

xDiff: Online Diffusion Model for Collaborative Inter-Cell Interference Management in 5G O-RAN

Peihao Yan, Huacheng Zeng, and Y. Thomas Hou

Abstract—Open Radio Access Network (O-RAN) is a key architectural paradigm for 5G and beyond cellular networks, enabling the adoption of intelligent and efficient resource management solutions. Meanwhile, diffusion models have demonstrated remarkable capabilities in image and video generation, making them attractive for network optimization tasks. In this paper, we propose xDiff, a diffusion-based reinforcement learning (RL) framework for inter-cell interference management (ICIM) in O-RAN. We first formulate ICIM as a resource allocation optimization problem aimed at maximizing a user-defined reward function and then develop an online learning solution by integrating a diffusion model into an RL framework for near-real-time policy generation. Particularly, we introduce a novel metric, *preference values*, as the policy representation to enable efficient policy-guided resource allocation within O-RAN distributed units (DUs). We have implemented xDiff on a 5G testbed consisting of three cells and a set of smartphones and evaluated its performance in various small-cell scenarios. Extensive experimental results demonstrate that xDiff outperforms state-of-the-art ICIM approaches, highlighting the potential of diffusion models for online optimization of O-RAN. Source code is available on GitHub [1].

Index Terms—O-RAN, interference management, conditional diffusion policy, deep reinforcement learning, online learning, 5G

I. INTRODUCTION

Open Radio Access Network (O-RAN) has emerged as a critical architecture for future cellular infrastructure. As bandwidth demands continue to surge, it is common that O-RAN small cells operate on the same frequency band (i.e., frequency reuse factor of 1) to maximize spectral efficiency [2]. However, the small-cell architecture exacerbates inter-cell interference, posing a significant challenge that requires careful interference management to ensure reliable performance. While extensive research exists on inter-cell interference management (ICIM), the innovative architectural paradigm of O-RAN unlocks new opportunities for deploying online learning approaches in real-world RAN systems by leveraging the xApps hosted within the Near-Real-Time (Near-RT) RAN Intelligent Controller (RIC). These approaches can rapidly adapt to interference conditions in the network to maximize the user-defined objectives while respecting the quality of service (QoS) demands of users.

P. Yan and H. Zeng are with the Department of Computer Science and Engineering at Michigan State University, East Lansing, MI 48824. Y.T. Hou is with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061. Corresponding author: Huacheng Zeng (hzeng@msu.edu).

The work of P. Yan and H. Zeng was supported in part by NSF Grants ECCS-2434001 and CNS-2312448 and NIST PWSICF Award 26-60-IF010. The work of Y.T. Hou was supported in part by NSF Grants ECCS-2434000 and CNS-2312447 and Virginia Commonwealth Cyber Initiative (CCI).

Diffusion models have demonstrated remarkable capabilities in content generation across diverse domains, such as producing high-quality images, videos, and text by iteratively refining noisy data into structured output. This generative prowess stems from their ability to model complex, high-dimensional distributions, making them a powerful tool beyond traditional applications. In the context of online interference management in O-RAN, diffusion models offer a compelling fit due to their capability to capture the stochastic nature of inter-cell interference and adaptively refine resource allocation policies. Despite their potential, diffusion models remain underexplored in O-RAN applications, as existing interference management strategies largely rely on conventional rule-based or learning-based approaches. This underutilization highlights an opportunity to harness the unique strengths of diffusion models to generate robust policies for the online optimization of O-RAN systems.

In this paper, we study the ICIM problem in an O-RAN system where adjacent small cells operate on the same frequency band.¹ We propose xDiff, a diffusion-based reinforcement learning (RL) framework that generates policies to guide the resource allocation at O-RAN distributed units (DUs). xDiff operates as an xApp within the Near-RT RIC, periodically querying DUs via the E2 interface to collect key performance metric (KPM) and media access control (MAC) data. Based on the KPM and MAC data observations, it dynamically refines its policies to guide the resource allocation at individual DUs, aiming to maximize a user-defined reward function while accounting for inter-cell interference, time-varying channel conditions, and dynamic user QoS demands.

One challenge in the design of xDiff lies in the time-scale discrepancy between RIC and DU operations. The Near-RT RIC updates its resource management policies based on the KPM and MAC data from multiple DUs, operating at a near-real-time scale of 10 ms to 1 s. In contrast, DUs operate at a real-time scale of 1 ms, performing subframe-by-subframe resource allocation based on the policies provided by the Near-RT RIC. In addition to this time-scale mismatch, individual DUs cannot cooperate in real time for joint interference management due to inter-DU communication latency. Instead, cooperation and coordination among DUs need to be achieved through the policies generated by the Near-RT RIC.

To address this challenge, we introduce a new metric called the *preference value* for each resource block (RB) at every DU. This metric mathematically represents the policy

¹We note that the frequency reuse factor of 1 is commonplace in real-world cellular networks as the operators push the boundary of spectrum utilization, especially in densely populated areas (see [3]–[5]).

generated by the Near-RT RIC, enabling interference-aware resource allocation at individual DUs. The numerical policy representation serves as a crucial bridge between near-real-time and real-time operations for efficient ICIM. For the Near-RT RIC, it simplifies the output format of the diffusion model, accelerating the convergence of its training process. This is particularly important for online learning, as the diffusion model must be continuously trained based on KPM and MAC data observations to adapt to network dynamics, such as interference conditions, time-varying channels, and fluctuating user demands. For individual DUs, this numerical policy representation acts as a set of scheduling-priority weights for resource block scheduling. It can be seamlessly integrated with existing DU scheduling algorithms, such as proportional fairness (PF).

Another challenge within the design of xDiff is the integration of the diffusion model with reinforcement learning. Unlike image and video generation tasks, where diffusion models are trained offline on large datasets to generate high-quality outputs without requiring continuous adaptation, online policy generation in O-RAN requires that diffusion models continuously adapt to dynamic network conditions. Moreover, unlike static image/video datasets, the input distribution of diffusion models in O-RAN evolves in real-time based on user demands, interference levels, and network states. This requires continuous training and updates of diffusion models using streaming KPM and MAC data from DUs.

To address this challenge, we propose an efficient architecture that integrates a diffusion model with RL for adaptive policy generation. Specifically, we use a conditional denoising diffusion probabilistic model (DDPM) to generate policies and employ a critic with double Q-learning networks to evaluate them. One difficulty lies in effectively aligning the diffusion model's generative process with the policy optimization objectives of Q-learning. To overcome this, we introduce a joint training approach that combines the Q-learning loss with the denoising loss. By combining the two losses, the diffusion model learns to generate policies that not only capture the underlying data distribution but also optimize the performance based on the critic's feedback. Moreover, this architecture leverages the strengths of both policy-based and value-based models, reducing the number of denoising steps required by the diffusion model. This makes it well-suited for Near-RT operations.

We have integrated xDiff into the OpenAirInterface (OAI) software suite and evaluated its performance on a 5G testbed consisting of three cells and a set of smartphones. Ablation studies show that the diffusion model plays a key role in generating policies to improve network performance in the face of inter-cell interference. Extensive experimental results demonstrate that xDiff is robust in Near-RT policy generation and outperforms the state-of-the-art (SOTA) RL approaches, showcasing the potential of diffusion models for online optimization in O-RAN.

This work advances the state-of-the-art as follows.

- We propose a new metric called *preference value* as the policy representation for the Near-RT RIC to manage inter-cell interference. This compact representation not only simplifies

TABLE I: 5G network ICI approaches in the literature.

Reference	Objective	Key Idea	OTA?	UE	RIC?
eICIC [13] CoMP [14] LTE-Adv [19]	Spectrum Efficiency	Almost Blank Subframe (ABS)	✓	✗	✗
CoCa [39]	Maximize Streams	Antenna Beamforming	✓	[40]	✗
CSRS [41]	feICIC	FAP clustering	✗	✗	✓
2-Layer IC [42]	Improve Link Capacity	RCGCA	✗	✗	✓
IAIS [24]	Enhance reliability	Interference prediction	✓	[43]	✓
ChARM [36] IM-rApp [23]	Improve Throughput	Machine Learning (Prediction)	✓	N/A	✓
MLMCOS [25]	Maintain High QoE	ML Classification	✗	[44]	✓
DRL-IM [20] mmLBRA [45] Dynamic-IC [28]	Increase Sum-Rate	Reinforcement Learning	✗	✗	✓
xDiff (ours)	Improve System QoS	Diffusion Policy	✓	✓	✓

Note: OTA = Over-The-Air.

the training process of the diffusion model but also seamlessly integrates with existing DU scheduling algorithms.

- We propose a framework that integrates diffusion model with RL. It appears to be robust and efficient in maximizing user-defined reward functions in the presence of inter-cell interference.
- Extensive experimental results demonstrate that xDiff outperforms the SOTA ICIM approaches, including deep Q-learning and actor-critic RL.

II. RELATED WORK

Interference is a fundamental problem in wireless networks. Even within cellular networks, the literature already has a large body of work on interference management, encompassing power-domain methods [6], [7], time-domain techniques [8], [9], frequency-domain approaches [10], spatial-domain solutions [11], and code-domain strategies [12]. Since it is impossible to review all existing work, our survey focuses on ICIM in 5G and O-RAN.

ICIM in 5G and O-RAN. A number of ICIM techniques have been proposed for 4G/5G cellular networks, such as enhanced inter-cell interference coordination (eICIC) [13], coordinated multi-point (CoMP) [14], and dynamic power control [15]. In recent years, the use of massive MIMO, beamforming, and carrier aggregation technologies has gained increasing attention for interference suppression [16]–[18]. Meanwhile, both data-driven optimization approaches [19], [20] and learning-based techniques [10], [21], [22] have advanced in increasingly sophisticated forms to enhance the management of inter-cell interference, leading to significant improvements in spectrum utilization for 5G networks [17], [23], [24].

The new architecture of O-RAN presents opportunities to deploy intelligent solutions for ICIM in realistic 5G networks, driving the rapid development of DNN models [24]–[27], RL solutions [28]–[32], and graph-based models [33] for interference prediction and management. The flexibility of O-RAN further enables inter-cell coordination via the RIC for interference mitigation [34]–[38], enhancing 5G network resilience, robustness, and scalability.

Although the literature contains a large body of work on ICIM, little progress has been made in the design of online learning solutions and their validation in realistic networks.

