Reinforcement Learning with Physics-Informed Symbolic Program Priors for Zero-Shot Wireless Indoor Navigation

Tao Li[1,†], **Haozhe Lei**[1], **Mingsheng Yin**[1], **Yaqi Hu**[1]
{taoli, hl4155, my1778, yh2829}@nyu.edu

[1]**Department of Electrical and Computer Engineering, New York University**

[†] Corresponding author

Abstract

When using reinforcement learning (RL) to tackle physical control tasks, inductive biases that encode physics priors can help improve sample efficiency during training and enhance generalization in testing. However, the current practice of incorporating these helpful physics-informed inductive biases inevitably runs into significant manual labor and domain expertise, making them prohibitive for general users. This work explores a symbolic approach to distill physics-informed inductive biases into RL agents, where the physics priors are expressed in a domain-specific language (DSL) that is human-readable and naturally explainable. Yet, the DSL priors do not translate directly into an implementable policy due to partial and noisy observations and additional physical constraints in navigation tasks. To address this gap, we develop a physics-informed program-guided RL (PiPRL) framework with applications to indoor navigation. PiPRL adopts a hierarchical and modularized neuro-symbolic integration, where a meta symbolic program receives semantically meaningful features from a neural perception module, which form the bases for symbolic programming that encodes physics priors and guides the RL process of a low-level neural controller. Extensive experiments demonstrate that PiPRL consistently outperforms purely symbolic or neural policies and reduces training time by over 26% with the help of the program-based inductive biases.

1 Introduction

Reinforcement learning (RL), while widely explored in a variety of engineering contexts (Zhao et al., 2020; Li et al., 2022a; Nguyen & Reddi, 2023), typically suffers from poor sample efficiency in training and limited generalization in testing (Mohanty et al., 2021), especially when facing sophisticated control tasks with high-dimensional sensor inputs and complex system dynamics (Lesort et al., 2018; Lauri et al., 2023; Li et al., 2024a; Hammar et al., 2025). Fortunately, even a small amount of prior knowledge encoded through inductive biases about the task and environment, often seemingly obvious, can dramatically improve learning (Baxter, 2000; Hessel et al., 2019).

While a universally agreed-upon taxonomy of inductive biases in RL has yet to emerge, some common choices are as follows. 1) *Representation bias*, which bears the same spirit of data augmentation, embeds underlying prior knowledge into the training data (Han et al., 2022; Zhao & Liu, 2022; Li et al., 2025a); 2) *Objective bias* modulates RL processes by shifting the objective function through, for instance, reward shaping (Ng et al., 1999; Gupta et al., 2022), regularization (Geist et al., 2019; Pan et al., 2024), intrinsic motivation and curiosity (Kulkarni et al., 2016; Jaques et al., 2019); 3) *Architectural bias*, which is embedded in the design of the neural networks, leads to a physics-informed architecture (Raissi et al., 2019; Cuomo et al., 2022) when handling data and environments

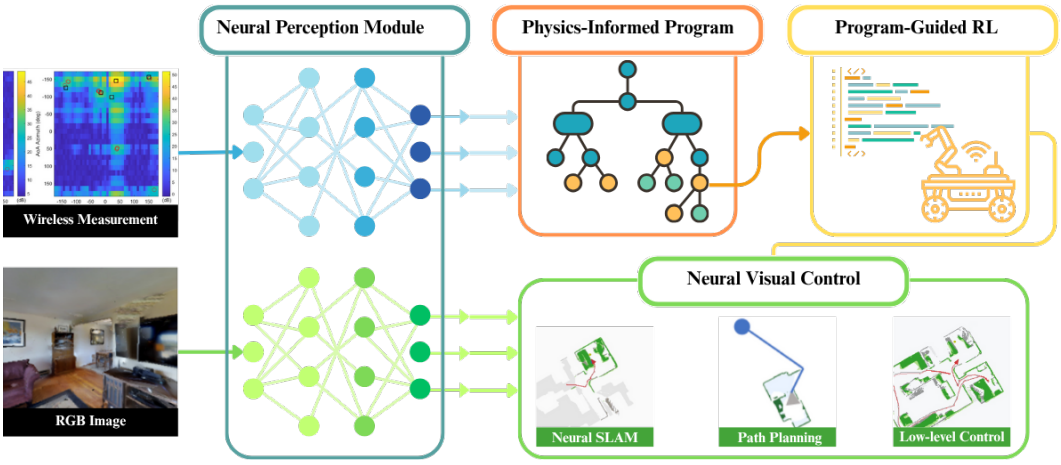

Figure 1: The workflow of the physics-informed program-guided RL (PiPRL). The core of PiPRL is the physics-informed symbolic program, which coordinates the other modules in the wireless indoor navigation tasks.

with physics constraints; and 4) *Algorithmic bias* points to the setup of learning algorithms, including hyperparameters (Hessel et al., 2019), initialization (Fallah et al., 2021; Pan et al., 2023a; Li et al., 2024b), and regularization (Mei et al., 2020; Agarwal et al., 2021; Pan et al., 2023b).

Although these inductive biases yield encouraging successes in both theoretical and applied domains, their designs and practical implementations, despite their distinct characteristics, all require intensive manual labor. This problem is even more exacerbated in robotic control tasks, where high-dimensional, noisy sensor readings, sophisticated system dynamics and environments, and physical constraints make it complicated and sometimes painful to introduce inductive biases to learning-based control. One example of such is the wireless indoor navigation task reviewed in Section 2.

To facilitate the distillation of inductive biases in RL for real-world robotic tasks, this work proposes to encode physics priors into symbolic programs written in a domain-specific language (DSL) (Rodriguez-Sanchez et al., 2023) as the inductive bias, which are human-readable and naturally explainable and can be easily generated from natural languages without handcrafting. Given the close structural alignment between natural language expressions and symbolic programs, our intuition is

when physics priors are easily accessible to humans in natural language,
so should it be to agents through symbolic programming languages.

Despite its ease of use and inherent explainability, the symbolic program, grounded in abstract physics knowledge, can only provide agents with high-level abstract navigation strategies, such as "moving along the wireless signal path", without directly engaging with the low-level sensory signals or motor control needed for real-world robotic operation, for which RL becomes essential.

To get the best of two worlds, we develop a Physics-informed Program-guided Reinforcement Learning (PiPRL) framework to achieve zero-shot generalization in wireless indoor navigation tasks (see Section 3). As shown in Figure 1, our neuro-symbolic RL framework consists of three components: 1) a pretrained **neural perception module** for processing raw sensor readings, 2) a **symbolic program module** that expresses physics priors in DSL, which i) directly maps the processed wireless signals to high-level navigation strategies based on physics principles of mmWave propagation and ii) modulate an RL process to search for navigation strategies when the physics priors do not prescribe a executable policy but rather desiderata or constraints that characterize effective policies, and 3) a pretrained low-level RL motion control module translating high-level navigation strategies into control commands.

