Causal Model-Based Reinforcement Learning for Sample-Efficient IoT Channel Access

Aswin Arun*, Christo Kurisummoottil Thomas†, Rimalpudi Sarvendranath*, and Walid Saad‡
* Department of Electrical Engineering, Indian Institute of Technology, Tirupati, Andhra Pradesh, India.
† Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, USA.
† Department of Electrical and Computer Engineering, Virginia Tech, Alexandria, VA, USA.
Emails:{ee22b055, sarvendranath}@iittp.ac.in, cthomas2@wpi.edu, walids@vt.edu.

ICC 2026 - IEEE International Conference on Communications | 979-8-3195-4209-0/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICC59461.2026.11587782

***Abstract*—Despite the advantages of multi-agent reinforcement learning (MARL) for wireless use case such as medium access control (MAC), their real-world deployment in Internet of Things (IoT) is hindered by their sample inefficiency. To alleviate this challenge, one can leverage model-based reinforcement learning (MBRL) solutions, however, conventional MBRL approaches rely on black-box models that are not interpretable and cannot reason. In contrast, in this paper, a novel causal model-based MARL framework is developed by leveraging tools from causal learning. In particular, the proposed model can explicitly represent causal dependencies between network variables using structural causal models (SCMs) and attention-based inference networks. Interpretable causal models are then developed to capture how MAC control messages influence observations, how transmission actions determine outcomes, and how channel observations affect rewards. Data augmentation techniques are then used to generate synthetic rollouts using the learned causal model for policy optimization via proximal policy optimization (PPO). Analytical results demonstrate exponential sample complexity gains of causal MBRL over black-box approaches. Extensive simulations demonstrate that, on average, the proposed approach can reduce environment interactions by 58%, and yield faster convergence compared to model-free baselines. The proposed approach inherently is also shown to provide interpretable scheduling decisions via attention-based causal attribution, revealing which network conditions drive the policy. The resulting combination of sample efficiency and interpretability establishes causal MBRL as a practical approach for resource-constrained wireless systems.**

I. Introduction

The proliferation of wireless sensor networks, internet of things (IoT) devices, and machine-to-machine (M2M) communication systems demands efficient medium access control (MAC) protocols that optimize performance under dynamic conditions [1]. While reinforcement learning (RL) [2]–[4] can automatically learn optimal scheduling policies by exploring the large state-action space of wireless environments, a fundamental barrier to AI-native wireless systems is *sample complexity*. Model-free RL requires extensive real-world interactions to converge, and each sample incurs significant communication overhead in resource-constrained networks. This sample inefficiency makes practical deployment of learning-based MAC protocols infeasible, necessitating approaches that achieve fast convergence with minimal real-world data.

A. Related Works

In multi-agent RL (MARL) settings [5], distributed sensor nodes can learn decentralized policies while coordinating through control signaling with the gateway. In contrast to model-free RL that are sample inefficient [2]–[5], model-based RL (MBRL) [6] and [7] can significantly improve sample efficiency, requiring a small number of samples generated through real environment interactions, by learning a model of the environment dynamics and using it to generate synthetic experience for policy training. Complementary RL approaches include diffusion-based data augmentation for replay buffers [8], [9], hierarchical MBRL with temporal abstraction [10], and large language model (LLM)-based knowledge integration through reward shaping [11]. However, standard MBRL and LLM-based approaches [6], [7] often learn black-box models whose performance are tied to the training data and does not provide an understanding of which variables causally influence which outcomes. This opacity creates three fundamental challenges. First, policies learned from black-box models may exploit *spurious correlations* that fail to transfer when network conditions change. Second, *model errors accumulate* over planning horizons, limiting the effectiveness of long-horizon autonomous control. Third, the learned policies *lack explainability*, making it impossible to verify their trustworthiness before deployment, a critical concern for mission-critical IoT systems. A promising approach here is to exploit causal inference [12] to understand the underlying causal structure of wireless dynamics, enabling model-based agents to generalize reliably under distribution shifts while requiring limited real-world data.

B. Contributions

The main contribution of this paper is a novel MBRL for device scheduling in IoT that leverages causal graphs to explicitly represent the cause and effect relations among states, actions, observations, and rewards. By exploiting this causal structure, we aim to improve *sample efficiency* through data augmentation using more accurate environment modeling; and facilitate *faster learning* via model-based rollouts using the causal structure. Our key contributions include:

- We formulate the wireless MAC problem for the uplink (UL) of a multi-user IoT network using structural causal models (SCMs) that explicitly represent the causal relationships between sensor node states, control messages, transmission actions, observations, and rewards.
- We develop gated recurrent units (GRU) and attention-based neural networks to learn causal dependencies modeled as a SCM, with the architecture *inherently providing*

The research of Walid Saad was supported by the US National Science Foundation under Grant NSF CNS-2225511.

979-8-3195-4209-0/26/$31.00 ©2026 IEEE
Authorized licensed use limited to: Indian Institute of Technology. Downloaded on August 27,2026 at 16:27:47 UTC from IEEE Xplore. Restrictions apply.

explainability through attention weights that reveal causal influences on scheduling decisions. Further, we integrate the learned causal model into a model-based RL algorithm using proximal policy optimization (PPO), where the causal model generates synthetic rollouts for policy training.

- We *analytically derive the sample efficiency gains* that can be achieved using a causal MBRL compared to traditional RL baseline, as a function of the SCM complexity and learned model accuracy.
- Through extensive simulations[1], we show that proposed causal MBRL achieves *faster convergence* compared to model-free baselines and on average *better sample efficiency* via 58% reduction in environment interactions.

