Received 10 February 2025; accepted 7 March 2025. Date of publication 18 March 2025; date of current version 7 April 2025.

Digital Object Identifier 10.1109/OJCOMS.2025.3552277

Digital Twin-Enhanced Wireless Indoor Navigation: Achieving Efficient Environment Sensing With Zero-Shot Reinforcement Learning

TAO LI[1] (Member, IEEE), HAOZHE LEI[1], HAO GUO[1,2] (Member, IEEE), MINGSHENG YIN[1], YAQI HU[1], QUANYAN ZHU[1] (Senior Member, IEEE), AND SUNDEEP RANGAN[1] (Fellow, IEEE)

[1] Department of Electrical and Computer Engineering, New York University, New York, NY 11201, USA
[2] Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden

CORRESPONDING AUTHOR: H. GUO (e-mail: hg2891@nyu.edu)

The work of Haozhe Lei, Mingsheng Yin, Yaqi Hu, and Sundeep Rangan was supported in part by NSF under Grant 2345139, Grant 2148293, Grant 2133662, Grant 1952180, and Grant 1904648; in part the NTIA Public Wireless Innovation Fund; and in part by the Industrial Affiliates of NYU Wireless.
The work of Hao Guo was supported by Swedish Research Council (VR) under Grant 2023-00272.
Data is available on-line at https://github.com/Panshark/PIRL-WIN.

ABSTRACT Millimeter-wave (mmWave) communication is a vital component of future generations of mobile networks, offering not only high data rates but also precise beams, making it ideal for indoor navigation in complex environments. However, the challenges of multipath propagation and noisy signal measurements in indoor spaces complicate the use of mmWave signals for navigation tasks. Traditional physics-based methods, such as following the angle of arrival (AoA), often fall short in complex scenarios, highlighting the need for more sophisticated approaches. Digital twins, as virtual replicas of physical environments, offer a powerful tool for simulating and optimizing mmWave signal propagation in such settings. By creating detailed, physics-based models of real-world spaces, digital twins enable the training of machine learning algorithms in virtual environments, reducing the costs and limitations of physical testing. Despite their advantages, current machine learning models trained in digital twins often overfit specific virtual environments and require costly retraining when applied to new scenarios. In this paper, we propose a physics-informed reinforcement learning (PIRL) approach that leverages the physical insights provided by digital twins to shape the reinforcement learning (RL) reward function. By integrating physics-based metrics such as signal strength, AoA, and path reflections into the learning process, PIRL enables efficient learning and improved generalization to new environments without retraining. Digital twins play a central role by providing a versatile and detailed simulation environment that informs the RL training process, reducing the computational overhead typically associated with end-to-end RL methods. Our experiments demonstrate that the proposed PIRL, supported by digital twin simulations, outperforms traditional heuristics and standard RL models, achieving zero-shot generalization in unseen environments and offering a cost-effective, scalable solution for wireless indoor navigation.

INDEX TERMS Digital twin, millimeter-wave (mmWave) communication, wireless indoor navigation, reinforcement learning (RL), physics-informed learning, zero-shot generalization.

I. INTRODUCTION

HIGH-FREQUENCY transmission in the millimeter-wave (mmWave) bands is a key component of modern fifth-generation (5G) wireless systems, enabling not only massive data rates but also highly accurate positioning and localization capabilities [1], [2]. The wide bandwidth of mmWave signals, combined with the use of large antenna arrays, allows for fine-grained temporal and angular resolution of signal paths, making mmWave a powerful tool for the use cases such as indoor navigation and simultaneous localization and mapping (SLAM) [3], [4]. Unlike traditional camera-based sensors, mmWave signals provide the added advantage of penetrating beyond line-of-sight, allowing for robust navigation in obstructed and complex indoor environments [5], [6].

Indoor navigation using wireless and radar signals is essential in environments where traditional vision-based systems face challenges, such as poor lighting, smoke, or occlusions. This approach is highly valuable for robotics in warehouses, hospitals, and disaster response scenarios, where precise and robust navigation is critical. For instance, radar-based systems can provide reliable positioning in non-line-of-sight (NLOS) conditions, enabling robots to perform tasks like inventory management, patient assistance, or search and rescue. Additionally, wireless signal-based navigation leverages existing infrastructure (e.g., Wi-Fi or 5G networks) to guide robots with minimal hardware requirements. In such a wireless indoor navigation (WIN) problem [6], a mobile robot (or agent) must navigate to a target that broadcasts periodic mmWave signals, while the environment is unknown. However, effectively leveraging mmWave signals for indoor navigation remains challenging because physics-based heuristics, such as following the angle of arrival (AoA), provide effective zero-shot generalization in simple settings without requiring training, but they often fall short in complex wireless environments where multipath propagation, including reflections and diffractions [7]. Additionally, the effectiveness of such heuristics can be diminished by noisy signal measurements, leading to suboptimal navigation decisions.

To better simulate and evaluate these complex real-world environments, digital twins offer an innovative solution. A digital twin is a virtual replica of a physical system, enabling real-time simulation, optimization, and monitoring across various domains [8]. In the context of wireless communications, digital twins are used to model intricate environments and predict how wireless signals behave under different conditions. For indoor navigation, this means a digital twin of a building can simulate how mmWave signals propagate through walls, floors, and other obstacles, providing valuable insights for refining navigation algorithms [9]. Digital twins also offer a cost-effective alternative to physical testing, allowing machine learning models to be trained in virtual environments, significantly reducing development costs and improving scalability. Despite these advantages, machine learning models trained in digital twins often suffer from overfitting to specific environments, making them less effective in new settings. Extensive retraining is often required to adapt to different environments, which can be both time-consuming and computationally expensive [8].

In tackling such complex navigation problems besides classic machine learning, deep reinforcement learning (RL) has emerged as a promising end-to-end (e2e) framework, capable of learning policies directly from multimodal input data, including both vision and wireless signals. However, e2e RL methods are data- and computation-intensive, often requiring vast amounts of training data and GPU hours [10]. These models also tend to overfit the training environment, leading to poor generalization when deployed in new, unseen settings [11], and they often require pre-exploration to function effectively in new environments [5].

To overcome these limitations, digital twins can play a pivotal role in making reinforcement learning models more efficient and generalizable. This work proposes a physics-informed reinforcement learning (PIRL) approach, where physical principles derived from the digital twin environment are incorporated into the RL reward structure. As shown in Fig. 1, the key idea is to augment the standard e2e RL framework with a reward function shaped by physics-based metrics such as signal strength, AoA, and path reflections. These physically-motivated rewards guide the agent towards actions that align with real-world wireless propagation phenomena, thereby enhancing learning efficiency and improving generalization across different environments. Since these physical principles hold across diverse wireless environments, PIRL enables zero-shot generalization, allowing the trained model to navigate new environments without requiring extensive retraining. Two recent studies [12] and [13] develop structural digital twins for autonomous aerial vehicles using a library of physics-based models to adapt to changing structural conditions. Reference [12] integrates Bayesian state estimation for model selection and [13] enhances the interpretability with optimal trees, facilitating explainable assessments and optimal sensor placement. While these two studies focus on model-driven adaptation, in this paper, we focus on using PIRL to demonstrate a learning-based approach that dynamically transfers knowledge across different environments. As an extension of our previous conference version of the work [14], in this paper, we introduce a generalized wireless digital twin (WDT) framework, providing a more comprehensive theoretical foundation. Additionally, a rigorous analysis of link state properties is conducted to enhance the understanding of the system's dynamic behavior. The motivations and methodological details have been further enhanced, ensuring greater clarity and reproducibility. Furthermore, the computational complexity analysis has been expanded with a more in-depth discussion, offering a thorough evaluation of the proposed approach's efficiency. Finally, the paper includes an extensive discussion on scalability and potential applications, broadening the practical implications of the proposed model.

The contributions of the paper are summarized as follows:

- We design and formulate a WDT framework specifically for the WIN problem, enabling detailed simulation of complex wireless environments.
- We propose a novel physics-informed reward shaping approach for RL, simplifying implementation and improving training efficiency by embedding physics-based constraints.
- We demonstrate that PIRL reduces the training time and computational overhead compared to vanilla e2e

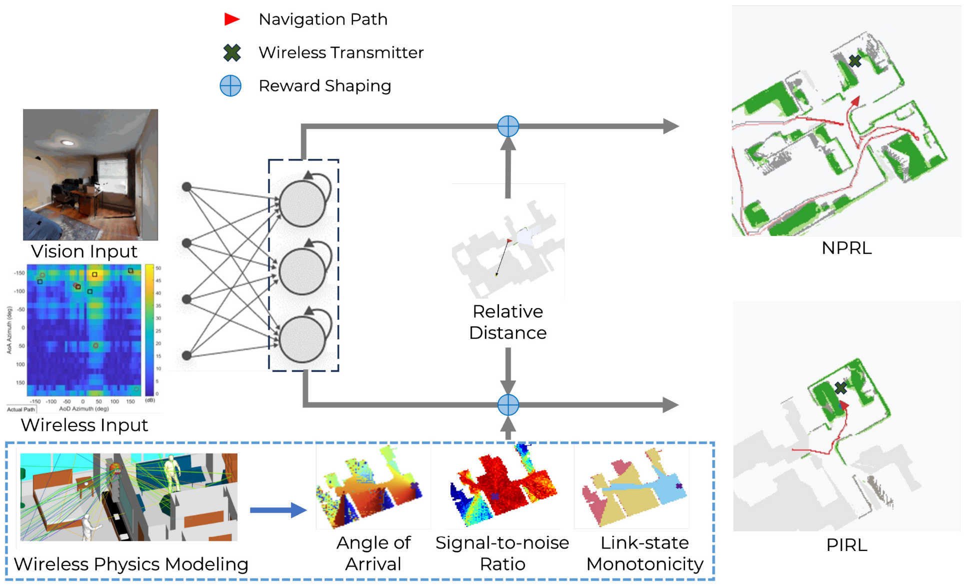


FIGURE 1. The wireless indoor navigation (WIN) requires the agent to navigate to the wireless transmitter in an unknown environment using multi-modal input, including vision and wireless. The non-physics e2e RL (NPRL), based on relative distance cost, fails to navigate efficiently in unseen scenarios. Trained to utilize physics prior, physics-informed RL (PIRL) acquires zero-shot generalization with interpretable policies.

RL, particularly valuable in scenarios where simulating wireless propagation is costly.

- Our experiments show that PIRL generalizes well to new environments in a zero-shot manner, outperforming existing heuristic and RL-based methods.
- Inspired by recent advances in explainable AI, we perform sensitivity analysis on the learned PIRL model, showing that its actions are interpretable and consistent with the underlying physics principles embedded in the reward function.

II. SYSTEM MODEL

In this section, we first introduce the indoor digital twin platform we developed. Then, we present the problem formulation of the WIN setup.

