Deep Reinforcement Learning-based User Scheduling Methods with Low-Complexity Beamforming for Massive MU-MIMO Systems

Yue Zhu, Shuang Li, Longxiang Guo, Wei Ge, and Li Wei

***Abstract*—As massive MU-MIMO systems scale, managing high computational demands and ensuring fairness among users become crucial challenges. Traditional scheduling methods often fall short in dynamically adapting to changing environments and balancing multi-dimensional performance metrics. To address these challenges, we enhance the advantage actor-critic (A2C) framework by integrating convolutional neural networks (CNNs) and Transformers for massive MU-MIMO systems. The CNN components specialize in extracting localized channel-state features, while the Transformers dynamically model inter-user dependencies through attention mechanisms. Specifically, we innovatively embed convolutional layers within the Transformer encoder and employ an auto-regressive decoder, reformulating the user group selection as a sequential decision-making process based on conditional probabilities. This represents the first application of a hybrid CNN-Transformer architecture for discrete scheduling decisions in MU-MIMO systems. To ensure balanced performance, we introduce a multi-metric reward function that incorporates multiple metrics rather than a single performance indicator. Our reward is calculated as the product of the selected user's spectral efficiency (SE) and the Jain fairness index (JFI) during scheduling. Simulations demonstrate the reward function's effectiveness in achieving both high SE and fair resource distribution. We further enhance reward convergence speed through an improved policy network that boosts user scheduling performance while accelerating reward convergence. The proposed model stabilizes reward curves faster than existing frameworks, enabling quicker convergence on optimal strategies and reducing training time, thereby enhancing the framework's suitability for dynamic MU-MIMO scenarios. Additionally, integrating digital and analog maximal ratio combining (MRC) and zero-forcing (ZF) beamforming techniques offers practical, scalable solutions tailored to future massive MU-MIMO systems.**

***Index Terms*—Advantage actor-critic, conformer, low-complexity beamforming, massive MU-MIMO, user scheduling.**

I. INTRODUCTION

MULTI-USER multiple input multiple output (MU-MIMO) systems can serve multiple users simultaneously using the same time-frequency resources through various beamforming techniques, greatly enhancing the spectral efficiency (SE) of wireless communication. To meet the growing demand for higher data rates, emerging MIMO systems are expected to use a large number of antennas, known as massive MU-MIMO [1]. Ideally, as the number of BS antennas increases, the system's capacity expands, allowing it to serve more users and leading to a configuration that is large in terms of both users and antennas [2]. Given the high computational demands due to the large number of antennas and users, these systems are likely to implement low-complexity beamforming (BF) techniques [3]. Furthermore, by implementing a carefully designed scheduling scheme, massive MU-MIMO systems can significantly improve performance through more efficient spectrum use and better adaptation to the wireless propagation environment. Additionally, user scheduling plays a key role in the fair distribution of resources among users. It ensures that all users receive an appropriate share of available resources, avoiding a situation where a few dominate the system and others face connectivity issues. This balanced resource allocation is essential for preserving a positive user experience within the massive MU-MIMO system [4]. Thus, effective user scheduling is essential for optimizing resource allocation, improving efficiency in resource utilization, and guaranteeing fairness among users.

Scheduling methods are fundamental to wireless communications and have been extensively researched. Traditional methods, including the Greedy method, round-robin (RR), and proportional fairness (PF), each have their characteristics. The Greedy scheduler [5], [6] boosts system throughput by prioritizing users with the best channel conditions, but this often compromises fairness by favoring users nearer the base station (BS) with optimal channel conditions. The RR scheduler [7] provides fairness and simplicity by cycling through users but results in lower throughput. The PF scheduler [8] balances throughput and fairness by considering both current user throughput and overall fairness, improving the chances of scheduling users who are less frequently selected.

K. Ko [9] presents a multiuser scheduling strategy with a focus on low computational complexity and near-optimal throughput. Despite its advantages, their method, which uses chordal distance for user selection, does not address user fairness, risking uneven service distribution. Similarly, [10] proposes the semi-orthogonal user scheduling (SUS) strategy, which improves efficiency by selecting users with quasi-orthogonal channels but falls short in terms of fairness, potentially leading to imbalanced resource access. The modified

Manuscript received November 20, 2024; revised May 14, 2025; approved for publication by Kim, Hyoil, Division 3 Editor, July 12, 2025.

The work of S. Li was supported by the Fundamental Research Funds for the Central Universities (XK2050021008).

The authors are with the College of Underwater Acoustic Engineering, Harbin Engineering University, Harbin 150001, China. email: {yuezhu, shuangli, gewei, weiliocean}@hrbeu.edu.cn, heu503@126.com.

S. Li. is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2025.000056

1229-2370/25/$10.00 © 2025 KICS

SUS strategy (SUS-M) proposed by [11] aims to address fairness by grouping users and considering individual rates. Nevertheless, these scheduling strategies are heuristic and may require many iterations with increasing users, often failing to find a global optimum or fully address all performance metrics.

Furthermore, these scheduling methods do not adapt well to complex wireless environments. Recent studies highlight that 6G+ networks must support a wide range of user needs and experiences throughout the scheduling cycle. This calls for the development of more adaptive scheduling approaches to boost system throughput and meet diverse user experience requirements.

In recent years, deep learning (DL) have shown promising results in wireless resource management, with pioneering works like [12] and [13] employing neural networks (NNs) to optimize user scheduling decisions. These approaches typically formulate scheduling as a combinatorial optimization problem, where NNs are trained to directly map channel state information (CSI) to optimal user subsets. However, this paradigm faces dual challenges of scalability and practicality. The inherent NP-hard nature of subset selection over M users creates combinatorial complexity that scales exponentially as $\mathcal{O}(2^M)$, forcing networks to approximate solutions from intractably large search spaces. Simultaneously, the reliance on high-quality training labels derived from exhaustive search or expert heuristics becomes increasingly impractical in dynamic environments, where channel non-stationarity induces distributional shifts between offline training and online deployment phases. This limitation is particularly pronounced in massive MIMO systems, where rapid channel variations caused by user mobility demand real-time adaptability that conventional supervised learning frameworks struggle to provide.

Reinforcement learning (RL) is increasingly utilized for complex decision-making problems due to its adaptability. It learns from previous experiences to determine the best actions and develop an optimal policy for resolving tasks [14]. RL training involves a reward function that guides actions to achieve the most efficient strategy in any state and maximize cumulative rewards over time.

User scheduling in massive MU-MIMO systems can be represented as a Markov decision process (MDP), where both the observation and action spaces are governed by clearly defined reward functions. RL agents can then be developed to interact with the wireless environment, aiming to find the optimal solution to the MDP.

To tackle these scheduling challenges, [15] provides a comprehensive review of both traditional and RL methods used in various wireless network scheduling schemes, outlining the advantages and performance improvements brought by RL. In addition, [16] frames user scheduling in massive MU-MIMO systems as a MDP and solves the problem using a Q-learning algorithm. Their approach, which includes a multi-agent system and an enhanced action selection policy, effectively promotes convergence towards optimal solutions. However, due to the high-dimensional space involved in observing and selecting users in massive MU-MIMO systems, the Q-learning process becomes extremely complex and difficult to converge.

To address the limitations of traditional RL in high-dimensional state spaces, DL has been integrated to extract and process complex input data, enhancing RL's effectiveness in these environments. Recent applications of deep reinforcement learning (DRL) in wireless networks include [17], which formulates contention window (CW) design for random-access networks as an RL problem. Using deep Q-learning (DQN), it selects the optimal CW values from local observations, achieving near-optimal performance and surpassing other learning-based and rule-based methods. However, DQN faces challenges with large discrete action spaces, known as action dimensionality disaster.

For continuous action spaces, [18] employs deep deterministic policy gradient (DDPG) to address scheduling in massive MU-MIMO and [19] models the power control problem as a MDP and proposes a DDPG-based algorithm to solve the faced problem of exponential growth of neurons in the output layer. Despite its potential, DDPG is sensitive to hyperparameter tuning and finds continuous actions harder to learn, leading to convergence issues.

To mitigate these problems, [20] utilizes multiple agents for resource allocation, reducing action space dimensionality but introducing communication losses between agents. The massive MU-MIMO user scheduling problem can be abstracted as a combinatorial optimization (CO) problem. For instance, [21] uses Transformers, leveraging self-attention mechanisms to handle sequence data, to solve the traveling salesman problem (TSP). Similarly, [22] proposes pointer network (PN) [23] for RL, transforming user combination into a sequence selection problem and effectively reducing action space dimensions. However, these methods typically focus on single-step decisions and throughput, lacking the iterative nature of RL.

The advantage actor-critic (A2C) framework excels in massive MU-MIMO systems due to its focus on long-term optimization, efficiency, and flexibility. It not only considers immediate rewards but also optimizes long-term gains, resulting in robust scheduling policies that adapt to varying network loads and user demands. Its gradient-based learning approach facilitates efficient updates of both the strategy and value function, balancing exploration and exploitation to speed up learning and enhance stability. Moreover, the AC framework is highly adaptable to dynamic environments, adjusting its strategy based on real-time feedback to optimize user scheduling decisions in response to changing network conditions.

Although considerable work has been done on user scheduling, there remains a significant gap in research specifically exploring user scheduling approaches that incorporate low-complexity beamforming methods. Maximal ratio combining (MRC) is widely regarded as an ideal candidate for massive MU-MIMO systems due to its simplicity. Additionally, analog MRC [3], the simplest form of beamforming, which only utilizes phase shifters, is also considered in this work.

Motivated by these considerations, we employ the A2C model for user scheduling. The advantage function enhances the algorithm's ability to learn both the strategy and value functions effectively, thereby improving performance and stability. When selecting users, it is crucial to consider not only

individual channel characteristics but also the correlations between users. To address this, we use a transformer-based network for the actor, leveraging its attention mechanism to capture user correlations. Additionally, a convolutional module is incorporated in the encoder to efficiently process channel matrix information and extract channel characteristics, as demonstrated in [12]. We also modify the decoder to operate in auto-regressive mode, facilitating probability-based user-ordered output.

In our wireless communication environment, we integrate low-complexity beamforming techniques. These techniques offer shorter computation times, enhance real-time system performance, reduce latency, and are easier to implement on existing hardware without extensive upgrades. We also design a reward function based on key communication performance indicators to guide user scheduling. This approach ensures not only a reasonable degree of SE but also fairness, thus maintaining high communication quality and preventing the degradation of user experience due to resource imbalances.

The novelty of our work lies in the following aspects:

- **Enhanced Policy Network**: We pioneer the integration of CNN and Transformer architectures into the A2C framework for massive MU-MIMO scheduling. The CNN layers extract spatially-correlated CSI through localized filtering operations, while Transformer attention mechanisms model user dependency patterns. Our architectural innovation embeds convolutional operations within the Transformer encoder and implements an auto-regressive decoder, transforming combinatorial user group selection into a sequential decision process. This represents the first successful application of CNN-Transformer fusion for discrete scheduling outputs in MU-MIMO systems.
- **Multi-Metric Reward Function**: Unlike traditional algorithms that optimize for a single performance metric, our approach combines multiple indicators in the reward function. We use the product of the selected user's SE and Jain fairness index (JFI) at timesteps as the environmental reward. This method ensures that as SE improves, the JFI may initially decrease but will eventually converge, balancing SE and fairness. Simulation results validate the effectiveness of this reward function in achieving a fair distribution of resources while optimizing SE.
- **Accelerated Reward Convergence**: Our enhanced policy network not only improves user scheduling performance in massive MU-MIMO systems but also achieves significantly faster convergence in reward accumulation compared to other frameworks. The CNN-Transformer network effectively balances exploration and exploitation in complex channel environments. Simulation results show that the improved reward curve reaches stability more quickly, enabling faster convergence on optimal scheduling strategies and reducing training time. This accelerated convergence demonstrates the model's efficiency in learning effective scheduling policies, making it highly suitable for dynamic MU-MIMO environment.
- **Beamforming Integration**: We incorporate both digital and analog MRC along with zero-forcing (ZF) beamforming techniques into our scheduling framework. These low-complexity beamforming methods provide practical, scalable solutions for massive MU-MIMO systems, ensuring that the proposed scheduler can be readily applied to real-world scenarios.