II. System Model and MARL Formulation

We consider an IoT network with U sensor nodes that communicate with a gateway over a shared UL channel. Each sensor node aims to transmit P data packets to the gateway while coordinating channel access through control messages. Similar to the model considered in [13], all sensor nodes transmit their UL data packets over the same frequency channel using time division multiple access (TDMA) scheme. The objective is to deliver all packets to the gateway with minimal latency, which is critical for time-sensitive IoT applications such as environmental monitoring, industrial control, and smart grid systems. We consider a packet erasure channel in which the transmitted data packets can be received in error with block error rate (BLER) $\rho \in [0, 1]$. For simplicity, we assume the BLER is identical for all UL data transmissions. Each sensor node has a dedicated error-free UL and downlink (DL) control channels for exchanging signaling messages with the gateway. We assume that control channels do not experience packet losses and have negligible transmission delay. The gateway implements a pre-defined MAC signaling protocol that is not learned. Sensor nodes need to learn to communicate with the gateway by understanding the channel access scheme and MAC signaling protocol of the gateway with an objective to successfully deliver the packets in minimum time possible.

In the UL, the gateway receives the scheduling requests (SRs) from the sensor nodes, and in the DL, it sends the scheduling grant (SG) to the specific node who has been granted the channel access. Furthermore, in the DL, it also sends an acknowledgment (ACK) when a packet is successfully received. Let $\mathcal{M}_L = \{0, 1, 2\}$ denote the set of three possible DL control messages (DCMs) from the gateway to each node and $m_t^u \in \mathcal{M}_L$ denote DCM sent to u^{th} node at time slot t. The gateway sending a null message is denoted by $m_t^u = 0$, releasing an SG to node u in the next time slot is denoted by $m_t^u = 1$, and sending an ACK is defined by $m_t^u = 2$. We define $\mathcal{U}_L = \{0, 1\}$ as the set of UL control messages (UCMs), where $n_t^u \in \mathcal{U}_L$ represents the UCM at time t from node u. The gateway will interpret $n_t^u = 1$ as receiving scheduling request and $n_t^u = 0$ as no request.

[1]Due to space constraints, the evaluation of generalizability is omitted and will be presented in a journal extension.

At each time step t, for each learner u, the environment delivers a scalar observation $o_t^u \in \mathcal{O}^U = [0, P]$ describing the number of packets that remain in the learner's UL transmit buffer. Here, P is the transmit buffer capacity, and all packets are assumed to be of the same size. Similarly, at each time step, the gateway receives a scalar observation $o_t^b \in \mathcal{O}^B = [0, L + 1]$ from the environment. $o_t^b = 0$ means that UL channel is idle, $o_t^b = u \in [1, U]$ represents collision-free UL transmission received from node u and $o_t^b = U + 1$, when collision occurred in the UL channel. The gateway also receives the SRs $\{n_t^u\}_{u=1}^U$ from the sensor nodes via the control channel. It selects one requesting node uniformly at random and sends an SG to that node: $m_t^u = 1$ for the selected node u, and $m_t^{u'} = 0$ for all other nodes $u' \neq u$. If a node successfully transmits data concurrently with sending an SR, the gateway sends an ACK instead of an SG to that node, and schedules a different node. For u^{th} sensor node at time step t, let $a_t^u \in \{0, 1, 2\}$ denotes the UL shared channel action. Action value 0 correspond to idle, and action values 1 and 2 correspond to transmitting and deleting the oldest packet in the buffer, respectively. We define the vector $\boldsymbol{d}_t^u = (n_t^u, a_t^u)$ as the combined decision made by the u^{th} sensor node. Further, the state of each sensor node $\boldsymbol{x}_t^u$ is defined, and it incorporates the observation history, previous decisions, and received control messages over a window of N time steps:

$$\boldsymbol{x}_t^u = \left[o_{t-1}^u, \boldsymbol{d}_{t-1}^u, m_{t-1}^u, \cdots, o_{t-N}^u, \boldsymbol{d}_{t-N}^u, m_{t-N}^u\right], \quad (1)$$

Similarly, the gateway maintains a state based on its observations and the control message history from all nodes:

$$\boldsymbol{x}_t^b = \left[o_{t-1}^b, \boldsymbol{n}_{t-1}, \boldsymbol{m}_{t-1}, \cdots, \boldsymbol{o}_{t-N}^b, \boldsymbol{n}_{t-N}, \boldsymbol{m}_{t-N}\right], \quad (2)$$

where $\boldsymbol{n}_t$ and $\boldsymbol{m}_t$ are the vector of UCMs from all sensor nodes and the vector of DCMs to sensor nodes, respectively. For notational convenience, we also define the global state $\boldsymbol{x}_t = (\boldsymbol{x}_t^b, \boldsymbol{x}_t^1, \cdots, \boldsymbol{x}_t^U)$, global observation $\boldsymbol{o}_t = (o_t^b, o_t^1, \cdots, o_t^U)$ and global actions $\boldsymbol{d}_t = (\boldsymbol{d}_t^1, \boldsymbol{d}_t^2, \cdots, \boldsymbol{d}_t^U)$.

A. Multi-Agent RL Framework

Following [13], we assign a reward of $r_t = -1$ once packets are successfully delivered and deleted. Reward is greater than $-T_{\max}$ when all packets are delivered and deleted in less number of steps. The total episode reward $R = \sum_{t=0}^{T-1} r_t$, where $T \leq T_{\max}$ is the actual episode length. Given the current state $\boldsymbol{x}_t^u$ and observation o_t^u, the RL agent at each sensor node selects shared channel action and UCM according to its policy $\pi_u(\boldsymbol{d}_t^u \mid \boldsymbol{x}_t^u)$. The objective is to learn policies $\{\pi_u(\boldsymbol{d}_t^u \mid \boldsymbol{x}_t^u)\}_{u=1}^U$ that maximize the expected cumulative discounted reward: $J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, where $\gamma \in [0, 1)$ is the discount factor that balances immediate and future rewards r_t. Computing node policies to maximize $J(\pi)$ using standard MARL has the limitation of requiring large number of real-world data samples, which may not be available in practice. To overcome this challenge, we now propose a causal MBRL approach that addresses these limitations via explicitly capturing the *causal relations* among RL variables and enabling *data augmentation* by generating synthetic data samples that nearly matches the real-world samples.

