

## Article

# Enhanced Deep Reinforcement Learning for Robustness Falsification of Partially Observable Cyber-Physical Systems

Yangwei Xing <sup>1</sup>, Ting Shu <sup>1,\*</sup> , Xuesong Yin <sup>2</sup> and Jinsong Xia <sup>1</sup>

<sup>1</sup> School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China; 2023220603078@mails.zstu.edu.cn (Y.X.); summerx@zstu.edu.cn (J.X.)

<sup>2</sup> School of Media and Design, Hangzhou Dianzi University, Hangzhou 310018, China; yinxs@hdu.edu.cn

\* Correspondence: shuting@zstu.edu.cn

## Abstract

Robustness falsification is a critical verification task for ensuring the safety of cyber-physical systems (CPS). Under partially observable conditions, where internal states are hidden and only input–output data is accessible, existing deep reinforcement learning (DRL) approaches for CPS robustness falsification face two key limitations: inadequate temporal modeling due to unidirectional network architectures, and sparse reward signals that impede efficient exploration. These limitations severely undermine the efficacy of DRL in black-box falsification, leading to low success rates and high computational costs. This study addresses these limitations by proposing DRL-BiT-MPR, a novel framework whose core innovation is the synergistic integration of a bidirectional temporal network with a multi-granularity reward function. Specifically, the bidirectional temporal network captures bidirectional temporal dependencies, remedies inadequate temporal modeling, and complements unobservable state information. The multi-granularity reward function includes fine-grained, medium-grained and coarse-grained layers, corresponding to single-step local feedback, phased progress feedback, and global result feedback, respectively, providing multi-time-scale incentives to resolve reward sparsity. Experiments are conducted on three benchmark CPS models: the continuous CARS model, the hybrid discrete-continuous AT model, and the controller-based PTC model. Results show that DRL-BiT-MPR increases the falsification success rate by an average of 39.6% compared to baseline methods and reduces the number of simulations by more than 50.2%. The framework’s robustness is further validated through theoretical analysis of convergence and soundness properties, along with systematic parameter sensitivity studies.

**Keywords:** cyber-physical systems; deep reinforcement learning; robustness falsification; bidirectional temporal network; multi-granularity reward function; metric temporal logic



Academic Editor: Shuang Xu

Received: 8 January 2026

Revised: 30 January 2026

Accepted: 4 February 2026

Published: 7 February 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

A Cyber-Physical System (CPS) [1] is a complex integrated system that integrates information technologies and physical entities such as sensors and actuators. Ensuring the safety of CPS is paramount, especially as they are increasingly deployed in safety-critical domains such as autonomous driving and industrial automation [2,3]. This necessitates effective verification of system robustness [4], which is the ability to maintain correct operation under perturbations and uncertainties. In practice, robustness falsification has emerged as an effective verification paradigm that actively searches for counterexamples to disprove robustness specifications, thereby uncovering potential design flaws and enhancing CPS

safety assessment [5]. However, a particularly demanding scenario arises when falsifying partially observable CPS, where internal states are hidden and only input–output data is accessible. This inherent partial observability presents a significant challenge for existing falsification techniques, motivating the research presented in this work.

Existing robustness falsification approaches for CPS can be categorized into heuristic methods and DRL-based techniques. Heuristic methods [6] do not require an exact mathematical model or gradient information of the system and only need to evaluate candidate solutions, which makes them broadly applicable. However, when falsifying partially observable CPS, they suffer from extremely low efficiency [7]. Their inefficiency stems from neglecting temporal correlations in inputs, often requiring tens of thousands of simulations. In contrast, DRL methods [6] promise high sample efficiency by learning to exploit the structure of counterexamples, potentially reducing the required simulations significantly. Nevertheless, DRL-based falsification faces two critical limitations in practical, partially observable CPS applications. First, they rely heavily on accurate system dynamic models. Second, DRL algorithms often encounter sparse reward signals [8,9] during the search process, leading to slow convergence and risks of falling into local optima. Thus, while heuristic methods offer model-free generality and DRL methods provide high sample efficiency, each approach exhibits fundamental limitations in the context of partially observable CPS falsification; heuristics lack the temporal reasoning needed for efficiency, whereas DRL requires unavailable state information and struggles with sparse rewards.

To overcome these inherent limitations of deep reinforcement learning in partially observable CPS falsification, recent research has primarily advanced along two distinct paths. On one hand, more specialized advanced falsification methods have been developed, such as Transformer-based sequence models, to enhance the capability of modeling long-range temporal dependencies. On the other hand, general Partially Observable Markov Decision Process (POMDP)-aware reinforcement learning methods have been introduced, employing memory mechanisms or belief-state modeling to better handle incomplete information. However, as elaborated in Section 2.3, these approaches still face specific challenges when applied to black-box CPS: they often fail to explicitly model the inherent bidirectional temporal causality of the system, lack predictive capability regarding future behavior to aid decision-making, and their reward mechanisms are typically not designed specifically to address the multi-scale sparsity problem induced by temporal logic specifications. Consequently, constructing a framework capable of simultaneously addressing bidirectional temporal perception and structured multi-granularity rewards is crucial for enhancing falsification efficacy in black-box environments. This work aims to address this challenge through the proposed DRL-BiT-MPR framework.