A. INDOOR DIGITAL TWIN FORMULATION

Wireless Information: Since wireless signals and RGB-based images are both multi-dimensional vectors associated with the agent's pose p, these two can be treated as vector fields in the considered WIN task. We refer to them as the wireless field $W(p)$ and the vision field $V(p)$, respectively. The agent pose is represented by

$$p = (x, y, \varphi), \tag{1}$$

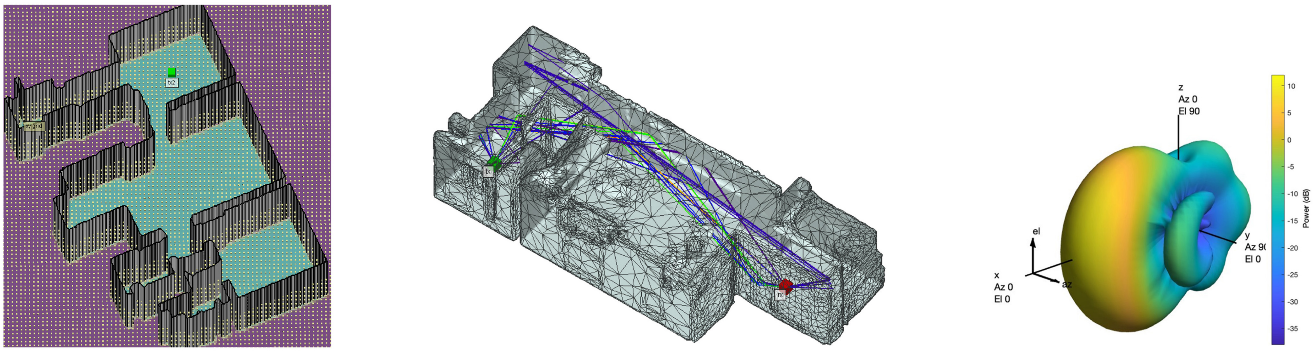
where x, y denotes the xy-coordinate of the agent measured in meters, and φ represents the orientation of the agent in radius (measured counter-clockwise from x-axis). For each pose p, the wireless field $W(p)$ describes the wireless signal received by the agent at the pose p, which includes the AoA and AoD (D: departure) for five channels. Following a similar setup in [6], selecting the strongest 5 paths for channel parameter estimation is essential due to the sparsity of wireless channels, especially in mmWave communications. Most of the received signal power is concentrated in a few dominant paths, while weaker paths contribute minimally and often fall within the noise floor. By focusing on these strongest paths, we can reduce computational complexity, improve estimation accuracy and align with practical 3GPP and wireless channel models. This approach ensures efficient modeling while maintaining the essential characteristics of the propagation environment. Here, AoA is the direction from which a wireless signal arrives at a receiving antenna, and AoD is the direction in which a wireless signal departs from a transmitting antenna. Several wireless methods are available to estimate paths from transmitted signals; we use a tensor decomposition method from [6] reviewed below in *Wireless Digital Twin*. Mathematically,

$$W(p) = \left(g_n, \Omega_n^{rx}, \Omega_n^{tx}\right)_{n=1}^{N} \in \mathbb{R}^{3\times N}, \tag{2}$$

where N is the maximum number of detected paths, and, for path n, g_n denotes its signal-to-noise ratio (SNR), Ω_n^{rx} and Ω_n^{tx} denote the AoA and AoD of the path n, respectively. Following the setup in [6], we use the top $N = 5$ paths.

Wireless Digital Twins: The genesis of WDTs proposed in this paper relies upon the Gibson model [15], a remarkable embodiment of real-world indoor reconstruction based on point clouds and RGB-D (D: Depth) cameras. The realism of the RGB input v_t in WDT surpasses that of the synthetic SUNCG dataset (a manually created large-scale dataset of synthetic 3D scenes with dense volumetric annotations), earlier utilized in exploration research [16]. The simulated wireless field $W(p)$ adheres to the mmWave simulation methodology expounded in [6].

To initiate the wireless data simulation, a mesh discretization of the 2D map with a cell width of 15 cm is implemented



(a) Receiver grid demo.

(b) Wireless link in ray tracing demo.

(c) Antenna pattern demo

FIGURE 2. **Wireless Channel Simulation Demos.**

as presented in Fig. 2(a), and wireless signals for each vertex point are generated. The simulation commences with the utilization of ray-tracing software such as Wireless InSite [17] to generate noise-free electromagnetic wave rays, as shown in Fig. 2(b). However, it is crucial to understand that these ray-tracing wireless propagation paths are not exact representations of real-world wireless channels that a robot could actually receive. That is to say, the robot cannot directly access the ray tracing paths in the real world, indicating a potential deviation between the simulated and real-world environments.

The subsequent phase involves the orchestration of antenna arrays, the induction of noise, and the subsequent disintegration of the channel, enabling the extraction of potential real-world robot-receivable wireless signal paths.

The rendering of high-resolution ray-tracing data to cover the entire map for channel sounding and robot navigation entails the use of a 2D receiver (RX) grid with a 15 cm interval, as shown in 2(a). Each task configuration includes one transmitter (TX) and an RX grid, referred to as a wireless link in wireless communication parlance. The strongest 25 rays out of 250 are chosen for each wireless link to simulate the wireless channel, as validated by numerous experiments [18], [19], [20].

The design of antenna arrays is shown in Fig. 2(c). Leveraging the theoretical foundations developed in Fig. 2c, three of 1x8 patch microstrip antenna arrays for the RX and three 2x4 patch microstrip antenna arrays for the TX are simulated. In this way, we can realize an omnidirectional coverage by 1) three 120° sectors in the azimuth plane, 2) elevation angles of 0°, 120°, and -120°, and 3) code-based beamforming for each array in each time index t. Moreover, to facilitate TX and RX detection, a known synchronization signal is transmitted by the TX, sweeping through a sequence of directions from the different TX arrays.

A 3D codebook of the mmWave system is designed following [21], [22] to obtain corresponding AoA and AoD in the channel decomposition post-medium wave. At this point, ray tracing data, antenna patterns, antenna group design, beamforming, and the codebook coalesce to simulate realistic indoor wireless channels. Notably, a loss of 6dB, inclusive of noise figures, is introduced during antenna group design, and additive white Gaussian noise (AWGN) assumed to be independent and identically distributed (i.i.d.) is added across the channel modeling RX antennas.

With the wireless channel acquired, the next step involves sub-channel (wireless path) estimation via low-rank tensor decomposition [23], [24]. This yields the wireless data

$$W(p) = \left(g_n, \Omega_n^{rx}, \Omega_n^{tx}\right)_{n=1}^{5} \in \mathbb{R}^{3\times 5}, \tag{3}$$

where g_n denotes the SNR of the n-th channel, and Ω_n^{rx} and Ω_n^{tx} denote the AoA and AoD of the n-th sub-channel.

Finally, the fusion of the wireless channel data with the Gibson indoor model culminates in the creation of WDTs, meticulously tailored for indoor navigation, as shown in Fig. 3. For additional details regarding the simulation process, readers may refer to the pertinent sections in [6].

B. WIRELESS INDOOR NAVIGATION: TASK SETUP

With the designed WDT, consider a WIN task setup as studied in [6], where a stationary target is positioned at an unknown location in an indoor environment. The target is equipped with a mmWave transmitter that broadcasts wireless signals at regular intervals. Equipped with a mmWave receiver, an RGB camera, and motion sensors, the agent aims to navigate to the target in minimal time. In contrast to the PointGoal task [25], WIN does not provide the agent with the target coordinates. The detailed environment setup and the agent's actuation/sensor models are presented below.

The agent starts from an initial pose p_1 and aims to locate and navigate to the target (the wireless transmitter) denoted by (x^*, y^*). We consider a WIN task where the agent operates in the presence of multiple kinds of information feedback that we denote with a vector

$$o_t = \left(m_t, \hat{p}_t, v_t, w_t\right), \tag{4}$$

where t is the time step, m_t is an estimate map, $\hat{p}_t = (\hat{x}_t, \hat{y}_t, \hat{\varphi}_t)$ is the estimated pose, $v_t = V(p_t)$ represents visual information, and $w_t = W(p_t)$ represents the wireless information. More details are listed below:

- *Map and pose estimation m_t and $\hat{p}_t$:* The map and pose estimates can come from any SLAM module. In

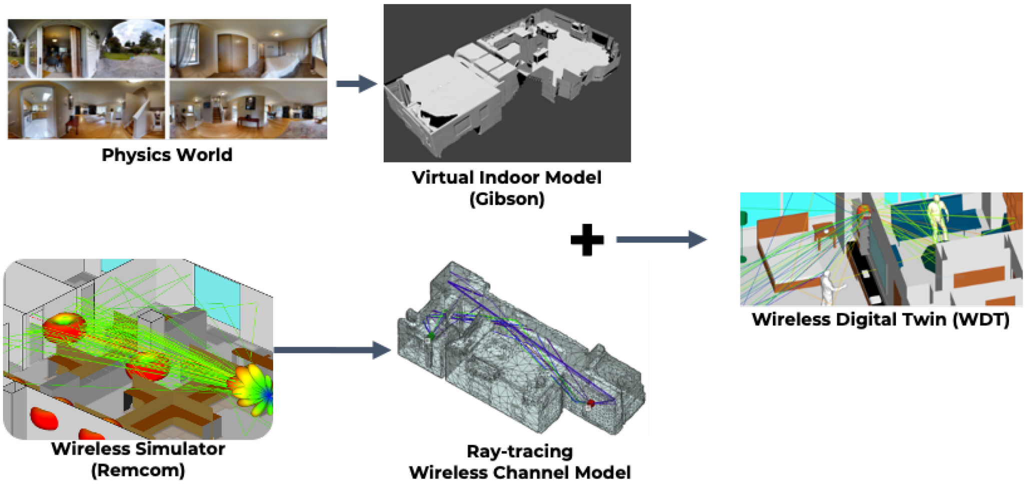

Physics World
Virtual Indoor Model (Gibson)
Wireless Digital Twin (WDT)
Wireless Simulator (Remcom)
Ray-tracing Wireless Channel Model

FIGURE 3. A summary of wireless digital twin (WDT).

the simulations below, we will use the state-of-the-art neural SLAM module proposed in [26] that provides robustness to the sensor noise during navigation. This SLAM module internally maintains a spatial map m_t and the agent's pose estimate $\hat{p}_t$ (different from the raw sensor reading $\bar{p}_t$) at each time step during the navigation process. The spatial map is represented as $m_t \in [0, 1]^{2\times M\times M}$ is a 2-channel $M \times M$ matrix, where $M \times M$ denotes the map size. Each element in the first channel represents the probability of an obstacle at the corresponding location, while those in the second channel denote the probability of that location being explored. We note that each "location" in the spatial map corresponds to a cell of size 5cm × 5cm in the physical world. The spatial map is initialized with all zeros at the beginning of the navigation. The agent starts at the center of the map facing east, i.e., $p_1 = (M/2, M/2, 0)$. Of particular note is that the map center may not coincide with the center of the floor plan; the agent can start with any location in the indoor environment. The map updates rely on a pre-trained neural SLAM module, which is essentially a neural network [25] that adjusts the entries of the two channels to update the obstacle information along the navigation.

- *Visual information v_t:* $V(p) \in \mathbb{R}^{3\times L_1\times L_2}$ is the 3-channel RGB camera image input at the pose p, where L_1 and L_2 denote the height and the width, respectively. In addition to the wireless sensor and the camera, the agent is also equipped with motion sensors. The sensor readings lead to the estimate of the agent pose $\hat{p} = (\hat{x}, \hat{y}, \hat{\varphi})$, which can be different from the agent's authentic pose p. The difference $\varepsilon_{\text{sen}} = \hat{p} - p$ is referred to as the sensor noise.
- *Wireless information w_t:*

$$W(p) = \left(g_n, \Omega_n^{rx}, \Omega_n^{tx}\right)_{n=1}^{N} \in \mathbb{R}^{3\times N}, \tag{5}$$

where N is the maximum number of detected paths along which signals propagate. For the n-th path, g_n denotes its signal-to-noise ratio (SNR), Ω_n^{rx} and Ω_n^{tx} denote the AoA and AoD, respectively. We consider the top $N = 5$ paths with the strongest signal strengths among all paths (see [6]).