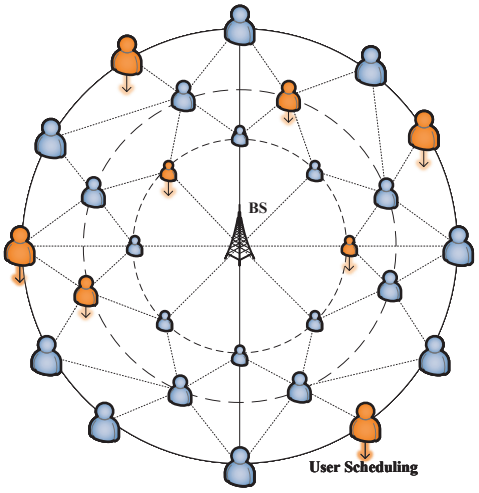


Fig. 1. Massive MU-MIMO communication system with user scheduling.

The remainder of the paper is organized as follows. Section II describes the system model and formulates the user scheduling problem. Section III presents the proposed CNN-Transformer enhanced A2C framework, including the policy network network, the multi-metric reward design, and the beamforming integration. Section IV provides simulation results and performance comparisons. Finally, Section V concludes the paper.

II. System Model and Problem Statement

In this work, we consider an uplink single-cell MU-MIMO system. The BS is equipped with M antennas and there are N users, each equipped with a single antenna, assuming M is less than N. Ideally, with the growth in the number of BS's antennas, more users can be served, resulting in a system that expands significantly in both users and antennas. Thus, the system need to dynamically select K users for simultaneous communication, where the value of K is typically not fixed, with $K \leq \min(N, M)$.

We assumed that the uplink CSI is available at the BS and it is used as input of the model. We consider a narrow band flat fading channel, where the $M \times 1$ channel vector for user i can be denoted by $\mathbf{h}_i$ and the received signal at BS is given by

$$\mathbf{y} = \sqrt{p}\sum_{i=1}^{K} \mathbf{h}_i x_i + \mathbf{n} = \sqrt{p}\mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{y} \in \mathbb{C}^{M\times 1}$ is the received signal vector at the BS, p is the average transmit power of the user, $\mathbf{H} = \left[\mathbf{h}_1, \cdots, \mathbf{h}_N\right]$, $\mathbf{x} = \left[x_1, \cdots, x_N\right]^T$, $\mathbf{h}_i \in \mathbb{C}^{M\times 1}$ is the channel matrix between the BS and user i, and x_i is the transmitted symbols vector by user i. Additionally, $\mathbf{n} \in \mathbb{C}^{M\times 1}$ is the receiver complex noise vector with a distribution, $\mathbf{n} \sim \mathcal{CN}(0, \sigma^2\mathbf{I})$.

Without loss of generality, we assume σ^2 to be 1, and $\mathbf{I}$ is the identity matrix.

A. Channel Modeling

We consider two channel models: Rayleigh fading channels and ray-based channels with doppler effect. Rayleigh channels effectively model environments dominated by scattering without a direct path, common in urban and indoor settings. Ray-based channels with doppler effect provide a more realistic representation of environments with both line-of-sight (LoS) and non-LoS paths, especially in high-mobility scenarios. Analyzing both models helps develop robust communication systems for diverse conditions.

1) Rayleigh Channel Model: In this scenario, the $M \times 1$ channel vector for user k can be written as

$$\mathbf{h}_k = \sqrt{\beta_k}\mathbf{u}_k, \tag{2}$$

where the entries of $\mathbf{u}_k$ are independent and identically distributed (i.i.d) Rayleigh fading variables, $\mathbf{u}_k \sim \mathcal{N}(0, \mathbf{I})$. The large-scale fading coefficients β_k are presented as follows:

$$\beta_k = A\zeta_k \left(\frac{d_0}{d_k}\right)^{\gamma}, \tag{3}$$

where A is a unit-less constant indicating the geometric attenuation at the reference distance d_0, d_k is the distance between the kth user and the base station, and γ is the pathloss attenuation exponent. ζ_k is a log-normal random variable, $10\log_{10}\zeta_k \sim \mathcal{N}(0, \sigma_{\text{sh}}^2)$, to model the effect of shadowing between the kth user and the BS.

2) Ray-Based Channels with Doppler Effect: In this scenario, we study scheduling in a high-speed mobile environment [24]. We also assume that the BS has predicted CSI, and that the propagation channel is a millimeter wave channel. Due to high degree-of-freedom space path loss, we can represent the millimeter wave channel using a geometric channel model with limited L scatterers for each user. In an environment where users are moving at a speed of v, since the doppler shift channel changes over time, and assuming an equal number of scatterers per user, the fading channel vector between the BS and the moving user k can be expressed as

$$\mathbf{h}_k(t) = \alpha \sum_{l=1}^{L} \beta_{kl} e^{j2\pi f_D t} \boldsymbol{a}_k(\theta_l) \in \mathbb{C}^{N\times 1}, \tag{4}$$

where $\alpha = \sqrt{M/L}$, β_{kl} is a channel gain of the lth path of the kth user which obeying zero-mean, unit-variance complex Gaussian distributed; $f_D = f_m \cos(\theta_l^r)$ is the doppler shift with maximum doppler frequency $f_m = v/\lambda$ and the angle-of-arrival (AOA) θ_l^r of lth path at the user. The antenna elements at the BS are placed by the inter-element distance d, which is typically half wavelength. $\boldsymbol{a}_k(\theta_l)$ represents the normalized steering vector of the uniform linear array (ULA) at angle-of-departure (AOD) θ_l of the kth user. It is denoted as:

$$\boldsymbol{a}_k(\theta_l) = \frac{1}{N}\left[1, e^{-j\frac{2\pi d}{\lambda}\sin(\theta_l)}, \cdots, e^{-j\frac{2\pi(N-1)d}{\lambda}\sin(\theta_l)}\right]^T. \tag{5}$$

B. Beamforming Techniques

Given the channel matrix $\mathbf{H}$, the BS calculates the ZF beamforming vector for user k is:

$$\mathbf{A} = \mathbf{H}^{\mathbf{H}}\left(\mathbf{H}\mathbf{H}^{\mathbf{H}}\right)^{-1}. \tag{6}$$

ZF beamforming aims to completely nullify the interference between different users achieved by projecting the transmitted signal into a subspace where interference is minimized.

The beamforming vector for user k (i.e., the kth column of $\mathbf{A}$) is:

$$\mathbf{a}_k^{\text{ZF}} = \mathbf{H}\left(\mathbf{H}^{\mathbf{H}}\mathbf{H}\right)^{-1}\mathbf{e}_k, \tag{7}$$

where $\mathbf{e}_k$ is a unit vector with the kth element being 1 and all other elements being 0. This design ensures that the beamforming vector for user k is orthogonal to the channel vectors of all other users, i.e., $\mathbf{a}_k^{\text{H}}\mathbf{h}_i = 0$ for $i \neq k$, thus eliminating interference. MRC is another beamforming technique aimed at maximizing the signal-to-interference-plus-noise ratio (SINR) of a desired signal in multiplexed environments. In MRC, the receiver aggregates signals from multiple antennas using a weighted sum, where the weights are dynamically adjusted according to the prevailing channel conditions. Although MRC does not fully eliminate multi-user interference, it significantly enhances the desired signal by maximizing its SNR. Due to its low computational complexity, MRC effectively mitigates signal attenuation, reduces noise, and improves signal quality, particularly in scenarios with weak signal strength. Integrating MRC beamforming techniques into a reinforcement learning framework is anticipated to improve overall system performance. MRC can be implemented in both digital and analog forms:

In digital MRC, the beamforming vector for user k is:

$$\mathbf{a}_k^{\text{digital}} = \mathbf{h}_k. \tag{8}$$

In analog MRC, the beamforming vector for user k is:

$$\mathbf{a}_k^{\text{analog}} = \exp(j\angle\mathbf{h}_k), \tag{9}$$

where $\angle\mathbf{h}_k$ represents the vector of angles of each element of $\mathbf{h}_k$.

The SINR for user k with general beamforming is given by:

$$\text{SINR}_k = \frac{p\left|\mathbf{a}_k^{\mathbf{H}}\mathbf{h}_k\right|^2}{p\sum_{i=1,i\neq k}^{K}\left|\mathbf{a}_k^{\mathbf{H}}\mathbf{h}_i\right|^2 + \left\|\mathbf{a}_k^{\mathbf{H}}\right\|^2}, \tag{10}$$

where $\mathbf{a}_k$ is the beamforming vector for user k, and $\mathbf{h}_i$ represents the channel vectors. This SINR formula accounts for the signal power, interference from other users, and noise.

When using the ZF algorithm, the SINR for user k simplifies to:

$$\text{SINR}_k^{ZF} = \frac{p}{\left[\left(\mathbf{H}^{\mathbf{H}}\mathbf{H}\right)^{-1}\right]_{kk}}. \tag{11}$$

C. Optimization Objective

Based on the above, we can get the instant SE of user k is:

$$R_k = \log_2(1 + \text{SINR}_k). \tag{12}$$

Then, we can derive the SE of the whole system as:

$$R_{sum} = \sum_{k\in[\mathcal{K}]} \log_2(1+\mathrm{SINR}_k). \tag{13}$$

User scheduling involves the selection of the optimal user set $\mathcal{K}$ from N. Our goal is to formulate a scheduler that can take into account both total system SE as described in (14) and widely-used metric for fairness Jain's fairness index [25] as described in (16).

$$\mathrm{SE}_t = \sum_{k\in\mathcal{K}_t} R_k, \tag{14}$$

$$j_t^k = \frac{\sum_{i=1}^{t} R_k(i)}{T}, \quad k \in \{1,\cdots,N\}, \tag{15}$$

$$fair_t = \frac{\left(\sum_{k=1}^{N} j_t^k\right)^2}{N\left(\sum_{k=1}^{N}(j_t^k)^2\right)}, \tag{16}$$

where SE_t represents the SE of the timestep t of scheduled users, T is the total time step of the schedule, j_t^k represents the average of cumulative throughput allocated to user k up to time t, $fair_t$ represents fairness index.

We wish to dynamically adjust these two metrics by multiplying SE with fair_t. It decreases when SE increases but certain users are dispatched very little, thus inducing a reward function to improve fairness, and also promotes SE when fairness is good but SE is not.

III. Deep Reinforcement Learning based Scheduler

In order to be able to consider SE and fairness and minimize the runtime, we propose an RL-based user scheduling scheme for MU-MIMO systems. In this paper, we introduce the CNN-Transformer into policy networks to address the user selection challenge under communication systems. We model this scheduling problem using an MDP and translate the optimization goals into states, rewards, and actions defined by the MDP. By intensively learning the continuous interaction between the agent and the MU-MIMO environment, we generate a better scheduling policy to optimize the gains of the user selection process.