As an attempt at the **hierarchical and modularized integration** of neuro policies and symbolic programs in a robotic control task, this work's **contributions** are as follows. 1) We develop PiPRL to encode physics priors through a symbolic DSL program that serves as an inductive bias for RL processes. We position our work by reviewing the literature in Appendix A. 2) Besides directly prescribing symbolic policies, PiPRL also employs programs to guide RL policy learning by substitut-

ing physics-compliant actions with their unbiased reward estimates. 3) We empirically demonstrate that PiPRL outperforms purely symbolic and RL policies consistently in terms of sample efficiency and generalizability in Gibson testbeds (Xia et al., 2018).

2 Related Works on Indoor Navigation and Limitations

The mobile robot agent aims to utilize the fine-grained temporal and angular resolution of mmWave signal paths to achieve wireless-based positioning and localization (Guidi et al., 2014), which offers a unique advantage of penetrating beyond line of sight over vision-based methods (Savva et al., 2019; Chaplot et al., 2020). The core challenge of WIN lies in its call for zero-shot generalization, where agents need to complete navigation tasks without fine-tuning when deployed in unseen testing environments.

According to radio frequency propagation, a simple physics-based heuristic of following the mmWave's angle of arrival (AoA) has proven effective and generalizable in structured testbeds (Yin et al., 2022). However, it fails to handle complex environments where mmWave signals propagate along multiple paths through reflections and diffractions, rendering the observations of signal paths highly noisy and inexact (Shahmansoori et al., 2018). Towards this end, Sutera et al. (2020) hand-crafted the state representation of wireless sensor measurements (*representation bias*) to indicate the nearest obstacles to the RL agent; Ayyalasomayajula et al. (2020) developed a hierarchical-decoder network (*architectural bias*) for wireless localization; and most recently, Yin et al. (2024) and Li et al. (2025b) explored an end-to-end deep RL approach with carefully designed reward functions (*objective bias*). While empirically successful, these existing attempts are ad hoc by nature and lead to black-box policies requiring additional efforts for interpretability (Li et al., 2025b).

3 Task Setup and Preliminary

Wireless Indoor Navigation. Consider a wireless indoor navigation (WIN) task setup as studied in (Yin et al., 2022), where a stationary target is positioned at an unknown location in an indoor environment. The target is equipped with an mmWave transmitter that broadcasts wireless signals at regular intervals. Equipped with an mmWave receiver, an RGB camera, and motion sensors, a mobile robot agent aims to navigate to the target in minimal time. In contrast to the PointGoal task (Anderson et al., 2018), WIN does not provide the agent with the target's relative coordinates.

Like most robotic control tasks (Li et al., 2022b; Lauri et al., 2023), one can formulate WIN as a partially observable Markov decision process (POMDP). The state corresponds to the agent's actual pose $p = (x, y, \varphi)$, where x, y denote the xy-coordinate of the agent measured in meters, and φ represents the orientation of the agent in degrees (measured counter-clockwise from the x-axis). The agent aims to locate and navigate to the target (the wireless transmitter) denoted by (x^*, y^*). Following (Chaplot et al., 2020), the agent employs three navigation actions: $\mathcal{A} = \{a_F, a_L, a_R\}$, where $a_F = (d, 0, 0)$ denotes the moving-forward command with a travel distance $d = 25\,\mathrm{cm}$, and $a_L = (0, 0, -10^\circ)$ and $a_R = (0, 0, 10^\circ)$ denote the turn-left and -right by 10° commands.

Mobile robots are typically equipped with motion sensors that estimate the pose at each time step $t \in \mathbb{N}_+$, which we denote by $\bar{p}_t = (\bar{x}_t, \bar{y}_t, \bar{\varphi}_t)$. In addition to the sensor readings, the robot also receives a 3-channel RGB camera image $v_t \in \mathbb{R}^{3\times L_1\times L_2}$ (L_1 and L_2 are the frame width and height) and mmWave measurement (after signal processing) $w_t \in \mathbb{R}^{N_1\times N_2\times N_3}$, a three-dimensional tensor whose entries pertain to the spatial-temporal correlation among antenna arrays' angles, received power, and delays (Yin et al., 2022). The tuple $o_t = (\bar{p}_t, v_t, w_t)$ constitutes the partial observation of the agent. Denote by $h_t = \{(o_k, a_k)_{k=1}^t, o_t\}$ the historical observations up to time t. The agent aims to find a (stochastic) policy that maps the history to a distribution over actions, $\pi(h_t) \in \Delta(\mathcal{A})$, which generates a sequence of actions $\{a_t\}_{t=1}^H$ finding the target withing the horizon H.

Deep Reinforcement Learning. Using the POMDP language, let $c_t = \|x_t - x^*\|^2 + \|y_t - y^*\|^2$ be the Euclidean distance (or any distance metric, e.g., geodesic distance) between the current pose

```
Factor pose := (x̄, ȳ, φ̄)
Factor vision := v
Factor wireless := w
```

```
Feature pose estimate := (x̂, ŷ, φ̂)
Feature path estimate := (g, Ω^rx, Ω^tx)
Feature link state estimate := ℓ̂
```

```
Policy random move:
    Execute a_F w/ P(0.33)
    or Execute a_L w/ P(0.33)
    or Execute a_R w/ P(0.33)
```

```
Effect move forward:
    if a_F executed:
        pose = pose + a_F
    Return pose
```

Figure 2: Examples of **Factor**, **Feature**, **Policy**, and **Effect** in RLang. **Factor** and **Feature** describe the agent's observations and related features. **Policy** and **Effect** represent the navigation policy and its consequence.

and the target position, the optimal policy corresponds to the minimizer of the expected cumulative cost, referred to as the value function, $\pi^* \in \arg\min J(\pi) \triangleq \mathbb{E}_\pi[\sum_{t=1}^{T} c_t | p_1]$. When facing high-dimensional state inputs in continuous spaces, one can leverage the representation learning power of deep neural networks and consider temporal difference learning and Q-learning with value function approximators (Tsitsiklis & Roy, 1997; Li & Zhu, 2019; Mnih et al., 2015; Liu et al., 2022).

Alternatively, parameterizing the policy through a deep neural network with parameters $\theta \in \mathbb{R}^n$, one can consider directly optimizing the parameterized policy π_θ through the policy gradient method (Sutton et al., 2000; Konda & Borkar, 1999; Mnih et al., 2016; Fujimoto et al., 2018; Liu et al., 2023) and policy gradient-based policy optimization methods (Schulman et al., 2015; 2017; Agarwal et al., 2021; Pan et al., 2025). This work instantiates the proposed framework using proximal policy optimization (PPO) by Schulman et al. (2017) to stay consistent with PPO-based RL baselines in the experiments. Note that PiPRL, as a modularized neuro-symbolic RL framework, is fundamentally algorithm-agnostic: the symbolic program component operates independently of the underlying RL optimizer. Any policy-based or value-based RL algorithm can be seamlessly integrated into PiPRL.