Finally, for agent actions, following [26], we assume the agent utilizes three default navigation actions,

$$\mathcal{A} := \{a_F, a_L, a_R\}. \tag{6}$$

Here, $a_F = (d, 0, 0)$ denotes the moving-forward command with a travel distance equal to the grid size $d = 25$ cm; and $a_L = (0, 0, -10°)$ and $a_R = (0, 0, 10°)$ stand for the control commands: turning left and right by 10 degrees, respectively.

C. WIN OBJECTIVE

Navigating within an unknown environment can be viewed as sequential decision-making using partial observations. The agent's state is given by its authentic pose p_t that remains hidden, and only partial information o_t, which includes the spatial map m_t, pose estimate $\hat{p}_t$, RGB images v_t, and wireless signal w_t, collected by sensors is available for decision-making at each time step. The state transition as shown in the actuation model presented in Section II-B is Markovian

$$p_{t+1} = p_t + a_t. \tag{7}$$

An important remark is that even though the agent acquires the initial pose and subsequent action sequence, it is unaware of its actual pose. Such unobservability of the underlying state is due to the fact that actuators may not perfectly carry out the control action under various sources of error and uncertainty, such as mechanical friction, voltage variations, and inertia. The agent has to utilize sensor readings from the motion sensor (wheel encoders) and RGB sensor to derive an estimate $\hat{p}_t$. Hence, the WIN task is a partially observable Markov decision process (POMDP), with the observation kernel o_t being too complicated to be analytically modeled.

The navigation performance can be measured through a cost function defined as the Euclidean distance (or any distance metric, e.g., geodesic distance) between the current pose and the target position

$$c_t = \|x_t - x^*\|^2 + \|y_t - y^*\|^2. \tag{8}$$

Denote $\mathcal{H}_t := \{(o_k, a_k)_{k=1}^{t-1}, o_t\}$ the set of all possible observations up to time t, and $\mathcal{H} := \cup_{t=1}^{H} \mathcal{H}_t$ the union of all histories, where H denotes the horizon length. The agent's objective in WIN is to find an optimal policy $\pi : \mathcal{H} \to \mathcal{A}$ such that the cumulative cost $\mathbb{E}_\pi[\sum_{t=1}^{H} c_t]$ is minimized, which is a customary setup in mobile robot navigation tasks [25]. Since the cost function is non-negative and attains zero if and only if the agent arrives at the target position, minimizing the cumulative cost is equivalent to finding the shortest path, implying that the agent aims to navigate to the target in minimal time.

III. PHYSICS-INFORMED REINFORCEMENT LEARNING: MOTIVATIONS AND REWARD SHAPING WITH THREE PHYSICS TERMS

With formulated WIN problem (4) and (6), in this section, we first present the enhanced state-of-the-art method using deep RL and its limitations. Then, we introduce our proposed PIRL with a focus on three specific physics-informed terms: link states, reversibility, and SNR.

A. CLASSIC DEEP RL

The planning algorithms for POMDP [27] are not suitable for WIN, since the state and the observation space are of high dimensions and continuum, and the observation kernel remains unknown. To create model-free learning-based navigation, one can apply deep reinforcement learning, such as proximal policy optimization (PPO) [28], to approximately solve the cost-minimization problem in (9), where the policy π is represented by an actor-critic neural network [29], and its model weights are denoted by $\theta \in \mathbb{R}^d$.

To address the partial observability in WIN, we incorporate a recurrent module [30] into the actor-critic network architecture (see Section IV-A). With the recurrent neural network (RNN), the policy $\pi(\theta)$ need not take in all past observations $\{(o_k, a_k)_{k=1}^{t-1}, o_t\}$, and instead, the current partial observation suffices, as RNN can memorize historical input and integrate information feedback across time [30]. We refer to RL with the loss function (9) as **non-physics-based RL** (NPRL), to differentiate it from the physics-informed RL to be described shortly

$$\min_\theta \mathcal{L}_{\mathrm{RL}}(\theta) := \mathbb{E}_{\pi(\theta)} C_{\mathrm{RL}}, \quad C_{\mathrm{RL}} = \sum_{t=1}^{H} c_t. \tag{9}$$

However, as presented in the literature [31], [32], we observe in the initial experiments that when NPRL policies are applied to the WIN problem, they exhibit poor generalization ability and sample efficiency. For example, the NPRL agent trained for one task (a given map and one target position within the map) even fails to navigate to another target within the same map. Due to multiple reflections and diffractions of mmWave, the wireless field $W(p)$ changes drastically when the transmitter moves from one location to another, especially when the indoor environment displays complex geometry. Consequently, model weights learned for (overfit) one task are barely relevant to another. In addition to limited generalization, the NPRL agent requires an astronomical amount of samples due to catastrophic forgetting. Since wireless fields vary across different tasks, knowledge of the previously learned task may be abruptly lost as information relevant to the current task is incorporated. Hence, the NPRL agent needs to be re-trained under previous tasks, leading to time-consuming shuffle training [10].

B. PHYSICS-INFORMED REINFORCEMENT LEARNING

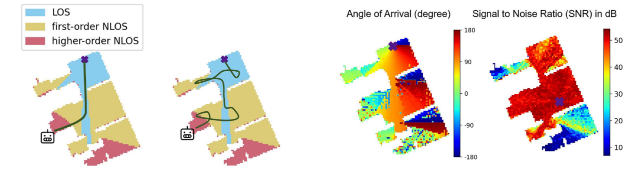
Physics-informed machine learning or RL has emerged as a promising approach to simulate and tackle multiphysics problems in a sample-efficient manner [33]. The gist is that neural networks can be trained from additional information obtained by enforcing physics laws. Existing general-purpose strategies of distilling the physics-domain-knowledge into the RL agent include supervised-learning approaches such as imitation learning [34], and RL approaches such as offline/batch RL [35], [36] and vanilla RL, i.e., online policy learning [37], where the agent repeatedly interact with the digital twin to acquire feedback. This work considers the simple online learning approach based on WDT because we need a fair comparison between our proposed PIRL and other baseline wireless navigation approaches that are based on online RL on sample efficiency and generalization.

Adopting online RL, we thus propose to simply augment the cost with *physically-motivated reward shaping*. Specifically, the augmented cost function is defined as

$$\mathcal{L}(\theta) := \mathbb{E}_{\pi(\theta)}[C_{\mathrm{RL}} + \lambda_{\mathrm{LS}} C_{\mathrm{LS}} + \lambda_{\mathrm{AoA}} C_{\mathrm{AoA}} + \lambda_{\mathrm{SNR}} C_{\mathrm{SNR}}], \tag{10}$$

where the additional terms are motivated by physics principles in WIN: C_{LS}, for link-state monotonicity, C_{AoA} for the angle of arrival direction following, and C_{SNR} for SNR increasing. λ_{LS}, λ_{AoA}, and λ_{SNR} are weighting constants. In the following, we present these three physics-informed terms in detail.

In the experiments, the configuration of hyperparameters and weighting constants before each reward term follows the intuition that each component is equally important and shall not dominate or be dominated by other reward components. Note that each reward component falls within intervals of similar ranges. For example, the values for C_{AoA} and C_{SNR} range from 0 to 180. The distance reward, varying across the maps and the target locations, typically assumes a positive value no greater than 200. Since these reward components are of similar ranges, we confine the reward weights to [0.5, 2], over which we utilize a grid search to explore various weight combinations. The detailed hyperparameter setups are deferred to Section V.



(a) The link state decreases monotonically along the shortest path.

(b) The agent can move reversely along the AoA and explore a high SNR area in NLOS.

FIGURE 4. **The physics principles in WIN.**

1) Monotonicity of Link States: In mmWave applications, link states are of great importance [1], [6], which are primarily categorized into Line-of-Sight (LOS) and NLOS. A location (x, y) (or equivalently a pose p) is said to be of *LOS* if there is a wireless signal path wherein electromagnetic waves traverse from the source to the receiver without encountering any hindrances. In contrast, *NLOS* signifies the absence of such a direct visual path. NLOS can further be subdivided into first-order, second-order, third-order, and so forth. First-order NLOS (1-NLOS) implies that at least one electromagnetic wave in the wireless link undergoes a single reflection or diffraction. Likewise, second-order NLOS (2-NLOS) suggests at least one electromagnetic wave undergoing two instances of reflection or diffraction. Similar arguments apply to higher-order NLOS, denoted by 2^+-NLOS. Define

$$\ell(p) \in \{0, 1, 2\} \tag{11}$$

as the link state of the pose p, where the link state evaluation 0, 1, and 2 represent the LOS (0-NLOS), 1-NLOS, and 2^+-NLOS scenarios, respectively. Note that the link state is a wireless terminology instead of the actual state input to be fed into RL models. Instead, the agent learns to infer the link state from the raw wireless inputs $W(p)$ [6].

Fig. 4(a) presents a distribution map of link state for indoor wireless signals. The purple cross represents the target location. The LOS coverage, by definition, is a connected area, unlike 1-NLOS, and 2^+-NLOS coverage. Hence, when the agent enters the LOS area, the shortest path to the target is the straight line connecting the two (see Fig. 4(a)), which remains within the LOS area. Another important observation is that the LOS area must be bordered by 1-NLOS, which is then bordered by 2-NLOS, and so forth. In other words, if the link state increases as the agent navigates, the resulting path cannot be optimal. This observation leads to the following Theorem.

Theorem 1: A necessary condition for a path to be optimal is that the link state decreases monotonically along the path, which motivates the term

$$C_{\text{LS}} = \sum_t \max\{0, \ell_t - \ell_{t-1}\}. \tag{12}$$

Mathematically, given navigation path $\vec{p} := (p_1, \ldots, p_H)$, p_t denotes the pose at time t, let $\ell_t = \ell(p_t)$ be the link state of the pose p_t. A necessary condition of $\vec{p}$ being the shortest path is that the link state ℓ is non-increasing along the path: $\ell_i \leq \ell_j$, for $0 \leq j < i \leq H$.

Proof: Consider a navigation path $\vec{p} := (p_1, \ldots, p_H)$, p_t denotes the pose at time t. Let $\ell_t = \ell(p_t)$ be the corresponding link state of the pose p_t. Suppose, for the sake of contradiction, that for the shortest path $\vec{p}$, there exists $0 \leq j < i \leq H$ such that $\ell_i > \ell_j$, and we consider two possible cases: 1) $\ell_i = 1 > 0 = \ell_j$, and 2) $\ell_i = 2 > 1 = \ell_j$. In the first case, when entering the LOS area, the agent shall remain in the LOS, as we discussed earlier. Hence, $\vec{p}$ is not optimal. In the second case, since ℓ cannot jump from 2 to 0, there must be some 1-NLOS after p_i. Let $k > i$ be the smallest index for which $\ell_k = 1$, then connecting ℓ_j and ℓ_k yields a shorter path, conflicting the optimality. Fig. 4(a) presents a visualization of the two cases. ■

2) Reversibility of mmWaves: Similar to the principle of reversibility of light, the mmWave follows the same path if the direction of travel is reversed. This reversibility principle leads to a simple yet effective navigation strategy: *following the AoA of the strongest path*, which experiences the least number of reflections. Besides, [6] shows that following the AoA of the strongest path in 1-NLOS cases (NLOS with a single reflection) generally leads to decent navigation since it tends to find a route around the obstacle. However, for 2-NLOS cases ($\ell_t = 2$), following the AoA may not be a reliable solution, since it arises from multiple reflections or diffractions. To impose this angle tracking, we add the term

$$C_{\text{AoA}} = \sum_{t=1}^{H} |\widetilde{\Omega}_t - \Omega_{1,t}^{rx}|^2 \cdot \mathbb{1}_{\{\ell_t \neq 2\}} \tag{13}$$

into (10) where $\widetilde{\Omega}_t$ is the agent's moving direction derived from the action and $\Omega_{1,t}^{rx}$ is the AoA of the strongest path included in the wireless information w_t.