Authorized licensed use limited to: Indian Institute of Technology. Downloaded on August 27,2026 at 16:27:47 UTC from IEEE Xplore. Restrictions apply.

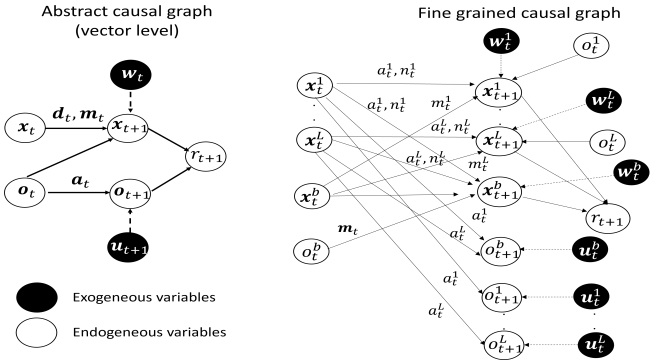


Fig. 1: Causal Graph $\mathcal{G}$ for MAC Scheduling. Abstract level causal graph is provided for simplicity of understanding. We use the fine grained causal graph for our proposed method.

III. Causal Model Based Wireless Scheduling

In this section, we present a rigorous framework for learning causal models in multi-user IoT device scheduling for channel access. We formalize the SCM using the known causal graph $\mathcal{G}$ (see Fig. 1). We then formulate a causal MBRL-based scheduling problem that learns the structural equations (SEs) part of causal model and optimize the scheduling policies.

A. Formal SCM Definition

We formalize the wireless scheduling environment as an SCM $\mathcal{M} = \langle \mathcal{U}, \mathcal{V}, \mathcal{F}, P(\mathcal{U}) \rangle$, where $\mathcal{U}$ is the set of exogenous variables (external noise and unobserved factors), $\mathcal{V}$ is the set of endogenous variables (system variables determined by the model), $\mathcal{F}$ is the set of SEs defining causal relationships, and $P(\mathcal{U})$ is the probability distribution over exogenous variables. The endogenous variables at time t, defined as $\mathcal{V}_t$, consist of agent states $\mathcal{X}_t = \{\boldsymbol{x}_t^1, \ldots, \boldsymbol{x}_t^U, \boldsymbol{x}_t^b\}$, observations $\mathcal{O}_t = \{o_t^1, \ldots, o_t^U, o_t^b\}$, UL actions $\mathcal{D}_t = \{\boldsymbol{d}_t^1, \ldots, \boldsymbol{d}_t^U\}$ (sensor node decisions), DCMs $\mathcal{M}_t = \{m_t^1, \ldots, m_t^U\}$ (gateway decisions) and reward $r_t \in \mathbb{R}$. The exogenous variables capture stochasticity and include state transition noise $\{w_t^1, \ldots, w_t^L, w_t^b\}$ (e.g., due to channel fading, interference), and external environmental factors $\{u_t^1, \ldots, u_t^U\}$. Each u_t^u corresponds to the packet arrival process that generate the observations o_t^u. Each endogenous variable $v \in \mathcal{V}_t$ is defined by a SE: $v := f_v(\text{Pa}(v), \mathcal{U}_v)$, where $\text{Pa}(v) \subseteq \mathcal{V}_{t-1} \cup \mathcal{V}_t$ are the parent variables of v, i.e., all variables that directly cause v, in the causal graph, and $\mathcal{U}_v$ represents the exogenous inputs affecting v. Specifically, we have *state transition equations* as

$$\begin{aligned} \boldsymbol{x}_{t+1}^u &:= f_x^u(\boldsymbol{x}_t^u, o_t^u, \boldsymbol{d}_t^u, m_t^u, w_t^u), \quad u = 1, \ldots, U, \\ \boldsymbol{x}_{t+1}^b &:= f_x^b(\boldsymbol{x}_t^b, o_t^b, \boldsymbol{d}_t, \boldsymbol{m}_t, w_t^b), \end{aligned} \tag{3}$$

and *observation equations* as

$$\begin{aligned} o_{t+1}^u &:= f_o^u(\boldsymbol{x}_t^u, a_t^u, u_t^u), \quad u = 1, \ldots, U, \\ o_{t+1}^b &:= f_o^b(\boldsymbol{x}_t^b, \boldsymbol{d}_t, w_t^b), \end{aligned} \tag{4}$$

and *reward equation* as $r_{t+1} := f_r(\boldsymbol{x}_t, \boldsymbol{d}_t, \boldsymbol{m}_t, \boldsymbol{o}_t)$. (5)

The causal relationships in the SCM are represented by a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes correspond to endogenous variables $\mathcal{V} = \mathcal{V}_t \cup \mathcal{V}_{t+1}$ and directed edges represent direct causal relationships: $(v_i, v_j) \in \mathcal{E}$ if $v_i \in \text{Pa}(v_j)$. Unlike knowledge graphs that represent semantic relationships or correlative associations [14], SCM is specifically designed to capture *direct causal influences* and avoids the exponential blowup of learning unstructured dependencies. It preserves *interventional* semantics, i.e., the graph remains valid when network conditions vary, enabling policies generalize across channel conditions. SCMs also satisfy the Markov property, i.e., each variable is conditionally independent of its non-descendants given its parents.