Domain-Specific Language. A domain-specific language (DSL) is a formal language tailored to a specific domain with precisely yet narrowly defined syntax and semantics; for instance, TEX by Knuth (1986) is a DSL for typesetting. Compared with general-purpose languages (GPL), such as Python (van Rossum, 1991), DSLs trade general expressivity for ease of use and customization in specific domains, which is exemplified by the Planning Domain Description Languages (PDDL) (Ghallab et al., 1998), an early attempt to standardize AI planning languages. This work considers RLang as the DSL, which consists of a set of declarations and element types, with each one corresponding to one or more components of POMDP and associated policy classes, such as options (Sutton et al., 1999); and its formal semantics and grammar are in (Rodriguez-Sanchez et al., 2023, Appendix A). We briefly review the main RLang element types relevant to our navigation tasks.

RLang uses **Factor** to specify the agent's states (MDP) and partial observations (POMDP). In the WIN task, the partial observation includes pose estimate, vision, and wireless sensor measurements: $o = (\bar{p}, v, w)$, for which RLang defines three factors, as shown in Figure 2. Closely related to **Factor** is **Feature**, depicting a function of states, which can be helpful when employing additional information processing, feature extraction, and representation learning to reduce the raw input dimensionality. As will be made clear in Section 4, we use deep learning (Chaplot et al., 2020) and low-rank tensor decomposition (Zhou et al., 2017) to extract features from motion and wireless sensor measurements. Besides the two basic elements, two core declarations in RLang are **Policy** for executing policies and **Effect** for describing rewards and transitions, i.e., subsequent factors. Figure 2 provides an instance of the random navigation policy and the movement effect when executing the move-forward action a_F. The ensuing section articulates the symbolic program using the two declarations built on physics priors to instruct the RL agent.

4 Physics-Informed Program-Guided Reinforcement Learning

Framework Overview. A PiPRL agent in WIN first relies on a neural perception module for simultaneous localization and mapping (SLAM), mmWave propagation path estimation. This neural-network module transforms the high-dimensional sensor measurements into the agent's understanding of the surrounding environment, extracting semantically meaningful features, including signal strength, arrival angles, and obstacle locations.

These features form the basis for symbolic programming that prescribes high-level navigation strategies aligned with physics priors using **`Policy`** in RLang. However, while some physics knowledge directly leads to executable policies, other knowledge only provides desiderata or necessary conditions that characterize good policies. These high-level principles narrow the policy search space but do not pinpoint an exact policy, for which reinforcement learning remains indispensable. Our physics-informed program (PiP) utilizes **`Effect`** to guide an RL process in searching for optimal policies, resulting in PiP-guided RL. Finally, a vision-based neural network controller breaks down high-level strategies into a sequence of navigation actions to avoid collision using visual information. The following discussion primarily addresses our core contribution on PiP and program-guided RL, while briefly touching on other modules which we adapt from previous works (Chaplot et al., 2020; Yin et al., 2022; Li et al., 2025b). Detailed neural network architectures are in Appendix B.

Neural Perception. A critical subtask of indoor navigation is mapping and pose estimation, which creates a map representation of surrounding obstacles and corrects pose readings from motion sensors based on additional visual information. We employ the pretrained neural SLAM model in (Chaplot et al., 2020), which provides robustness to the sensor noise during navigation. This SLAM module internally maintains a spatial map m_t and the agent's pose estimate $\hat{p}_t$ that is different from the raw sensor reading $\bar{p}_t$. 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. Note that each "location" in the spatial map corresponds to a cell of size $5\,\mathrm{cm} \times 5\,\mathrm{cm}$ in the physical world. The neural SLAM module, denoted by $\boldsymbol{F}_{\text{SLAM}}$, is parameterized by a residual network (θ_{SLAM}) (He et al., 2016) and generates new pose estimates and maps auto-regressively using current the RGB image, the two most recent motion sensor readings, and previous maps and pose estimates: $m_t, \hat{p}_t = \boldsymbol{F}_{\text{SLAM}}(v_t, \bar{p}_{t-1:t}, \hat{p}_{t-1}, m_{t-1}|\theta_{\text{SLAM}})$.

Wireless measurements also require additional processing to extract hidden information about the signal propagation path. The three-dimensional tensor $w_t \in \mathbb{R}^{N_1\times N_2\times N_3}$, after path estimation through low-rank decomposition (Zhou et al., 2017; Yin et al., 2022), produces tuples $\{g_n, \Omega_n^{\text{rx}}, \Omega_n^{\text{tx}}\}_{n=1}^N$, where 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 angle of arrival (AoA) and departure (AoD), respectively. We denote by $(g_1, \Omega_1^{\text{rx}}, \Omega_1^{\text{tx}})$ the tuple under the strongest signal strength, indicating that the path experiences the least number of reflections.

Another quantity crucial to WIN is the link state, which is categorized into Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS). A location (x, y) (or pose p) is said to be of LOS if there is a wireless signal path wherein electromagnetic waves traverse from the transmitter to the receiver without encountering any obstacles. 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. X-order NLOS ($X \in \mathbb{N}_+$) implies that at least one electromagnetic wave in the wireless link undergoes X-time reflection or diffraction. Note that the link state, denoted by $\ell_t = 1, 2, \ldots$, is a wireless terminology instead of the actual state input to be fed into RL models. Instead, the agent learns to estimate the link state, $\hat{\ell}_t$, from the path estimates using a vanilla fully connected network (Yin et al., 2022). In summary, one can view this neural wireless information processing as a neural network-parameterized function: $\{g_n(t), \Omega_n^{\text{rx}}(t), \Omega_n^{\text{tx}}(t)\}_{n=1}^N, \hat{\ell}_t = \boldsymbol{F}_{\text{wireless}}(w_t|\theta_{\text{wls}})$.