3) Navigation in 2^+-NLOS and the Gradient Field of SNR: Due to reflections, diffractions, and measurement noises, the reversibility principle is less effective in 2^+-NLOS. Denote by $g(p) = \sum_i g_i(p)$ the overall SNR at the pose p, or equivalently, the location (x, y). A key observation is that g displays remarkable declines in the transit from the LOS and 1-NLOS to 2^+-NLOS areas, see Fig. 4(b). Upon statistically analyzing 21 maps, it is observed that navigating from the 1-NLOS position to the nearest 2^+-NLOS position leads to an average decline of 25.2 dB in SNR. Hence, we propose a navigation heuristic in 2^+-NLOS scenarios: the agent should move along the reverse direction of the SNR gradient field $-(\nabla_x g, \nabla_y g)$ (finite differences in practice), i.e., toward the direction with the stronger signal strength. We remark that such a heuristic is less helpful in the LOS and 1-NLOS, where ∇g is relatively insubstantial: the difference between SNRs of two adjacent mesh vertices is mostly within 3 dB. To encourage the policy to increase in SNR, we add the cost

$$C_{\text{SNR}} = \sum_{t=1}^{H} |\widetilde{\Omega}_t - \nu_t|^2 \tag{14}$$

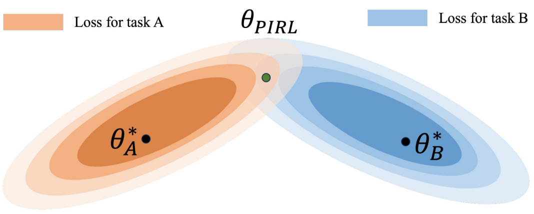


FIGURE 5. PIRL targets the suboptimal θ_{PIRL} shared by various tasks, instead of optimal policies θ_A^*, θ_B^* for individual tasks.

where ν_t denotes the angle between $-\nabla_{x,y} g(p_t)$ and the x-axis. In numerical implementations, ν_t is replaced by the steepest descent direction approximated using finite differences of the mesh points. Consider a discretization of the angle range $\{-180, -170, \ldots, 0, \ldots, 170, 180\}$. For each relative angle from the discrete set, we compute the average of SNR evaluations at all mesh points (where the wireless data is collected, see Section II-A) along the direction. The highest SNR direction is set to be ν_t.

One important observation is that the physics-based reward shaping is not a potential-based transformation [38]. To see this, consider a sequence of poses $p_1 \to p_2 \to \cdots \to p_n \to \cdots \to p_1$ such that the agent can travel through them in a cycle, which can incur a net positive cost, e.g., C_{LS} is strictly positive when traversing from LOS to NLOS and then back to LOS. Hence, the policy invariance theorem [38] tells that (10) leads to a navigation policy distinct from the shortest path prescribed by (9). For example, following AoA in the 1-NLOS may yield a detour around a corner rather than the shortest path. Even though PIRL is not optimal, it targets suboptimal solution θ_{PIRL} shared across various tasks (because physics principles are invariant) as shown in Fig. 5. The shared suboptimality alleviates catastrophic forgetting in training and creates zero-shot generalization in testing.

IV. PROPOSED PIRL ALGORITHM AND IMPLEMENTATION

To accommodate the heterogeneous information (vision and wireless), we design a hierarchical RL policy inspired by [26]. The RL policy consists of two separate neural networks,

$$\pi(\theta) = (\pi_G(\theta_G), \pi_L(\theta_L)). \tag{15}$$

Here, π_G is a global policy network that sets a long-term goal location, which does not represent the agent's estimate of the target position but rather a waypoint on the navigation path. π_L is a local policy that takes in the long-term goal and generates a sequence of navigation actions. A schematic illustration is presented in Fig. 6, and the pseudocode is summarized in Algo. 1. The following subsections present the key components of the proposed PIRL algorithm.

A. OVERALL PIRL HIERARCHICAL POLICY STRUCTURE

Since the wireless information w_t in (5) is directly generated by the transmitter, the global policy needs to produce a series of waypoints using such information. Specifically, denote by

$$\alpha_t = \pi_G(w_t \mid \theta_G) \tag{16}$$

the output from the global policy, which consists of an estimated angle $\hat{\Omega}_t$ and link-state estimates $\hat{\ell}_t$ based on the current wireless input w_t. Given the global policy output α_t and the agent current pose estimate $\hat{x}_t, \hat{y}_t$, the long-term goal $p_t^L = (x_t^L, y_t^L)$ can be expressed as

$$x_t^L = \hat{x}_t + \delta_t \cos \hat{\Omega}_t, \quad y_t^L = \hat{y}_t + \delta_t \sin \hat{\Omega}_t, \tag{17}$$

where δ_t is a predicted distance depending on the link state estimate $\hat{\ell}_t$. Also, the predicted distance is given by

$$\delta_t = \mathbb{1}_{\{\hat{\ell}_t = 2\}} \cdot D_b + \left(1 - \mathbb{1}_{\{\hat{\ell}_t = 2\}}\right) \cdot D_s, \tag{18}$$

where $D_b = 7.5$ leading to aggressive exploration and $D_s = 2.5$ to a conservative one. The intuition behind this setting is as follows: if the agent is in a state of 2^+-NLOS, it prefers to search for the goal aggressively; if not, the agent prefers to move slowly, being more cautious.

Once the global policy determines the long-term goal, a path planner denoted by f_{plan}, based on the Fast Marching method [39], computes the shortest path from the current location to the goal using the spatial map m_t and the pose estimate $\hat{p}_t$ from the SLAM module. The unexplored area is considered a free space for planning. The output of the planner is a short-term goal

$$p_t^S = f_{\text{plan}}\left(p_t^L, m_t, \hat{p}_t\right), \tag{19}$$

which is the farthest point on the path within the grid size $d = 0.25$m from the agent. Then, the local policy takes in the path-planning output and the camera images, producing navigation actions

$$a_t = \pi_L\left(v_t, p_t^S | \theta_L\right) \tag{20}$$

for collision avoidance.

Since the navigation actions correspond to small movements (e.g., turn left/right by 10°), the agent needs to implement a sequence of actions to move from the current position to the short-term goal before calling the global policy to update the long-term goal. Hence, our PIRL operates the global and local policy at different timescales. Denote by H_g the global decision horizon, indicating the total number of calls to the global policy. At each global time step $t \in \{1, 2, \ldots, H_g\}$ (i.e., global policy call), the local policy operates in a local decision horizon denoted by H_l: at each local time step $\tau \in \{1, 2, \ldots, H_l\}$, the policy takes in the visual information and executes an action (line 11-14) in Algo. 1.

We finally conclude the policy network overview by presenting its neural network architecture. The global policy

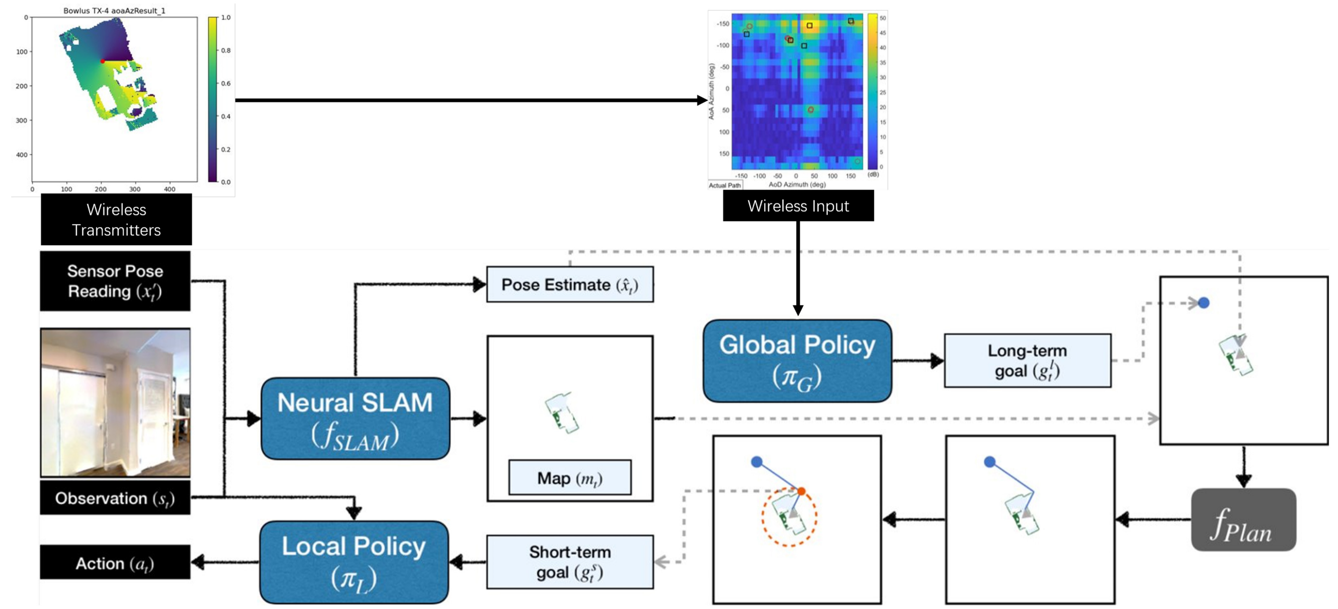


FIGURE 6. The hierarchical structure of the RL policy. The global policy takes in the wireless input and produces long-term goals (blue dot) fed to the local policy that generates a sequence of navigation actions to an associated short-term goal (red dot). The local policy relies on the active neural-SLAM module [26] for pose and map estimates that are utilized later by the planner to produce short-term goals.

comprises a recurrent neural network architecture, which includes a linear sequential wireless encoder network with two layers, followed by fully connected layers and a Gated Recurrent Unit (GRU) layer [40]. Additionally, there are two distinct layers at the end, referred to as the actor output layer and the critic output layer. The local policy is constructed using a recurrent neural network architecture. It incorporates a pre-trained ResNet18 [41] as the visual encoder, which is followed by fully connected layers and a GRU layer.

B. GLOBAL AND LOCAL POLICY TRAINING

We employ different training processes for the global and local policies since the two policy networks are of different functionalities. For the local policy training, we follow the practice in [26], where the local reward is determined by the agent's proximity to the short-term objective and the cross-entropy loss is utilized. The local policy undergoes training via imitation learning, specifically through behavioral cloning, and we refer the reader to [26] for more details.

Our contribution mainly lies in global policy training that applies PPO [28] with the global reward shaped by physics terms in Section III. While (10) lays down the general principle for physics-informed reward shaping, the implementation further enforces the monotonicity of link states in the LOS and 1-NLOS, and the resulting global reward function design is as follows.

$$r_t^g = \begin{cases} \lambda_{\text{LS}} C_{\text{LS}} \cdot \left(\zeta_1 e^{-\zeta_2 c_t} - \lambda_{\text{AoA}} C_{\text{AoA}}\right), & \hat{\ell}_t \in \{0, 1\} \\ -\lambda_{\text{SNR}} C_{\text{SNR}}, & \hat{\ell}_t \in \{2\}, \end{cases} \tag{21}$$

where ζ_1 and ζ_2 are hyperparameters.

