioannis Pefkianakis, Sayantan Choudhury, Shugong Xu, and Songwu Lu. Exploiting spatial, frequency, and multiuser diversity in 3gpp lte cellular networks. *IEEE Transactions on Mobile Computing*, 11 (11):1652–1665, 2012. doi: 10.1109/TMC.2011.206.

Yuhang Li, Yang Lu, Bo Ai, Octavia A Dobre, Zhiguo Ding, and Dusit Niyato. Gnn-based beamforming for sum-rate maximization in mu-miso networks. *IEEE Transactions on Wireless Communications*, 23(8): 9251–9264, 2024.

Lingjia Liu, Young-Han Nam, and Jianzhong Zhang. Proportional fair scheduling for multi-cell multi-user mimo systems. In *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, 2010. doi: 10.1109/CISS.2010.5464834.

Shengjie Liu and Chenyang Yang. Learning user scheduling and hybrid precoding with sequential graph neural network. In *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2024.

Juha Meinilä, Pekka Kyösti, Tommi Jämsä, and Lassi Hentilä. Winner ii channel models. *Radio Technologies and Concepts for IMT-Advanced*, pages 39–92, 2009.

Xiaoxu Meng, Yongyu Chang, Yafeng Wang, and Jingzhou Wu. Multi-user grouping based scheduling algorithm in massive mimo uplink networks. In *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pages 409–413, 2018. doi: 10.1109/CompComm.2018.8780961.

C. Oestges. Validity of the kronecker model for mimo correlated channels. In *2006 IEEE 63rd Vehicular Technology Conference*, volume 6, pages 2818–2822, 2006. doi: 10.1109/VETECS.2006.1683382.

Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Narayan Prasad, Honghai Zhang, Hao Zhu, and Sampath Rangarajan. Multi-user mimo scheduling in the fourth generation cellular uplink. *IEEE transactions on wireless communications*, 12(9):4272–4285, 2013.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2022.

Sahand Rezaei-Shoshtari, Charlotte Morissette, Francois R Hogan, Gregory Dudek, and David Meger. Hypernetworks for zero-shot transfer in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9579–9587, 2023.

Yifei Shen, Jun Zhang, S. H. Song, and Khaled B. Letaief. Graph neural networks for wireless communications: From theory to practice. *IEEE Transactions on Wireless Communications*, 22(5):3554–3569, 2023. doi: 10.1109/TWC.2022.3219840.

Hong Yang. User scheduling in massive mimo. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2018.

Taesang Yoo and A. Goldsmith. On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming. *IEEE Journal on Selected Areas in Communications*, 24(3):528–541, 2006. doi: 10.1109/JSAC.2005.862421.

Zhongze Zhang, Tao Jiang, and Wei Yu. User scheduling using graph neural networks for reconfigurable intelligent surface assisted multiuser downlink communications. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8892–8896, 2022. doi: 10.1109/ICASSP43922.2022.9746441.

Zhongyuan Zhao, Gunjan Verma, Chirag Rao, Ananthram Swami, and Santiago Segarra. Link scheduling using graph neural networks. *IEEE Transactions on Wireless Communications*, 22(6):3997–4012, 2022.

Yuan-Xin Zhu, Do-Yup Kim, and Jang-Won Lee. Joint antenna and user scheduling in the massive mimo system over time-varying fading channels. *IEEE Access*, 9:92431–92445, 2021. doi: 10.1109/ACCESS.2021.3092754.