Physics-Informed Program. We first articulate the physics priors that are helpful in shaping an effective navigation strategy. *Prior* #1: The **principle of reversibility** states that an electromagnetic wave traversing from the source to the target will follow the same path if its direction is reversed. Such a principle leads to a simple yet effective strategy: following the angle of arrival (AoA) of the strongest path, since the strongest path experiences the least number of reflections and naturally detours around obstacles. We create a symbolic policy, referred to as `reverse AoA`, to encapsulate this strategy, where `intermediate` is a waypoint along the AoA (the second entry of `path estimate`) by an offset distance of $D = 2.5\,\mathrm{m}$ (hyperparameter). The **`Option`**, a declaration in RLang, prescribes a sequence of actions starting from the initial condition **`init`** to the termination **`until`**. The **`Option`** represents a low-level vision-based controller to be introduced later.

```
Policy: reverse AoA
    if not pose estimate == goal:
        intermediate[1] := pose estimate[1] + D * cos(path estimate[2])
        intermediate[2] := pose estimate[2] + D * sin(path estimate[2])
        Execute Option: Visual Control: init := pose estimate until :=
    intermediate
```

The reversibility prior is less effective in higher-order NLOS, for which we consider *Prior* #2: the source of an electromagnetic wave acquires the **maximum signal strength**, which declines along the path. In other words, if one considers the overall SNR at a pose $g(p) = \sum_{n=1}^{N} g_n(p)$, the closer the agent is to the transmitter, the higher its received SNR is. In contrast to *Prior* #1, this prior knowledge does not directly prescribe a navigation strategy, since the neural perception does not provide an SNR ascent direction, unlike AoA. It rather lays down a desideratum that the optimal strategy should meet. Similarly, *Prior* #3, rooted in the **link state monotonicity**, states that a necessary condition for a navigation path to be optimal is that the link state decreases monotonically.

Program-Guided RL. Since *Prior* #2 and #3 do not induce a precise symbolic policy, we turn to PPO, which is referred to as the **`Neural Policy`**, to search for an RL policy that complies with the priors. To be consistent with the symbolic policy `reverse AoA`, the output of **`Neural Policy`** is also an intermediate waypoint (equivalently, the offset angle $\hat{\Omega}_t$ w.r.t. x-axis), and the input is the processed wireless data $\boldsymbol{F}_{\text{wireless}}(w_t)$. In practical implementations, we consider a discretization of the angle range $\hat{\Omega}_t \in \Pi = \{-170, \ldots, 0, \ldots, 170, 180\}$. Substitute into the distance function c_t the coordinate of the waypoint determined by $\hat{\Omega}_t$, and we obtain the reward.

The central question in guiding the PPO process is how to program those desiderata using RLang. At first glance, it seems that this question bears a similar spirit of guided policy search (Levine & Abbeel, 2014; Li et al., 2023) and exploration (Kang et al., 2018; Yang et al., 2023), where physics priors provide guidance. However, it is not straightforward to translate such priors into concrete policy, which is a prerequisite for the importance sampling (IS) technique commonly employed in guided search. We consider directly replacing the sampled action from the PPO network with a physics prior-compliant one (if available) and replacing the actual reward with the IS-weighted one (Paine et al., 2020; Libardi et al., 2021). The reweighted rewards only apply to the action loss without changing the target value in the value loss as in (Espeholt et al., 2018; Libardi et al., 2021).

Denote by π_ϕ the neural network policy, and the action selection distribution is given by $\pi_\phi(\cdot|\boldsymbol{F}_{\text{wireless}}(w_t))$. We now define the set of prior-compliant actions through RLang element **`ActionRestriction`** and the IS-weighted reward through **`Effect`**. Given the current and last pose estimates $\hat{p}_{t-1:t}$ and the associated SNR $g_t = g(\hat{p}_t)$, the angle of the vector (`movement angle`) connecting two poses is given by $\nu_t = \arctan(\hat{y}_t - \hat{y}_{t-1}/\hat{x}_t - \hat{x}_{t-1})$ measured in degrees with possible $\pm 180°$, depending on which quadrant the vector is within. The compliant actions are those along the SNR ascent direction with deviation no more than $10°$ (discretization resolution).

```
ActionRestriction SNR prior:
    if SNR > Last SNR:
        Return angle in Π and movement angle - 10° <= angle <= movement
    angle + 10°
    if SNR <= Last SNR:
        Return angle in Π and movement angle - 190° <= angle <= movement
    angle - 170°
Effect Cost Correction:
    Return cost = cost * \# SNR prior *  P(Neural Policy)(angle)
Effect Link State Prior:
    if Line State Estimate > Last Link State Estimate:
        terminate Neural Policy and reset
```

Once the action set is settled, whenever the action $\hat{\Omega}_t$ from **`Neural Policy`** is not compliant, the PiPRL agent randomly picks one $\hat{\Omega}'_t$ from the restricted set and incurs a cost c_t that is biased with respect to the original action. To obtain an unbiased estimate, one can employ the importance sampling technique often seen in off-policy evaluation (Xie et al., 2019; Bannon et al., 2020) and

online learning with bandit feedback (Lattimore & Szepesvári, 2020; Li et al., 2021). The probability of choosing the compliant action is $1/\#\text{compliant}$ under the uniform distribution, while its counterpart under **`Neural Policy`** is given by $\pi_\phi(\hat{\Omega}'_t)$. The quotient $\pi_\phi(\hat{\Omega}'_t)/(1/\#\text{compliant})$ accounts for the correction in the change of probability measure. Hence, the corrected cost is $\hat{c}_t = \#\text{compliant} \cdot \pi_\phi(\hat{\Omega}'_t) \cdot c_t$,as shown in the **`Effect`** above. Finally, to enforce the *Prior* #3 on link state monotonicity, we simply define an **`Effect`** that terminates the PPO training if the link state estimate increases and restarts a new episode. We summarize the entire PiPRL workflow in the following, where a meta symbolic program instructs other policy programs and guidance programs.

```
Policy meta-program:
    if link state estimate <= 2:
            Execute Policy: reverse AoA
    else:
        Execute Neural Policy: PPO and ActionRestriction: SNR prior  and
        Effect: Cost Correction, Link State Prior
```

Visual Controller. Once the intermediate waypoint is determined either by the symolic or neural policy, a path planner denoted by $\boldsymbol{F}_{\text{planner}}$, based on the Fast Marching method (Sethian, 1996), computes the shortest path from the current location to the waypoint 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 $p_t^S = \boldsymbol{F}_{\text{planner}}(p_t^L, m_t, \hat{p}_t)$, which is the farthest point on the path within the map. Then, the visual controller takes in the path-planning output and the camera images, producing navigation actions $a_t = \pi_{\text{ctrl}}(v_t, p_t^S|\theta_{\text{ctrl}})$ for collision avoidance. The visual controller, parameterized by a recurrent neural network consisting of ResNet18 (He et al., 2016), is a task-invariant neural policy pretrained through behavioral cloning (Chaplot et al., 2020).

5 Experiments

This section evaluates the proposed PiPRL for WIN tasks, aiming to answer the following questions. 1) **Sample Efficiency**: Does PiPRL take fewer training data than the non-physics-based baselines? 2) **Zero-shot Generalization**: Can the PiPRL agent navigate in unseen environments without fine-tuning? Due to the page limit, we move the ablation study on the SNR and link state priors to Appendix C. We follow the same setup in (Li et al., 2025b) and briefly touch upon some key aspects. We remind the reader that $\boldsymbol{F}_{\text{SLAM}}$, $\boldsymbol{F}_{\text{wireless}}$ and $\boldsymbol{F}_{\text{planner}}$ are all pre-trained models and the program-guided PPO is our main focus, whose hyperparameters are listed in Appendix B.