Algorithm 1 PIRL Algorithm

```
Require: Global policy $\pi_G$, pre-trained Local policy $\pi_L$, and
    the planner module $f_{PLAN}$; time horizon $H_g, H_l$
Initialize global policy parameters $\theta_0$;
while not converged do
    Reset environment and agent state;
    Set global time $t = 0$;
    Sample initial time $w_t, v_t$ for policies;
    while $t < H_g$ do
        Set local time $\tau = 0$;
        Sample action $\alpha_t$ from global policy $\pi_G(w_t|\theta)$;
        Compute long-term goal $p_t^L$ using $\alpha_t$;
        Compute short-term goal $p_t^S$ using planner $f_{PLAN}$;
        while $\tau < H_l$ do
            Sample action set $a_l$ from local policy
    $\pi_L(v_t, p_t^S)$;
            Execute action $a_l$ and observe next state $v_{\tau+1}$;
            Update local time $\tau = \tau + 1$;
        Update global policy parameter $\theta$ using collected
    data and the PPO algorithm;
        Observe next state $w_{t+1}$;
        Update global time $t = t + 1$;
Output $\theta_t$
```

We are now ready to illustrate the PPO algorithm for global policy training. Denoting

$$G_t = \sum_{k=0}^{H_g - 1} \gamma^k r_{t+k+1}^g \tag{22}$$

as the discounted future reward starting from t, we can derive the state-value function under a global policy $\pi(\theta)$ (also denoted by π_θ) as

$$V_\theta(w) = \mathbb{E}_{\alpha\sim\pi}[G_t|w_t]. \tag{23}$$

Similarly, we can determine the value function of a (state, action) pair (i.e., Q function) under such policy as

$$Q_\theta(w,\alpha) = \mathbb{E}_{\alpha\sim\pi}[G_t|w_t,\alpha_t]. \tag{24}$$

To measure the performance of an action at a certain state, we can use the advantage function defined as [28]

$$A_\theta(w,\alpha) = Q_\theta(w,\alpha) - V_\theta(w). \tag{25}$$

Unlike the vanilla policy gradient method [42] that optimizes the value function, PPO considers a clipped surrogate objective. Let $\pi_{\theta_{\text{old}}}$ represent the old policy from the last update, and π_θ denote the new policy. The probability ratio is denoted as

$$\mu(\theta) = \frac{\pi_\theta(\alpha|w)}{\pi_{\theta_{\text{old}}}(\alpha|w)}. \tag{26}$$

Additionally, we introduce a small hyperparameter ϵ. To ensure the ratio remains within a certain range, we define the clipping function as

$$\text{clip}(\mu(\theta), 1-\epsilon, 1+\epsilon). \tag{27}$$

This function restricts the ratio to be no greater than $1+\epsilon$ and no less than $1-\epsilon$. Therefore, the objective function under this clipping is:

$$J^{\text{CLIP}}(\theta) = \mathbb{E}\Bigg[\min\Bigg(\mu(\theta)\hat{A}_{\theta_{\text{old}}}(w,\alpha), \text{clip}(\mu(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\theta_{\text{old}}}(w,\alpha)\Bigg)\Bigg], \tag{28}$$

where $\hat{A}_{\theta_{\text{old}}}(\cdot)$ represents the estimated advantage for the old policy using sample rewards G_t. The objective function $J^{\text{CLIP}}(\theta)$ calculates the expectation over the minimum value between two terms: the first term is the product of the ratio and the estimated advantage under the old policy, while the second term is the product of the clipped ratio and the estimated advantage under the old policy. Such an operation addresses the training instability with extremely large parameter updates and big policy ratios.

When implementing PPO on a network architecture with shared parameters for both the policy (actor) and value (critic) functions, the critic is responsible for updating the value function to obtain the estimated advantage function $\hat{A}_{\theta_{\text{old}}}(\cdot)$. On the other hand, the actor serves as our policy model. To promote sufficient exploration in the learning process, an error term, $(V_\theta - V_{\text{target}})^2$, and an entropy bonus $H(w, \pi_\theta(\cdot|w))$, is introduced for value estimation and exploration encouragement, where V_{target} represents the discounted cumulative reward associated with a sample trajectory. When a given trajectory ends, target state values are computed as

$$V_t^{\text{target}} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V_{\theta_{\text{old}}}(w_{t+k}), \tag{29}$$

where k is the length of trajectory segment. Such segmentation breaks a large sample trajectory into multiple segments, leading to multiple PPO updates along the whole sample trajectory [28]. In summary, the overall PPO objective function can be written by

$$J^{\text{PPO}} = \mathbb{E}\Big[J^{\text{CLIP}}(\theta) - \xi_1\big(V_\theta - V_{\text{target}}\big)^2 + \xi_2 H(\pi_\theta(\cdot|w))\Big], \tag{30}$$

where ξ_1 and ξ_2 are two hyperparameters. By optimizing the objective function $J^{\text{PPO}}(\theta)$, we obtain the optimal policy π.

V. EXPERIMENTS

This section evaluates the proposed PIRL approach for WIN tasks, aiming to answer the following questions.

- *Sample Efficiency:* does the PIRL take fewer training data than the non-physics-based baseline?
- *Zero-shot Generalization:* can PIRL navigate in unseen wireless environments without fine-tuning?
- *Interpretablility:* does the PIRL conform to the physics principles, leading to interpretable navigation?

We briefly touch upon the training procedure and the experiment setup in the ensuing paragraphs. All experiments are conducted using a Linux GPU workstation with an AMD Threadripper 3990X (64 Cores, 2.90 GHz) and an NVidia RTX.

A. EXPERIMENT SETUP

The experiment includes 21 different indoor maps (15 for training and 6 for testing) from the Gibson dataset [15] labeled using the first 21 characters in the Latin alphabet (A, B, ..., U). Table 1 presents the label map correspondence, where the left-hand side displays the maps used for training, while the right-hand side displays those for testing. Each map includes ten different target positions labeled using numbers (1,2, ..., 10). The agent's starting position is fixed for each map regardless of the target position, depending on which, the ten targets for each map are classified into three categories. The first three targets (1-3) are of LOS (i.e., the agent's starting position is within the LOS area), the next three (4-6) belong to 1-NLOS, and the rest four (7-10) correspond to 2^+-NLOS scenarios. For each task (e.g., A1), the maximum number of training episodes is 1000, and the training process terminates if the agent completes the task in more than 6 episodes out of 10 consecutive ones.

During the training phase, the first 15 maps (A-O) with associated 10 task positions are utilized to learn a PIRL policy in sequential order. The training process follows a specific sequence, starting with task A and progressing to A10, followed by training under tasks B1 to B10. Each task consists of 1000 training episodes. This procedure is repeated until the agent has been exposed to all 15 maps with all target positions. The intuition behind this sequential training approach is to gradually increase the complexity of the tasks.

TABLE 1. Label-map correspondence.

Label	Map Name	Label	Map Name	Label	Map Name
A	Bowlus	I	Capistrano	P	Woonsocket
B	Arkansaw	J	Delton	Q	Dryville
C	Andrian	K	Bolton	R	Dunmor
D	Anaheim	L	Goffs	S	Hambleton
E	Andover	M	Hainesburg	T	Colebrook
F	Annawan	N	Kerrtown	U	Hometown
G	Azusa	O	Micanopy		
H	Ballou				

It begins with LOS cases, which are relatively simple, then proceeds to 1-NLOS cases, and finally to 2^+-NLOS cases, which pose a higher level of difficulty.

We consider three baseline navigation algorithms, namely,

- Non-physics-based RL (NPRL): the RL policy is of the same architecture as our proposed PIRL, whereas the reward function is not physics-informed, i.e., only $\mathcal{L}_{RL}$ in (9).
- Wireless-assisted navigation (WAN): this non-RL-based method, put forth in [6], relies on a physics-based heuristic that utilizes wireless signals (following AoAs) exclusively within LOS and 1-NLOS scenarios while exploring randomly in 2^+-NLOS. WAN uses a pre-trained classification model to infer the link state.
- Vision-augmented SLAM (V-SLAM), which is a vision-augmented version of the active neural SLAM (AN-SLAM) in [26].

The first two are primary baselines since our PIRL is a hybrid of the two methodologies. Additionally, to highlight the necessity of leveraging wireless signals in indoor navigation, we consider the third baseline where V-SLAM only takes in RGB image data without wireless inputs. The V-SLAM agent is capable of localizing the target once it falls within the visual (LOS), whereas in the NLOS, V-SLAM reduces to the AN-SLAM, aiming to explore as much space as possible.

For the first NPRL baseline policy, we consider rotation training to alleviate catastrophic forgetting. The rotation training generally follows the task sequence as the sequential training in PIRL. Yet, after finishing the training on the current task, we randomly select a set of previous tasks to re-train the model before moving to the next task in the sequence to refresh NPRL's "memory". The number of re-train tasks is set to be half of the total number of finished tasks. Our experiments use the pre-trained vision model and neural-SLAM module in the other two baselines. We report experimental results based on 20 repeated tests with different random seeds. Moreover, the system parameters, including their detailed descriptions and values in the simulations, are summarized in Table 2.

B. SAMPLE EFFICIENCY

We first evaluate the sample efficiency of the PIRL training process by comparing the number of training episodes of PIRL in LOS, 1-NLOS, and 2^+-NLOS with those of NPRL.

TABLE 2. Hyperparameters used in the PIRL algorithm.

Hyperparameter	Value
γ, discount factor for future rewards	0.99
ϵ, clipping parameter for PPO	0.2
λ_{LS}, weight for link-state cost component	1.0
λ_{AoA}, weight for AoA tracking cost component	0.5
λ_{SNR}, weight for SNR gradient cost component	0.3
ξ_1, weight for value loss in PPO	0.5
ξ_2, weight for entropy bonus in PPO	0.01
ζ_1, reward hyperparameter for AoA adherence	0.7
ζ_2, reward hyperparameter for SNR optimization	0.3
Learning Rate	3×10^{-4}
Batch Size, Number of samples per training batch	64
α, exploration rate (e.g., epsilon in epsilon-greedy)	0.1
β, regularization parameter for policy entropy	0.01
H_g, H_l, Global and local decision horizons	100, 50

The bar plot in Fig. 7 gives a visualization of the sample efficiency in the training phase on map A (the first map used in the training) and I (midway in the training). In the early stage of the training, no remarkable difference between the two is observed. However, as the training proceeds, PIRL demonstrates a superior sample efficiency on map I, compared with NPRL. This is because the PIRL agent learns to utilize the physics principles that persist across different wireless fields, after being trained on first a few maps. One can see that the PIRL policy already acquires generalization ability to some extent at this point, such that lightweight training would be sufficient for navigating in new environments. In contrast, the NPRL agent, using vanilla end-to-end learning, may be confused when exposed to drastically different wireless fields. Hence, the prior experience does not carry over to the new environment, and NPRL needs to learn almost from scratch.