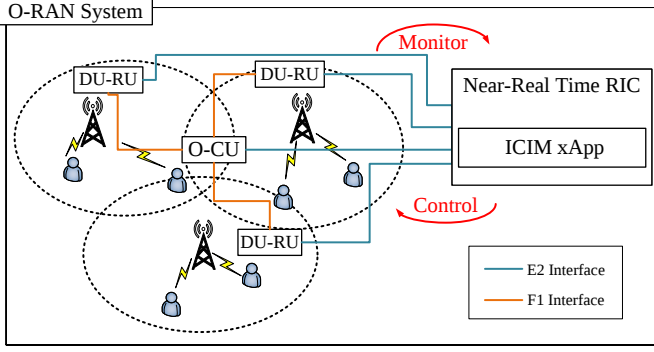


Fig. 1: Inter-cell interference among O-RAN small cells.

xDiff fills this gap. Table I highlights the most relevant work and positions xDiff uniquely in the literature. Specifically, xDiff differs from prior work in that it is the first to use diffusion policies for ICIM and has been validated on realistic 5G testbeds.

xApps for Resource Allocation in O-RAN. Many xApps have been developed to optimize various network performance metrics—such as spectral efficiency, latency, fairness, and reliability—by dynamically enforcing resource allocation policies at DUs (see, e.g., [34], [46]). Among these approaches, online learning [47], particularly RL [28], [45], has emerged as an appealing technique for xApps due to its ability to adaptively refine resource allocation policies in response to time-varying network conditions. Actually, a considerable volume of work has explored RL for network management and optimization in O-RAN, including network slicing [32], [48], flow scheduling [49], MCS selection [50], [51]. However, despite these prior efforts, limited progress has been made in developing RL-based xApps for ICIM in O-RAN. xDiff fills this gap by introducing a novel RL-based xApp for ICIM.

Diffusion Policy for Reinforcement Learning. Diffusion models, originally developed for image and video synthesis [52], have recently gained attention in RL for their ability to model complex, high-dimensional policy distributions [53]–[55]. Unlike traditional RL approaches that rely on deterministic or stochastic policy networks, diffusion models leverage a generative process that iteratively refines noisy inputs into structured outputs, enabling more expressive and flexible policy representations. In an RL framework, diffusion models can be used to learn a distribution over optimal actions, enhancing agents’ adaptation in dynamic environments. For instance, pioneering work has integrated diffusion models with RL to model complex distributions of possible actions [56]–[58], make multi-step predictions about the next state [59], [60], and expand the database [61]. To the best of our knowledge, diffusion models have not been explored for online resource optimization in O-RAN.

III. PROBLEM STATEMENT

Consider an O-RAN system as shown in Fig. 1, where the small cells operate on the same frequency band for both uplink and downlink transmissions. Each small cell is equipped with one RU and one DU. A central RIC is connected to all DU devices for remote control and management. Due to the physical

distance between RIC and DU, the control and management of DUs can only be executed at the Near-RT level. The primary role of RIC in this architecture is to generate Near-RT policies that guide resource allocation operations in individual DUs, optimizing the user-defined reward function. For this network setting, we have the following two notes.

Frequency Reuse. Given the limited availability of spectrum, frequency reuse is essential for accommodating the growing demand for wireless services. By reusing the same frequency in adjacent cells, operators can deploy a higher density of base stations, such as small cells, to improve coverage and capacity without requiring additional spectrum resources. In fact, real 5G networks commonly reuse the same frequency bands for neighboring cells—a practice known as frequency reuse or universal frequency reuse (Reuse-1). Ericsson’s study shows that Reuse-1 will play an important role in meeting the growing capacity demands in dense 5G networks and has the potential to double the network capacity compared to Reuse-2 [62]. The aggressive frequency reuse calls for efficient and intelligent interference management techniques.

Multi-DU Multi-Cell Architecture: The flexibility of O-RAN allows for the support of multiple cells (i.e., multiple RUs) using either a single DU or multiple DUs. In the former case, signals from all RUs are jointly processed at a single DU, making it well-suited for CoMP and massive MIMO transmissions. However, this approach places a heavy computational burden on the DU and requires high-capacity front-haul links, posing challenges for practical deployment. In the latter case, PHY-layer signal processing is performed independently at individual DUs, with coordination managed at the Near-RT RIC. While this architecture does not fully optimize network capacity, it offers great flexibility in network operation and maintenance. In this work, we consider the multi-DU architecture for our design.

A. Experimental Observations

To understand the impacts of inter-cell interference, we conduct experiments on an OAI testbed that comprises two cells and two commercial smartphones (see details in §V). The core network generates persistent downlink data traffic at 50 Mbps for smartphone 1 and at 60 Mbps for smartphone 2 using *iperf*. The real throughput (TP) and queue delay of both smartphones were measured at their respective DUs. Fig. 2 presents our observations in three cases.

- **Case I:** We intentionally avoid inter-cell interference by assigning the two cells to different frequency bands: one on n78 and the other on n41. Fig. 2a presents our measurement results. Evidently, the two smartphones maintain stable and reliable link connections with an acceptable delay of 1 ms.
- **Case II:** We assign both cells to the same frequency band, i.e., n78. Fig. 2b shows our measurement results. It can be seen that smartphone 2 experiences an unstable link connection with significant queue delays (up to 5 s). This connection instability is caused by interference from the other cell.
- **Case III:** In contrast to Case II, we use the RIC to deploy a simple resource allocation policy for ICIM: cell 1 uses 0–50 RBs while cell 2 uses 51–106 RBs. Fig. 2c presents

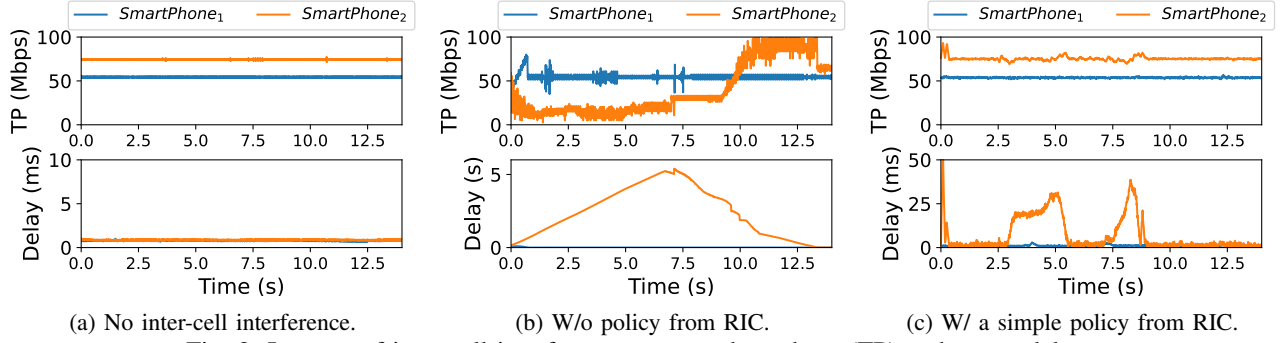


Fig. 2: Impacts of inter-cell interference on user throughput (TP) and queue delay.

our measurement results in this case. It shows that both smartphones maintain stable connections with acceptable delay.

These experimental results demonstrate the destructiveness of inter-cell interference and showcase the effectiveness of the RIC policy in ICIM for meeting users' throughput and delay demands.

B. Reward-Based Formulation

The objective of this work is to develop an xApp for Near-RT RIC that can generate an efficient policy to guide the resource allocation of DUs in the presence of inter-cell interference. Denote $\mathcal{K}$ as the set of small cells, with RU_k and DU_k denoting its radio and distributed units. Denote $\mathcal{U}_k$ as the set of UEs served by DU_k , with $\mathcal{U} = \bigcup_{k \in \mathcal{K}} \mathcal{U}_k$ denoting the set of all UEs in the network. While DU performs resource allocation subframe-by-subframe at the time scale of 1 ms, the policy from the RIC updates at a slower pace, with the time scale from 10 ms to 1 s.

For UE $i \in \mathcal{U}$, denote P_i as its throughput demand. Denote $\rho_i(t)$ as its achieved throughput in time slot t , where $t = 0, 1, 2, \dots$. If $\rho_i(t) \geq P_i$, then the UE's throughput demand is met and the regret is zero; otherwise, we define the regret as its normalized throughput deficit, i.e., $\frac{P_i - \rho_i(t)}{P_i}$. Combining these two cases, we model the regret of UE i in time slot t as follows: $\max(\frac{P_i - \rho_i(t)}{P_i}, 0)$.

Similarly, for UE $i \in \mathcal{U}$, denote D_i as its delay demand. Denote $\tau_i(t)$ as the achieved average delay of UE i 's data packets in time slot t . If $\tau_i(t) \leq D_i$, then the UE's delay demand is met and its delay regret is zero; otherwise, we define the delay regret as its normalized delay deficit, i.e., $\frac{\tau_i(t) - D_i}{D_i}$. Combining these two cases, we model the delay regret of UE i in time slot t as follows: $\max(\frac{\tau_i(t) - D_i}{D_i}, 0)$.

Following the convention, we use reward instead of regret as our optimization objective function. To do so, we define the reward value as the inverse of the regret value. Specifically, let $r_k^{[p]}(t)$ denote the throughput reward of small cell k in time slot t , and $r_k^{[d]}(t)$ denote its delay reward. Then, we have:

$$r_k^{[p]}(t) = \sum_{i \in \mathcal{U}_k} \min(\frac{\rho_i(t) - P_i}{P_i}, 0), \quad (1)$$

$$r_k^{[d]}(t) = \sum_{i \in \mathcal{U}_k} \min(\frac{D_i - \tau_i(t)}{D_i}, 0), \quad (2)$$

where both $r_k^{[p]}(t)$ and $r_k^{[d]}(t)$ are non-positive values.

To model the varying QoS demands of different UEs, we introduce a non-negative vector to denote the weights of their throughput and delay rewards. Specifically, let $\lambda_k^{[p]}$, $\lambda_k^{[d]}$ denote the throughput and delay weights in small cell k . Network operators can use these weights to adjust the priorities of throughput and delay during online optimization. Incorporating the weights, the QoS metric, representing the total reward of all UEs in all cells, can be written as:

$$r(t) = \sum_{k \in \mathcal{K}} (\lambda_k^{[p]} r_k^{[p]}(t) + \lambda_k^{[d]} r_k^{[d]}(t)). \quad (3)$$

Based on the above reward functions, we formulate this problem as a Markov decision process (MDP). The objective is to find the optimal policy π^* that maximizes the expected cumulative reward (discounted sum of rewards), i.e.,

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r(t) \right]. \quad (4)$$

In this optimization problem, DUs are the environment; Near-RT RIC is the AI agent that makes policy decisions. UE QoS demands P_i and D_i , $i \in \mathcal{U}$, are given values that may change over time. UE QoS metrics ρ_i and τ_i are observable variables. Throughput and delay weights, $\lambda_k^{[p]}$ and $\lambda_k^{[d]}$, are given constants for all $k \in \mathcal{K}$.

IV. xDIFF: DESIGN

We propose xDiff to address the optimization problem in Eq. (4). Fig. 3 shows the key components of xDiff in the architecture of O-RAN. xDiff is an xApp residing within the Near-RT RIC, generating policies to guide the resource allocation at individual DUs. It should be stressed that the policy is generated by xDiff in Near-RT fashion (in the time scale of 10 ms to 1 s), while the resource is allocated at the DUs in real-time fashion (every one millisecond). Model input, output, and structure are three key components of xDiff, which we describe as follows.