B. Problem Formulation: Causal MBRL Based Scheduling

Given $\mathcal{G}$, we learn the functional forms of the SEs in (3)-(5) using parameterized inference networks.[2] For each endogenous variable $v_j \in \mathcal{V}$, the objective is to learn a conditional distribution: $q_\phi(v_j \mid \text{Pa}(v_j)) \approx p(v_j \mid \text{Pa}(v_j))$, by minimizing the negative log-likelihood $-\sum_{v_j \in \mathcal{V}_{t+1}} \mathbb{E}_{\delta \sim \mathcal{D}}[\log q_\phi(v_j | \text{Pa}(v_j))]$, where q_ϕ is parameterized by neural network weights ϕ, and p represents the true conditional distribution in the environment. Here, the expectation is over the empirical distribution $p(v_j \mid \text{Pa}(v_j))$ obtained using the wireless data $\mathcal{D}$ that involves the stat-action-reward trajectories. Challenges here involve heterogeneous parent variable types and dimensions across $\text{Pa}(v_j)$, dynamic structural changes as the causal graph evolves over time, and complex temporal dependencies among sequential actions. To handle heterogeneous inputs, we introduce a *variable encoding layer* that maps all variables to a common embedding space. This variable encoder is a two-layer MLP with ReLU activation function.

Given the current-time variables $\mathcal{V}_t$, we obtain the encoded set $\hat{\mathcal{V}}_t = \{\hat{s}_i = f(s_i) : s_i \in \mathcal{V}_t\}$. Encoders are shared across all inference networks, enabling transfer learning. The architecture for inference network is inspired from the causal model discussed in [15]. For each target variable $v_j \in \mathcal{V}$, we construct an inference network $f^j : \text{Pa}(v_j) \to \mathcal{P}(v_j)$ that outputs a probability distribution over v_j. We partition the parent set into state variables and action variables: $\text{Pa}(v_j) = \mathcal{S}_j \cup \mathcal{A}_j$, where $\mathcal{S}_j = \text{Pa}(v_j) \cap (\mathcal{X}_t \cup \mathcal{O}_t)$ are the states and observations and $\mathcal{A}_j = \text{Pa}(v_j) \cap (\mathcal{D}_t \cup \mathcal{M}_t)$ are the actions and control messages. This partitioning is considered because states are processed independently, while actions are processed sequentially to capture temporal dependencies. Further, the integrated causal MBRL problem aims at optimizing the causal model neural network parameters ϕ and policy model parameters θ:

$$\min_{\theta,\phi} \sum_{i=1}^{n_{\text{epoch}}} \left[\mathcal{L}_{\text{RL}}^{(i)}(\theta, \phi) + \mathcal{L}_{\text{model}}^{(i)}(\phi) \right], \tag{7}$$

where the RL loss $\mathcal{L}_{\text{RL}}^{(i)}(\theta, \phi) = -\mathbb{E}_{\pi_\theta, q_\phi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$ is averaged over the real-world interaction with the wireless environment, n_{epoch} is the number of real-world data interactions stored in $\mathcal{D}$, and the causal model loss is

$$\mathcal{L}_{\text{model}}^{(i)}(\phi) = -\sum_{v_j \in \mathcal{V}_{t+1}} \mathbb{E}_{\delta \sim \mathcal{D}^{(i)}}[\log q_\phi(v_j | \text{Pa}(v_j))] + \lambda \|\phi\|_2^2.$$

For a batch $\mathcal{B} \subset \mathcal{D}$ of size B, $\mathcal{L}_{\text{model}}(\phi) = -\frac{1}{B}\sum_{\delta \in \mathcal{B}} \sum_{v_j \in \mathcal{V}_{t+1}} \log q_\phi(v_j^\delta \mid \text{Pa}(v_j)^\delta)$, where v_j^δ is the observed value of v_j in transition δ.

[2] We note that constraint-based methods (PC algorithm) or score-based approaches (greedy equivalence search, NOTEARS) could be employed in future work to infer $\mathcal{G}$ from wireless data.

Authorized licensed use limited to: Indian Institute of Technology. Downloaded on August 27,2026 at 16:27:47 UTC from IEEE Xplore. Restrictions apply.

$$f^j : \mathrm{Pa}(v_j) \xrightarrow{\text{encode}} \mathrm{Pa}(v_j)_{\text{encoded}} \xrightarrow{\text{separate}} \begin{cases} \text{States } \mathcal{S}_j \\ \text{Actions } \mathcal{A}_j \end{cases} \xrightarrow{\text{process}} \boldsymbol{h}^j \xrightarrow{\text{decode}} q_\phi(v_j \mid \mathrm{Pa}(v_j)). \tag{6}$$

IV. Proposed Solution: Causal MBRL Algorithm

We now describe the MBRL algorithm and analyze sample efficiency gains. We propose to solve ϕ and θ in an alternating way via a three-phase approach executed each epoch to optimize (7): (i) collect real environment trajectories using π_θ, (ii) periodically update the causal model parameters ϕ using accumulated data, and (iii) optimize the policy network parameters θ using both real and synthetic data generated from the learned model. The detailed steps are provided in Algorithm 1. We model the causal inference network using attention based neural networks.

A. Causal Inference Networks

An overview of the architecture is represented by (6).

1) Value vector computation: For each state variable $s_i \in \mathcal{S}_j$, we compute a contribution vector (aka value vector) $\boldsymbol{c}_i^j$ using a single layer MLP. $\boldsymbol{c}_i^j$ represents the potential contribution of state s_i to predicting v_j.