C. GENERALIZATION

We first highlight that our testing environments (new maps with different target positions) are structurally different from training cases. Different room topologies and wireless source locations create drastically different wireless fields unseen in the training phase, as the reflection and diffraction patterns are distinct across each setup. We collect the testing performance of three baselines and our PIRL on maps P to U, and report the average results of 20 repeat tests under different random seeds. Since baselines and PIRL use different reward designs, we consider the metric normalized path length (NPL) defined as the ratio of the actual path length (the number of navigation actions) over the shortest path length of the testing task (the minimal number of actions). The closer NPL is to 1, the more efficient the navigation is. The comprehensive comparison is summarized in Table 3, and Fig. 8 gives a visualization of NPLs averaged over the LOS task (e.g., P1-3), the 1-NLOS (e.g., P4-6), and the 2^+-NLOS (e.g., P7-10) on testing maps P and T. Our PIRL policy generalizes well to these unseen tasks

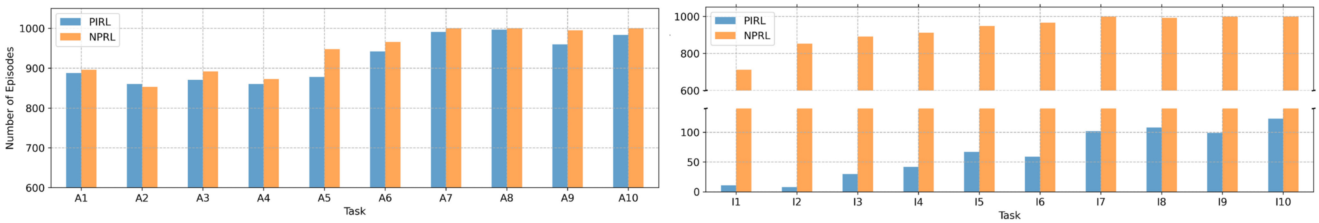


FIGURE 7. The number of episodes for ten tasks in map A and I. For each map, task number 1-3, 4-6, and 7-10 are tasks of LOS, 1-NLOS, and 2^+-NLOS case, respectively. Compared with NPRL, PIRL requires fewer and fewer episodes on each case as the training progresses.

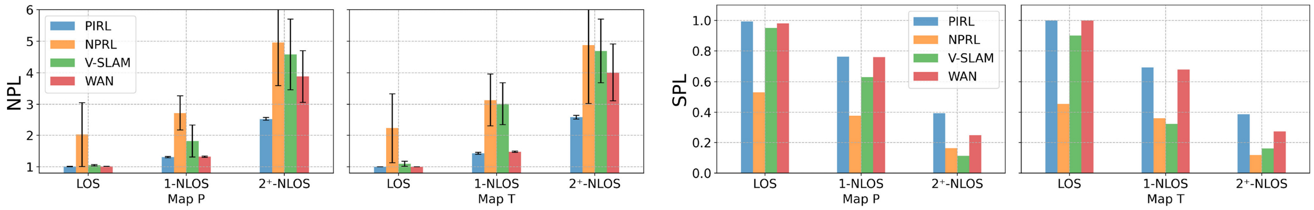


FIGURE 8. Average NPLs and the standard deviations (left) and SPLs (right) returned by navigation policies in the testing. Unlike NPL, SPL uses the inverse of the path length, and hence, the smaller the SPL one returns, the better it is. Since SPL assigns zeros to unsuccessful navigation instances, we do not report its standard deviation.

and achieves the smallest NPLs across all three scenarios. In addition to NPL, we also report in Fig. 8 the Success weighted by (normalized inverse) Path Length (SPL) and in Table 4 the quantitative values, which is customary in the literature [25]. Compared to NPRL, PIRL consistently achieves lower NPL and higher SPL across all link states, demonstrating superior performance. In LOS, PIRL reduces NPL by approximately (55% (P) and 51% (T)), comparable to V-SLAM, and achieve a higher SPL improvement (119% vs. 109% for P, 88% vs. 70% for T). In 1-NLOS, PIRL outperforms V-SLAM with a greater NPL reduction (58% vs. 42% for P, 47% vs. 11% for T) and a significantly higher SPL gain (111% vs. 75% for P, 83% vs. -15% for T). The advantage is most pronounced in 2-NLOS, where PIRL reduces NPL by 48% (P) and 40% (T) while achieving an SPL improvement over 230% (P) and 120% (T), far exceeding V-SLAM's performance. Overall, PIRL consistently delivers both lower NPL and significantly higher SPL, proving its robustness and efficiency across varying link conditions.

Moreover, to compare the running cost, in Table 5, we present the time consumption of PIRL and NPRL across 15 different maps, labeled A to O. The last column of the table reports the improvement of PIRL over NPRL in terms of GPU hours. The number of episodes represents the average across 10 different tasks per map. Since the training process follows a sequential order based on the map index, PIRL exhibits faster convergence in later maps, whereas NPRL maintains a consistent convergence time regardless of the map.

Furthermore, we compare the computational complexity of PIRL, NPRL, WAN, and V-SLAM in both training and testing. In training, all methods leverage the pre-trained neural SLAM module and local policy from prior work, which was trained on 10 million samples. Since WAN and V-SLAM are heuristic-based, their training complexity is limited to these shared components. However, PIRL and NPRL require additional training for their global policy, with PIRL demonstrating significantly lower sample complexity than NPRL as training progresses. In response to R3.1, we provide GPU hour comparisons in Table 1 (Table 5 in the manuscript) to quantify this difference. For testing, the primary computational cost comes from processing input data and goal generation. Unlike PIRL and NPRL, WAN and V-SLAM do not require a trained global policy, as they follow heuristic rules (e.g., AoA-based navigation, and RGB-based target localization). However, since PIRL's global policy network is lightweight (only five layers), the dominant computation overhead across all methods remains the neural SLAM and local policy, leading to comparable operational complexity in testing.

D. INTERPRETABLE NAVIGATION

We provide empirical evidence that the PIRL leverages the principles stated in Section III in the sense that the agent's behavior is well aligned with the physics principles. Specifically, we focus on 1) the reversibility principle: whether the agent follows the AoA, and 2) the gradient heuristic: whether the agent moves toward the high-SNR direction. Fig. 9(a), 9(b), and 9(c) present the histograms of 1000 sample angle outputs $\hat{\Omega}$ (i.e., moving directions) at a LOS, a 1-NLOS, and a 2^+-NLOS position, respectively. One can see from these figures that the PIRL obeys the physics principles enforced by C_{AoA} and C_{SNR}.

Furthermore, we attempt to interpret the PIRL model using explainable AI (XAI) methodologies. Since our physics-informed reward shaping bears distinct physics principles depending on the specific link states, we opt for post-hoc local XAI methods that provide explanations for specific

TABLE 3. A comparison of NPLs under 6 testing maps. PIRL achieves impressively efficient navigation in the challenging scenario 2^+-NLOS, compared with baselines.

	Map P			Map Q			Map R		
	LOS	1-NLOS	2^+-NLOS	LOS	1-NLOS	2^+-NLOS	LOS	1-NLOS	2^+-NLOS
PIRL	1.01 ± 0.01	1.31 ± 0.02	2.53 ± 0.05	1.01 ± 0.01	1.50 ± 0.04	2.61 ± 0.05	1.01 ± 0.00	1.23 ± 0.03	2.55 ± 0.06
NPRL	2.03 ± 1.02	2.72 ± 0.55	4.96 ± 1.37	2.12 ± 1.00	3.08 ± 0.68	5.00 ± 1.41	2.28 ± 1.03	2.49 ± 0.81	4.99 ± 1.20
V-SLAM	1.05 ± 0.02	1.82 ± 0.51	4.58 ± 1.12	1.11 ± 0.03	2.89 ± 0.73	4.89 ± 1.00	1.09 ± 0.03	1.68 ± 0.6	4.68 ± 1.01
WAN	1.02 ± 0.00	1.32 ± 0.02	3.88 ± 0.82	1.01 ± 0.01	1.63 ± 0.05	3.71 ± 0.71	1.01 ± 0.00	1.23 ± 0.02	3.97 ± 0.83
	Map S			Map T			Map U		
	LOS	1-NLOS	2^+-NLOS	LOS	1-NLOS	2^+-NLOS	LOS	1-NLOS	2^+-NLOS
PIRL	1.01 ± 0.01	1.23 ± 0.01	2.82 ± 0.04	1.00 ± 0.00	1.43 ± 0.03	2.59 ± 0.06	1.01 ± 0.01	1.73 ± 0.03	2.46 ± 0.05
NPRL	2.01 ± 0.99	2.81 ± 0.83	5.14 ± 1.21	2.23 ± 1.10	3.13 ± 0.83	4.88 ± 1.86	1.90 ± 0.89	3.25 ± 0.64	4.50 ± 1.05
V-SLAM	1.04 ± 0.03	1.99 ± 0.58	4.98 ± 1.00	1.10 ± 0.08	3.01 ± 0.67	4.69 ± 1.01	1.06 ± 0.04	3.19 ± 0.56	4.43 ± 1.00
WAN	1.01 ± 0.00	1.32 ± 0.03	3.78 ± 0.90	1.00 ± 0.00	1.48 ± 0.02	4.01 ± 0.90	1.01 ± 0.01	1.74 ± 0.04	3.63 ± 0.70

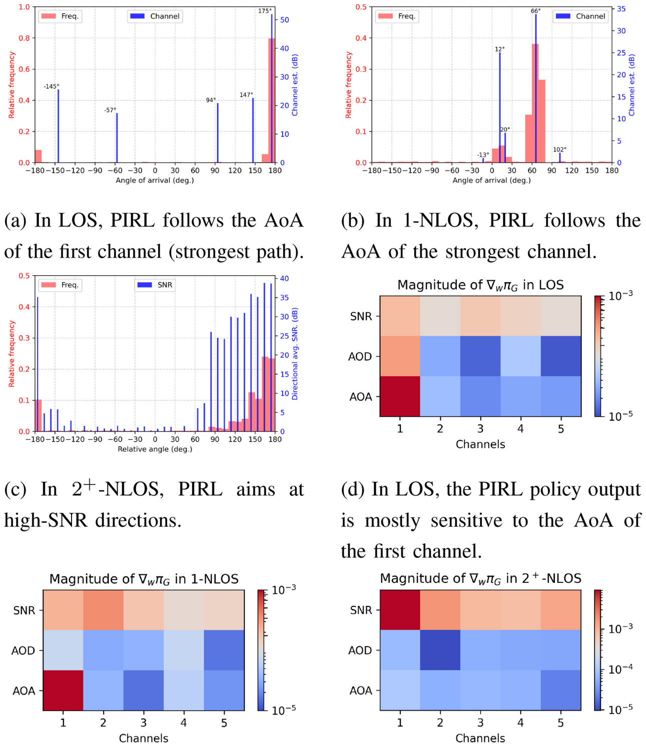
TABLE 4. Performance improvement (NPL: the lower, the better; SPL: the higher, the better) of policies in Map P and Map T compared to NPRL.

Map	Link State	NPL (%)			SPL (%)		
		PIRL	V-SLAM	WAN	PIRL	V-SLAM	WAN
P	LOS	**-54.71**	-52.91	-54.26	**118.81**	109.47	116.14
	1 NLOS	**-58.15**	-41.85	-57.83	111.32	74.64	**110.75**
	2 NLOS	**-48.16**	-6.15	-20.49	**232.30**	-3.97	108.74
T	LOS	-50.71	-45.81	**-50.74**	88.25	69.77	**88.42**
	1 NLOS	**-47.43**	10.66	-45.59	**83.37**	-15.20	79.32
	2 NLOS	**-40.15**	-5.89	-18.75	**120.45**	30.11	95.87

TABLE 5. GPU hours and episodes of PIRL and NPRL for maps A-O.