Model Input. xDiff obtains the RAN state information from the DUs via the E2 interface and uses the information as the input to infer the policy for the resource allocation of individual DUs. The RAN state information includes both KPM and MAC data at each DU. Table II lists the KPM and MAC data that are used as input to xDiff. Denote B as the number of (KPM and MAC data) samples that xDiff obtains

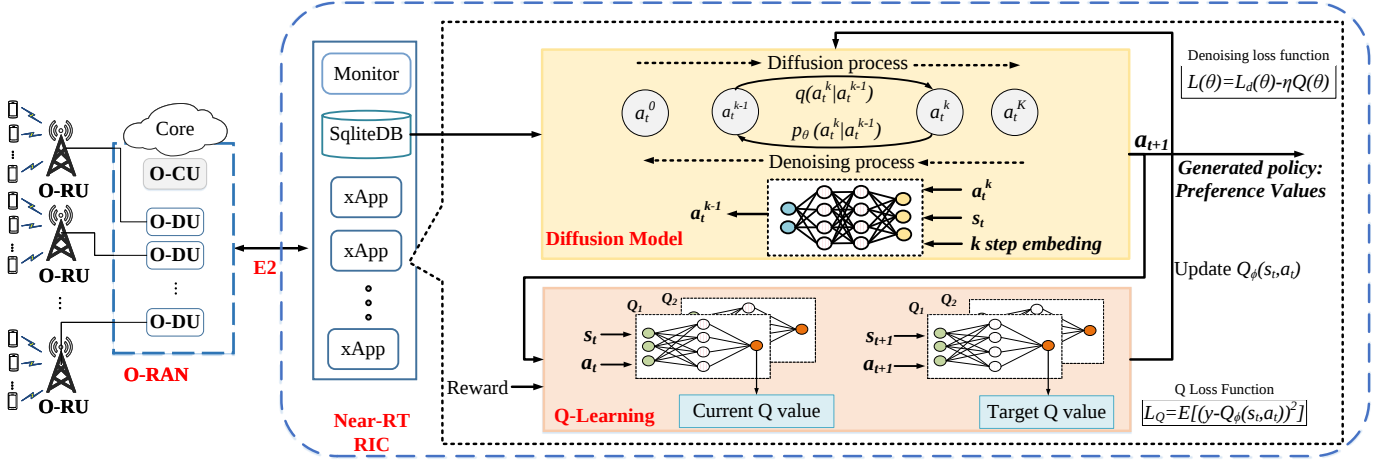


Fig. 3: Architecture of xDiff.

TABLE II: The list of KPM and MAC data that xDiff obtains for each UE.

	Data	UL or DL?	Explanation
KPM data	Per-UE TP	DL	Average data rate achieved by UE
	Per-UE delay	DL	Delay of SDU after being requested.
	Per-UE PRBs	DL	# of PRBs assigned to each UE in 10 ms.
MAC data	PUSCH SNR	UL	Quality of signal transmitted by UE.
	PHR	UL	Max Tx power - current usage power.
	MCS	DL&UL	Modulation index and coding rate for data.
	BLER	DL&UL	Percentage of blocks received with errors.
	Current TBs	DL	# of Transport Blocks being Tx-ed.
	Scheduled RBs	DL	# of PRBs scheduled for transmission.

per second from one DU. We observed that B varies from 100 to 1000 samples per second.

Model Output. The model output is a resource allocation policy for individual DUs, which plays a critical role in the management of inter-cell interference. The time-scale discrepancy between RIC and DU operations makes it infeasible for RIC to directly manage resource allocation for individual DUs, necessitating an efficient policy representation to bridge the near-real-time and real-time operations. The policy representation should capture both short-term and long-term network dynamics, including interference levels, channel conditions, user mobility, and user demands, to guide real-time resource allocation at DUs. Additionally, it should be lightweight to minimize communication overhead on the E2 interface. To address these challenges, we propose an elegant mathematical formulation as the policy representation, which will be explained in §IV-A.

Policy Agent Structure. Even with a well-defined policy representation, the policy agent design is a challenging task as it must generate a policy in a near-real-time manner. The policy agent must rapidly adapt to network dynamics while meeting stringent latency requirements. Additionally, it must ensure fast policy convergence, allowing the learning algorithm to quickly adjust to time-varying network conditions without prolonged training periods. Furthermore, due to the physical separation between RIC and DU, the policy agent in RIC operates with partial observability and incomplete network information, necessitating an efficient learning architecture for policy generation. To address the above challenges, we propose a diffusion model for policy generation within an RL framework, as shown in Fig. 3. Details will be presented in §IV-B.

A. Policy Representation

Each DU is responsible for the subframe-by-subframe (real-time) resource allocation of its own cell, in the presence of inter-cell interference. Due to the inter-DU communication latency, joint resource allocation cannot be achieved in a real-time fashion. Therefore, the DUs need to be coordinated by the Near-RT RIC for interference-aware resource allocation.

Hard Policy. A natural approach for the Near-RT RIC to coordinate resource allocation is by controlling the use of resource blocks at each DU. Specifically, the policy agent generates a *preference value* for each UE in each cell on each RB, i.e.,

$$p(DU_k, UE_i, RGB_j) = \begin{cases} 1, & \text{For } DU_k, \text{ allocate } RB_j \text{ to } UE_i, \\ 0, & \text{No preference, decide it by } DU_k, \\ -1, & \text{For } DU_k, \text{ do not allocate } RB_j \text{ to } UE_i, \end{cases} \quad (5)$$

where $i \in \mathcal{U}_k$, $k \in \mathcal{K}$, and $j \in \mathcal{J}$. For instance, $\mathcal{J} = \{1, 2, \dots, 106\}$ for 5G NR with 40 MHz bandwidth. Through learning the interference patterns in the network, the policy agent intends to use “1” and “-1” as the recommendation values to avoid strong inter-cell interference on the same RB. If there is no strong inter-cell interference on an RB, the policy agent does not make a recommendation for this RB, and the DU can allocate this RB using its own scheduling algorithm. The Near-RT RIC sends the generated recommendation values to each DU on the E2 interface. Upon receiving the recommendation values, DU_k follows the recommendation for its resource allocation.

While this policy has a simple representation, it does not perform well in practice as confirmed by our experiments. We observed that DUs frequently struggled to find sufficient RBs for UE scheduling, resulting in a high delay of UE communications. This may be attributed to its *hard* recommendation reflected by its discrete policy representation values, which explicitly excludes a subset of RBs for the UEs and thus limits the scheduling flexibility of DUs.

Soft Policy. Based on the experimental observations, we relax the policy representation values from discrete numbers to continuous numbers within the range from -1 to 1, i.e.,

$$p(DU_k, UE_i, RB_j) \in [-1, 1]. \quad (6)$$

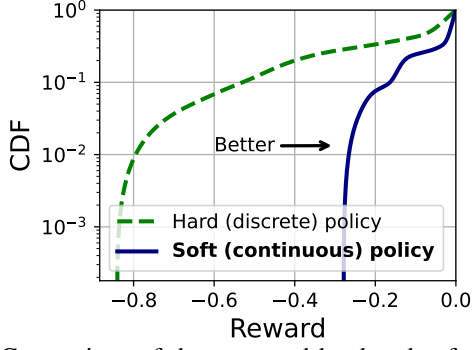


Fig. 4: Comparison of the proposed hard and soft policies.

A large value means that an RB has a high preference for the UE, while a small value means the RB has a low preference for the UE. We thus call the policy representation as *preference values*. This *soft* policy allows for flexible real-time resource allocation at the DUs while respecting the recommendation from the policy agent.

Experimental Comparison. We conduct experiments on a 5G O-RAN testbed that comprises three cells and ten smartphones (see §V for details). Each UE maintains a consistent position throughout the experiments while generating continuous data traffic demands. This controlled scenario allows us to isolate the impact of policy variations without mobility-induced effects. The hard and soft policy agents share the same model structure as described in §IV-B. Fig. 4 presents our measured reward. It is evident that the soft policy representation significantly outperforms its hard counterpart, echoing the above analysis.

B. Diffusion-Based Policy Generation

Based on the above policy representation, we propose a diffusion-based RL framework for online policy generation.

Why Diffusion Model for RL? Recently, diffusion-based RL has emerged as a strong candidate for the online decision-making process [58], [63]. It is well-suited for ICIM in O-RAN due to its ability to efficiently explore high-dimensional policy spaces while maintaining smooth and adaptive policy evolution. Unlike conventional RL approaches that may struggle with non-stationary interference patterns and dynamic resource constraints, diffusion-based RL leverages generative modeling techniques to learn a diverse distribution of optimal policies. This allows the policy agent to generalize across different network conditions and adapt to time-varying interference scenarios.

Consider two UEs, UE_1 and UE_2 , for example. Each UE belongs to a different small cell. To mitigate inter-cell interference, multiple optimal resource allocation solutions may exist. For example, $(UE_1 \rightarrow RB_1, UE_2 \rightarrow RB_2)$ is an optimal allocation; $(UE_1 \rightarrow RB_2, UE_2 \rightarrow RB_1)$ is another optimal solution. As illustrated in Fig. 5, the diffusion model is particularly well-suited for capturing the complex distribution of optimal actions. It outperforms conventional Gaussian policies by enabling more accurate policy generation and offering faster convergence.

Diffusion-Based RL Framework. This framework takes the KPM and MAC data in Table II as its input to generate

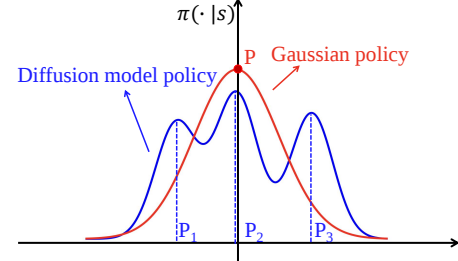


Fig. 5: Gaussian modeling vs. diffusion modeling.

$p(DU_k, UE_i, RGB_j)$ for all $k \in \mathcal{K}$, $i \in \mathcal{U}_k$, and $j \in \mathcal{J}$. As shown in Fig. 3, our RL framework integrates a diffusion model with double Q-learning networks to generate resource allocation policies (i.e., *preference values*). The diffusion model acts as the policy generator, while the Q-learning networks serve as the evaluator and critic to ensure policy quality and stability. We train the diffusion model using the Q-values to progressively denoise the *preference values*. At each denoising step, the model refines its output. We apply the principle of “Guidance for Maximizing the Q-Function” [64], [65], where the diffusion model prioritizes decisions that maximize the expected future reward (as indicated by the Q-value) during the optimization iterations. By adding this guidance at each iteration, the diffusion model is trained to maximize the reward.

Double Q-learning. In the diffusion-based RL framework, we use a value-based model as the evaluator (critic), which generates values to guide the training of the diffusion model. Traditional Q-learning algorithms select the maximum target Q-value to update the current Q-value. This may lead to overestimation of the Q-value due to noise and bias in data samples. To address this issue, we employ double-deep Q-learning networks to predict the Q-value of the policy generated by the diffusion model. Of these two Q-networks, one predicts the current Q-values while the other predicts the target Q-value.