A. A2C MU-MIMO User Scheduling Scheme

A2C [26] is a DRL model combining a policy network π_θ and an value network V_ω, where θ and ω are parametres of policy and value network. The overall framework is illustrated in Fig. 2, the policy network acts as an "actor," interacting with the environment and learning an improving strategy via the policy gradient, which is guided by the value function. The value network serves as a "critic," assessing the actor's performance and influencing its subsequent actions. The objective of the actor network is to maximize the cumulative reward R, while the critic network aims to evaluate the state value function. Through (17) and (18), the two networks collaborate to determine the parameters that optimize the cumulative reward.

$$\omega^* = \arg\min_{\omega}(r+\gamma V_\omega(s') - V_\omega(s)), \tag{17}$$

$$\delta_w = r+\gamma V_w(s') - V_w(s), \tag{18}$$

$$\theta^* = \arg\max_{\theta} \mathbb{E}_{\pi_\theta}[\log \pi_\theta(s,a)\delta_w], \tag{19}$$

where r represents immediate reward of current state s, γ is the discount factor, by setting this value, we can allow the model to consider rewards for multiple steps afterward. s' is the next state and δ_w is an unbiased estimate of the dominance function.

In previous articles, only one-shot decision has been considered, where scheduling trajectory ends when the user sequence is selected once, so $V_\omega(s') = 0$. However, we assume that the user reality is in constant motion and include a component for the state of the environment that affected by the scheduling operation, so that the scheduling program can be used to select the appropriate group of users in successive time steps.

B. A2C Model Definition and Key Components

In this section, we apply the A2C framework to formulate a MDP model to solve the user scheduling problem in massive MU-MIMO networks. According to the optimization target describedthe in Section II, state space, action space and reward function in MDP can be formulated as follows:

State. The state space $\mathcal{S}$ includes important information about the environment. We denote the state at timestep t as $s_t \in \mathcal{S}$, which consists of the channel matrix $\mathbf{H} \in \mathbb{C}^{M\times N}$. We split the real and imaginary parts of the channel matrix, resulting in $\mathbf{H} \in \mathbb{C}^{M\times N\times 2}$. Additionally, we record the average scheduling rate for all users until current time step $\boldsymbol{J}_t = [j_t^1, j_t^2, \cdots, j_t^N]$as part of the state. Based on these components, the system state at timestep t is defined as:

$$s_t = [\mathbf{H}_t, \boldsymbol{J}_t]. \tag{20}$$

The channel translation model integrates two key components: user mobility patterns and channel state dynamics. The user position $\mathbf{x}_u(t)$ evolves according to a constrained Markov process:

$$\mathbf{x}_u(t) = \mathbf{x}_u(t-1) + \Delta\mathbf{x}_u(t), \tag{21}$$

where the stochastic displacement $\Delta\mathbf{x}_u(t)$ in polar coordinates is defined by:

$$r \sim \sqrt{\mathrm{Uniform}(d_0^2, r_0^2)}, \tag{22}$$

$$\sigma \sim \mathrm{Uniform}(0, 2\pi), \tag{23}$$

with spatial constraint $\|\mathbf{x}_u(t)\| \in [d_0, r_0]$, where r_0 denotes the BS coverage radius. The channel matrix $\mathbf{H}_t$ is determined by the nonlinear mapping:

$$\mathbf{H}_t = f_{\mathrm{ch}}\left(\mathbf{H}_{t-1},\, \mathbf{x}_u(t),\, \mathbf{x}_u(t-1),\, v_u(t)\right), \tag{24}$$

where $v_u(t)$ represents the instantaneous user velocity. The channel transition function $f_{\mathrm{ch}}(\cdot)$ incorporates the following mechanisms:

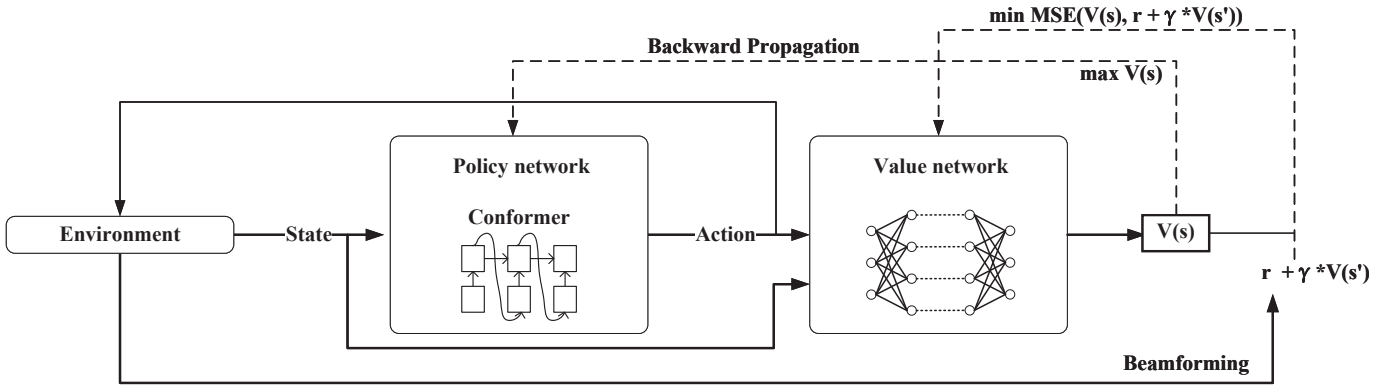


Fig. 2. Advantage actor-critic framework.

- **Path Loss and Shadowing:**

$$\beta_k = A\zeta_k \left(\frac{d_0}{\|\mathbf{x}_u(t)\|} \right)^{\gamma}, \tag{25}$$

where A denotes the reference path gain at distance d_0, ζ_k represents shadowing effects, and γ is the path-loss exponent.

- **Doppler Effect:**

$$f_m = \frac{v_u(t)}{\lambda}, \quad f_D = f_m \cos\theta_{\mathrm{AOA}}, \tag{26}$$

where λ is the carrier wavelength and θ_{AOA} denotes the angle of arrival.

This integrated modeling approach provides a rigorous foundation for analyzing RL-based scheduling strategies in different channels.

Action. In RL learning, discrete and multidiscrete are both types of action spaces that define the nature of actions an agent can take. A discrete action space is a single, one-dimensional space where the agent can choose from a finite number of actions. This is used when the action can be described by a single decision from a set of mutually exclusive options. Multi-discrete action space is a space where each dimension is a separate discrete action space. The agent can choose a specific action from each dimension independently. This is used when the agent's action involves making multiple independent choices, each from its own discrete set of options. In our problem, The action space $\mathcal{A}$ is defined as an multi-discrete, comparing with discrete, multi-discrete spaces is simpler to implement and more suitable for multi-user scheduling problems. With the idea of one-hot coding, each user is treated as a dimension. We denote the action of step t as $a_t \in \mathcal{A}$, which represents the set of selected users.

Reward. Our reward function is designed to balance SE and user fairness while enforcing practical scheduling constraints. The composite reward consists of three components:

$$R_t = \underbrace{\mathrm{SE}_t \cdot \mathrm{fair}_t}_{\text{Main objective}} + \underbrace{\mathrm{fairpenalty}_t}_{\text{Immediate constraint}} + \underbrace{\mathrm{endpenalty}_t}_{\text{Long-term constraint}}, \tag{27}$$

1) Component Analysis: Main objective ensures tradeoffs between instantaneous SE and cumulative fairness. The product formulation inherently penalizes solutions that maximize one metric at the extreme expense of the other.

$\mathrm{fairpenalty}_t$ enforces the BS's maximum simultaneous user capacity M:

$$\mathrm{fairpenalty}_t = \begin{cases} -1.0, & \text{if } K_t > M \\ 0, & \text{otherwise} \end{cases}. \tag{28}$$

The penalty magnitude was set empirically to 20% of the average per-step reward, providing sufficient deterrence without dominating the learning signal. $\mathrm{endpenalty}_t$ addresses persistent constraint violations over time:

$$\mathrm{endpenalty}_t = \begin{cases} -\frac{l_t}{L}, & \text{if } t > \frac{L}{5} \text{ and } K_t > M \\ 0, & \text{otherwise} \end{cases}. \tag{29}$$

where l_t counts violations up to step t, and L is the maximum episode length. The delayed activation (after $L/5$ steps) preserves early exploration capability.

2) Design Rationale: The combined formulation addresses three critical challenges:

- **Short-Term Optimization**: The main term encourages greedy SE-fairness balance.
- **Hardware Constraints**: Fair penalty prevents physical layer infeasibility.
- **Temporal Consistency**: End penalty ensures sustained policy compliance.

3) Ablation Study Validation: The progressive penalty activation presents three learning phases:

- **Exploration Phase**: Unpenalized constraint violations enable discovery of high-SE strategies.
- **Transition Phase**: Gradually increasing penalties shape feasible policies.
- **Exploitation Phase**: Stable constraint compliance with fine-tuned optimization.

As can be seen from Table I, removing both penalties at the same time will greatly increase the violation rate, causing the model to select a large number of users during scheduling, which will degenerate into only focusing on one metric, SE, and fairness will not be a factor affecting rewards, which is

TABLE I
IMPACT OF PENALTY COMPONENTS (70 USERS, 32 ANTENNAS)

Configuration	Violation Rate (%)	Fairness	SE (bps/Hz)	Episode Reward
No Penalty	47.5	0.69	9.31	0.67
Fair Penalty Only	17.0	0.78	7.56	0.93
End Penalty Only	17.2	0.73	7.62	0.79
Full Reward	**12.4**	**0.94**	**8.20**	**0.85**

contrary to the actual scheduling logic, and will also lead to the user being in a poorer channel state not being able to be scheduled. Combining these two penalties reduces the violation rate and is logical, as the number of scheduling grows, the model achieves scheduling that does not exceed the constraints in the mid-scheduling phase, when the number of users is roughly the same each time they are scheduled, and the order of magnitude of the SEs obtained each time does not change too much, the model gradually begins to pursue the maximization of the fairness, and in the convergence phase the two will reach a balance, which will ensure both the each scheduling moment maximizes the SE as much as possible, while at the same time maintaining the long-term fairness in the scheduling cycle.

C. Policy Network And Critic Network

1) Policy Network: The design of policy networks is fundamentally governed by the spatiotemporal characteristics inherent to MU-MIMO scheduling environments. The wireless channel matrix, as a critical environmental state component, exhibits strong spatial correlations arising from the geographical distribution of user equipment and multipath scattering environments. These spatial features can effectively captured through convolutional operations, which have demonstrated feasibility in processing structured channel matrices. Meanwhile, optimal scheduling decisions require modeling long-term temporal dependencies, including historical resource allocation trajectories and dynamic inter-user interference relationships, these capabilities provided by global attention mechanisms.

Conventional neural architectures face inherent limitations in addressing these dual requirements. Such as PN lack the capacity for structured spatial feature extraction from complex channel matrices, CNNs prove inadequate for modeling sequential decision dependencies. Transformers, though powerful in sequence modeling, fail to effectively leverage spatial correlations in channel data. To overcome these limitations, we adopt the Conformer architecture, originally pioneered in speech processing. This hybrid design synergistically integrates convolutional layers for spatial feature extraction with multi-head self-attention (MHSA) mechanisms for temporal dependency modeling, establishing an ideal framework for MU-MIMO scheduling tasks requiring joint spatiotemporal awareness.