2) Action embedding with GRU: Given encoded actions $\{\hat{a}_i\}_{a_i \in \mathrm{PA}(v_j)}$, the GRU computes the action embedding $\boldsymbol{e}^j = \mathrm{GRU}^j(\{\hat{a}_i\})$. From this, we derive the query vector $\boldsymbol{q}^j = W_q^j \boldsymbol{e}^j + \boldsymbol{b}_q^j$ and action contribution vector $\boldsymbol{c}_a^j = W_a^j \boldsymbol{e}^j + \boldsymbol{b}_a^j$, where weights W_q^j, W_a^j and biases are learnable parameters. Each state variable s_i has a learnable key vector $\boldsymbol{k}_i$ (shared across all inference networks to ensure consistent variable importance representations). The attention score $e_i^j = \boldsymbol{k}_i^T \boldsymbol{q}^j$ measures compatibility between state s_i and query $\boldsymbol{q}^j$. We normalize attention weights using:

$$\alpha_i^j = \frac{\exp(e_i^j)}{1 + \sum_{s_k \in \mathcal{S}_j} \exp(e_k^j)}, \;\; \alpha_a^j \;\; = \frac{1}{1 + \sum_{s_k \in \mathcal{S}_j} \exp(e_k^j)}.$$

The attention weights provide interpretability of the learned causal model. High attention weight α_i^j indicates that state s_i has strong causal influence on v_j given the current action context. We then compute the hidden representation of the posterior distribution as a weighted sum of the value vectors:

$$\boldsymbol{h}^j = \sum_{s_i \in PA(v_j)} \alpha_i^j \cdot \boldsymbol{c}_i^j + \alpha_a^j \cdot \boldsymbol{c}_a^j. \tag{8}$$

The final component maps the hidden representation to distribution parameters. The decoder $D^j : \mathbb{R}^{d_h} \to \Theta_{v_j}$ is a two-layer MLP, where Θ_{v_j} is the parameter space for the distribution of v_j. For continuous variables (e.g., states, observations): $q_\phi(v_j \mid \mathrm{Pa}(v_j)) = \mathcal{N}(\mu_j, \sigma_j^2)$, where $[\mu_j, \log \sigma_j] = D^j(\boldsymbol{h}^j)$. We predict $\log \sigma_j$ to ensure positivity. For discrete variables (e.g., discrete actions): $q_\phi(v_j \mid \mathrm{Pa}(v_j)) = \mathrm{Categorical}(\boldsymbol{\pi}_j)$ where $\boldsymbol{\pi}_j = \mathrm{softmax}(D^j(\boldsymbol{h}^j))$.

3) Policy update using data augmentation: We now integrate the learned causal model into a complete MBRL framework. We leverage the learned causal model to generate synthetic data matching real-world dynamics, reducing the need for extensive wireless samples during training. Short rollouts ($k \in [3, 10]$) for synthetic data reduce model error accumulation for higher accuracy but sacrifice sample efficiency, whereas long rollouts $k \in [10, 50]$ improve efficiency at the cost of compounding errors. Further, PPO is used to optimize the network parameters because it safely handles training on both real and simulated data through importance weighting and clipping, which is critical for MBRL. Using both real and simulated data, our loss function $\mathcal{L}^{\text{PPO}}(\theta)$ is equal to

$$\mathbb{E}_{(\boldsymbol{x}_t, \boldsymbol{a}_t) \sim D}\Big[\min\big(\rho_t(\theta),\, \mathrm{clip}\big(\rho_t(\theta), 1-\epsilon, 1+\epsilon\big)\big)\, \hat{A}_t\Big] \tag{9}$$

where $\rho_t(\theta) = \frac{\pi_\theta(\boldsymbol{d}_t|\boldsymbol{x}_t)}{\pi_{\theta_{\text{old}}}(\boldsymbol{d}_t|\boldsymbol{x}_t)}$, $\hat{A}_t$ is the advantage estimate. The advantage function is formally defined as $A^\pi(\boldsymbol{x}_t, \boldsymbol{d}_t) = Q^\pi(\boldsymbol{x}_t, \boldsymbol{d}_t) - V^\pi(\boldsymbol{x}_t)$, where $Q^\pi(\boldsymbol{x}_t, \boldsymbol{d}_t)$ is the expected return when taking action $\boldsymbol{d}_t$ in state $\boldsymbol{x}_t$ and then following policy π, and $V^\pi(\boldsymbol{x}_t)$ is the expected return in state $\boldsymbol{x}_t$ and following policy π. clip() function constraints ρ_t within $\{1-\epsilon, 1+\epsilon\}$.