To enhance the learning stability and efficiency, following the design in [66], we adopt two current Q-networks (Q_{ϕ_1}, Q_{ϕ_2}) for Q-value prediction and two target networks ($Q_{\phi'_1}, Q_{\phi'_2}$) to provide stable reference for current Q-value prediction. The smaller value from the two target networks is used to compute the target Q value. This redundancy helps reduce Q-value overestimation. Denote $Q_{\phi'_i}(s_{t+1}, \mathbf{a}_{t+1}^0)$ as the predicted value from the target Q network, where $\mathbf{a}_{t+1}^0 \sim \pi_{\theta'}$. Then, we compute the target Q-value as follows: $y = r(s_t, \mathbf{a}_t) + \gamma \min_{i=1,2} Q_{\phi'_i}(s_{t+1}, \mathbf{a}_{t+1}^0)$, where $r(s_t, \mathbf{a}_t)$ is the reward of taking action $\mathbf{a}_t$ at state s_t and γ is the discount rate. Based on the target Q value, we optimize ϕ_i , i.e., the weights in Q_i , by minimizing the following loss function:

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s_t, \mathbf{a}_t, s_{t+1}) \sim \mathcal{D}} \left[(y - Q_{\phi_i}(s_t, \mathbf{a}_t))^2 \right], i \in \{1, 2\}, \quad (7)$$

where $\mathcal{D}$ is the online dataset collected under policy π_θ .

Diffusion and Denoising Process. The training process of a diffusion model involves corrupting data with Gaussian noise at increasing levels and training a DNN to predict the added noise at each step. By minimizing the difference between the

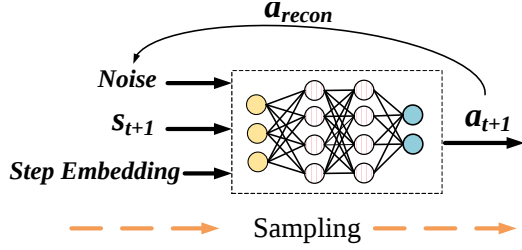


Fig. 6: Iterative sampling process of diffusion model.

predicted and actual noise, the model implicitly learns the score function, enabling it to generate new samples by reversing the diffusion process. The inference process of a diffusion model starts with pure Gaussian noise and iteratively denoises it using the trained model's noise predictions. By reversing the diffusion process step by step, it gradually reconstructs a high-quality sample from the learned data distribution. When using diffusion model within an RL framework, it should be stressed that there are two different timesteps: one for diffusion noise-adding/denoising steps, and the other for RL iterations. We use superscript $k \in \{1, \dots, K\}$ to denote diffusion step index and subscript $t \in \{1, \dots, T\}$ to denote RL iteration index.

In RL, the policy/action must be generated based on the current system state. This conditioning process is critical for policy generation. Therefore, we define the conditional diffusion policy as follows:

$$\pi_{\theta}(\mathbf{a} \mid \mathbf{s}) = p_{\theta}(\mathbf{a}^{0:K} \mid \mathbf{s}) = \mathcal{N}(\mathbf{a}^K; \mathbf{0}, \mathbf{I}) \prod_{k=1}^K p_{\theta}(\mathbf{a}^{k-1} \mid \mathbf{a}^k, \mathbf{s}), \quad (8)$$

where $\mathbf{a}$ and $\mathbf{s}$ are the action and state of an RL process, with k denoting its step index. $\mathcal{N}(\cdot; \cdot, \cdot)$ denotes Gaussian noise following the given parameters. Generally speaking, $p_{\theta}(\mathbf{a}^{k-1} \mid \mathbf{a}^k, \mathbf{s})$ could be modeled as a Gaussian distribution.

To solve the multi-optimization problem, we follow [55] by parameterizing $p_{\theta}(\mathbf{a}^{k-1} \mid \mathbf{a}^k, \mathbf{s})$ as a conditional noise prediction model (see Fig. 6). Once the final action $\mathbf{a}_t^K$ is obtained, it is used to determine the next action $\mathbf{a}_{t+1}$. The *preference values* update the objective function $O(\mathbf{s}, \mathbf{a})$, ensuring that the action selection process accounts for real-time network conditions. The optimization of the diffusion model is driven by a loss function, which balances denoising reconstruction and policy learning.

Following the Denoising Diffusion Probabilistic Model (DDPM) in [52], we train our conditional ϵ -model—a Multi-Layer Perceptron (MLP) parameterized by θ —based on the below loss function:

$$\mathcal{L}_d(\theta) = \mathbb{E}_{k \sim U(K), \epsilon \sim \mathcal{N}(0, \mathbf{I})} [\|\epsilon - \epsilon_{\theta}(\sqrt{\alpha_k} \mathbf{a} + \sqrt{1 - \alpha_k} \epsilon, \mathbf{s}, k)\|^2],$$

where ϵ is the noise, $U(K)$ is a uniform distribution over the discrete set as $\{1, \dots, K\}$ and $\mathcal{D}$ denotes the data samples in the database.

To improve learning efficiency, we inject the Q-value function from the Q-learning networks into the denoising process. Specifically, following the approach in [55], we define

Algorithm 1 xDiff: Diffusion-based RL algorithm.

Initialize: Policy network π_{θ} ; critic networks Q_{ϕ_1}, Q_{ϕ_2} ; target networks $\pi_{\theta'}, Q_{\phi'_1}, Q_{\phi'_2}$.

Input: Action $\mathbf{a}_t$; current state $\mathbf{s}_t$; and next state $\mathbf{s}_{t+1}$.

Output: preference value $\mathbf{a}_{t+1}$.

```

1: // Database initialization
2:  $\mathcal{D} \leftarrow \emptyset$ 
3: for  $b \in \{0, \dots, \text{batchsize}\}$  do
4:   Generate  $\mathbf{a}_t^0 \sim \pi_{\theta'}(\mathbf{a}_t | \mathbf{s}_t)$  by Eq. (8).
5:   Send  $\mathbf{a}_t^0$  to RAN and get  $\mathbf{s}_{t+1}$  and  $r_t$  from RAN
6:    $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathbf{s}_t, \mathbf{a}_t^0, r_t, \mathbf{s}_{t+1}\}$ 
7: end for
8: // Online policy learning and generation
9: for each iteration do
10:  Repeat steps 4–6
11:  Sample mini-batch  $B = \{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})\} \sim \mathcal{D}$ 
12:  Sample  $\mathbf{a}_{t+1}^0 \sim \pi_{\theta'}(\mathbf{a}_{t+1} | \mathbf{s}_{t+1})$  by Eq. (8).
13:  Update  $Q_{\phi'_1}$  and  $Q_{\phi'_2}$  based on Eq. (7)
14:  Update policy by minimizing Eq. (9)
15:   $\theta' \leftarrow \rho\theta + (1 - \rho)\theta'$ 
16:   $\phi'_i \leftarrow \rho\phi_i + (1 - \rho)\phi'_i$  for  $i \in \{1, 2\}$ 
17: end for

```

$\mathcal{Q}(\theta) = \frac{\mathbb{E}_{\mathbf{s} \sim \mathcal{D}, \mathbf{a}^0 \sim \pi_{\theta'}[\mathcal{Q}(\mathbf{s}, \mathbf{a}^0)]}}{\mathbb{E}_{(\mathbf{s}, \mathbf{a}) \sim \mathcal{D}}[|\mathcal{Q}(\mathbf{s}, \mathbf{a})|]}. Then, the loss function that we use to train the diffusion network is as follows:$

$$\mathcal{L}(\theta) = \mathcal{L}_d(\theta) - \eta \mathcal{Q}(\theta), \quad (9)$$

where η is a hyperparameter that balances the two loss terms. η plays a key role in model training. We will study it through experiments.

Sampling Process. Sampling refers to the process of generating new action (i.e., new policy for DUs' resource allocation) by reversing a noise-injection process, gradually refining random noise into a coherent output by following learned diffusion steps. Fig. 6 illustrates our sampling process. Initially, the input includes randomly generated noise, state $\mathbf{s}_{t+1}$, and the embedding of denoising step index k . The state information acts as a conditional guide, directing the network through iterative denoising processes to generate the action/policy $\mathbf{a}_{t+1}$ for state $\mathbf{s}_{t+1}$.

Summary of Workflow. The complete workflow of xDiff is detailed in Alg. 1. The parameter ρ controls the soft update of target networks using exponential moving average (EMA), ensuring a gradual adaptation to the latest policy while maintaining stability. A small ρ (e.g., 0.05) prevents abrupt changes.

C. Policy-Guided Resource Allocation at Individual DUs

The policy generated by xDiff must be taken by the DUs to guide their resource allocation. Fig. 7 shows the architecture of an DU, where a real-time scheduler operates at the MAC layer for resource allocation and user scheduling. In what follows, we first introduce the conventional UE scheduling algorithm in DU, and then explain how the policy generated by xDiff is integrated into the existing scheduling algorithm.

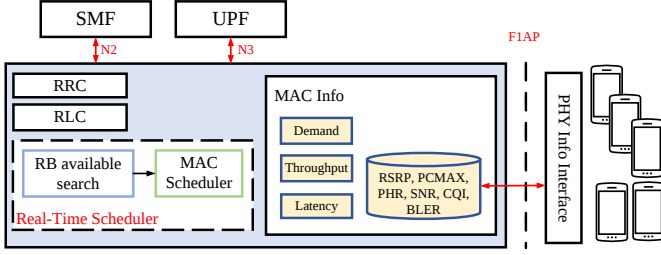


Fig. 7: Architecture of 5G DU.

Proportional Fairness (PF) Scheduler. PF is a popular scheduler that has been widely used in real-world cellular networks. It allocates resources to users based on a balance between maximizing throughput and ensuring fairness, by giving more resources to users with higher channel quality while preventing the system from favoring only the best users. It aims to optimize network performance while maintaining equitable access for all users. Specifically, the PF scheduler calculates a metric for each user, which is a ratio of their instantaneous throughput (or CQI) to their historical throughput. This metric is used to decide which user should be allocated resources. The metric for user i at time slot t is:

$$\text{PF_Metric}_i(t) = \frac{r_i(t)}{R_i(t-1)}, \quad (10)$$

where $r_i(t)$ is the instantaneous achievable data rate of user i at time t , which is predicted based on the CQI feedback. $R_i(t)$ is the average historical throughput of user i in the previous time slot. The PF scheduler computes the PF metric for each user in each time slot, and selects the user with the highest PF metric for resource allocation.

Integrate Diffusion Policy into PF Scheduler. The MAC scheduler receives the policy (i.e., *preference values*) generated by the diffusion model via the E2 interface and allocates RBs to UEs according to the policy. Specifically, for each UE, the scheduler aims to utilize the RBs with a high *Preference Value* while avoiding those with a low *Preference Value*. The scheduler achieves this objective through the below two steps.

- **Step I: Adjust PF Metric.** Recall that $p(DU_k, UE_i, RB_j)$ indicates the UE_i 's preference for RB_j . A negative value means that this RB is not favorable due to the inter-cell interference. We wish to exclude those RBs when calculating a UE's PF Metric. To do so, we define a weight for UE i by letting $w_i = \frac{|\{j \in \mathcal{J} : p(DU_k, UE_i, RB_j) < 0\}|}{N_{RB}}$. It represents the percentage of favorable RBs for UE_i . Then, we incorporate the UE weight into the PF metric as follows.

$$\text{PF_Metric}_i(t) = \frac{r_i(t)}{R_i(t-1)} \times w_i. \quad (11)$$

We note that, since DUs perform resource allocation independently, we omit DU index k for simplicity.