The structure of policy network is shown in Fig. 3, where $\mathbf{x}_i$ is the pre-processed channel vector $\mathbf{h}$ of user i, and $\mathbf{z}$ is the pre-processed vector of $\mathbf{J}_t$. Then the encoder maps all user channel vectors to the coding space at once, referring to (30). The Conformer comprises two feed-forward (FFN) modules, with a MHSA module and a convolution module sandwiched between them, creating a macaron structure. We use this structure to train the real and imaginary components of the channel matrix separately. A convolution operation is performed on these two components to generate the channel embedding matrix, which can efficiently represent the channel information. For a given input $\mathbf{x}_i$ to a Conformer block, the corresponding output can be expressed mathematically as follows:

$$\begin{aligned}
\tilde{\mathbf{x}}_i &= \mathbf{x}_i + \frac{1}{2}\mathrm{FFN}(\mathbf{x}_i),\\
\mathbf{x}'_i &= \tilde{\mathbf{x}}_i + \mathrm{MHSA}(\tilde{\mathbf{x}}_i),\\
\mathbf{x}''_i &= \mathbf{x}'_i + \mathrm{Conv}(\mathbf{x}'_i),\\
\mathbf{h}_i^{enc} &= \text{Layernorm}\left(\mathbf{x}''_i + \frac{1}{2}\mathrm{FFN}(\mathbf{x}''_i)\right).
\end{aligned} \tag{30}$$

With this procedure, the network can extract the correlation between users and also extract the local features of the user's channel using the convolution operation.

The decoder outputs one user index at a time. Assuming that the first t users have been decoded and we want to predict the next user, the decoding process of the decoder consists of four timesteps:

$$\begin{aligned}
\mathbf{h}_t^{\mathrm{dec}} &= \mathbf{x}_{i_t} + \mathrm{PE}_t \in \mathbb{R}^d,\\
\mathbf{h}_{t=0}^{\mathrm{dec}} &= \mathbf{h}_{\mathrm{st}}^{\mathrm{dec}} = \mathbf{x}_{\mathrm{mean}} + \mathrm{PE} \in \mathbb{R}^d,
\end{aligned} \tag{31}$$

where $\mathbf{x}_{\mathrm{mean}}$ is the mean of all user channel vectors. Position encoding (PE) is the positional encoding from [27]. We utilize PE to process the user's sequential information. Then, using two self-attention modules, the first aggregates the information of the decoded users. The query is taken from the user t embedding, while the key and value are taken from each user embedding that has been decoded.

$$\begin{aligned}
\hat{\mathbf{h}}_t^{\ell+1} &= \mathrm{softmax}\left(\frac{\mathbf{q}^\ell \mathbf{K}^{\ell^T}}{\sqrt{M}}\right)\mathbf{V}^\ell,\ \ell = 0, \cdots, L^{\mathrm{dec}} - 1,\\
\mathbf{q}^\ell &= \hat{\mathbf{h}}_t^\ell \hat{\mathbf{W}}_q^\ell,\\
\mathbf{K}^\ell &= \hat{\mathbf{H}}_{1,t}^\ell \hat{\mathbf{W}}_K^\ell,\\
\mathbf{V}^\ell &= \hat{\mathbf{H}}_{1,t}^\ell \hat{\mathbf{W}}_V^\ell,\\
\hat{\mathbf{H}}_{1,t}^\ell &= [\hat{\mathbf{h}}_1^\ell, \cdots, \hat{\mathbf{h}}_t^\ell],
\end{aligned} \tag{32}$$

$$\hat{\mathbf{h}}_t^\ell = \begin{cases} \mathbf{h}_t^{\mathrm{dec}} & \text{if } \ell = 0\\ \mathbf{h}_t^{\mathrm{q},\ell} & \text{if } \ell > 0. \end{cases} \tag{33}$$

This is followed by attention on the undecoded user. The query is taken from the previous output embedding, while the key and value are taken from the unselected users' embeddings. A mask $\mathcal{M}_t$ is needed here to determine whether the user has been selected or not.

$$\begin{aligned}\mathbf{h}_t^{\mathrm{q},\ell+1} &= \mathrm{softmax}\left(\frac{\mathbf{q}^\ell \mathbf{K}^{\ell^T}}{\sqrt{M}} \odot \mathcal{M}_t\right)\mathbf{V}^\ell,\ \ell = 0,\cdots,L^{\mathrm{dec}}-1,\\ \mathbf{q}^\ell &= \hat{\mathbf{h}}_t^{\ell+1}\tilde{\mathbf{W}}_q^\ell,\\ \mathbf{K}^\ell &= \mathbf{H}^{\mathrm{enc}}\tilde{\mathbf{W}}_K^\ell,\\ \mathbf{V}^\ell &= \mathbf{H}^{\mathrm{enc}}\tilde{\mathbf{W}}_V^\ell. \end{aligned} \tag{34}$$

The final timestep entails executing a query with single-head attention to obtain the distribution of unselected users. Subsequently, the next user is sampled from it based on the probability distribution. During testing, users are sampled using the maximum probability indexing. Scheduling ends when $\mathbf{x}_{\mathrm{mean}}$ is returned.

$$\begin{aligned} p_t^{\mathrm{dec}} &= \mathrm{softmax}\left(\tanh\left(\frac{\mathbf{q}\mathbf{K}^T}{\sqrt{d}} \odot \mathcal{M}_t\right)\right),\\ \mathbf{q} &= \mathbf{h}_t^{\mathbf{q}}\bar{\mathbf{W}}_q,\\ \mathbf{K} &= \mathbf{H}^{\mathrm{enc}}\bar{\mathbf{W}}_K, \end{aligned} \tag{35}$$

where p_t^{dec} is the probability that the user corresponds. Due to the chain rule of policy network, the joint probability of an action when the state is given can be estimated by the following equation:

$$\pi_\theta(a|s) = \prod_{t=1}^{n} p_t^{dec}\left(i_t \mid i_{t-1}, i_{t-2}, \cdots, i_1, \mathbf{H}\right). \tag{36}$$

Combined with our goal function, the loss function of the policy network is:

$$J(\theta) = -\mathbb{E}_{\pi_\theta}[\log \pi_\theta(s,a)\delta_w]. \tag{37}$$

We approximate the TD error δ_w using an approximate value function V_w with parameter w. The policy gradient can be represented as follows:

$$\nabla_\theta J(\theta) = -\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\delta_w]. \tag{38}$$

Based on the above procedure, we acquire a Conformer-based policy network, as summarized in Algorithm 1, where $\mathbf{E} \in \mathbb{R}^{M\times d}$ denotes the intermediate embedding matrix produced during each layer of the Conformer encoder and the d denotes the embedding dimension. Given the complex-valued channel matrix $\mathbf{H}$ and the state vector $\boldsymbol{J}_t$, the network outputs a user scheduling sequence $\boldsymbol{S} = [i_1, \cdots, i_K]$. The process begins by splitting the channel matrix into its real and imaginary components and projecting them into a high-dimensional embedding space. These embeddings are then refined through multiple Conformer encoder layers, each consisting of FFN, MHSA, and convolutional blocks. After encoder processing, the decoding phase is initialized using the average of the initial embeddings and position encoding. In the autoregressive decoding stage, users are sequentially selected using self-attention over the previously selected users and cross-attention with the encoded channel features, while applying masking to prevent duplicate selections. The decoding continues until the desired number of users is selected or a termination condition is met.

Algorithm 1 : Conformer-based policy network for MU-MIMO scheduling

Input: Channel matrix $\mathbf{H} \in \mathbb{C}^{M\times N}$, joint state vector $\boldsymbol{J}_t$
Output: Selected user sequence $\boldsymbol{S} = [i_1, \cdots, i_K]$

1. Input Preprocessing
Split $\mathbf{H}$ into real/imaginary parts; stack into $\mathbf{X} \in \mathbb{R}^{M\times N\times 2}$
Embed: $\mathbf{E}_0 \leftarrow \mathrm{Linear}(\mathbf{X}) \in \mathbb{R}^{M\times d}$
2. Conformer Encoder
for $l = 1$ **to** L_{enc} **do**
  $\mathbf{E} \leftarrow \mathbf{E}_{l-1} + \frac{1}{2}\,\mathrm{FFN}(\mathbf{E}_{l-1})$
  $\mathbf{E} \leftarrow \mathbf{E} + \mathrm{MHSA}(\mathbf{E})$
  $\mathbf{E} \leftarrow \mathbf{E} + \mathrm{Conv}(\mathbf{E})$
  $\mathbf{E}_l \leftarrow \mathbf{E} + \frac{1}{2}\,\mathrm{FFN}(\mathbf{E})$
end for
$\mathbf{H}^{\mathrm{enc}} \leftarrow \mathrm{LayerNorm}(\mathbf{E}_{L_{\mathrm{enc}}})$
3. Decoder Initialization
Initialize $S \leftarrow [\,]$, $\mathbf{h}_0 \leftarrow \frac{1}{M}\sum_{i=1}^{M}\mathbf{E}_0[i] + \mathrm{PE}(0)$
4. Auto-Regressive Encoder
for $t = 1$ **to** K **do**
  if $t > 1$ **then**
    $\mathbf{h}_{t-1} \leftarrow \mathbf{E}_0[i_{t-1}] + \mathrm{PE}(t-1)$
  end if
  Self-attention over $\{\mathbf{h}_0, \cdots, \mathbf{h}_{t-1}\}$
  Cross-attention (with masking) between $\mathbf{h}_{t-1}$ and $\mathbf{H}^{\mathrm{enc}}$
  Compute logits ℓ_t and $p_t \leftarrow \mathrm{Softmax}(\ell_t)$
  Select $i_t \leftarrow \arg\max_j p_t[j]$ (test) or $\mathrm{Sample}(p_t)$ (train)
  Append i_t to S; **break** if termination token
end for
return $\boldsymbol{S} = [i_1, \cdots, i_K]$

2) Critic Network: The Critic network is a Multi-Layer Perceptron (MLP), which evaluates the value of action taken by the agent in a given state. It comprises several fully-connected layers that extract input features layer by layer, culminating in the desired value. When the policy network converges to the optimal parameter, the loss function of critic network is:

$$J(\omega) = \frac{1}{2}(r + \gamma * V_\omega(s_{t+1}) - V_\omega(s_t))^2, \tag{39}$$

where γ is the discount factor, afterwards the parameter ω can be updated in the following form:

$$\nabla_\omega J(\omega) = -(r + \gamma * V_\omega(s_{t+1}) - V_\omega(s_t))\nabla_\omega V_\omega(s_t). \tag{40}$$

D. Complexity Analysis

Our framework integrates beamforming-aware scheduling with RL, requiring careful analysis of computational complexity across two key components: policy network inference, beamforming computation. Below we provide a detailed breakdown:

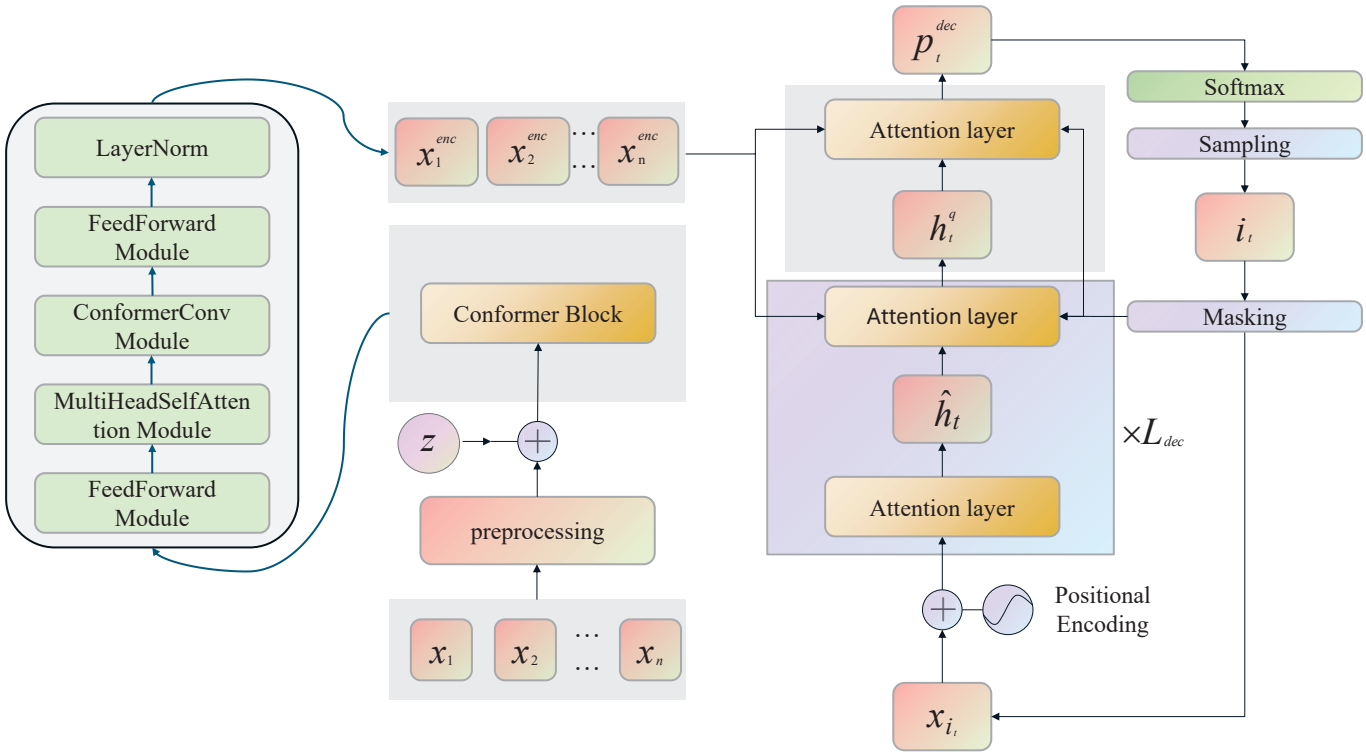


Fig. 3. Policy network that combine Conformer module and auto-regressive module.

1) Policy Network Complexity: The Conformer-based policy network consists of:

- **Convolutional Layers**:

$$O_{\text{conv}} = n \times K^2 \times C_{\text{in}} \times C_{\text{out}}. \tag{41}$$

- **Self-Attention**:

$$O_{\text{attn}} = n^2 \times d. \tag{42}$$

- **Feed-Forward Networks**:

$$O_{\text{ffn}} = n \times d^2. \tag{43}$$

- **Autoregressive Decoder**:

$$O_{\text{dec}} = n \times d. \tag{44}$$

where n is the total number of users to be scheduled in the system, and the total policy complexity per decision is:

$$O_{\text{policy}} = O_{\text{conv}} + O_{\text{attn}} + O_{\text{ffn}} + O_{\text{dec}}. \tag{45}$$

It should be noted that the complexity described above primarily applies during the pre-training phase. During this phase, the model must perform large-scale computations to update the parameters and learn the optimal policy. However, once the model is trained and has converged, the inference time during the actual scheduling process is significantly reduced. This is because, at this point, the model's weights are fixed, and the main computational burden comes from the user input data (such as CSI) and beamforming calculations. Hence, during the inference phase, the time complexity is much lower compared to the pre-training phase.

2) Beamforming Complexity: For ZF, the complexity is dominated by the matrix inversion operation, which requires $O(N^3)$ for an $N \times N$ matrix. This operation becomes computationally expensive as the number of antennas or users increases. In Digital MRC, the complexity is linear in the number of users or antennas, requiring just a simple matrix transpose. The complexity is thus $O(N)$, where N is the number of users or antennas. Analog MRC requires extracting the phase of each user's channel and performing phase matching. This operation is also linear in complexity, resulting in $O(N)$ complexity.

3) Total Complexity: Combining all components yields:

$$O_{\text{total}} = O_{\text{policy}} + O_{\text{beamforming}}. \tag{46}$$

The total complexity reflects both the policy network's complexity and the beamforming computations. However, as mentioned earlier, the computational load during the inference phase is primarily determined by the fixed policy network and the beamforming algorithm in use, which has a significantly reduced time complexity compared to the pre-training phase.

IV. Performance Evaluation

In this section, we evaluate the performance of both traditional and learning-based method in environment. We first introduce the training details, followed by our simulation results.

A. Simulation Setup and Parameters

Our simulation environment models a single-cell massive MIMO system where users are randomly distributed within a

0.1 km radius disk centered at the BS, as illustrated in Fig. 1. Users move at speeds between -5 m and 10 m per timestep, with direction reversal at the cell edge to maintain continuous motion within the coverage area. We consider two channel models: Rayleigh fading and a ray-based model with Doppler effects. Each episode consists of a maximum of 600 timesteps, and the performance is evaluated using the average episode reward:

$$\eta(\theta) = \frac{1}{M}\frac{1}{T}\sum_{i=1}^{T} R_{\text{sum}}(\boldsymbol{H}_i, \mathcal{K}_i), \tag{47}$$

where T is the episode length, and R_{sum} is the sum reward. For the ray-based channel model, we incorporate higher user mobility with single-antenna users uniformly distributed within the cell. Each user's communication environment is modeled using $L = 20$ scatterers, with angles of departure (AOD) and arrival (AOA) drawn from a Laplacian distribution. The mean direction μ_k is uniformly distributed over $[0, 2\pi]$, capturing the angular dispersion due to scattering and random user orientations while accounting for Doppler-induced frequency shifts.

1) Hyperparameter Tuning: The DRL framework is implemented using a Conformer encoder [28] and an autoregressive decoder [21] for sequential decision making. The critic network employs a three-layer MLP with weights initialized using the MSRA method [29]. Training is conducted using the Adam optimizer [30] within the PyTorch framework [31] on an NVIDIA 4060Ti GPU.

Key hyperparameters include a learning rate of 1×10^{-4} for the actor network to ensure stable policy updates and 1×10^{-3} for the critic network to accelerate value estimation convergence. The exploration parameter ϵ is set to 0.1, introducing moderate randomness in action selection to prevent premature convergence to suboptimal policies while avoiding excessive exploration. According to [21], a shallow decoder suffices for this combinatorial optimization task. hence, the number of decoding layers L_{dec} is set to 2.

To find these suitable hyperparameters, we first selected representative candidate values for each and then performed a grid search followed by empirical validation on a held-out set of channel data. Specifically, we evaluated the actor learning rate from 10^{-5} to 10^{-2}, the critic learning rate from 10^{-4} to 10^{-1}. The ϵ is selected from $\{0.05, 0.1, 0.2\}$ based on the empirical value, and 0.1 is found to be more appropriate, which is a value that is often chosen for most tasks. We also tested decoder depths $L_{\text{dec}} \in \{1, 2, 3\}$. Finally, the hyperparameters listed in Table II achieve an optimal trade-off between training stability, convergence speed, and average performance.

TABLE II
SIMULATION AND TRAINING PARAMETERS

Parameter	Value
Channel Model	Rayleigh Fading, Ray-based
BS Antennas	16 / 32
UEs	40 / 70
Cell Radius	100 m
Actor Learning Rate	$1e^{-4}$
Critic Learning Rate	$1e^{-3}$
Optimizer	Adam
L_{dec}	2
ϵ	0.1
Max Steps per Episode	600
Training Epochs	600

B. Simulation Analysis

In this section, we compare traditional methods with learning-based schedulers. To ensure a fair evaluation, the learning-based schedulers are trained in environment with a relatively small number of users. Simulation results indicate that the performance of our scheduler closely approximates that of the PF algorithm. Widely recognized as a balanced scheduling strategy, the PF algorithm effectively manages the trade-off between system throughput and fairness. The comparable performance with the PF algorithm highlights our method's ability to achieve a similarly effective balance among these critical metrics.

1) Training Rewards and Convergence Analysis: This section presents the reward, as defined in (27), achieved throughout the training episodes. For a more comprehensive comparison, we evaluated multiple policy network architectures, including the existing PN, MLP network, our custom-designed CNN and CNN-Transformer network, as illustrated in the figure. All policy networks are based on the A2C framework.

The reward curves of model training in a Rayleigh environment are shown in the Fig. 4, where Figs. 4(a) and 4(b) compare the situation when the number of system users is set to 40 and 70, respectively. In Fig. 4(a), A2C-Proposed-40 that means when the number of users is 40, the reward obtained by using the CNN-Transformer network trained in the A2C framework and it achieves higher and more stable value, indicating superior convergence and performance over other models. A2C-PN-40, which uses a PN, reaches a high reward curve but falls short of the A2C-Proposed-40, while A2C-MLP-40 that uses MLP network and A2C-CNN-40 that uses CNN both stabilize at lower rewards, suggesting limited ability to learn effective scheduling patterns with this scale user number. In Fig. 4(b), as the number of users increases to 70, the probability of discovering users with favorable channel conditions also rises, the reward obtained by A2C-PN-70 gradually converges toward that of A2C-Proposed-70. However, the increased exploration space causes greater fluctuations in the reward curves. Despite this, A2C-Proposed-70 continues to demonstrate faster convergence. And the reward curves of A2C-MLP-70 and A2C-CNN-70 similarly begin to approach those of A2C-PN-70.

The reward curves of different models are compared in the ray channel environment. Figs. 5(a) and 5(b) correspond to a total user numbers of 40 and 70, respectively. In Fig. 5(a), We observed that, despite increased fluctuations in the overall reward curve, A2C-Proposed-40 quickly stabilizes at a high reward level, outperforming other methods in both convergence speed and peak reward. While A2C-PN-40 reaches a comparable reward level after a period of training, its

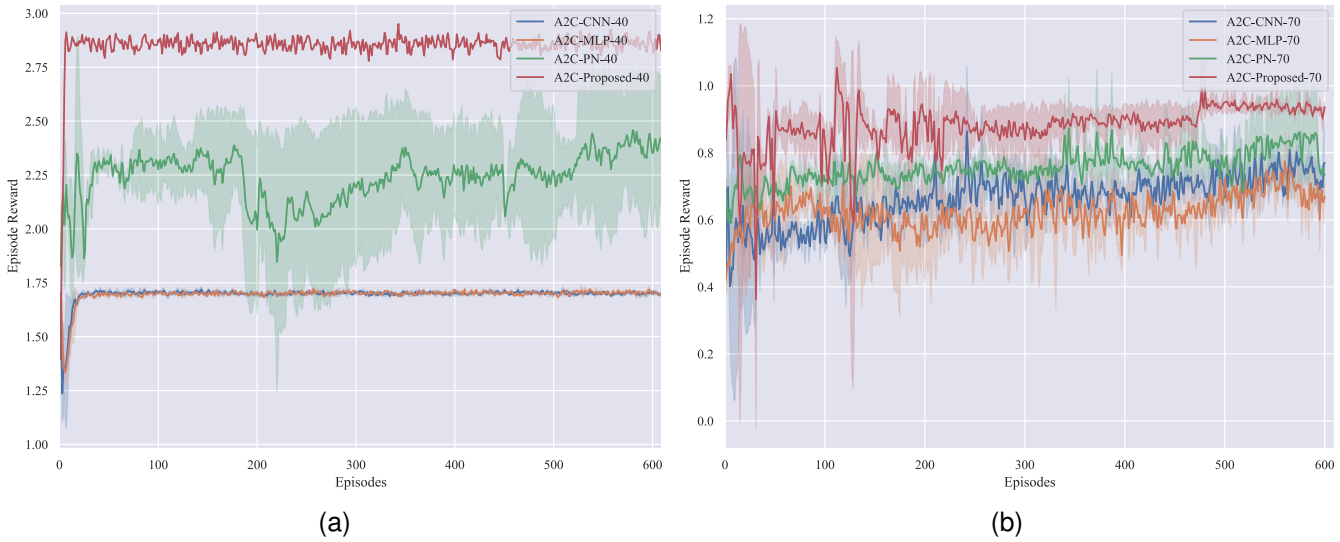


Fig. 4. Learning curves for Rayleigh channel environment.