Algorithm 1: Proposed Causal Model-Based RL

Input: Environment with unknown dynamics; initial policy parameters θ_{policy}; hyperparameters $n_{\text{epoch}}, n_{\text{graph}}, n_{\text{round}}, k_{\text{rollout}}$
Output: Optimized policy π_θ
Initialization: Replay buffer $\mathcal{D} \leftarrow \emptyset$, causal graph $\mathcal{G}$ as in Fig. 1, inference network parameters $\phi \sim \mathcal{N}(0, 0.01)$, policy networks $\theta_{\text{policy}} \sim \mathcal{N}(0, 0.01)$, value network $\theta_{\text{value}} \sim \mathcal{N}(0, 0.01)$.
for $i = 1$ to n_{epoch} **do**
  Phase 1: Real Environment Interaction
  for each episode $e = 1$ to n_{episodes} **do**
    Initialize state $\boldsymbol{x}_0 \sim p_0(\boldsymbol{x})$
    for $t = 0$ to $T-1$ **do**
      Sample actions: $\boldsymbol{d}_t^u \sim \pi_u(\cdot|\boldsymbol{x}_t^u)\ \forall u$; $\boldsymbol{m}_t \sim \pi_b(\cdot|\boldsymbol{x}_t^b)$
      Execute actions; observe $\boldsymbol{x}_{t+1}, \boldsymbol{o}_{t+1}, r_t$, store in $\mathcal{D}$
    end for
  end for
  Phase 2: Causal Model Update
  if $i \bmod n_{\text{graph}} = 0$ **then**
    for gradient step $= 1$ to N_{model} **do**
      Sample minibatch $B \subset \mathcal{D}$
      Compute $\mathcal{L}_{\text{model}}(\phi)$, and update $\phi \leftarrow \phi - \eta_{\text{model}} \nabla_\phi \mathcal{L}_{\text{model}}$
    end for
  end if
  Phase 3: Policy Optimization
  for $j = 1$ to n_{round} **do**
    Generate synthetic data: $\mathcal{D}_{\text{sim}} \leftarrow \emptyset$
    Sample initial states $\{x_0^{(i)}\}_{i=1}^{N_{\text{rollout}}}$ from $\mathcal{D}$
    for $i = 1$ to N_{rollout} **do**
      trajectory $\leftarrow k_{\text{rollout}}(\boldsymbol{x}_0^{(i)}, \pi, k_{\text{rollout}})$ using Algorithm 2
      $\mathcal{D}_{\text{sim}} \leftarrow \mathcal{D}_{\text{sim}} \cup$ trajectory
    end for
    $\mathcal{D}_{\text{combined}} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{sim}}$
    for ppo_epoch $= 1$ to N_{ppo} **do**
      Sample minibatch $B \subset \mathcal{D}_{\text{combined}}$, Compute $\{\hat{A}_t\}$
      Update policy: $\theta_{\text{policy}} \leftarrow \theta_{\text{policy}} - \eta_{\text{policy}} \nabla_\theta \mathcal{L}_{\text{PPO}}$
      Update value: $\theta_{\text{value}} \leftarrow \theta_{\text{value}} - \eta_{\text{value}} \nabla_\theta \mathcal{L}_{\text{value}}$
    end for
  end for
end for
Return: Learned policy π_θ

Authorized licensed use limited to: Indian Institute of Technology. Downloaded on August 27,2026 at 16:27:47 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: k-Step Model Rollout

Initialize $\mathcal{D}_{\text{sim}} \leftarrow []$, $\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
for $\tau = 0$ to $k-1$ **do**
  $\boldsymbol{d}_\tau^u \sim \pi_u(\cdot \mid \boldsymbol{x}_\tau^u), \forall u$. // Sample actions from current policy:
  $\boldsymbol{m}_\tau \sim \pi_b(\cdot \mid \boldsymbol{x}_\tau^b)$
  // Predict next state and observations using causal model
  for $u = 1$ to U **do**
    $\boldsymbol{x}_{\tau+1}^u \sim q_\phi(\boldsymbol{x}_{\tau+1}^u \,|\, \text{Pa}(\boldsymbol{x}_{\tau+1}^u))$, $o_{\tau+1}^u \sim q_\phi(o_{\tau+1}^u \,|\, \text{Pa}(o_{\tau+1}^u))$
  end for
  $r_\tau = f_r(\boldsymbol{x}_\tau, \boldsymbol{d}_\tau, \boldsymbol{m}_\tau, \boldsymbol{o}_\tau)$ // Compute reward
  $\mathcal{D}_{\text{sim}} \leftarrow \mathcal{D}_{\text{sim}} \cup \{(\boldsymbol{x}_\tau, \boldsymbol{d}_\tau, \boldsymbol{m}_\tau, r_\tau, \boldsymbol{x}_{\tau+1}, \boldsymbol{o}_{\tau+1})\}$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}_{\tau+1}$ // Update current state
end for
return $\mathcal{D}_{\text{sim}}$

B. Theoretical Analysis

We justify sample efficiency through causal factorization. Under standard assumptions (identifiability, bounded state, and action spaces, Markov property, Lipschitz dynamics), the sample complexity for causal MBRL is:

Theorem 1. *For a wireless MAC system with $|\mathcal{V}|$ variables (includes state, action and reward), maximum support $V_{\max}$, causal in-degree d_{in} (maximum number of parents for a single node in $\mathcal{G}$), and a desired accuracy of the causal model $\epsilon > 0$, causal MBRL achieves sample complexity:*

$$N_{causal} = \mathcal{O}\left(|V|^2 \cdot V_{\max}^{d_{in}} / \left(\epsilon^2 (1-\gamma)\right)\right) \tag{10}$$

versus black-box MBRL's $\mathcal{O}(V_{\max}^{|V|}/\epsilon^2)$, yielding exponential improvement $\sim V_{\max}^{|\mathcal{V}|-d_{in}}/|\mathcal{V}|^2$.

Proof: Detailed proof is omitted due to space constraints. The key insight is that the joint distribution over all variables can be factorized using causal structure $p(\mathcal{V}) = \prod_{v_j \in \mathcal{V}} p(v_j|\text{Pa}(v_j))$. Each conditional $p(v_j|\text{Pa}(v_j))$ depends only on $d_{\text{in}}(v_j)$ parents, where $d_{\text{in}}(v_j) \leq d_{\text{in}} \ll |\mathcal{V}|$ by the acyclic structure constraint. By standard results in learning theory (Hoeffding's inequality and finite hypothesis classes) [16], [17], to learn a single conditional $p(v_j|\text{Pa}(v_j))$ with accuracy ϵ, we require samples proportional to the support size: $N_j = \mathcal{O}\left(V_{\max}^{d_{\text{in}}(v_j)}/\epsilon^2\right)$. This is because the conditional probability table has $\mathcal{V}_{\max}^{d_{\text{in}}}$ entries (one for each parent combination), each requiring $\mathcal{O}(\log(1/\epsilon)/\epsilon^2)$ samples for accurate estimation. Since we must learn all $|\mathcal{V}|$ conditionals, and learning one variable does not immediately improve learning others (by causality), the total complexity is:

$$N_{\text{total}} = \sum_{v_j \in \mathcal{V}} N_j = |\mathcal{V}| \cdot \mathcal{O}\left(\frac{\mathcal{V}_{\max}^{d_{\text{in}}}}{\epsilon^2}\right) = \mathcal{O}\left(\frac{|\mathcal{V}| \cdot \mathcal{V}_{\max}^{d_{\text{in}}}}{\epsilon^2}\right)$$