- **Step II: RB Allocation.** The scheduler first prioritizes all UEs based on their PF_Metric values and then allocates RB for each UE based on their priority. For each selected UE, the scheduler assigns RBs with the highest *preference values* for transmission. This allocation process continues iteratively until either the transmission demand of the current UE is fulfilled or all available RBs have been allocated.

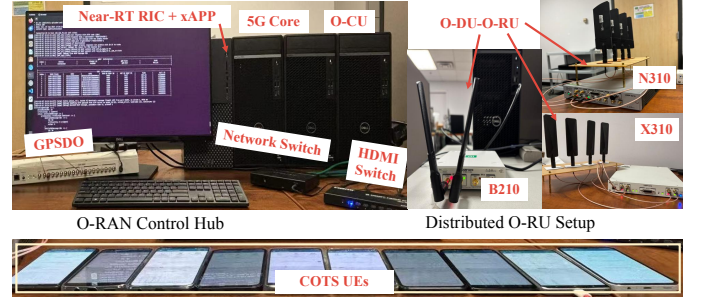


Fig. 8: A 5G O-RAN testbed that includes three RUs and tens of smartphones.

V. IMPLEMENTATION

Testbed Setup. Fig. 8 illustrates our O-RAN experimental testbed, which consists of a 5G core network, one CU, three DUs, three RUs, Near-RT RIC, and ten commercial smartphones. The system operates on the n78 frequency band. The center frequency is 3319.68 MHz, and the subcarrier spacing is 30 kHz. The testbed is configured so that all smartphones can access the Internet. The three RUs were implemented using different USRP devices—N310, X310, and B210—to emulate the diversity of commercial RU equipment. Both N310 and X310 support 2x2 MIMO, while B210 supports 1x1 transmission only. The three USRP are synchronized in both frequency and time for TDD operations on the n78 band. The ten smartphones are from various vendors, including Google Pixel, OnePlus, Motorola, Xiaomi, and Samsung.

OpenAirInterface (OAI) Modifications. We use OAI [67] 5G RAN for our experiments. The DU then assigns a Radio Network Temporary Identifier (RNTI) to each of the newly admitted UEs as identification. OAI only supports the PF algorithm, which cannot meet our requirements. Therefore, we modified the downlink scheduler functions in `oai/openair2/LAYER2/NR_MAC_gNB` and the E2 interface in `oai/openair2/E2AP/RAN_FUNCTION`.

Near-RT RIC. We use Mosaic5g Flexric [68]–[70] as our near-RT RIC. Flexric supports E2 Node agent, near-RT RIC, and xApp. It provides a flatbuffers encoding/decoding scheme as an alternative to ASN.1. We use SWIG as an interface generator to enable C/C++ and Python development for xApps. We built our xApp with E2AP v2.03 and KPM v2.03.

xDiff Implementation. We built an xAPP within the Near-RT RIC for our conditional diffusion model using a fully-connected DNN following the DDPM method in [52]. To accelerate the online training process of the diffusion model and meet the timing requirements, we grouped the 106 RBs into 10 clusters for the diffusion model to generate the *preference values*. We employed a DNN with a 4-layer MLP and mish activations for the diffusion model. We use 256 hidden units for all layers. The input of $\epsilon(\theta)$ is the concatenation of the last step action vector, the current state vector, and the sinusoidal positional embedding of timestep k . The output of $\epsilon(\theta)$ is the predicted residual at diffusion timestep K . For the double Q-learning networks, we adopt a similar MLP architecture as the diffusion policy; however, we use 4-layer MLPs with 256 hidden units. For xDiff, we need to define a BWP for each UE, including its starting position and bandwidth size. The starting

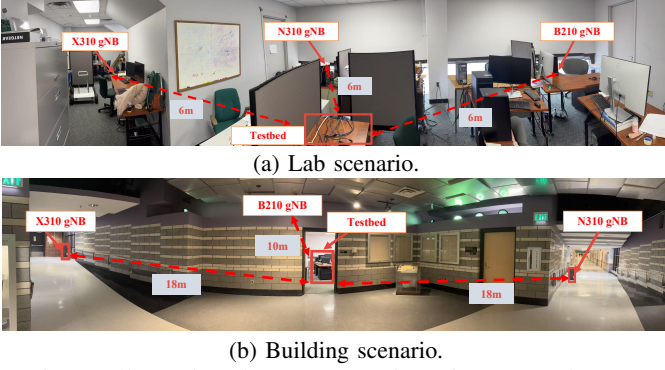


Fig. 9: Illustrating the two scenarios of our experiments.

position can be obtained through xDiff, while the bandwidth size is determined by the *preference* values and traversing through all the RBs.

Open-Source Code: The source code of xDiff is available on GitHub [1].

VI. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of xDiff in lab and building scenarios as shown in Fig. 9. We use three metrics for the evaluation: throughput, delay, and the defined reward function. We note that the reward value is non-positive as it was defined to be the inverse of throughput/delay regret. We would like to answer the following questions through the evaluation.

- **Q1:** How are KPM and MAC data affected by inter-cell interference? This is important as xDiff generates Near-RT ICIM policy based on KPM and MAC data.
- **Q2:** Is diffusion model critical for xDiff? What are the best values for its key parameters?
- **Q3:** How does xDiff perform compared to the state-of-the-art ICIM solutions?

A. A Case Study

We consider a simple network in the lab scenario as a case study to examine the operations of xDiff.

Two-Cell Case. We first consider a network with two cells, each serving a single smartphone. The core network generates persistent data traffic for the two smartphones at 50 Mbps and 60 Mbps, respectively. We aim to observe the MAC-layer data and evaluate its performance under inter-cell interference. To do this, we first activate smartphone 1 at time moment T_1 , followed by smartphone 2 at time moment T_2 . We collect smartphone 1's MAC-layer data at its serving DU, including its Power Headroom (PHR), Channel Quality Indicator (CQI), Modulation and Coding Scheme (MCS), BLER, throughput (TP), and queueing delay.

Fig. 10 presents our measurement data, with the time moments T_1 and T_2 marked along the x-axis. We make the following observations. Between T_1 and T_2 , since only one smartphone is active, there is no inter-cell interference. As a result, smartphone 1 achieves ideal connection performance. During this period, it maintains a stable link with high PHR

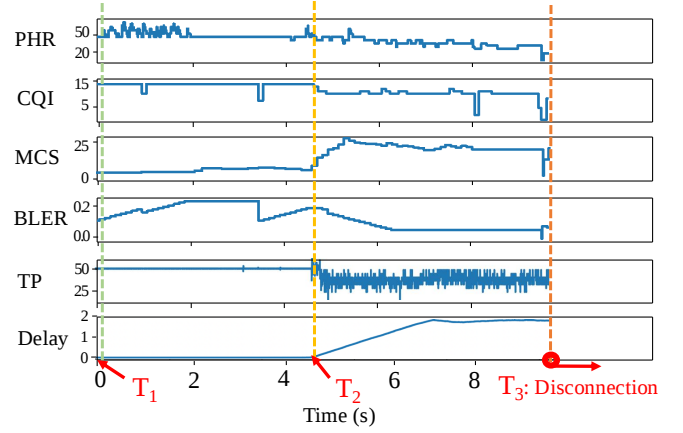


Fig. 10: MAC-layer observations. UE_1 and UE_2 start their downloading requests at 50 Mbps at T_1 and T_2 , respectively. Throughput (TP) was measured in Mbps, and delay was measured in seconds.

TABLE III: RU power parameters.

	USRP	In-sync PH (dB)	PCMAX (dBm)	Average RSRP (dBm)
$UE_1 \in gNB_1$	B210	55	17	-88
$UE_2 \in gNB_2$	N310	42	21	-73
$UE_3 \in gNB_3$	X310	32	17	-89

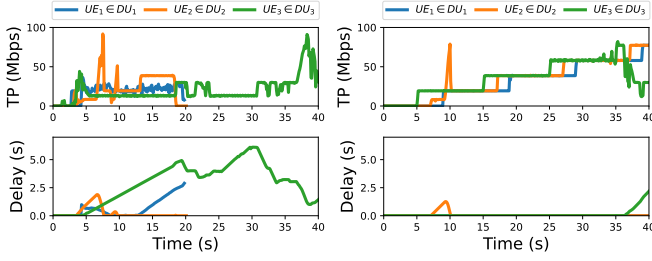
values of around 50, high CQI values of around 15, and consistent MCS values. Its real throughput remains stable at 50 Mbps, consistently meeting its demand. The queueing delay stays low at around a few milliseconds.

From time moment T_2 , the network has two active smartphones with aggressive traffic demands, generating significant inter-cell interference. The impact of interference is reflected in smartphone 1's MAC data. As shown in Fig. 10, since time moment T_2 , smartphone 1 experiences a considerable performance drop, including decreased PHR, declined CQI, reduced throughput, and increased delay. At time moment T_3 , smartphone 1 disconnects from the network due to the overwhelming interference.

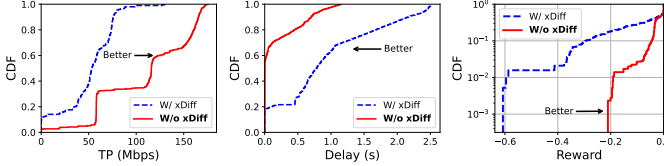
These observations confirm the underlying relationship between inter-cell interference and MAC data, supporting our design of the diffusion model that generates ICIM policies based on MAC/KPM observations.

Three-Cell Case. We now examine the impact of interference in a three-cell network, where each cell serves a single UE (smartphone). Table III presents the key measured parameters of the three UEs, including their in-sync PHR, maximum transmission power (PCMAX), and average Received Signal Reference Power (RSRP). These measurements reflect the link quality of each UE. For each UE in each cell, we gradually increased its throughput demand from 0 to 80 Mbps using the *iperf* and measured its throughput and delay performance in the network with and without xDiff.

Fig. 11 presents an instance of our measurement results. We make the following observations. When the three network cells allocate resources independently (i.e., without xDiff), the throughput of the three UEs fluctuates dramatically and frequently falls below their throughput demands. This is due to the independent resource allocation at each cell, which leads



(a) W/o xDiff. (b) W/ xDiff.
Fig. 11: Performance comparison of O-RAN w/ and w/o xDiff.



(a) Throughput (b) Delay (c) Reward
Fig. 12: Measured performance in the case study.

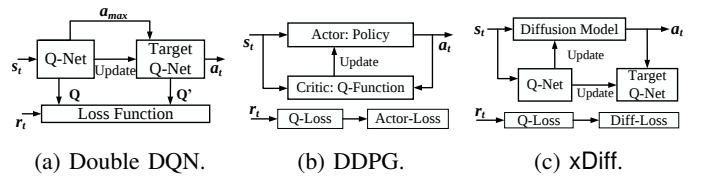
to dynamic inter-cell interference. Compared to throughput, the impact of interference on communication delay is even more significant. The delay for one UE spikes to as high as 5 seconds. More critically, inter-cell interference causes UE disconnections when each UE continues to increase its throughput demand. As shown in the figure, UE_1 and UE_2 lose connection at the 20-second mark due to severe interference. In contrast, the network with xDiff exhibits much more stable throughput and delay performance. Moreover, the network meets the throughput and delay demands of the three UEs most of the time, and no UE disconnections are observed. This demonstrates the effectiveness of xDiff in ICIM.