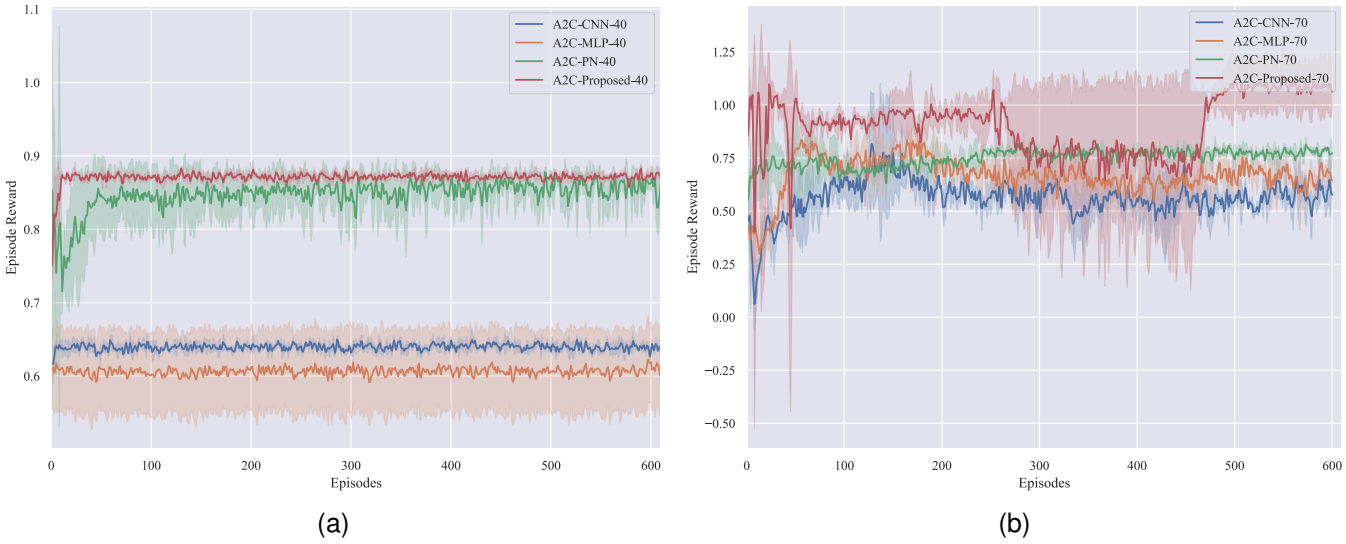


Fig. 5. Learning curves for ray-based channel environment.

average reward remains lower than that of A2C-Proposed-40. Meanwhile, A2C-MLP-40 and A2C-CNN-40 still continue to demonstrate suboptimal performance. In Fig. 5(b), with 70 users, A2C-Proposed-70 continues to lead, exhibiting faster convergence and higher rewards, underscoring its effectiveness as user count increases. Although A2C-PN-70 achieves high rewards, it falls short of A2C-Proposed-70 in terms of average reward and convergence speed. A2C-MLP-40 and A2C-CNN-40 show stability after convergence, but do not outperform A2C-Proposed-70 and A2C-PN-70 in general, which again illustrates their limitations in scheduling under high user density conditions.

Overall, these models experiences significant fluctuations during the initial learning phase, gradually stabilizing as training progresses, which is a training process consistent with reinforcement learning. However, as environmental complexity increases, scheduling performance tends to be constrained. For instance, the A2C-PN that based on PN shows more pronounced fluctuations in the ray-based channel environment compared to the Rayleigh channel. In contrast, our A2C-Proposed model exhibits relatively stable fluctuations across varying conditions, maintaining this characteristic even as the user number increases. These results underscore the exceptional adaptability and robustness of our scheduler, especially in complex environments with varying user densities, while the A2C-MLP and A2C-CNN models reveal notable limitations.

The reward curves shown by A2C-MLP and A2C-CNN are low that can be well explained, both CNN-based scheduler and MLP architectures establish direct input-output mappings. As the number of users increases, the policy network struggles to fit to optimal results. In contrast, the seq2seq structure offers significant advantages for user scheduling tasks. The scheduler

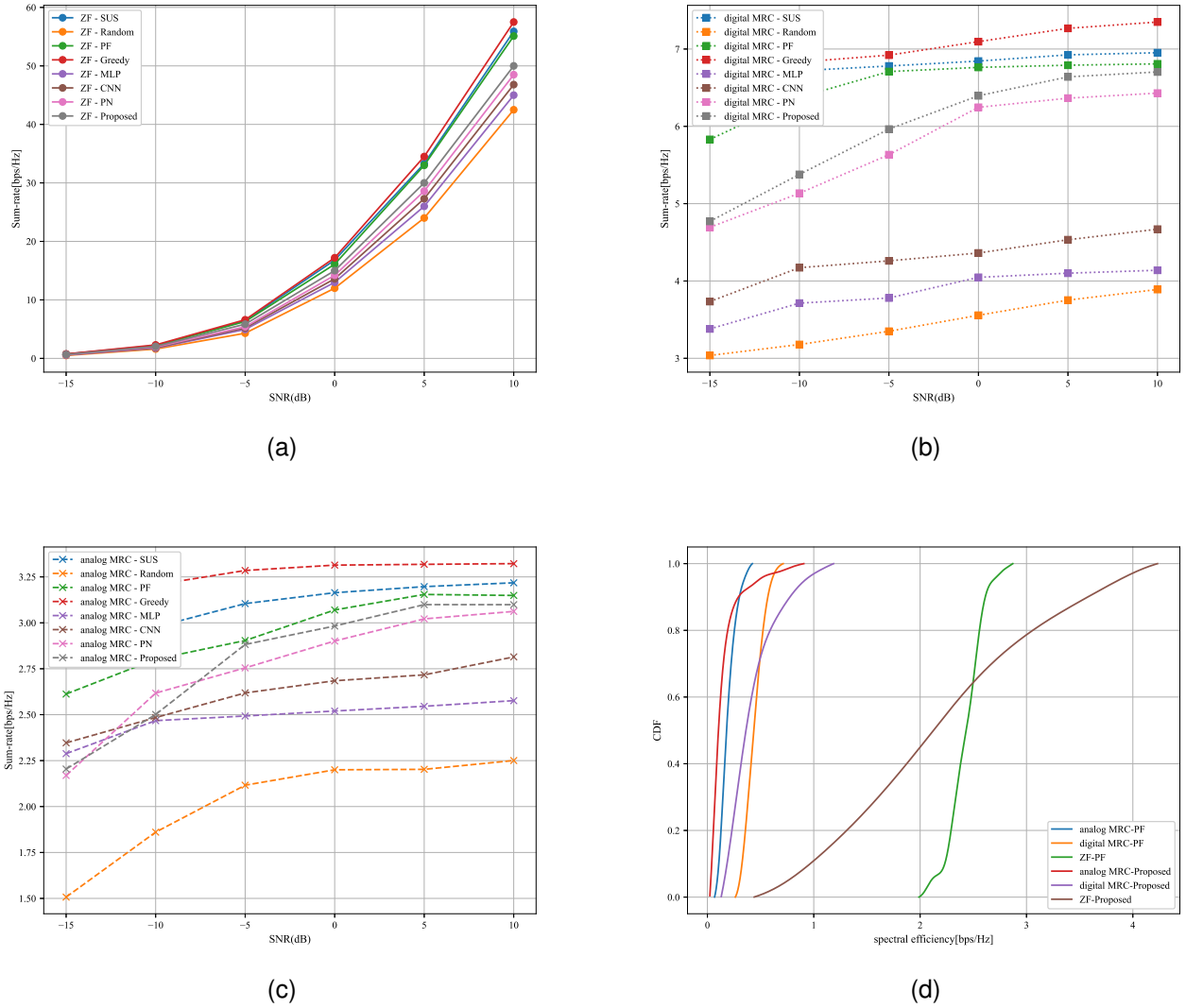


Fig. 6. SE comparison of scheduling at N = 40 and Rayleigh channels.

based on PN learns the dependencies in the user sequence, making decisions about selecting the next user based on the information of already chosen users, thereby improving the selection of a user group. In comparison, our CNN-Transformer network not only leverages the Transformer's ability to capture long-range dependencies but the convolutional module's strength in extracting local channel features. This combination results in a higher reward, faster convergence during training, and an overall higher average reward compared to the PN-based model.

During training, we also apply an ϵ-greedy strategy to improve exploration by introducing controlled randomness in action selection. While this randomness can contribute to fluctuations even during convergence, the overall trend in the reward curves indicates that the seq2seq style output achieved by our architecture that combines a CNN-Transformer network with an autoregressive decoder achieves superior exploration, faster convergence, and higher rewards.

2) Performance Comparison of Proposed Scheduler with other Models: Throughput, Fairness, and Computational Efficiency: Considering the characteristics of different beamforming technologies. ZF excels at canceling interference and achieving high capacity in ideal conditions, but it suffers from noise enhancement and high computational complexity in ill-conditioned channels. Digital MRC is simple and efficient, maximizing SNR with low complexity, but performs poorly in high interference environments and can be less effective at low SNR. Analog MRC offers low complexity and works well in high SNR scenarios, but lacks flexibility in adapting to dynamic channel conditions and is suboptimal in low SNR. During the testing stage, we test additional 600 timesteps and combine ZF, digital MRC and analog MRC to compare the sum rate at different SNR levels as illustrated in Fig. 6 and Fig. 7. The scheduling methods compared likewise include those of MLP, CNN, PN, and CNN-Transformer (referred to as Proposed in the figure), as well as traditional SUS, Random, PF and Greedy algorithms.