Label	Map Name	PIRL		NPRL		Improvement (GPU Hours%)
		GPU Hours	EPs (Average)	GPU Hours	EPs (Average)	
A	Bowlus	20.35	798	27.70	1000	26.56
B	Arkansaw	17.81	579	25.32	1000	29.67
C	Andrian	15.21	440	24.90	996	38.93
D	Anaheim	13.34	384	25.11	1000	46.86
E	Andover	5.21	208	23.50	940	77.83
F	Annawan	5.02	201	23.32	933	78.47
G	Azusa	4.67	187	22.11	884	78.87
H	Ballou	4.42	177	21.27	851	79.21
I	Capistrano	6.83	273	24.33	1000	71.93
J	Delton	4.87	195	22.58	903	78.42
K	Bolton	5.28	211	24.00	960	77.99
L	Goffs	3.91	156	21.37	855	81.70
M	Hainesburg	4.85	194	23.00	920	78.91
N	Kerrtown	4.72	189	23.10	924	79.56
O	Micanopy	3.93	157	23.17	927	83.03

instances instead of creating a interpretable surrogate model, e.g., linear and rule-based models [43], [44], [45], to explain the global navigation behaviors over the entire map. Among existing post-hoc local XAI approaches, model-agnostic methods, such as LIME [46], SHAP [47], and Ancors [48], enjoy border applicability since they apply to generic machine learning models. However, these methods often need to generate auditing data under stringent requirements and additional training and computation to examine the non-interpretable model and key features, which is challenging to fulfill in our digital twin environment. For example, for a given input, LIME requires perturbed data around the



(a) In LOS, PIRL follows the AoA of the first channel (strongest path).

(b) In 1-NLOS, PIRL follows the AoA of the strongest channel.

(c) In 2^+-NLOS, PIRL aims at high-SNR directions.

(d) In LOS, the PIRL policy output is mostly sensitive to the AoA of the first channel.

(e) In 1-NLOS, the PIRL policy output is mostly sensitive to the AoA of the first channel.

(f) In 2^+-NLOS, the PIRL policy output is mostly sensitive to the SNR of the first channel.

FIGURE 9. The interpretability experiments on the reversibility principle (Section III-B(b)) and the SNR heuristic (Section III-B(c)). (a)(b)(d)(e) confirms that the PIRL agent traverses reversely along the angle of arrival. (c)(f) indicates that SNR is instrumental when navigating in high NLOS areas.

neighborhood of the input instance subject to a proximity requirement. However, due to diffractions and reflections in mmWave propagation, a slight offset to the transmitter location can create drastically different wireless fields.

Therefore, we resort to gradient-based attribution methods specific to neural networks [49] due to their lightweight operation without additional data generation. Following the gradient explanation technique in [50], we compute the global policy's partial derivative with respect to each feature (e.g., link state, AOA, and SNR) to inspect the importance of each feature when deciding the policy network output. Denote by $w(p)$ the wireless information input at position p defined in (5). For a given testing map, we randomly sample 10 locations in LOS, 1-NLOS, and 2^+-NLOS areas, respectively. We evaluate the mean value of $\nabla_w \pi_G(w)$ over these 10 inputs in the three areas. We repeat the same procedure for all testing maps and plot the magnitude of each entry of average $\nabla_w \pi_G$ in Fig. 9. Similar to the saliency map in [50], the heat map in Fig. 9 indicates the PIRL policy sensitivity to each feature: the darker red blocks suggest more important features.

Ideally, the policy network should be mostly sensitive to the first-channel AoA in LOS and 1-NLOS, since our reward-shaping encourages the agent to travel reversely along the AoA direction. In higher-order NLOS, the policy should be sensitive to SNR as we employ SNR gradient directions as a navigation heuristic. Our intuition is confirmed by the heat map in Fig. 9, and the PIRL model indeed leverages the wireless information as instructed by the principles, which points to another advantage of incorporating the physics information into RL: the physics-based reward components lead to interpretable navigation.

TABLE 6. Ablation studies on the SNR and link state terms. The metric is NPL averaged over all testing tasks.

	LOS	1-NLOS	2^+-NLOS
WAN	1.01 ± 0.01	1.45 ± 0.03	3.83 ± 0.81
PIRL	1.01 ± 0.01	1.41 ± 0.03	2.60 ± 0.05
SNR Ablation	1.02 ± 0.02	1.46 ± 0.04	4.62 ± 1.15
Link State Ablation	1.02 ± 0.02	1.47 ± 0.05	3.90 ± 1.02

E. ABLATION STUDY

Recall that PIRL differs from WAN in its use of link state and SNR information. We conduct ablation studies regarding C_{LS} and C_{SNR}, for which we report the NPL results. For the SNR ablation, we replace C_{SNR} with the relative distance cost in 2^+-NLOS to see whether the SNR heuristic helps the agent navigate efficiently in such a scenario. As one can see from Table 6, the answer to the question is affirmative, as the SNR ablation returns significantly higher NPLs in 2^+-NLOS. We also replace C_{LS} with a constant number to investigate whether the link-state penalty discourages the agent from entering the higher-order NLOS area from the lower-order NLOS. The third row in Table 6 indicates that without C_{LS}, the agent frequently revisits the high-order NLOS areas in testing, which yields higher NPLs in NLOS scenarios. In summary, C_{SNR} contributes to PIRL's success in 2^+-NLOS, and C_{LS} helps stabilize the navigation (less variance).

VI. RELATED WORKS

Due to necessary exploration [51], RL typically suffers from poor sample efficiency [52], especially when facing sophisticated tasks such as WIN. PIRL emerges as a promising remedy through integrating data and mathematical physics models. Even though no census has been established on the exact definition, PIRL amounts to introducing appropriate **observational**, **inductive**, and **learning** biases that can speed up the learning process [33]. Introducing observation biases bears the same spirit of data augmentation, where the underlying physics law is embedded into the training data. For example, [53] trains an RL model using a combination of historical data and synthetic data generated from a traffic flow model for ramp metering. By incorporating into the RL training a predicted conflict zone visualized by a physics-based prediction algorithm, [54] develops a physics-informed aircraft conflict resolution strategy. Inductive biases correspond to interventions to the RL model architecture [55], and the resulting outputs are guaranteed to implicitly satisfy a set of given physics laws. One example is the physics-informed model-based RL considered in [56], where the physics constraints are imposed on the model learning. Our proposed PIRL method belongs to the third class: introducing learning biases. By selecting appropriate loss functions and constraints to modulate the training, this class of PIRL favors convergence to solutions adhering to the underlying physics. The advantage of our reward-shaping approach lies in its lightweight design: without generating synthetic data (first class) or modifying the RL policy network structure (second class), the physics-informed reward-shaping features seamless compatibility with existing RL training paradigms. Moreover, what distinguishes our PIRL from other straightforward reward-shaping methods [57] is that our proposed PIRL features a two-timescale operation to accommodate the multi-modal information in WIN, where the physics-informed learning biases are injected into the slow-scale global policy learning.

VII. CONCLUSION AND FUTURE WORK

This work develops a physics-informed RL (PIRL) for wireless indoor navigation and is evaluated by the proposed digital twin. By incorporating physics prior to reward shaping, PIRL introduces learning biases to modulate policy learning, favoring those adhering to physics principles. As these principles are invariant across training/testing tasks, PIRL alleviates catastrophic forgetting in training and displays zero-shot generalization in testing.

This paper opens up several directions for future work. First, we point out that one failure case our PIRL agent may encounter is when it enters into a power-outage region (e.g., a balcony) where no or negligible wireless signals are available due to complex spatial geometry and high-order reflections. One future extension to address this issue is to synthesize vision and wireless information. Once the agent is trapped in a power outage case, it switches to the exploration mode and aims to maximize the covered area (e.g., areas scanned by the RGB camera). By doing so, the agent is motivated to move out of the region and explore another area. Once jumping out of the power-outage area, the wireless-based PIRL is again activated to take over the long-term goal generation. Through such an event-triggered control framework, the agent can navigate across the entire indoor environment regardless of the wireless signal availability.

Another promising extension is to consider the height factor in navigation. Even though our WIN is a 2-dimensional navigation task, it is likely in reality for the agent to observe a line-of-sight (LOS) link state but with an obstacle between the agent and the target. Our proposed PIRL can still cope with such a scenario. The intuition is that if the agent observes an LOS link state but also an obstacle through the RGB camera, it should start to follow the AoA of the second strongest channel, which corresponds to the propagation path within the xy-pane that experiences the least reflections.

Bearing such intuition, when designing the reward-shaping, we can incorporate a symbolic trigger (i.e., if-else rule) into the C_{AoA} in the loss function in eq. (10). In LOS, if no obstacle is observed, then we use the current C_{AoA} design; otherwise, the trigger is active, and we replace the AoA direction in C_{AoA} with the corresponding entry from the second strongest channel.

PIRL's generic design makes it inherently scalable to larger environments and more complex wireless setups, such as FR3/upper mid-band scenarios. In larger spaces, such as expanded rooms or multi-room environments, PIRL can adapt by efficiently managing interference, beamforming, and resource allocation, ensuring robust communication and accurate navigation even with an increased number of antennas and robots. Furthermore, FR3 scenarios introduce additional complexity due to the coexistence of short- and long-range propagation characteristics. PIRL's adaptable framework allows it to learn and generalize across diverse multipath profiles and dynamic link conditions, making it well-suited for real-world deployment. To further enhance scalability, PIRL can be extended to heterogeneous wireless setups, incorporating varying antenna configurations, dense user environments, and rapidly changing network conditions, ensuring flexibility and reliability in complex, large-scale deployments.

REFERENCES

[1] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proc. IEEE*, vol. 102, no. 3, pp. 366–385, Mar. 2014.

[2] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology*. Cambridge, MA, USA: Academic Press, 2020.

[3] A. Shahmansoori, G. E. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, "5G position and orientation estimation through millimeter wave MIMO," in *Proc. IEEE GLOBECOM Workshops*, San Diego, CA, USA, 2015, pp. 1–6.

[4] F. Guidi, A. Guerra, and D. Dardari, "Millimeter-wave massive arrays for indoor SLAM," in *Proc. IEEE ICC Workshops*, 2014, pp. 114–120.

[5] R. Ayyalasomayajula et al., "Deep learning based wireless localization for indoor navigation," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2020, pp. 1–14.

[6] M. Yin et al., "Millimeter wave wireless assisted robot navigation with link state classification," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 493–507, 2022.

[7] T. S. Rappaport, R. W. Heath Jr., R. C. Daniels, and J. N. Murdock, *Millimeter Wave Wireless Communications*. London, U.K.: Pearson Educ., 2015.

[8] L. U. Khan, Z. Han, W. Saad, E. Hossain, M. Guizani, and C. S. Hong, "Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 4, pp. 2230–2254, 4th Quart., 2022.

[9] T. Kobayashi et al., "Personal and intuitive indoor navigation system based on digital twin," in *Proc. IEEE ICCE*, Berlin, Germany, 2021, pp. 1–6.

[10] M. Savva et al., "Habitat: A platform for embodied AI research," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 9339–9347.

[11] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, Feb. 2017.

[12] M. G. Kapteyn, D. J. Knezevic, D. Huynh, M. Tran, and K. E. Willcox, "Data-driven physics-based digital twins via a library of component-based reduced-order models," *Int. J. Numer. Methods Eng.*, vol. 123, no. 13, pp. 2986–3003, 2022.

[13] M. G. Kapteyn and K. E. Willcox, "From physics-based models to predictive digital twins via interpretable machine learning," 2020, *arXiv:2004.11356*.

[14] M. Yin, T. Li, H. Lei, Y. Hu, S. Rangan, and Q. Zhu, "Zero-shot wireless indoor navigation through physics-informed reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Yokohama, Japan, 2024, pp. 5111–5118.

[15] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9068–9079.

[16] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 190–198.

[17] "Remcom." Mar. 2022. [Online]. Available: https://www.remcom.com

[18] W. Khawaja, O. Ozdemir, and I. Guvenc, "UAV air-to-ground channel characterization for mmWave systems," in *Proc. 86th IEEE Veh. Technol. Conf.*, 2017, pp. 1–5.