We repeated the above measurements multiple times, placing the three smartphones in different locations to collect their throughput and delay data. Fig. 12 presents our measurement results. It is evident that using xDiff significantly improves the network's throughput, delay, and reward performance. This further confirms the effectiveness of xDiff in ICIM.

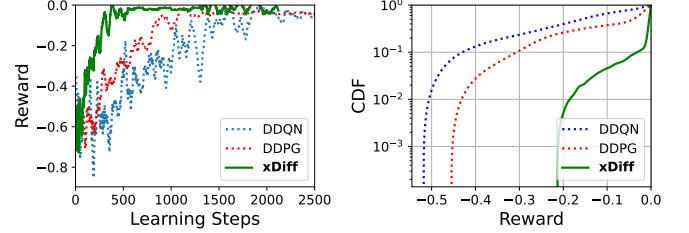
B. Ablation and Parameter Studies

Ablation Study. xDiff consists of multiple key components, including a diffusion model and Q-networks. To ensure that the performance gain in xDiff is primarily attributed to the diffusion model rather than other components, we conduct an ablation study. Specifically, we compare xDiff with two variant models obtained by removing its diffusion components. Fig. 13 highlights the network architecture comparison between xDiff and its two counterpart models, which we describe as follows.

- **Double Deep Q-learning Algorithm (DDQN):** We remove the diffusion model in xDiff to create this DDQN model. It uses a Q-network to generate the actions and another Q-network to predict the target function values. Similar to xDiff, it employs double Q-networks to mitigate overestimation bias. The target Q-network is updated based on the next state and the loss function is computed using the reward signal.
- **Deep Deterministic Policy Gradient (DDPG):** DDPG is a specialized Actor-Critic RL method designed for continuous



(a) Double DQN. (b) DDPG. (c) xDiff.
Fig. 13: Comparison of Different RL Algorithms



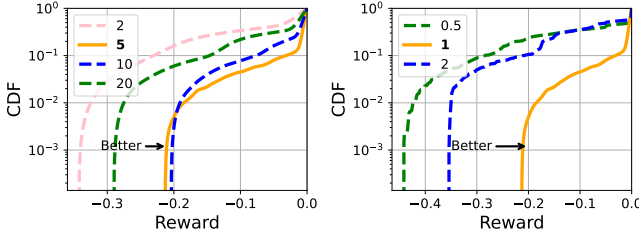
(a) Convergence. (b) Reward.
Fig. 14: Experimental results of ablation study.

action spaces. It is a model-free actor-critic framework, consisting of an actor network implemented as an MLP and a critic network that evaluates the Q-function. It does not have a diffusion model but also uses two Q-networks to improve stability.

We implemented these three policy generation models on our testbed and evaluated their performance under identical scenarios. The experiments involved three cells and ten smartphones. Fig. 14 presents our experimental results in terms of convergence speed and reward distribution. Fig. 14a shows that xDiff achieves the fastest convergence speed when there is a change in the network state (e.g., increased UE throughput demands). Fig. 14b demonstrates that xDiff attains the highest reward compared to its two counterparts. These observations confirm that xDiff outperforms its ablated counterparts and highlight the critical role of the diffusion model in generating efficient policies.

Denoising Step Number K . The number of denoising steps is a key parameter for diffusion models, directly affecting their generation performance. In time-insensitive tasks such as image and video generation, a large number of denoising steps is preferable as it generally leads to better generation quality. However, policy-making for ICIM in O-RAN is a time-sensitive task. A small number of denoising steps weakens the diffusion model's learning ability, resulting in suboptimal policies and degraded network performance. Conversely, a large number of denoising steps delays policy updates at DUs, leading to outdated policies that hinder resource allocation. To address this trade-off, we conduct experiments to empirically determine the optimal number of denoising steps that balance performance and computational delay.

Fig. 15a presents our experimental results. When increasing the number of denoising steps from 2 to 5, the network reward improves significantly due to enhanced policy quality. Beyond 5 steps, the reward gain diminishes; increasing the steps from 5 to 10 yields little improvement because the policy update delay at the DU offsets the benefit of improved policy quality. When the step number increases from 10 to 20, the reward declines.



(a) # of denoising steps, i.e., N . (b) Loss-balance parameter η .
 Fig. 15: Ablation study of hyperparameters

This occurs because the additional steps provide marginal policy enhancement while significantly prolonging computation at the Near-RT RIC, leading to outdated policies and reduced network performance. Based on these observations, we set the number of denoising steps to 5 in our extensive experiments.

Loss-Balancing Parameter η . The loss function plays a critical role in training the diffusion model, influencing both convergence speed and generation quality. As shown in Eq. (9), xDiff employs a combination of two loss functions: the diffusion model's internal loss $\mathcal{L}_d$ and an external loss $\mathcal{L}_q$. The hyperparameter η controls the balance between these two loss components during training. We conduct experiments to evaluate the impact of η on network performance.

Fig. 15b presents our experimental results for $\eta = 0.5, 1$, and 2. The results indicate that $\eta = 1$ yields the best network performance. We also observed that a lower η (e.g., 0.5) results in unstable network performance, whereas a higher η (e.g., 2) slows down convergence. Based on these findings, we set $\eta = 1$ in our extensive experiments.

C. Performance Comparison

We now evaluate the performance of xDiff by comparing it against existing methods, which we describe as follows.

- **Qualcomm's IAIS method [24].** The Interference-aware Intelligent Scheduling (IAIS) method proposes an ML-based interference prediction technique that utilizes CSI reported by 5G UE [43], [71], enabling it to schedule transmissions over dynamic air interfaces intelligently.
- **CSRS method [41]:** This is a Cellular Spectrum Resource Sharing (CSRS) algorithm that dynamically allocates spectrum resources in cellular networks using a cluster-based approach. Center UEs share spectrum, while edge UEs use different channels. The resource allocated to a UE is proportional to its demand to minimize ICI.
- **Cell-Independent Resource Allocation (CIRA):** This is a widely-used approach where no inter-cell coordination exists for interference management. Instead, each DU allocates the entire set of available RBs to its UEs using the PF scheduler.
- **One-Third Frequency Reuse (OTFR):** In this method, the frequency spectrum is allocated equally among three DU-cells, with each cell receiving one-third of the total spectrum. This fixed allocation ensures that each cell has a dedicated portion of the spectrum, with minimal interference between them.

Lab-Scale Scenario. Fig. 9a shows the testbed setup in a lab scenario for this evaluation. This is a scenario for typical

indoor 5G small cells, rich in obstacles, reflectors, blockage, human routing activities, and UE mobility. The experiments involve three cells and a total of ten smartphones. During the experiments, we use *iperf* to generate time-varying data traffic for each smartphone, ranging from light traffic (35 Mbps) to heavy traffic (165 Mbps). We repeated the same experiments for the five ICIM methods in the same settings and collected data for 5 hours.

Fig. 16 plots the throughput, queueing delay, BLER, and reward performance of the five ICIM methods. **Throughput:** It can be observed that xDiff and CSRS achieve similar performance, both successfully meet the throughput demands of the UEs over time. In contrast, the throughput of the other three methods falls behind xDiff and CSRS, indicating that they fail to meet the UEs' throughput demands over time. **Queueing Delay:** The results show that xDiff achieves superior delay performance compared to the other four methods. This can be attributed to the fast adaptation of the diffusion model under dynamic network conditions. **BLER:** It can be seen that CIRA achieves the highest BLER, while OTFR achieves the lowest BLER. xDiff yields moderate BLER compared to other methods. We note that a low BLER does not necessarily indicate better performance from a networking perspective. Instead, it may indicate that the transmission is not aggressive enough to explore the channel capacity via MCS adaptation. **Reward:** While throughput, delay, and BLER were measured during the experiments, the reward was calculated based on the measured throughput and delay values. It can be observed that xDiff considerably outperforms CSRS and is significantly better than the other three methods in terms of reward. Again, this is attributed to the strong adaptability and robustness of the diffusion model in xDiff in dynamic networks under time-varying interference conditions.

Building-Scale Scenario. We increase the cell radius by placing RUs at different locations in a building. The distance between two RUs is about 18 m, with thick concrete walls in between. We use the existing Wi-Fi network to establish the E2 interface between Near-RT RIC and DUs, with a communication delay of about 200 ms. Compared to the lab scenario, this setting has a lighter interference due to the larger cell size. We repeated the same experiments from the previous scenario to measure the performance of the five ICIM methods.

Fig. 17 plots the measured per-UE throughput, queueing delay, BLER, and the calculated reward, based on which we have the following observations. **Throughput:** The experimental results show that xDiff offers better throughput performance compared to the other four ICIM methods. However, unlike the previous scenario, the throughput gain of xDiff is marginal. This could be attributed to the light interference in the building-scale scenario. **Queueing Delay:** xDiff offers the minimum delay compared to the other four ICIM methods. This observation is consistent with that in the previous scenario. **BLER:** xDiff has a moderate BLER compared to the other methods. This observation is also consistent with the previous scenario. **Reward:** xDiff considerably outperforms IAIS and significantly outperforms the other three methods. This confirms that xDiff is superior to the SOTA methods in both strong and light inter-cell interference networks.

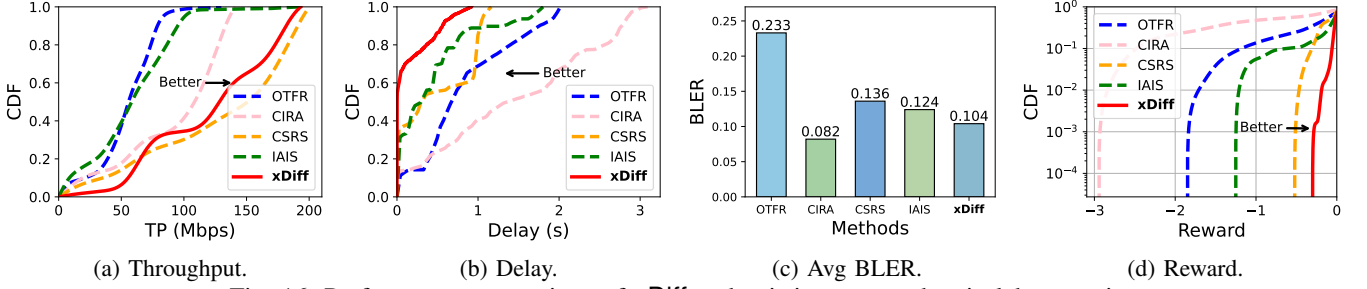


Fig. 16: Performance comparison of xDiff and existing approaches in lab scenario

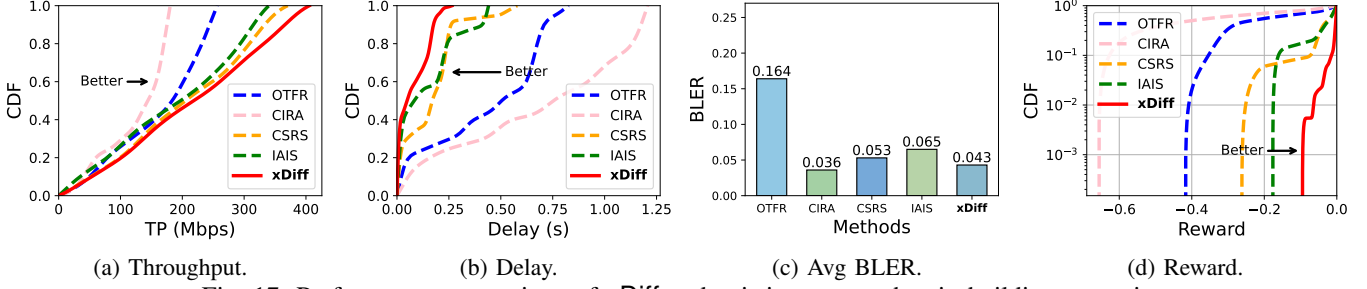


Fig. 17: Performance comparison of xDiff and existing approaches in building scenario

TABLE IV: Comparison of computational complexity and inference time.