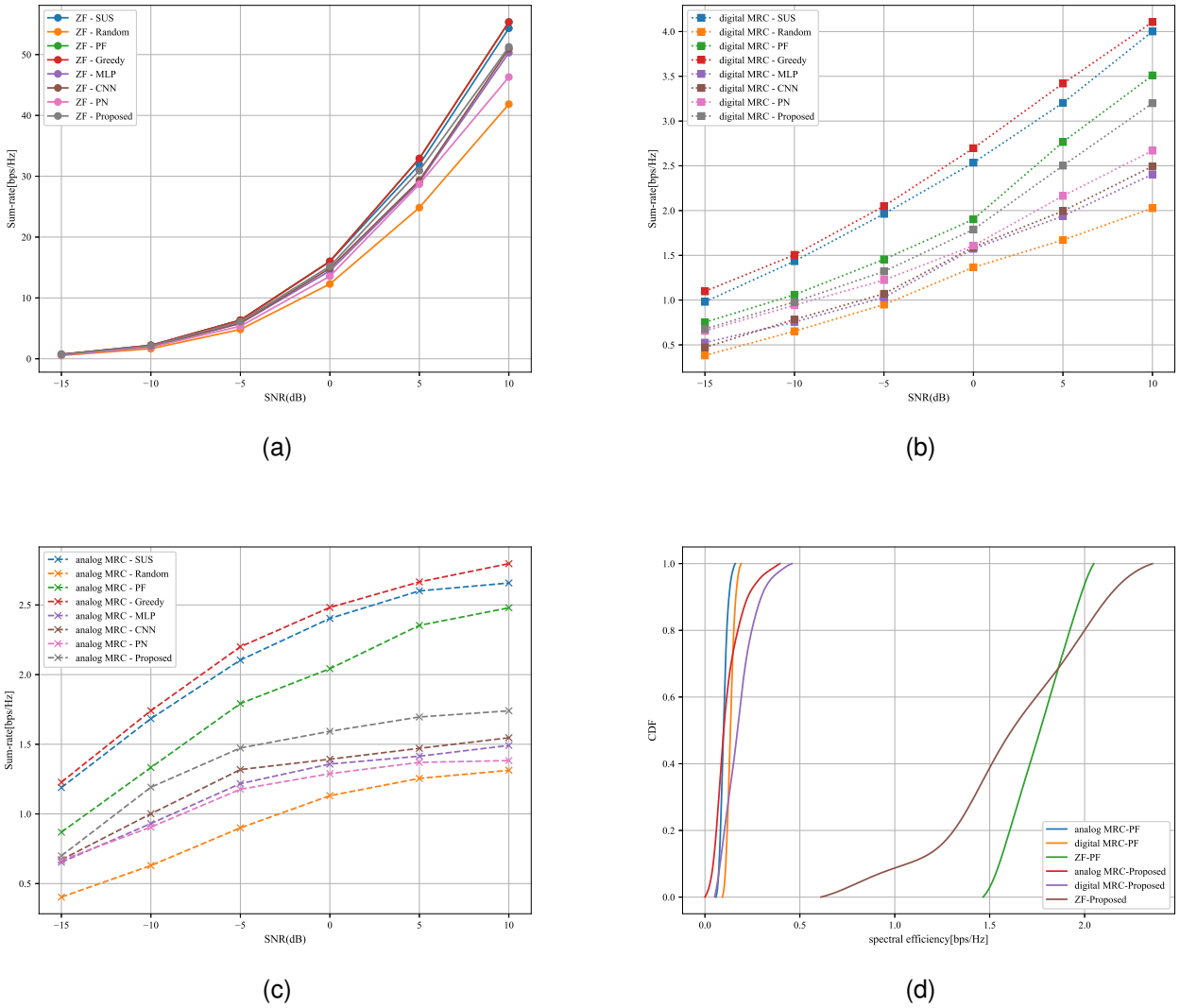


Fig. 7. SE comparison of scheduling at N =40 and Ray-based channels.

From Fig. 6, we observed that the scheduler utilizing the CNN-Transformer network achieves a competitive sum rate among learning-based schedulers, regardless of whether ZF or MRC beamforming is employed.

We compare the sum rate across different scheduling methods using ZF beamforming in Fig. 6(a). It is evident that across all SNR levels, ZF-Proposed, the sum rate obtained by the CNN-Transformer implemented under ZF beamforming, achieves results that are very close to the ZF-PF method and outperforms other learning-based approaches. As the SNR increases, the Proposed method demonstrates greater gains, indicating its ability to leverage channel information to maximize SE. This also further reflects the model's generalization performance. In Fig. 6(b), a similar trend is observed when applying digital MRC beamforming. Compared to other scheduling methods based on digital MRC, the Proposed scheduler outperforms other learning-based algorithms across the entire SNR range. While the PN also demonstrates commendable performance, it still falls short of our method. Notably, even at low SNR levels, the Proposed approach maintains a high sum rate, indicating its robustness under poor channel conditions. This robustness underscores the adaptability to diverse SNR environments, ensuring reliable performance across a wide range of signal quality conditions. In Fig. 6(c), the results are presented for analog MRC beamforming. Here, the Proposed scheduler maintains its advantage, outperforming the MLP-based, CNN-based, PN-based and Random. At mid-to-low SNR levels, the sum rate of our CNN-Transformer method that labeled as analog MRC-Proposed improvement becomes more pronounced.

To further illustrate the performance gap between our method and PF method, we also plotted the cumulative distribution function (CDF) of the achieved spectral efficiency under an SNR setting of 5 dB, as shown in Fig. 6(d). The CDF highlights the superior performance of the our method, consistently achieves a wider percentile distribution of SE, but stays close in terms of sum rates. This trend suggests that the Proposed approach not only ensures a degree of fairness, but

also improves the rate distribution among users. By providing users with appropriate service rates under different channel conditions, the scheduler shows great adaptability and can effectively meet the different needs of different users.

Fig. 7 is the result under the ray-based channel, where Fig. 7(a) compares the sum rate of various scheduling methods when ZF beamforming is employed. As shown in the figure, the Proposed scheduler achieves good results, and the performance gap becomes progressively larger as the SNR value increases. It is worth noting that the MLP-based and CNN-based schedulers perform poorly though compared to the conventional schedulers. However, both the CNN-Transformer that labeled Proposed and PN achieve better results, proving that the combination of reinforcement learning and ZF beamforming is still feasible. Fig. 7(b) shows the sum rate of each scheduling scheme under digital MRC beamforming. Similar to the results observed with ZF beamforming, the Proposed scheduler achieves strong sum rate performance across all SNR levels. Although the overall sum rate under digital MRC beamforming is lower compared to ZF beamforming, the relative performance trends remain consistent. The Proposed scheduler outperforms other learning-based methods, further demonstrating its adaptability to different beamforming techniques. Fig. 7(c) displays the sum rate comparison when analog MRC beamforming is applied. The results indicate that the Proposed scheduler outperforms other methods, particularly at higher SNR values. While the analog MRC scheme is easier to implement compared to ZF and digital MRC, it exhibits suboptimal performance in interference-limited environments, which explains the relatively lower sum rate observed in Fig. 7(a) and Fig. 7(b). Despite this limitation, the Proposed scheduler successfully maximizes the sum rate under analog MRC, further demonstrating the adaptability of the reinforcement learning framework to different beamforming techniques.

The CDF plot in Fig. 7(d) shows the distribution of achievable SE for users under the PF and Proposed schedulers with SNR set to 5 dB, using both ZF and MRC beamforming techniques. The results reveal that the CDF curves of the Proposed scheduler are notably more dispersed, indicating a broader range of user rates, which ensures overall fairness. In contrast, the curves obtained by the PF-based algorithm are relatively concentrated, suggesting a tendency to schedule users with better channel conditions, thereby offering only limited fairness. Nevertheless, the PF-based scheduler and our Proposed scheduling procedure share similar design principles: the PF-based scheduling want ensure a certain level of fairness in practical applications even though it selects users with relatively good channel conditions, while the our scheduling procedure also seeks to maximize the total rate while maintaining a certain level of fairness.

The observed performance trends in Fig. 6 and Fig. 7 can be attributed to the ploicy network of each scheduler and the unique characteristics of different beamforming techniques. The reward curves of the Proposed scheduler highlight its capacity to leverage RL to dynamically adjust to channel conditions and optimize resource allocation.

From these results, we find PF scheduler is explicitly designed to balance fairness and spectral efficiency, but in practical applications, it tends to prioritize scheduling users with better average channel conditions, which results in a relatively concentrated rate distribution, as shown in the CDF plots (Fig. 6(d) and Fig. 7(d)). The key difference between the Proposed scheduler and the PF lies in its objective: it seeks to maximize the sum rate while ensuring fairness. As a result, it can fully utilize select users, thus enabling a wider distribution reach. Unlike MLP-based or CNN-based schedulers, which struggle to generalize well due to their limited ability to map complex input-output relationships, the Proposed scheduler benefits from a Transformer-like structure that excels at capturing inter-user relationships and determining user priorities based on network conditions, enabling the scheduler to achieve higher sum rate across different beamforming methods.

Additionally, the differences between ZF, digital MRC and analog MRC beamforming techniques impact each scheduler's performance, as shown in Fig. 6(a)–(c) and Fig. 7(a)–(c). With ZF, which offers robust interference mitigation, the Proposed scheduler exhibits significant performance gains by exploiting high SNR conditions to maximize the sum rate. Digital MRC and analog MRC, while less effective in suppressing interference, still enable the Proposed scheduler to outperform other methods by leveraging spatial diversity more effectively. This adaptability suggests that the RL-based framework can be fine-tuned to work across various beamforming strategies, making it a versatile solution in diverse communication scenarios.

We recorded information about fairness as well as average run time in the Table III, the Proposed scheduler demonstrates superior computational efficiency and runtime stability under dynamic user loads. With a runtime of 0.27 seconds, the Proposed method significantly outperforms traditional schedulers like Greedy and PF, which exhibit higher execution times at 0.73 and 0.61 seconds, respectively. The Greedy scheduler, while achieving a high reward, suffers from the highest runtime due to its iterative nature and increased computational complexity when scaling to larger user sets. In contrast, the Proposed scheduler leverages a reinforcement learning framework that requires only a single forward propagation, ensuring consistent execution times regardless of user number. This efficiency is achieved without compromising on performance metrics, as the Proposed scheduler maintains competitive reward and fairness scores, at 0.95 and 0.94, respectively. Compared to MLP and CNN, the Proposed method balances both runtime efficiency and scheduling quality. For real-time applications, this balance is essential, as it enables the scheduler to respond swiftly to changes in user loads while ensuring equitable and high-throughput resource allocation. Therefore, we believe that the Proposed model is a practical and scalable solution for environments that require low latency and strong scheduling performance.

The reward and fairness for the Proposed scheduler reflect its design philosophy of maximizing sum rate while preserving an acceptable level of fairness. By leveraging a transformer-like structure for inter-user information extraction and adapting to various beamforming methods, the Proposed model achieves a unique balance between performance and computational efficiency. This adaptability makes it a promising approach for handling diverse SNR conditions and user distributions,

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT SCHEDULERS

Scheduler	Random	Greedy	SUS	PF	MLP	CNN	PN	Proposed
Reward	0.65	1.00	1.04	1.13	0.71	0.76	0.72	0.95
Fairness	0.95	0.52	0.64	0.96	0.91	0.93	0.89	0.94
Run Time (s)	0.36	0.73	0.43	0.61	0.14	0.17	0.26	0.27

Fig. 8. A2C vs PPO: Training reward.

TABLE IV
DRL METHOD COMPARISON (70 USERS, 32 ANTENNA BS)

Method	Sum-rate (bps/Hz)	Fairness	Reward
A2C (Ours)	6.80	**0.93**	**0.85**
PPO	**7.26**	0.88	0.73

ensuring reliable performance in both low and high SNR environments.

Although different policy-based schedulers demonstrate improved performance over conventional methods, their optimization remains limited by the need for extensive offline training and fixed inference strategies. To further enhance adaptability and learning efficiency, we explore DRL approaches, namely A2C and PPO, and present a comparative analysis of their effectiveness.

Table IV presents a comparison of A2C and PPO in terms of sum-rate and fairness. PPO achieves a marginally higher peak sum-rate (approximately 6.7% higher than that of A2C), but its fairness and reward are lower, suggesting that A2C maintains a more balanced resource allocation over extended scheduling periods.

Furthermore, the reward curves in Fig. 8 illustrate that PPO undergoes greater fluctuations, possibly due to its sensitivity to policy updates and hyperparameter choices. Although the clipping mechanism in PPO helps prevent excessive policy changes, it may also hinder the discovery of optimal scheduling policies. In contrast, A2C benefits from its synchronous updates, resulting in a better convergence trajectory.