[19] Y. Hu, M. Yin, W. Xia, S. Rangan, and M. Mezzavilla, "Multi-frequency channel modeling for millimeter wave and THz wireless communication via generative adversarial networks," Dec. 2022, *arXiv:2212.11858*.

[20] J. Thrane, D. Zibar, and H. L. Christiansen, "Model-aided deep learning method for path loss prediction in mobile communication systems at 2.6 GHz," *IEEE Access*, vol. 8, pp. 7925–7936, 2020.

[21] J. Song, J. Choi, and D. J. Love, "Codebook design for hybrid beamforming in millimeter wave systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2015, pp. 1298–1303.

[22] W. Xia, V. Semkin, M. Mezzavilla, G. Loianno, and S. Rangan, "Multi-array designs for mmWave and sub-THz communication to UAVs," in *Proc. 21st IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2020, pp. 1–5.

[23] F. Wen, N. Garcia, J. Kulmer, K. Witrisal, and H. Wymeersch, "Tensor decomposition based beamspace ESPRIT for millimeter wave MIMO channel estimation," in *Proc. IEEE Conf. Global Commun. (GLOBECOM)*, 2018, pp. 1–7.

[24] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave MIMO-OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1524–1538, Jul. 2017.

[25] P. Anderson et al., "On evaluation of embodied navigation agents," Jul. 2018, *arXiv:1807.06757*.

[26] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural SLAM," Apr. 2020, *arXiv:2004.05155*.

[27] H. Kurniawati, "Partially observable Markov decision processes and robotics," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, no. 1, pp. 1–25, Jan. 2022.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[29] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1928–1937.

[30] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," in *Proc. Assoc. Adv. Artif. Intell. Fall Symp. Series*, 2015, pp. 1–9.

[31] T. Li and Q. Zhu, "On convergence rate of adaptive multiscale value function approximation for reinforcement learning," in *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, 2019, pp. 1–6.

[32] T. Li, Y. Zhao, and Q. Zhu, "The role of information structures in game-theoretic multi-agent learning," *Annu. Rev. Control*, vol. 53, pp. 296–314, May 2022.

[33] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, no. 6, pp. 422–440, May 2021.

[34] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–35, 2018.

[35] J. Bannon, B. Windsor, W. Song, and T. Li, "Causality and batch reinforcement learning: Complementary approaches to planning in unknown domains," Jun. 2020, *arXiv:2006.02579*.

[36] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," Nov. 2020, *arXiv:2005.01643*.

[37] T. Li, G. Peng, and Q. Zhu, "Blackwell online learning for Markov decision processes," in *Proc. 55th Annu. Conf. Inf. Sci. Syst. (CISS)*, 2021, pp. 1–6.

[38] A. Y. Ng, D. Harada, and S. J. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. 16th Int. Conf. Mach. Learn. (ICML)*, 1999, pp. 278–287.

[39] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci.*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996.

[40] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1–12.

[42] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 1–7.

[43] B. Ustun and C. Rudin, "Supersparse linear integer models for optimized medical scoring systems," *Mach. Learn.*, vol. 102, no. 3, pp. 349–391, 2016.

[44] D. Wei, S. Dash, T. Gao, and O. Gunluk, "Generalized linear rule models," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6687–6696. [Online]. Available: https://proceedings.mlr.press/v97/wei19a.html

[45] Y. Ge, T. Li, and Q. Zhu, "Scenario-agnostic zero-trust defense with explainable threshold policy: A meta-learning approach," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–6.

[46] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 1135–1144.

[47] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4768–4777. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[48] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, Apr. 2018, pp. 1527–1535. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11491

[49] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable AI: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, 2020.

[50] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–8.

[51] T. Li, G. Peng, Q. Zhu, and T. Başar, "The confluence of networks, games, and learning a game-theoretic framework for multiagent decision making over networks," *IEEE Control Syst.*, vol. 42, no. 4, pp. 35–67, Aug. 2022.

[52] S. Mohanty et al., "Measuring sample efficiency and generalization in reinforcement learning benchmarks: NeurIPS 2020 Procgen benchmark," in *Proc. NeurIPS Compet. Demonstr. Track*, 2021, pp. 361–395. [Online]. Available: https://proceedings.mlr.press/v133/mohanty21a.html

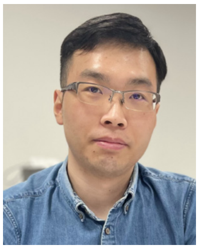
[53] Y. Han, M. Wang, L. Li, C. Roncoli, J. Gao, and P. Liu, "A physics-informed reinforcement learning-based strategy for local and coordinated ramp metering," *Transp. Res. Part C, Emerg. Technol.*, vol. 137, Apr. 2022, Art. no. 103584.

[54] P. Zhao and Y. Liu, "Physics informed deep reinforcement learning for aircraft conflict resolution," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8288–8301, Jul. 2022.

[55] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: A review," *Acta Mechanica Sinica*, vol. 37, no. 12, pp. 1727–1738, 2021. [Online]. Available: https://doi.org/10.1007/s10409-021-01148-1

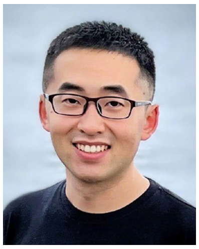
[56] X.-Y. Liu and J.-X. Wang, "Physics-informed Dyna-style model-based deep reinforcement learning for dynamic control," *Proc. Roy. Soc. A*, vol. 477, no. 2255, 2021, Art. no. 20210618.

[57] M. I. Radaideh et al., "Physics-informed reinforcement learning optimization of nuclear assembly design," *Nucl. Eng. Design*, vol. 372, Feb. 2021, Art. no. 110966.



TAO LI (Member, IEEE) received the B.S. degree in mathematics from Xiamen University, Fujian, China, in 2018. He is currently pursuing the Ph.D. degree in electrical engineering with New York University (NYU), NY, USA. Spanning across game theory, online optimization, and learning theory, his research advances novel methodologies and frameworks on predictive learning, non-equilibrium analysis, and meta-learning control for resilient cyber–physical system design, defense, and management. His continued enthusiasm and efforts have won him the NSERC Mitacs-Globalink Research Award and the NYU Dante Youla Award for research excellence.

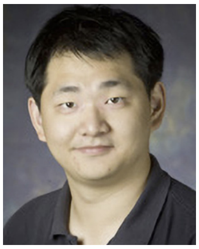
HAOZHE LEI received the B.S. degree in electrical engineering from China Agricultural University, Beijing, China, in 2019, and the M.S. degree in computer engineering from New York University, New York, where he is currently pursuing the Ph.D. degree in electrical engineering with NYU WIRELESS, under the supervision of Dr. S. Rangan. His research interests include wireless indoor navigation, outdoor autonomous navigation, wireless radar sensing, and transportation mobility management.



HAO GUO (Member, IEEE) received the B.Sc. degree in electrical engineering from Nankai University, Tianjin, China, in 2015, and the M.Sc. and Ph.D. degrees in communication engineering from the Chalmers University of Technology, Gothenburg, Sweden, in 2017 and 2022, respectively, where he is currently a Postdoctoral Researcher with the Department of Electrical Engineering. He is also a Postdoctoral Visiting Scholar with the Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY, USA. As an active contributor to the research community, he has reviewed over 300 papers for top journals and conferences in the field. His research interests include integrated sensing and communication, robotic navigation, and digital twinning. He has been awarded the Swedish Research Council International Postdoc Grant, the Ericsson Research Grant, and the Chalmers ICT SEED and Collaborative Grants. He also received the Pedagogical Prize 2020 from the Department of Electrical Engineering at Chalmers. In recognition of his outstanding service, he was named an Exemplary Reviewer of IEEE WIRELESS COMMUNICATIONS LETTERS in 2020, 2021, 2022, and 2024, as well as for IEEE COMMUNICATIONS LETTERS in 2023. He has served as a Guest Editor for *IEEE Vehicular Technology Magazine* and as a Technical Program Committee member for leading conferences, including IEEE GLOBECOM and IEEE ICC.

MINGSHENG YIN received the B.Eng. degree from the Beijing University of Posts and Telecommunications, the B.Sc. degree from the Queen Mary University of London in 2018, and the M.Sc. degree in electrical and computer engineering and the Ph.D. degree under the supervision of Dr. S. Rangan from New York University in 2020 and 2024, respectively. He has conducted research with Nokia Bell Labs, AT&T Labs, and Google. His research interests include localization, mmWave and THz communication, UAV wireless communication, and high-frequency ray tracing.

YAQI HU received the B.S. degree in telecommunication engineering from the Beijing University of Posts and Telecommunications, and the M.S. degree in electrical and computer engineering and the Ph.D. degree under the supervision of Prof. S. Rangan from New York University in 2024. She has conducted research with AT&T Labs and Microsoft. Her research interests include machine learning and wireless channel modeling.



QUANYAN ZHU (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electrical engineering from McGill University in 2006, the M.A.Sc. degree from the University of Toronto in 2008, and the Ph.D. degree from the University of Illinois at Urbana-Champaign in 2013. After stints with Princeton University, he is currently an Associate Professor with the Department of Electrical and Computer Engineering, New York University (NYU), where he is an Affiliated Faculty Member with the Center for Urban Science and Progress. He is the coauthor of two recent books published by Springer: *Cyber-Security in Critical Infrastructures: A Game-Theoretic Approach* (with S. Rass, S. Schauer, and S. König) and *A Game- and Decision-Theoretic Approach to Resilient Interdependent Network Analysis and Design* (with J. Chen). His current research interests include game theory, machine learning, cyber deception, network optimization and control, smart cities, the Internet of Things, and cyber–physical systems. He was a recipient of many awards, including the NSF CAREER Award, the NYU Goddard Junior Faculty Fellowship, the NSERC Postdoctoral Fellowship, the NSERC Canada Graduate Scholarship, and the Mavis Future Faculty Fellowships. He spearheaded and chaired the INFOCOM Workshop on Communications and Control on Smart Energy Systems, the Midwest Workshop on Control and Game Theory, and the ICRA workshop on Security and Privacy of Robotics. He served as the General Chair or the TPC Chair for the Seventh and the 11th Conference on Decision and Game Theory for Security in 2016 and 2020, the Ninth International Conference on Network Games, Control and Optimization in 2018, the Fifth International Conference on Artificial Intelligence and Security in 2019, and the 2020 IEEE Workshop on Information Forensics and Security. He also spearheaded the IEEE Control System Society Technical Committee on Security, Privacy, and Resilience in 2020.

SUNDEEP RANGAN (Fellow, IEEE) received the B.A.Sc. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of California at Berkeley, Berkeley, CA, USA. He has held Postdoctoral appointments with the University of Michigan, Ann Arbor, Ann Arbor, MI, USA, and Bell Labs. In 2000, he co-founded (with four others) Flarion Technologies, a spin-off of Bell Labs, that developed Flash OFDM, the first cellular OFDM data system and precursor to 4G cellular systems, including LTE and WiMAX. In 2006, Flarion was acquired by Qualcomm Technologies. He was a Senior Director of Engineering with Qualcomm involved in OFDM infrastructure products. He joined New York University Tandon (formerly NYU Polytechnic), Brooklyn, NY, USA, in 2010, where he is currently a Professor of Electrical and Computer Engineering. He is the Associate Director of NYU WIRELESS, Brooklyn, NY, USA, an industry-academic research center on next-generation wireless systems.