Method	Computational Complexity	Inference Time
OTFR	$\mathcal{O}(N_{\text{UEs}} \cdot N_{\text{RBs}})$	Subframe ($\leq 1\text{ms}$)
CIRA	$\mathcal{O}(1)$	Subframe ($\leq 1\text{ms}$)
CSRS [41]	$\mathcal{O}(K \cdot N_{\text{UEs}} \cdot \log N_{\text{UEs}})$	5.4 ms
IAIS [24]	$\mathcal{O}(N_{\text{UEs}} \cdot f_{\text{ML}})$	38.3 ms
xDiff	$\mathcal{O}(N_{\text{Denoising Steps}} \cdot N_{\text{UEs}} \cdot N_{\text{RBs}})$	21.8 ms

Comparison of Computational Complexity: We now compare the computational complexity and inference time (i.e., policy generation time) of xDiff against the other four methods. Table IV presents our results, where the inference time was measured on a desktop with a 14th gen i9 CPU. Since CIRA and OTFR are rule-based scheduling methods, they offer low computational complexity and can be completed within one subframe. CSRS demonstrates moderate complexity with an average inference time of 5.4 ms, striking a good balance between performance and computational complexity. IAIS relies on machine learning for interference prediction. It has high computational complexity with an observed inference time of 38.3 ms. xDiff has an average inference time of 21.8 ms. All these methods meet the Near-RT requirements in O-RAN systems.

VII. CONCLUSION

In this paper, we presented xDiff, an online learning-based xApp for ICIM in 5G O-RANs. xDiff consists of two key components: a diffusion model and an RL framework. We formulated the ICIM problem as a reward optimization problem and employed a diffusion-based RL framework for resource allocation policy generation. To address the time-scale discrepancy between the Near-RT RIC and real-time DUs, we introduced a new concept, the *preference values*, as the policy representation to bridge the operations of the RIC

and DUs. We implemented xDiff on a 5G testbed and evaluated its performance against state-of-the-art methods. Experimental results demonstrate that xDiff outperforms existing methods, highlighting the potential of diffusion models for the online optimization of O-RAN.

REFERENCES

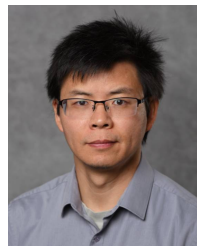
- [1] P. Yan, "O-RAN ICIM xDiff Source Code." https://github.com/peihaoY/Diffusion_5G_ORAN, 2025. Accessed on February-10-2025.
- [2] O-RAN ALLIANCE, "O-RAN Empowering Vertical Industry: Scenarios, Solutions and Best Practice." <https://www.o-ran.org>, December 2023. O-RAN WG1 Vertical Industry White Paper.
- [3] R. A. K. Fezeu, C. Fiandrino, E. Ramadan, J. Carpenter, L. C. De Freitas, F. Bilal, W. Ye, J. Widmer, F. Qian, and Z.-L. Zhang, "Unveiling the 5g mid-band landscape: From network deployment to performance and application qoe," in *Proceedings of the ACM SIGCOMM 2024 Conference*, (New York, NY, USA), pp. 358–372, Association for Computing Machinery, 2024.
- [4] A. Hassan, A. Narayanan, A. Zhang, W. Ye, R. Zhu, S. Jin, J. Carpenter, Z. M. Mao, F. Qian, and Z.-L. Zhang, "Vivisection mobility management in 5g cellular networks," in *Proceedings of the ACM SIGCOMM 2022 Conference*, (New York, NY, USA), pp. 86–100, Association for Computing Machinery, 2022.
- [5] A. Narayanan, X. Zhang, R. Zhu, A. Hassan, S. Jin, X. Zhu, X. Zhang, D. Rybkin, Z. Yang, Z. M. Mao, F. Qian, and Z.-L. Zhang, "A variegated look at 5g in the wild: performance, power, and qoe implications," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, SIGCOMM '21, (New York, NY, USA), p. 610–625, Association for Computing Machinery, 2021.
- [6] D. Lopez-Perez, X. Chu, and İ. Guvenc, "On the expanded region of picocells in heterogeneous networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 3, pp. 281–294, 2012.
- [7] D. Lopez-Perez, X. Chu, A. V. Vasilakos, and H. Claussen, "Power minimization based resource allocation for interference mitigation in ofdma femtocell networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 2, pp. 333–344, 2013.
- [8] C. Altay and M. Koca, "Design and analysis of energy efficient inter-tier interference coordination in heterogeneous networks," *Wireless Networks*, vol. 27, no. 6, pp. 3857–3872, 2021.
- [9] J. A. Ayala-Romero, J. J. Alcaraz, A. Zanella, and M. Zorzi, "Online learning for energy saving and interference coordination in hetnets," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1374–1388, 2019.

- [10] M. Yan, J. Yang, K. Chen, Y. Sun, and G. Feng, "Self-imitation learning-based inter-cell interference coordination in autonomous hetnets," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4589–4601, 2021.
- [11] S. Chaudhari and H. Kwon, "Machine learning based interference whitening in 5g nr mimo receiver," in *2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring)*, (Helsinki, Finland), pp. 1–6, IEEE, 2022.
- [12] Y. Li, X. Lei, P. Fan, and D. Chen, "An scma-based uplink inter-cell interference cancellation technique for 5g wireless systems," in *2015 International Conference on Wireless Communications & Signal Processing (WCSP)*, (Nanjing, China), pp. 1–5, IEEE, 2015.
- [13] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, "Algorithms for enhanced inter-cell interference coordination (eicic) in lte hetnets," *IEEE/ACM transactions on networking*, vol. 22, no. 1, pp. 137–150, 2013.
- [14] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, "Coordinated multipoint transmission and reception in lte-advanced: deployment scenarios and operational challenges," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 148–155, 2012.
- [15] J. Gu, S. J. Bae, B.-G. Choi, and M. Y. Chung, "Dynamic power control mechanism for interference coordination of device-to-device communication in cellular networks," in *2011 Third International Conference on Ubiquitous and Future Networks (ICUFN)*, (Dalian, China), pp. 71–75, IEEE, 2011.
- [16] Q. Li, Z. Zhang, Y. Liu, Z. Tan, C. Peng, and S. Lu, "Ca++: Enhancing carrier aggregation beyond 5g," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, (New York, NY, USA), pp. 1–14, Association for Computing Machinery, 2023.
- [17] W. Ye, X. Hu, S. Slender, A. Zhang, U. K. Dayalan, A. Hassan, R. A. K. Fezeu, A. Ajoo, M. Lee, E. Ramadan, F. Qian, and Z.-L. Zhang, "Dissecting carrier aggregation in 5g networks: Measurement, que implications and prediction," in *Proceedings of the ACM SIGCOMM 2024 Conference*, ACM SIGCOMM '24, (New York, NY, USA), p. 340–357, Association for Computing Machinery, 2024.
- [18] S. Naser, L. Bariah, S. Muhaidat, M. Al-Qutayri, M. Uysal, and P. C. Sofotasios, "Interference management strategies for multiuser multicell mimo vlc systems," *IEEE Transactions on Communications*, vol. 70, no. 9, pp. 6002–6019, 2022.
- [19] Y. Liu, C. S. Chen, and C. W. Sung, "Joint optimization on inter-cell interference management and user attachment in lte-a hetnets," in *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, (Mumbai, India), pp. 62–69, IEEE, 2015.
- [20] R. M. Sohaib, S. T. Shah, O. Onireti, Y. Sambo, Q. H. Abbasi, and M. Imran, "Drl-based joint resource scheduling of embb and urllc in o-ran," in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, (Denver, CO, USA), pp. 1523–1528, IEEE, 2024.
- [21] M. Elsayed, K. Shimotakahara, and M. Erol-Kantarci, "Machine learning-based inter-beam inter-cell interference mitigation in mmwave," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, (Dublin, Ireland), pp. 1–6, IEEE, 2020.
- [22] Z. Wang, Y. Zhou, Y. Shi, and W. Zhuang, "Interference management for over-the-air federated learning in multi-cell wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 8, pp. 2361–2377, 2022.
- [23] M. V. Ngo, Y. H. Pua, T.-L. Le, B. Chen, T. Quek, *et al.*, "Ai-driven rapps for reducing radio access network interference in real-world 5g deployment," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, (Vancouver, BC, Canada), pp. 1–2, IEEE, 2024.
- [24] B. Akgun, D. S. M. Singh, S. Kotla, V. Jain, S. Namdeo, R. Acharya, M. Jayabalan, A. Kumar, V. Chande, A. Kannan, J. Swami, Y. Chen, J. Boyd, and X. Zhang, "Interference-aware intelligent scheduling for virtualized private 5g networks," *IEEE Access*, vol. 12, pp. 7987–8003, 2024.
- [25] D. Anand, M. A. Togou, and G.-M. Muntean, "A machine learning-based approach for interference mitigation to enhance qos and qoe in 5g o-ran networks," in *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, (Valencia, Spain), pp. 1–6, IEEE, 2024.
- [26] D. Anand, M. A. Togou, and G.-M. Muntean, "A machine learning-based xapp for 5g o-ran to mitigate co-tier interference and improve qoe for various services in a hetnet environment," in *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, (Beijing, China), pp. 1–6, IEEE, 2023.
- [27] D. Anand, M. A. Togou, and G.-M. Muntean, "Enhancing qoe diversity in hetnets through interference mitigation with ml-based xapp in a 5g o-ran architecture," in *ICC 2024 - IEEE International Conference on Communications*, (Denver, CO, USA), pp. 5473–5478, IEEE, 2024.
- [28] M. Simsek, M. Bennis, and A. Czylik, "Dynamic inter-cell interference coordination in hetnets: A reinforcement learning approach," in *2012 IEEE Global Communications Conference (GLOBECOM)*, (Guilin, China), pp. 5446–5450, IEEE, 2012.
- [29] V. H. L. Lopes, G. M. Almeida, A. Klautau, and K. V. Cardoso, "O-ran-oriented approach for dynamic vnf placement focused on interference mitigation," in *ICC 2024 - IEEE International Conference on Communications*, (Denver, CO, USA), pp. 5479–5484, IEEE, 2024.
- [30] L. Xiao, H. Zhang, Y. Xiao, X. Wan, S. Liu, L.-C. Wang, and H. V. Poor, "Reinforcement learning-based downlink interference control for ultra-dense small cells," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 423–434, 2019.
- [31] M. Elsayed, M. Erol-Kantarci, and H. Yanikomeroglu, "Transfer reinforcement learning for 5g new radio mmwave networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 2838–2849, 2020.
- [32] T. Hu, Q. Liao, Q. Liu, D. Wellington, and G. Carle, "Inter-cell slicing resource partitioning via coordinated multi-agent deep reinforcement learning," in *ICC 2022-IEEE International Conference on Communications*, (Seoul, Korea), pp. 3202–3207, IEEE, 2022.
- [33] Z. Gu, B. Vucetic, K. Chikham, P. Aliberti, and W. Hardjawana, "Graph representation learning for contention and interference management in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 32, no. 3, pp. 2479–2494, 2024.
- [34] H. T. Cheng and W. Zhuang, "Qos-driven mac-layer resource allocation for wireless mesh networks with non-altruistic node cooperation and service differentiation," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 6089–6103, 2009.
- [35] V.-D. Nguyen, T. X. Vu, N. T. Nguyen, D. C. Nguyen, M. Juntti, N. C. Luong, D. T. Hoang, D. N. Nguyen, and S. Chatzinotas, "Network-aided intelligent traffic steering in 6g o-ran: A multi-layer optimization framework," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 2, pp. 389–405, 2024.
- [36] L. Baldesi, F. Restuccia, and T. Melodia, "Charm: Nextg spectrum sharing through data-driven real-time o-ran dynamic control," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, (London, United Kingdom), pp. 240–249, IEEE, 2022.
- [37] L. L. Schiavo, G. Garcia-Aviles, A. Garcia-Saavedra, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Cloudric: Open radio access network (o-ran) virtualization with shared heterogeneous computing," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '24, (New York, NY, USA), p. 558–572, Association for Computing Machinery, 2024.
- [38] W.-H. Ko, U. Ghosh, U. Dinesha, R. Wu, S. Shakkottai, and D. Bhargava, "{EdgeRIC}: Empowering real-time intelligent optimization and control in {NextG} cellular networks," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, (Santa Clara, CA), pp. 1315–1330, USENIX Association, 2024.
- [39] H. Yu, O. Bejarano, and L. Zhong, "Combating inter-cell interference in 802.11ac-based multi-user mimo networks," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, (New York, NY, USA), p. 141–152, Association for Computing Machinery, 2014.
- [40] R. University, "Wireless Open Access Research Platform (WARP).," <http://warp.rice.edu/trac/>, 2014. Accessed on 2014[Online].
- [41] S.-H. Wu, C.-H. Ko, and H.-L. Chao, "On-demand coordinated spectrum and resource provisioning under an open c-ran architecture for dense small cell networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 1, pp. 673–688, 2022.
- [42] C. Ge, S. Xia, Q. Chen, and F. Adachi, "2-layer interference coordination framework based on graph coloring algorithm for a cellular system with distributed mu-mimo," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3557–3568, 2022.
- [43] Q. T. Inc., "Snapdragon X60 5G Modem-RF System." <https://www.qualcomm.com/products/technology/modems/snapdragon-x60-5gmodem>, 2023. Accessed on 2023[Online].
- [44] srsRAN Project, "Open source O-RAN 5G CU/DU solution from Software Radio Systems (SRS).," <https://docs.srsran.com/projects/4g/en/latest/usermanuals/source/srsue/source/index.html>, 2024. Accessed on April-22-2023.
- [45] C.-H. Lai, L.-H. Shen, and K.-T. Feng, "Intelligent load balancing and resource allocation in o-ran: A multi-agent multi-armed bandit approach," in *2023 IEEE 34th Annual International Symposium on Per-*