Given these observations, A2C remains a preferable choice in scheduling scenarios where fairness and stability are of primary concern. Future work will explore alternative methods, such as Soft Actor-Critic (SAC), to assess whether improved exploration strategies can further enhance performance in discrete action spaces.

C. Future Research Directions in Scheduler Design

The results presented in this study present the potential of RL-based schedulers in balancing computational efficiency and scheduling performance. Building upon these findings, future research can explore advanced methodologies and novel frameworks to further optimize scheduler design and functionality. Here, we identify two promising directions for future improvements: the integration of graph neural networks (GNNs) to better capture spatial-temporal user dependencies, and the development of hybrid reinforcement learning frameworks to enhance scheduling scalability and adaptability.

1) GNNs for Enhanced Spectral Efficiency: While our current scheduler demonstrates significant improvements in SE compared to other learning-based methods, a measurable gap persists when compared to the PF scheduler. Closing this gap requires refining both the policy network architecture and the reward function.

User scheduling in massive MU-MIMO systems inherently exhibits a graph-like structure, where each user can be represented as a node with associated attributes such as spatial coordinates, channel state information, and historical throughput. The interference relationships between users naturally form edges in this graph. Traditional scheduling methods struggle to explicitly model such dependencies. GNNs, with their ability to model structured data through message passing and spatial-temporal aggregation, offer a promising approach to improve scheduling performance.

To fully exploit the potential of GNNs, future research could explore the following aspects:

- **Graph Representation Learning:** Develop a unified graphical representation for MU-MIMO scheduling, incorporating key user attributes (e.g., channel state, mobility patterns) while preserving the underlying interference relationships.
- **Adaptive Graph Construction:** Investigate strategies for dynamically constructing user graphs based on real-time network conditions, such as adaptive edge weighting mechanisms to capture inter-user interference more accurately.
- **GNN-RL Integration:** Embed GNN-based feature extraction modules into policy networks, enabling more effective scheduling decisions through enhanced spatial-awareness.

2) Advancing RL Frameworks for Scalable User Scheduling: The dynamic nature of wireless communication networks demands more flexible and scalable scheduling strategies. Model-free RL algorithms, while effective in learning optimal policies, often suffer from slow convergence and require extensive interactions with the environment. Model-based approaches, on the other hand, leverage predictive models to improve sample efficiency but may struggle with generalization due to modeling inaccuracies. A hybrid RL framework that combines these approaches could offer a compelling solution.

Future studies could explore the following key enhancements:

- **Dimensionality Reduction for Scheduling Efficiency:** Employ self-encoding techniques or clustering-based pre-scheduling to reduce the decision space, thereby improving training efficiency and inference speed.
- **Model-Enhanced Policy Optimization:** Incorporate channel prediction models to anticipate near-future CSI variations, allowing the scheduler to proactively adjust resource allocation strategies.
- **Offline Pretraining with Online Adaptation:** Utilize offline model-based pretraining to accelerate RL convergence, followed by real-time online adaptation using model-free updates to ensure robustness in dynamic environments.

3) Potential Impact on Wireless Communication and Scheduling: We believe that these technologies will be highly impactful in next-generation wireless systems. By integrating graph-based learning with DRL, the architecture enables explicit modeling of spatial temporal interference patterns through topological relationships—a critical capability for spectrum coordination in ultra-dense networks. This capability supports the vision of context-aware networks, where joint PHY-MAC optimization yields substantial area capacity gains in dense urban deployments. Moreover, a hybrid RL framework that leverages offline pre-training followed by online adaptation significantly reduces real-time computational latency compared to purely model-free methods, meeting the low latency, high scalability demands of massive machine type communications (mMTC) in 5G-Advanced networks.

V. Conclusion

To address the challenges of performance degradation and computational complexity in user scheduling for massive MU-MIMO scenarios, we propose a deep reinforcement learning-based user scheduler. This scheduler is capable of continuously selecting user groups under dynamic channel conditions. Our findings indicate that combining convolution module with transformer structures significantly enhances the model's ability to capture spatio-temporal relationships among users. Additionally, integrating beamforming techniques offers valuable insights for designing future schedulers.

In the evolving IoT landscape, with increasing numbers of transmitting antennas and users, traditional algorithms fall short of meeting the demands of 6G+ networks due to their slower decision-making processes. In contrast, RL-based schedulers present a promising alternative, delivering faster and more efficient scheduling capabilities suited for next-generation network requirements.

References

[1] N. H. M. Adnan, I. M. Rafiqul, and A. H. M. Z. Alam, "Massive MIMO for fifth generation (5G): Opportunities and challenges," in *Proc. ICCCE*, 2016.

[2] S. Li *et al.*, and J. Yin, "Massive MIMO asymptotics for ray-based propagation channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3977–3991, 2020.

[3] S. Li, P. J. Smith, and P. A. Dmochowski, "Hybrid distributed MRC with imperfect CSI for MU-MIMO systems," *IEEE Commun. Lett.*, vol. 25, no. 9, pp. 3109–3113, 2021.

[4] E. Castañeda, A. Silva, A. Gameiro, and M. Kountouris, "An overview on resource allocation techniques for multi-user MIMO systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 239–284, 2017.

[5] G. Dimić and N. D. Sidiropoulos, "On downlink beamforming with greedy user selection: performance analysis and a simple new algorithm," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3857–3868, 2005.

[6] S. Lee and J. S. Thompson, "QoS-guaranteed sequential user selection in multiuser MIMO downlink channels," in *Proc. IEEE VTC*, 2007.

[7] T. Qi, Y. Wang, and X. Feng, "Performance analysis of downlink user selection in multiuser MIMO system," in *Proc. IEEE ITNEC*, 2017.

[8] M. Torabzadeh and W. Ajib, "Proportional fairness packet scheduling with transmit beamforming for multi-user MIMO systems," in *Proc. IEEE RWS*, 2009.

[9] K. Ko and J. Lee, "Multiuser MIMO user selection based on chordal distance," *IEEE Trans. Commun.*, vol. 60, no. 3, pp. 649–654, 2012.

[10] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 528–541, 2006.

[11] M. Kuerbis, N. M. Balasubramanya, L. Lampe, and A. Lampe, "User scheduling in massive MIMO systems with a large number of devices," in *Proc. IEEE PIMRC*, 2017.

[12] S. Han, G. Kong, D. Kim, and S. Choi, "CNN-based user selection in MIMO broadcasting channel," in *Proc. ITC-CSCC*, 2019.

[13] Y. Yang *et al.*, "DECCO: Deep-learning enabled coverage and capacity optimization for massive MIMO systems," *IEEE Access*, vol. 6, pp. 23361–23371, 2018.

[14] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 2019.

[15] K.-L. A. Yau, K. H. Kwong, and C. Shen, "Reinforcement learning models for scheduling in wireless networks," *Front. Comput. Sci.*, vol. 7, no. 5, pp. 754–766, 2013.

[16] G. Bu and J. Jiang, "Reinforcement learning-based user scheduling and resource allocation for massive MU-MIMO system," in *Proc. IEEE/CIC ICCC*, 2019.

[17] A. Kumar, G. Verma, C. Rao, A. Swami, and S. Segarra, "Adaptive contention window design using deep Q-learning," in *Proc. IEEE ICASSP*, 2021.

[18] X. Guo *et al.*, "A novel user selection massive MIMO scheduling algorithm via real time DDPG," in *Proc. IEEE GLOBECOM*, 2020.

[19] C. Wang, D. Deng, L. Xu, W. Wang, and F. Gao, "Joint interference alignment and power control for dense networks via deep reinforcement learning," *IEEE Wireless Commun. Lett.*, vol. 10, no. 5, pp. 966–970, 2021.

[20] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, 2019.

[21] X. Bresson and T. Laurent, "The transformer network for the traveling salesman problem," 2021, *arXiv preprint arXiv:2103.03012*.

[22] L. Chen *et al.*, "Deep reinforcement learning for resource allocation in massive MIMO," in *Proc. EUSIPCO*, 2021.

[23] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. NeurIPS*, 2015.

[24] N. W. M. Thet, T. Baykas, and M. K. Ozdemir, "Performance analysis of user scheduling in massive MIMO with fast moving users," in *Proc. IEEE PIMRC*, 2019.

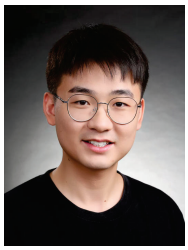
[25] R. K. Jain, D. M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Digital Equipment Corporation, MA, Tech. Rep. DEC-TR-301, 1984.

[26] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Proc. NeurIPS*, 1999.
[27] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017.
[28] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," 2020, *arXiv preprint arXiv:2005.08100*.
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE ICCV*, 2015.
[30] D. Kinga and J. Ba, "A method for stochastic optimization," in *Proc. ICLR*, 2015.
[31] A. Paszke *et al.*, "Automatic differentiation in pytorch," in *Proc. NeurIPS*, 2017.

Yue Zhu received the M.S. degree in Information and Communication Engineering from Harbin Engineering University, Harbin, China, in 2025. His research interests include MU-MIMO, resource optimization, and machine learning for communication.

Shuang Li received the B.E. degree in Electronic Information Science and Technology from Harbin Engineering University, China, in 2007, and the M.E. degree in Signal and Information Processing from the same university in 2010. From 2010 to 2015, she worked as a system engineer at Huawei (Chengdu) and also held positions in the public sector. She received her Ph.D. degree in Electronic Engineering from Victoria University of Wellington, New Zealand, in 2020. She is currently a tenure-track Associate Professor at Harbin Engineering University, China. She has published a number of high-quality research articles in the field of wireless communications, with her work appearing in top-tier journals and flagship conferences such as IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, and the IEEE International Conference on Communications (ICC). Her research interests include system performance analysis in multi-user MIMO (MU-MIMO) wireless communication systems, intelligent communication systems, underwater acoustic communications, and machine learning for future communication systems.

Longxiang Guo (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in Underwater Acoustic Engineering from Harbin Engineering University, Harbin, China, in 1998, 2001, and 2006, respectively. He is currently a Professor with the College of Underwater Acoustic Engineering, Harbin Engineering University. He is also a Visiting Scholar with the Laboratory of Environment and Acoustics, Department of Applied Sciences, Free University of Brussels, Belgium, from 2013 to 2015. He is currently presiding over the subprojects of the National Key Research and Development Program of China, the general program of the National Natural Science Foundation of China. His current research interests include underwater acoustic signal processing, underwater acoustic array signal processing, underwater acoustic detection, sonar system simulation, and polar acoustics.



Wei Ge received the Ph.D. degree in Underwater Acoustic Engineering from Harbin Engineering University, Harbin, China, in 2021. From 2019 to 2020, he was a Visiting Student with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI, USA. Since 2024, he has been an Associate Professor with College of Underwater Acoustic Engineering, Harbin Engineering University. His current research interests include underwater acoustic signal processing and communications.

Li Wei (Member, IEEE) is currently an Associate Professor with the College of Underwater Acoustic Engineering, Harbin Engineering University, Harbin, China. He received his B.E. degree in Electrical Engineering and Automation from Xi'an Jiaotong University, Xi'an, China, in 2011. He received his M.S. degree from University of Connecticut, Storrs, CT, U.S.A., in 2016. He received his Ph.D. degree from Michigan Technological University, Houghton, MI, U.S.A., in 2022. His current research interests include underwater acoustic communication and networking, deep generative models for underwater acoustic channels, AI for science.