Experiment Setup. The experiment includes 21 different indoor maps (the first 15, A–O, for training and the remaining 6, P–U, for testing) from the Gibson dataset (Xia et al., 2018) labeled using the first 21 characters in the Latin alphabet (A, B, . . ., U). Table 3 in Appendix B presents the complete list. For each map, we pick ten target locations that 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 remaining four (7-10) correspond to 2^+-NLOS scenarios.

Baselines. We consider three baseline navigation algorithms: 1) non-physics-based end-to-end RL (NPRL): the RL policy is purely the **`Neural Policy`** without priors. 2) Wireless-assisted navigation (WAN): This non-RL-based method, put forth in (Yin et al., 2022), only utilizes the symbolic policy **`reverse AoA`** in LOS and 1-NLOS, while switching to random exploration in 2^+-NLOS using the neural SLAM as suggested by (Chaplot et al., 2020). 3) Vision-augmented SLAM (V-SLAM): This policy leverages an object-detection algorithm (Bochkovskiy et al., 2020) in computer vision: once the target transmitter is within the view, the agent can localize and approach the target. It reduces to neural SLAM otherwise. The first two are primary baselines since our PiPRL is a hybrid of neural and symbolic policies. Additionally, to highlight the necessity of leveraging wireless signals in indoor navigation, we consider the third baseline where V-SLAM only takes in RGB images without wireless inputs.

Training. The first 15 maps (A-O) with associated 10 task positions are utilized to learn a PiPRL 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. One training instance terminates if the agent completes the task more than 6 episodes out of 10. This procedure is repeated until the agent has been exposed to all 15 maps with all target positions. In contrast, NPRL follows *rotation* training to alleviate catastrophic forgetting: after finishing the training on the current task, we randomly select a few previous tasks (Li et al., 2025b) to re-train the model before moving to the next task to refresh NPRL's "memory".

Sample Efficiency. We first evaluate the sample efficiency of the PiPRL training process by comparing the number of training episodes and GPU hours of PiPRL in LOS, 1-NLOS, and 2^+-NLOS with those of NPRL. As one can see from Table 1 (full table in Appendix C), the encoding physics priors via symbolic programs does reduce training samples. The #GPU hours are reduced by 26%, which is even more so as training progresses, when the neural policy starts to acquire the physics prior guided by the program.

Table 1: GPU hours and Episodes (Eps) of PiPRL and NPRL for Maps A-D (More in Table 6). Experiments are conducted on a Linux GPU workstation with an AMD Threadripper 3990X (64 Cores, 2.90 GHz) and an NVidia RTX 8000.

Label	Map Name	PiPRL		NPRL		Reduction (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

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. Table 2 summarizes part of the testing results (the first three testing maps; the full table is in Appendix C). We report the average *normalized path length* (NPL) of 20 repeated tests, which is defined as the quotient 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. One can see from Table 2 that PiPRL consistently outperforms others.

Table 2: A comparison of NPLs under 3 testing maps (More in Table 7) 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
PiPRL	1.01 $\pm$ 0.01	1.25 $\pm$ 0.02	2.13 $\pm$ 0.04	1.01 $\pm$ 0.01	1.37 $\pm$ 0.02	2.43 $\pm$ 0.05	1.01 $\pm$ 0.00	1.13 $\pm$ 0.03	2.65 $\pm$ 0.06
NPRL	2.03 $\pm$ 1.02	2.72 $\pm$ 0.55	4.96 $\pm$ 1.37	2.12 $\pm$ 1.00	3.08 $\pm$ 0.68	5.00 $\pm$ 1.41	2.28 $\pm$ 1.03	2.49 $\pm$ 0.81	4.99 $\pm$ 1.20
V-SLAM	1.05 $\pm$ 0.02	1.82 $\pm$ 0.51	4.58 $\pm$ 1.12	1.11 $\pm$ 0.03	2.89 $\pm$ 0.73	4.89 $\pm$ 1.00	1.09 $\pm$ 0.03	1.68 $\pm$ 0.6	4.68 $\pm$ 1.01
WAN	1.02 $\pm$ 0.00	1.32 $\pm$ 0.02	3.88 $\pm$ 0.82	1.01 $\pm$ 0.01	1.63 $\pm$ 0.05	3.71 $\pm$ 0.71	1.01 $\pm$ 0.00	1.23 $\pm$ 0.02	3.97 $\pm$ 0.83

6 Conclusion

We introduced PiPRL, a neuro-symbolic reinforcement learning framework that integrates physics-informed symbolic programs with neural network-based perception and control to enable zero-shot generalization in wireless indoor navigation tasks. At the core of PiPRL is a domain-specific language (RLang) for encoding symbolic programs that express physics priors–ranging from directly executable policies to high-level desiderata that characterize effective navigation strategies. These symbolic programs serve as structured, interpretable inductive biases that guide the reinforcement learning process when explicit policy specification is infeasible.

A Related Works on Neuro-Symbolic Learning

We have reviewed recent advancements in physics-informed inductive biases in RL and machine learning in general. Their specific applications in indoor navigation have also been discussed in the main text. This section is devoted to the recent developments on program-guided agents and neuro-symbolic learning.

Program-Guided Agent. There has been a recent surge of interest in learning from languages and instructions (Luketina et al., 2019). Many of them attempt to learn mappings for the semantic meaning of natural language to grounded information for agents (Kaplan et al., 2017; Bahdanau et al., 2018), which can also be viewed as inductive biases. Some other endeavors include natural-language-based task specification (Tellex et al., 2011; Fried et al., 2018) and policy conditioning (Zhong et al., 2020).

However, natural language statements can often be ambiguous to humans, which motivates the usage of formal languages, including logic programming (Jothimurugan et al., 2019; Jiang & Luo, 2019) and domain-specific languages (DSLs) (Ghallab et al., 1998; Rodriguez-Sanchez et al., 2023). Our work leverages the DSL program. Yet, unlike those prior works (Ghallab et al., 1998; Rodriguez-Sanchez et al., 2023; Sun et al., 2020), our program does not focus on task-specification, grounding information, but rather defining a symbolic policy coherent with physics priors.

Neuro-Symbolic RL The incorporation of DSL programs also distinguishes our work from contemporary efforts on neuro-symbolic RL. One of the common approaches in combining neuro learning and symbolic planning relies on integrating differentiable logic programming with deep RL policies (Jiang & Luo, 2019; Evans & Grefenstette, 2018; Shindo et al., 2025). The advantage of this approach is that gradient-based optimization techniques apply to both symbolic planning and RL (Evans & Grefenstette, 2018; Delfosse et al., 2023). However, it is less user-friendly than DSL programs in terms of readability and accessibility.