- sonal, *Indoor and Mobile Radio Communications (PIMRC)*, (Toronto, ON, Canada), pp. 1–6, IEEE, 2023.
- [46] Y. Xu, G. Gui, H. Gacanin, and F. Adachi, “A survey on resource allocation for 5g heterogeneous networks: Current research, future trends, and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 668–695, 2021.
- [47] F. Aslan, G. Iosifidis, J. A. Ayala-Romero, A. Garcia-Saavedra, and X. Costa-Perez, “Fair resource allocation in virtualized o-ran platforms,” *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 8, no. 1, pp. 1–34, 2024.
- [48] A. Filali, Z. Mlika, S. Cherkaoui, and A. Kobbane, “Dynamic sdn-based radio access network slicing with deep reinforcement learning for urllc and embb services,” *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2174–2187, 2022.
- [49] S. Bai, Z. Gao, X. Liao, and X. Sun, “Distributed adaptive multiuser scheduling via multi-agent reinforcement learning in multicell mimo cellular networks,” in *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, (Singapore, Singapore), pp. 1–6, IEEE, 2024.
- [50] Q. An, R. Doost-Mohammady, R. Yang, and K. Sridhar, “Dragon: A drl-based mimo layer and mcs adapter in open ran 5g networks,” in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, (New York, NY, USA), pp. 2323–2328, Association for Computing Machinery, 2024.
- [51] S. S. R. Jonnavithula, I. K. Jain, and D. Bharadia, “Mimo-ric: Ran intelligent controller for mimo xapps,” in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, (New York, NY, USA), pp. 2315–2322, Association for Computing Machinery, 2024.
- [52] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [53] Z. Zhu, H. Zhao, H. He, Y. Zhong, S. Zhang, H. Guo, T. Chen, and W. Zhang, “Diffusion models for reinforcement learning: A survey,” 2024.
- [54] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, vol. 0, no. 0, p. 02783649241273668, 2023.
- [55] Z. Wang, J. J. Hunt, and M. Zhou, “Diffusion policies as an expressive policy class for offline reinforcement learning,” 2023.
- [56] M. Rigter, J. Yamada, and I. Posner, “World models via policy-guided trajectory diffusion,” 2024.
- [57] B. Mazouze, W. Talbott, M. A. Bautista, D. Hjelm, A. Toshev, and J. Susskind, “Value function estimation using conditional diffusion models for control,” 2023.
- [58] L. Yang, Z. Huang, F. Lei, Y. Zhong, Y. Yang, C. Fang, S. Wen, B. Zhou, and Z. Lin, “Policy representation via diffusion probability model for reinforcement learning,” 2023.
- [59] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, “Planning with diffusion for flexible behavior synthesis,” 2022.
- [60] H. He, C. Bai, K. Xu, Z. Yang, W. Zhang, D. Wang, B. Zhao, and X. Li, “Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning,” *Advances in neural information processing systems*, vol. 36, pp. 64896–64917, 2023.
- [61] C. Lu, P. J. Ball, Y. W. Teh, and J. Parker-Holder, “Synthetic experience replay,” in *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS ’23*, (Red Hook, NY, USA), Curran Associates Inc., 2023.
- [62] ScienceDirect, “Frequency reuse - computer science topic,” 2025. Accessed: 2025-03-17.
- [63] Y. Chen, H. Li, and D. Zhao, “Boosting continuous control with consistency policy,” in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS ’24*, (Richland, SC), p. 335–344, International Foundation for Autonomous Agents and Multiagent Systems, 2024.
- [64] J. Clifton and E. Laber, “Q-learning: Theory and applications,” *Annual Review of Statistics and Its Application*, vol. 7, no. 1, pp. 279–301, 2020.
- [65] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, pp. 279–292, 1992.
- [66] S. Fujimoto, H. Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *International conference on machine learning*, (Stockholm, Sweden), pp. 1587–1596, PMLR, 2018.
- [67] openairinterface5g, “Openairinterface 5G Wireless Implementation.” <https://gitlab.eurecom.fr/oi/openairinterface5g>, 2024. Accessed on February-16-2024.
- [68] mosaic5g, “Flexible RAN Intelligent Controller (FlexRIC) and E2 Agent.” <https://www.intel.com/content/www/us/en/developer/topic-technology/edge-5g/tools/flexran.html>, 2021. 2022.
- [69] P. Group, “Picochip femtocell solutions.” <https://www.picochip.com/products/femtocells>, 2024. Accessed on 2024[Online].
- [70] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, “Flexran: A flexible and programmable platform for software-defined radio access networks,” in *Proceedings of the 12th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT ’16*, (New York, NY, USA), p. 427–441, Association for Computing Machinery, 2016.
- [71] Qualcomm Technologies Inc., “FSM100 5G RAN Platform for Small Cells.” <https://www.qualcomm.com/products/application/wireless-networks/small-cells/fsm100xx>, 2023. Accessed on 2023 [Online].



Peihao Yan is currently a PhD student in the Department of Computer Science and Engineering at Michigan State University (MSU), East Lansing, MI. She received her B.E. degree in communication engineering in 2020 and her M.S. degree in Computer Science and Technology in 2023 from China University of Petroleum (East China), Qingdao, China. Her current research interests include wireless networking and communications, with emphasis on 5G systems and O-RAN.



Huacheng Zeng (SM’20) is an Associate Professor in the Department of Computer Science and Engineering at Michigan State University (MSU). Prior to joining MSU, Dr. Zeng was an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Louisville. He also worked as a Senior System Engineer at Marvell Semiconductor, where he contributed to the development of wireless system solutions. He received his Ph.D. in Computer Engineering from Virginia Polytechnic Institute and State University (Virginia Tech). Dr. Zeng is a recipient of the NSF CAREER Award (2019), the Best Paper Award at IEEE SECON (2021), and the Best Student Paper Award at ACM WUWNET (2014). His research interests broadly include wireless networking and mobile sensing systems. He is serving on the editorial board of IEEE Transactions of Wireless Communications.



Y. Thomas Hou (Fellow, IEEE) received his Ph.D. from NYU Tandon School of Engineering in 1998. He is currently Bradley Distinguished Professor of Electrical and Computer Engineering at Virginia Tech, Blacksburg, VA, USA, which he joined in 2002. He was a Member of Research Staff at Fujitsu Laboratories of America in Sunnyvale, CA from 1997 to 2002. His current research focuses on developing real-time optimal solutions to complex science and engineering problems arising from wireless and mobile networks. He is also interested in wireless security. He authored/co-authored two textbooks and has published over 400 papers in IEEE/ACM journals and conferences. His papers were recognized by 12 best paper awards from IEEE and ACM, including an IEEE INFOCOM Test of Time Paper Award in 2023. He holds six U.S. patents. Prof. Hou was named an IEEE Fellow for contributions to modeling and optimization of wireless networks. He was/is on the editorial boards of a number of IEEE and ACM transactions and journals. He was Steering Committee Chair of IEEE INFOCOM conference and was a member of the IEEE Communications Society Board of Governors. He was also a Distinguished Lecturer of the IEEE Communications Society.