The $|\mathcal{V}|^2$ factor appears when accounting for variance reduction across epochs and confidence bound union over variables (Hoeffding-style bounds) [18]. ∎

By leveraging causal structure, Theorem 1 shows that learning complexity scales with the in-degree d_{in} rather than the total number of variables $|\mathcal{V}|$; since wireless dynamics are sparse (few variables directly influence each outcome), this causal factorization dramatically reduces sample requirements compared to unstructured black-box approaches.

TABLE I: Hyperparameters and algorithm configuration

Hyperparameter	Value	Hyperparameter	Value ($U = 1, 2$)
Learning rate α	$1e{-}4$	N_{rollout}	$50, 60$
GAE parameter λ	0.95	Rollout horizon k_{rollout}	$6, 8$
Discount factor γ	0.99	N_{model}	$100, 120$
Batch size	64	N_{PPO}	$5, 20$
Episodes per epoch	20	n_{graph}	$1, 2$
Evaluation episodes	128	n_{round}	$4, 8$
		History N	$1, 3$

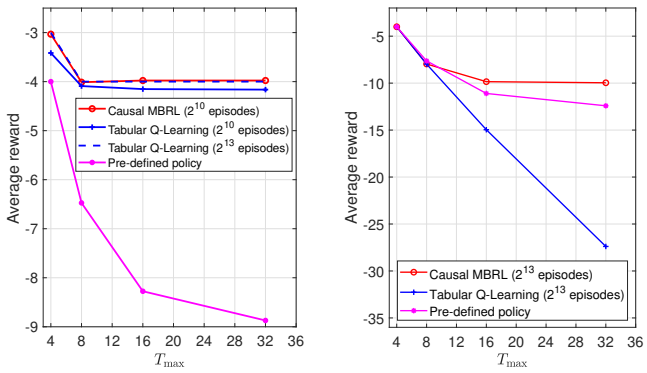
V. Simulation Results and Analysis

We validate the sample efficiency and performance of the proposed causal MBRL based wireless scheduling using extensive Monte-Carlo simulations. We implemented and tested the training procedure described in Algorithms 1 and 2. We have collected the performance, i.e., total reward R, per episode. We start our simulations with a full buffer at the user and repeated the training for maximum timesteps per episode $T_{\max} = 4, 8, 16, 32$. The dimension of the variable encoder embedding vector is chosen to be $d = 64$. For the inference networks, the input dimension is 64 and the dimension of one hidden layer $d_h = 128$. The dimension of the query vector is $d_q = 32$, and the ReLU activation function is used. The output dimension is 2 representing mean and variance of the causal node with softmax as the activation function and cross-entropy loss function. The policy network is similar with input dimension 64, one hidden layer with dimension equal to 128, and $tanh$ is used as the activation function. The output dimension is 3, outputting the UL shared channel action, UCM, and the logarithm of the sum of probabilities of channel and UL actions. Cross-entropy loss function is employed. Training parameters are listed in Table I.

Pre-defined policy at the node: We compare the reward of our proposed causal MBRL with the one obtained when the nodes implement a pre-defined MAC signaling policy known to the gateway as described in Algorithms 1 and 2 of [13]. *Tabular Q-learner:* We also compare with [13] that chooses the action and uplink signaling based on Q-tables.

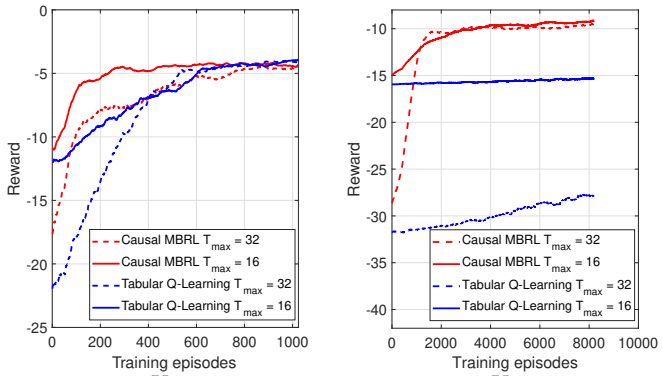
Sample efficiency analysis: Figs. 2a and 2b shows the average reward as a function of the maximum timesteps per episode $T_{\max}$ for $U = 1$ and $U = 2$, respectively. They compare the performance of the proposed causal MBRL, the tabular Q-learning approach [13], and the pre-defined policy. MBRL training is done for 2^{10} and 2^{13} episodes for $U = 1$ and $U = 2$, respectively. Furthermore, the reward shown is averaged over 8 random seeds. For $U = 1$, we see that after 2^{10} episodes, the performance of the causal MBRL converges to higher value than tabular Q-learning due to the synthetic rollouts from the learned causal model. Furthermore, tabular Q-learning requires 2^{13} episodes to converge, in turn, requiring a higher number of samples. Pre-defined policy performs worse because it waits for the schedule grant, unlike the learning approaches. For $U = 2$, we have shown the performance for 2^{13} episodes. Here, the causal MBRL is significantly better than Q-learning and the pre-defined policy. For $T_{\max} = 32$, the reward is -10 and -27 for the causal MBRL and Q-learning, respectively. Here, the pre-defined policy performs better than Q-learning as more number of training episodes are required for it to converge.