The modularized and hierarchical neuro-symbolic integration has also been explored in the literature (Sun et al., 2020; Kokel et al., 2021; Kuo et al., 2020). One of the shared characteristics is that a symbolic program takes the role of a high-level planner, while RL fulfills the commands from symbolic planners. Our PiPRL, even though still hierarchical, also employs RL, guided by symbolic programs, to search for high-level strategies, which is motivated by the complexity of real-world applications where no single symbolic policy is sufficient.

B Experiment Setup

This section provides additional setup details. To begin with, Table 3 presents our map labeling. In addition, we also summarize the neural network architectures employed in this work.

- **PPO Policy** 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 Cho et al. (2014). Additionally, there are two distinct layers at the end, referred to as the actor output layer and the critic output layer. We follow the standard PPO implementation as in (Schulman et al., 2017) and summarize its hyperparameters in Table 4.
- **Visual Control Policy** is constructed using a recurrent neural network architecture. It incorporates a pre-trained ResNet18 (He et al., 2016) as the visual encoder, which is followed by fully connected layers and a GRU layer.
- **Neural SLAM** consists of Resnet18 convolutional layers followed by two fully-connected layers, then followed by 3 deconvolutional layers.
- **Link State Classifier** is simply a 2-layer fully-connected network with ReLU activation (Nair & Hinton, 2010).

Table 3: The 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				

Table 4: PPO Hyperparameters.

Hyperparameters	**Value**
discounting factor	0.99
clipping parameter	0.2
weight for value loss	0.5
weight for entropy bonus	0.01
learning Rate	3×10^{-4}
Batch Size	64
exploration rate (e.g., epsilon-greedy)	0.1

C Additional Experiments

This section supplements additional experimental results. The comprehensive examinations of sample efficiency and generalization are in Table 6 and Table 7, respectively. Additionally, we want to answer the question: *To what extent do the symbolic priors help?*

C.1 Ablation Study

We simply erase the corresponding program when conducting the ablation. For example, when studying SNR prior, we remove **`ActionRestriction`** and **`Effect`** on cost correction. As one can see from Table 5, the answer to the question is affirmative, as the SNR ablation returns significantly higher NPLs in 2^+-NLOS. Similarly, the third row in Table 5 indicates that without the link state prior, the agent frequently revisits the high-order NLOS areas in testing, which yields higher NPLs in NLOS scenarios.

Table 5: 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
PiPRL	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

Table 6: GPU hours and Episodes (Eps) of PiPRL and NPRL for Maps A-O. Experiments are conducted on a Linux GPU workstation with an AMD Threadripper 3990X (64 Cores, 2.90 GHz) and an NVidia RTX 8000.

Label	Map Name	PiPRL		NPRL		Reduction (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

Table 7: 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
PiPRL	1.01 ± 0.01	1.25 ± 0.02	2.13 ± 0.04	1.01 ± 0.01	1.37 ± 0.02	2.43 ± 0.05	1.01 ± 0.00	1.13 ± 0.03	2.65 ± 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

References

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021. URL http://jmlr.org/papers/v22/19-736.html.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. DOI: 10.48550/arxiv.1807.06757. [Online] Available at https://arxiv.org/abs/1807.06757.

Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pp. 1–14, New York, NY, USA, 2020. Association for Computing Machinery. ISBN

9781450370851. DOI: 10.1145/3372224.3380894. URL https://doi.org/10.1145/3372224.3380894.

Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. Learning to understand goal specifications by modelling reward. *arXiv*, 2018. DOI: 10.48550/arxiv.1806.01946.

James Bannon, Brad Windsor, Wenbo Song, and Tao Li. Causality and batch reinforcement learning: Complementary approaches to planning in unknown domains. *arXiv preprint arXiv:2006.02579*, 2020. DOI: 10.48550/arxiv.2006.02579. [Online] Available at https://arxiv.org/pdf/2006.02579.

Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12: 149–198, 2000. DOI: 10.1613/jair.731.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. [Online] Available at https://arxiv.org/pdf/2004.10934.

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural SLAM. In *International Conference on Learning Representations*, 2020.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN rncoder–decoder for statistical machine translation. In *2014 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014.

Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022. ISSN 0885-7474. DOI: 10.1007/s10915-022-01939-z.

Quentin Delfosse, Hikaru Shindo, Devendra Dhami, and Kristian Kersting. Interpretable and explainable logical policies via neurally guided symbolic abstraction. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 50838–50858, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9f42f06a54ce3b709ad78d34c73e4363-Paper-Conference.pdf.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1407–1416, 2018. URL https://proceedings.mlr.press/v80/espeholt18a.html.

Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. *J. Artif. Int. Res.*, 61(1):1–64, January 2018. ISSN 1076-9757.

Alireza Fallah, Kristian Georgiev, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of debiased model-agnostic meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 24, pp. 3096—3107, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/18085327b86002fc604c323b9a07f997-Paper.pdf.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower

models for vision-and-language navigation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 3318–3329, 2018.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1587–1596, 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2160–2169, 2019. URL https://proceedings.mlr.press/v97/geist19a.html.

Malik Ghallab, Adele Howe, Craig Knoblock, Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. *PDDL—The Planing Domain Definition Language*. The International Conference on Artificial Intelligence Planning Systems Planning Competition Committee, 1998. [Online] Available at https://nergmada.github.io/planning-wiki/.

Francesco Guidi, Anna Guerra, and Davide Dardari. Millimeter-wave massive arrays for indoor SLAM. In *2014 IEEE International Conference on Communications Workshops (ICC)*, pp. 114–120, 2014. DOI: 10.1109/ICCW.2014.6881182.

Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15281–15295, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6255f22349da5f2126dfc0b007075450-Paper-Conference.pdf.

Kim Hammar, Tao Li, Rolf Stadler, Quanyan Zhu, and Kim Hammar. Adaptive security response strategies through conjectural online learning. *IEEE Transactions on Information Forensics and Security*, 20:4055–4070, 2025. ISSN 1556-6013. DOI: 10.1109/tifs.2025.3558600.

Yu Han, Meng Wang, Linghui Li, Claudio Roncoli, Jinda Gao, and Pan Liu. A physics-informed reinforcement learning-based strategy for local and coordinated ramp metering. *Transportation Research Part C: Emerging Technologies*, 137:103584, 2022. ISSN 0968-090X. DOI: 10.1016/j.trc.2022.103584.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

Matteo Hessel, Hado van Hasselt, Joseph Modayil, and David Silver. On inductive biases in deep reinforcement learning. *arXiv*, 2019. DOI: 10.48550/arxiv.1907.02908.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A Ortega, DJ Strouse, Joel Z Leibo, and Nando de Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3040—3049, 2019.