Authorized licensed use limited to: Indian Institute of Technology. Downloaded on August 27,2026 at 16:27:47 UTC from IEEE Xplore. Restrictions apply.



(a) $U = 1$ (b) $U = 2$

Fig. 2: Causal MBRL wireless scheduling: Average episode rewards vs $T_{\max}$ showing performance across different U (BLER= 0.5).



(a) $U = 1$ (b) $U = 2$

Fig. 3: Average rewards vs training episodes for different $T_{\max}$.

TABLE II: Comparison of number of samples required

Nodes	Q-Learning	Causal MBRL	Efficiency
$U = 1$	37.5K	12.5K	66.66%
$U = 2$	243.1K	126.7K	47.9%

All values represent total environment interactions (real-world samples) required to reach 95% optimal reward.

Table II compares the number of samples required. It shows that the number of real-world samples required until convergence for causal MBRL is 66.66% and 47.9% (and an average 58%) less than that of tabular Q-learning for $U = 1$ and $U = 2$, respectively. This shows the superior sample efficiency of the proposed causal MBRL method.

Fig. 3 shows average reward as a function of the training episodes for the causal MBRL and tabular Q-learning. This is done for $T_{\max} = 16$ and $T_{\max} = 32$. We see that the reward increases and then saturates for both the learning approaches. For the smaller number of episodes, the increase is higher for the causal MBRL compared to the tabular Q-learning, which shows that our approach learns faster with fewer samples. We see that for the $U = 1$ case, the reward of the causal MBRL saturates faster for $T_{\max} = 16$ compared to $T_{\max} = 32$ as the number of steps in each episode is smaller. However, tabular Q-learning takes around the same number of episodes to saturate.

VI. Conclusion

We have introduced causal models for IoT device scheduling, addressing fundamental limitations of existing MBRL approaches. Unlike black-box neural network models, our method discovers sparse DAGs that explicitly represent causal dependencies between states, actions, and rewards. The resulting causal structure enables (i) improved sample efficiency through reduced parameterization, (ii) better generalization via transferable causal mechanisms, and (iii) interpretability through attention-based causal attribution. Our attention-based inference networks learn to understand unknown MAC signaling protocols while coordinating channel access across multiple agents. Our simulations validate the approach and demonstrate significant improvements in convergence speed, throughput, and collision avoidance compared to model-free baselines.

References

[1] "Medium access control (mac) protocol specification," Tech. Rep. 3GPP TS 38.321, 3GPP, 2020, Release 16.

[2] N. Naderializadeh, J. J. Sydir, M. Simsek, and H. Nikopour, "Resource management in wireless networks via multi-agent deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3507–3523, Jun. 2021.

[3] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, Jun. 2019.

[4] H. Ye, G. Y. Li, and B. H. F. Juang, "Deep reinforcement learning for resource allocation in V2V communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[5] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for ai-enabled wireless networks: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1226–1252, 2021.

[6] J. I. Park and K. W. Choi, "Model-based reinforcement learning for wireless channel access," in *Proceedings of the IEEE 13th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, 2022, pp. 1393–1395.

[7] C. Hasan, A. Agapitos, D. Lynch, A. Castagna, G. Cruciata, H. Wang, and A. Milenovic, "Continual model-based reinforcement learning for data efficient wireless network optimisation," in *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2024.

[8] T. Gao, K. A. Neusypin, D. D. Dmitriev, B. Yang, and S. Rao, "Enhancing sample efficiency and exploration in reinforcement learning through the integration of diffusion models and proximal policy optimization," *arXiv preprint arXiv:2409.01427*, 2024.

[9] Y. Dong and J. Zhang, "Enhancing sample efficiency in online reinforcement learning via policy-guided diffusion models," in *Proceedings of the IEEE 13th International Conference on Computing and Pattern Recognition*, Oct. 2024, pp. 334–339.

[10] R. Schiewer, A. Subramoney, and L. Wiskott, "Exploring the limits of hierarchical world models in reinforcement learning," *Scientific Reports*, vol. 14, no. 1, pp. 26856, 2024.

[11] F. Zhang, J. Li, Y. C. Li, Z. Zhang, Y. Yu, and D. Ye, "Improving sample efficiency of reinforcement learning with background knowledge from large language models," *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[12] C. K. Thomas, C. Chaccour, W. Saad, M. Debbah, and C. S. Hong, "Causal reasoning: Charting a revolutionary course for next-generation AI-native wireless networks," *IEEE Vehicular Technology Magazine*, vol. 19, no. 1, pp. 16–31, 2024.

[13] A. Valcarce and J. Hoydis, "Toward joint learning of optimal MAC signaling and wireless channel access," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1233–1243, 2021.

[14] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, "Causal effect inference with deep latent-variable models," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.

[15] Z. Yu, J. Ruan, and D. Xing, "Explainable reinforcement learning via a causal world model," *arXiv preprint arXiv:2305.02749*, 2023.

[16] Wassily Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[17] S. S-Shwartz and S. B-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.

[18] S. Boucheron, G. Lugosi, and O. Bousquet, "Concentration inequalities," in *Summer School on Machine Learning*, pp. 208–240. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

Authorized licensed use limited to: Indian Institute of Technology. Downloaded on August 27,2026 at 16:27:47 UTC from IEEE Xplore. Restrictions apply.