Zhengyao Jiang and Shan Luo. Neural logic reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3110–3119, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/jiang19a.html.

Kishor Jothimurugan, Rajeev Alur, and Osbert Bastani. A composable specification language for reinforcement learning tasks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f5aa4bd09c07d8b2f65bad6c7cd3358f-Paper.pdf.

Bingyi Kang, Zequn Jie, and Jiashi Feng. Policy optimization with demonstrations. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2469–2478, 2018. URL https://proceedings.mlr.press/v80/kang18a.html.

Russell Kaplan, Christopher Sauer, and Alexander Sosa. Beating atari with natural language guided reinforcement learning. *arXiv*, 2017. DOI: 10.48550/arxiv.1704.05539.

Donald Knuth. *The Computers & Typesetting, Vol. A: The Texbook*. Addison-Wesley Longman Publishing Co., Inc., MA, USA, 1986. ISBN 0201134470.

Harsha Kokel, Arjun Manoharan, Sriraam Natarajan, Balaraman Ravindran, and Prasad Tadepalli. Reprel: Integrating relational planning and reinforcement learning for effective abstraction. *Proceedings of the International Conference on Automated Planning and Scheduling*, 31(1):533–541, May 2021. DOI: 10.1609/icaps.v31i1.16001. URL https://ojs.aaai.org/index.php/ICAPS/article/view/16001.

Vijaymohan R Konda and Vivek S Borkar. Actor-critic–type learning algorithms for markov decision processes. *SIAM Journal on Control and Optimization*, 38(1):94–123, 1999. ISSN 0363-0129. DOI: 10.1137/s036301299731669x.

Tejas D Kulkarni, Karthik R Narasimhan, Ardavan Saeedi, and Joshua B Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Yen-Ling Kuo, Boris Katz, and Andrei Barbu. Encoding formulas as deep networks: Reinforcement learning for zero-shot execution of ltl formulas. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5604–5610, 2020. DOI: 10.1109/IROS45743.2020.9341325.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge Core. Cambridge University Press, Cambridge, 01 2020. ISBN 9781108486828. DOI: 10.1017/9781108571401. URL https://www.cambridge.org/core/books/bandit-algorithms/8E39FD004E6CE036680F90DD0C6F09FC.

Mikko Lauri, David Hsu, and Joni Pajarinen. Partially observable markov decision processes in robotics: A survey. *IEEE Transactions on Robotics*, 39(1):21–40, 2023. ISSN 1552-3098. DOI: 10.1109/tro.2022.3200138.

Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franọis Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018. ISSN 0893-6080. DOI: 10.1016/j.neunet.2018.07.006.

Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/c7c9344b5a3c0533e29fa69ce807cf08-Paper.pdf.

Tao Li and Quanyan Zhu. On convergence rate of adaptive multiscale value function approximation for reinforcement learning. *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2019. DOI: 10.1109/mlsp.2019.8918816.

Tao Li, Guanze Peng, and Quanyan Zhu. Blackwell online learning for markov decision processes. In *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, 2021. DOI: 10.1109/CISS50987.2021.9400319.

Tao Li, Guanze Peng, Quanyan Zhu, and Tamer Baar. The confluence of networks, games, and learning a game-theoretic framework for multiagent decision making over networks. *IEEE Control Systems*, 42(4):35–67, 2022a. ISSN 1066-033X. DOI: 10.1109/mcs.2022.3171478.

Tao Li, Yuhan Zhao, and Quanyan Zhu. The role of information structures in game-theoretic multi-agent learning. *Annual Reviews in Control*, 53:296–314, 2022b. ISSN 1367-5788. DOI: 10.1016/j.arcontrol.2022.03.003.

Tao Li, Haozhe Lei, and Quanyan Zhu. Self-adaptive driving in nonstationary environments through conjectural online lookahead adaptation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7205–7211, 2023. DOI: 10.1109/ICRA48891.2023.10161368.

Tao Li, Kim Hammar, Rolf Stadler, and Quanyan Zhu. Conjectural online learning with first-order beliefs in asymmetric information stochastic games. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 6780–6785, 2024a. DOI: 10.1109/cdc56724.2024.10886479.

Tao Li, Henger Li, Yunian Pan, Tianyi Xu, Zizhan Zheng, and Quanyan Zhu. Meta stackelberg game: Robust federated learning against adaptive and mixed poisoning attacks. *arXiv preprint arXiv:2410.17431*, 2024b. [Online] Available at https://arxiv.org/pdf/2410.17431.

Tao Li, Juan Guevara, Xinghong Xie, and Quanyan Zhu. Self-confirming transformer for belief-conditioned adaptation in offline multi-agent reinforcement learning. In *Proceedings of the Seventh Workshop on Adaptive and Learning Agents, the Twenty Fourth International Conference on Autonomous Agents and Multiagent Systems*, pp. 1–10, 2025a. DOI: 10.48550/arxiv.2310.04579. URL https://openreview.net/forum?id=kMaYSsEwCT.

Tao Li, Haozhe Lei, Hao Guo, Mingsheng Yin, Yaqi Hu, Quanyan Zhu, and Sundeep Rangan. Digital twin-enhanced wireless indoor navigation: Achieving efficient environment sensing with zero-shot reinforcement learning. *IEEE Open Journal of the Communications Society*, 6:2356–2372, 2025b. DOI: 10.1109/ojcoms.2025.3552277.

Gabriele Libardi, Gianni De Fabritiis, and Sebastian Dittert. Guided exploration with proximal policy optimization using a single demonstration. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 6611–6620, 2021. URL https://proceedings.mlr.press/v139/libardi21a.html.

Fanghui Liu, Luca Viano, and Volkan Cevher. Understanding deep neural function approximation in reinforcement learning via \epsilon-greedy exploration. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 5093–5108, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/2119b5ac365c30dfac17a840c2755c30-Paper-Conference.pdf.

Shutian Liu, Tao Li, and Quanyan Zhu. Game-theoretic distributed empirical risk minimization with strategic network design. *IEEE Transactions on Signal and Information Processing over Networks*, 9:542–556, 2023. ISSN 2373-776X. DOI: 10.1109/tsipn.2023.3306106.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A survey of reinforcement learning informed by natural language. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 6309–6317, 2019. DOI: 10.24963/ijcai.2019/880. URL https://doi.org/10.24963/ijcai.2019/880.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 6820–6829, 2020. URL https://proceedings.mlr.press/v119/mei20b.html.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. ISSN 0028-0836. DOI: 10.1038/nature14236.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 1928–1937, 2016. URL https://proceedings.mlr.press/v48/mniha16.html.

Sharada Mohanty, Jyotish Poonganam, Adrien Gaidon, Andrey Kolobov, Blake Wulfe, Dipam Chakraborty, Gražvydas Šemetulskis, João Schapke, Jonas Kubilius, Jurgis Paükonis, Linas Klimas, Matthew Hausknecht, Patrick MacAlpine, Quang Nhat Tran, Thomas Tumiel, Xiaocheng Tang, Xinwei Chen, Christopher Hesse, Jacob Hilton, William Hebgen Guss, Sahika Genc, John Schulman, and Karl Cobbe. Measuring sample efficiency and generalization in reinforcement learning benchmarks: Neurips 2020 procgen benchmark. In Hugo Jair Escalante and Katja Hofmann (eds.), *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133, pp. 361–395, 2021. URL https://proceedings.mlr.press/v133/mohanty21a.html.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 807–814, 2010. ISBN 9781605589077.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287, 1999. ISBN 1558606122.

Thanh Thi Nguyen and Vijay Janapa Reddi. Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):3779–3795, 2023. ISSN 2162-237X. DOI: 10.1109/tnnls.2021.3121870.

Tom Le Paine, Caglar Gulcehre, Bobak Shahriari, Misha Denil, Matt Hoffman, Hubert Soyer, Richard Tanburn, Steven Kapturowski, Neil Rabinowitz, Duncan Williams, Gabriel Barth-Maron, Ziyu Wang, Nando de Freitas, and Worlds Team. Making efficient use of demonstrations to solve hard exploration problems. In *International Conference on Learning Representations*, 2020. DOI: 10.48550/arxiv.1909.01387.

Yunian Pan, Tao Li, Henger Li, Tianyi Xu, Zizhan Zheng, and Quanyan Zhu. A first order meta Stackelberg method for robust federated learning. In *Adversarial Machine Learning Frontiers Workshop at 40th International Conference on Machine Learning*, 2023a. DOI: 10.48550/arxiv.2306.13800.

Yunian Pan, Tao Li, and Quanyan Zhu. Is stochastic mirror descent vulnerable to adversarial delay attacks? a traffic assignment resilience study. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 8328–8333, 2023b. DOI: 10.1109/cdc49753.2023.10384003.

Yunian Pan, Tao Li, and Quanyan Zhu. On the variational interpretation of mirror play in monotone games. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 6799–6804, 2024. DOI: 10.1109/cdc56724.2024.10885800. URL https://ieeexplore.ieee.org/document/10885800.

Yunian Pan, Tao Li, and Quanyan Zhu. Model-agnostic meta-policy optimization via zeroth-order estimation: A linear quadratic regulator perspective. *arXiv preprint arXiv:2503.00385*, 2025. [Online] Available at https://arxiv.org/pdf/2503.00385.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. DOI: 10.1016/j.jcp.2018.10.045.

Rafael Rodriguez-Sanchez, Benjamin Adin Spiegel, Jennifer Wang, Roma Patel, Stefanie Tellex, and George Konidaris. RLang: A declarative language for describing partial world knowledge to reinforcement learning agents. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 29161–29178, 2023. URL https://proceedings.mlr.press/v202/rodriguez-sanchez23a.html.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346, 2019. DOI: 10.1109/ICCV.2019.00943.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 1889–1897, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. [Online] Available at https://arxiv.org/pdf/1707.06347.

J A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences (PNAS)*, 93(4):1591–1595, Feb. 1996. ISSN 0027-8424. DOI: 10.1073/pnas.93.4.1591.

Arash Shahmansoori, Gabriel E. Garcia, Giuseppe Destino, Gonzalo Seco-Granados, and Henk Wymeersch. Position and orientation estimation through millimeter-wave MIMO in 5g systems. *IEEE Transactions on Wireless Communications*, 17(3):1822–1835, 2018. ISSN 1536-1276. DOI: 10.1109/twc.2017.2785788.

Hikaru Shindo, Quentin Delfosse, Devendra Singh Dhami, and Kristian Kersting. BlendRL: A framework for merging symbolic and neural policy learning. In *International Conference on Learning Representations*, 2025. DOI: 10.48550/arxiv.2410.11689.

Shao-Hua Sun, Te-Lin Wu, and Joseph J. Lim. Program guided agent. In *International Conference on Learning Representations*, 2020.

Enrico Sutera, Vittorio Mazzia, Francesco Salvetti, Giovanni Fantin, and Marcello Chiaberge. Indoor point-to-point navigation with deep reinforcement learning and ultra-wideband. In *12th International Conference on Agents and Artificial Intelligence*, 2020. [Online] Available at https://arxiv.org/abs/2011.09241.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999. ISSN 0004-3702. DOI: 10.1016/s0004-3702(99)00052-1.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057—1063, 2000.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pp. 1507–1514. AAAI Press, 2011.

John N Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997. DOI: 10.1109/9.580874.

Guido van Rossum. Python programming language. https://www.python.org, 1991. Accessed: 2025-06-06.

Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9068–9079, 2018.

Tengyang Xie, Yifei Ma, and Yu-Xiang Wang. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/4ffb0d2ba92f664c2281970110a2e071-Paper.pdf.

Hanlin Yang, Chao Yu, peng sun, and Siji Chen. Hybrid policy optimization from imperfect demonstrations. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 4653–4663, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/0f0a30c7b46be23a83317c5cb721fc43-Paper-Conference.pdf.

Mingsheng Yin, Akshaj Kumar Veldanda, Amee Trivedi, Jeff Zhang, Kai Pfeiffer, Yaqi Hu, Siddharth Garg, Elza Erkip, Ludovic Righetti, and Sundeep Rangan. Millimeter wave wireless assisted robot navigation with link state classification. *IEEE Open Journal of the Communications Society*, 3:493–507, 2022.

Mingsheng Yin, Tao Li, Haozhe Lei, Yaqi Hu, Sundeep Rangan, and Quanyan Zhu. Zero-shot wireless indoor navigation through physics-informed reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5111–5118, 2024. DOI: 10.1109/icra57147.2024.10611229.

Peng Zhao and Yongming Liu. Physics informed deep reinforcement learning for aircraft conflict resolution. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8288–8301, 2022. ISSN 1524-9050. DOI: 10.1109/tits.2021.3077572.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 737–744, 2020. DOI: 10.1109/SSCI47803.2020.9308468.

Victor Zhong, Tim Rocktaschel, and Edward Grefenstette. RTFM: Generalising to new environment dynamics via reading. In *International Conference on Learning Representations*, 2020. DOI: 10.5555/2900423.2900661.

Zhou Zhou, Jun Fang, Linxiao Yang, Hongbin Li, Zhi Chen, and Rick S. Blum. Low-rank tensor decomposition-aided channel estimation for millimeter wave mimo-ofdm systems. *IEEE Journal on Selected Areas in Communications*, 35(7):1524–1538, 2017. DOI: 10.1109/JSAC.2017.2699338.