To address the core challenges of DRL-based CPS robustness falsification that include reliance on accurate system models, inadequate utilization of temporal correlations, and difficulties in handling partially observable internal dynamics, we propose a unified framework integrating two key components: a bidirectional temporal network (BiT) and a multi-granularity reward function. The BiT network serves as the state perception and temporal feature extraction module, addressing the limitations of traditional unidirectional networks in CPS modeling. Traditional networks can only process historical data and fail to leverage future contextual information, making them unable to capture the bidirectional temporal-causal dependencies inherent to CPS [10]. In contrast, the BiT network fuses past and future action sequences and system outputs to achieve more accurate state estimation. This capability aligns with the intricate temporal and causal correlations of CPS data, which unidirectional networks struggle to model effectively [11]. Through bidirectional information propagation, the BiT network explicitly interprets how past data shapes the current state with the aid of future contextual cues. It enables reliable state representation even

in black-box CPS scenarios where precise dynamic models are unavailable [12], directly reducing the algorithm's dependence on accurate system modeling while enhancing the exploitation of temporal correlations.

In addition to the BiT network, we design a multi-granularity reward function to address the sparse reward problem that hinders DRL performance in black-box environments [13]. Unlike traditional reward mechanisms that only provide feedback when the target state changes, our designed reward function delivers step-wise reward signals related to task progress. These signals are derived from the fine-grained state features extracted by the BiT network, ensuring that the reward feedback is closely aligned with the system's temporal-robustness correlation [14]. By receiving continuous task-relevant rewards, the DRL agent can quickly identify the optimization direction and reduce exploration inefficiency. In complex CPS tasks, the agent adjusts its strategy incrementally based on these granular signals [15], accelerating convergence to effective counterexample search strategies and mitigating the risk of falling into local optima.

The principal methodological advance presented in this work addresses a gap in prior research by coupling the solutions to two interdependent challenges. Previous methods often treat the issues of state uncertainty under partial observability and extreme reward sparsity as separate concerns. Here, the bidirectional temporal network and the multi-granularity reward function are conceived as an integrated unit. The network provides the temporal state awareness necessary to compute progress-sensitive rewards, while the reward function supplies the graded feedback required to steer and improve state estimation. This coupled approach is specifically devised to overcome the core limitation in black-box falsification, where incomplete perception and sparse signals reinforce each other negatively.

Collectively, the novelty and strength of the DRL-BiT-MPR framework stem from its integrated approach to solving the two fundamental limitations. The bidirectional temporal network provides a more accurate temporal representation under partial observability, while the multi-granularity reward function ensures efficient exploration guided by dense feedback. This dual design effectively reduces dependency on precise system models and overcomes the reward sparsity problem, setting it apart from prior DRL-based falsification methods.

The contributions of this study are as follows:

- We propose a novel bidirectional temporal network to address state uncertainty under partial observability. Its core innovation is the integration of historical and predictive information to explicitly model temporal causality, reducing reliance on precise system models.
- To overcome extreme reward sparsity, we design a multi-granularity reward mechanism. It decomposes the long-horizon falsification task to provide dense, multi-scale feedback, fundamentally improving exploration efficiency.
- The integrated DRL-BiT-MPR framework is empirically validated on three CPS benchmarks. It achieves an average improvement of 39.6% in success rate and reduces required simulations by 50.2%, with ablation studies confirming the contribution of each component.

The remaining parts of the paper are organized as follows: Sections 2 and 3 discuss related work and preliminaries, respectively. Section 4 introduces the details of our method and the evaluation methodology. Section 5 presents the evaluation results on three widely adopted CPS models, provides an in-depth analysis of the experimental results, and discusses the factors affecting the performance of our method. Finally, Section 6 summarizes our method and explores potential future work.

## 2. Related Work

### 2.1. Formal Verification Techniques

Formal verification methods serve as the traditional core techniques for CPS robustness verification. Their core idea, rooted in constructing a system mathematical model [16], involves applying strict logical reasoning to verify whether the system meets safety properties under all possible states [17]. Novak et al. proposed combining statistical model checking with cross-entropy methods [18], improving verification efficiency through rare event sampling; Zhang et al. designed a Bayesian statistical model checking technique [19] to reduce verification time for stochastic discrete-time hybrid systems; Modrak introduced Monte Carlo techniques [20] to guide state space random walks and assist model checking. These methods' greatest strength lies in their theoretical rigor as they ensure the reliability of verification results through exhaustive search or mathematical induction, making them well-suited for simple CPS scenarios with extremely high safety demands.

However, formal verification methods have insurmountable limitations [21]. On one hand, most physical components of CPS are continuous variables, forming an infinite state space [22] that requires discrete approximation, which either causes accuracy loss or computational explosion. On the other hand, these methods heavily depend on pre-constructed global mathematical models [23]. As CPS component complexity increases, model construction difficulty grows exponentially [24], and verification time often exceeds the acceptable engineering range.

### 2.2. Heuristic Methods

Heuristic methods are widely used black-box techniques in robustness-guided falsification, including simulated annealing, genetic algorithms [25], Tabu search, cross-entropy methods, and Gaussian regression. Their core advantage is that they do not require precise modeling and can optimize robustness values solely through interaction with the system's input and output. For example, the cross-entropy method approximates the input distribution induced by robustness through sampling [26], the Gaussian regression method transforms falsification into a region estimation problem [27] and uses Gaussian processes to construct the probabilistic semantics of temporal formulas, enhancing the targeting of counterexample search, and simulated annealing searches for the minimum robustness value in complex spaces through a random search strategy that simulates the physical annealing process [28]. The flexibility of these methods makes them applicable in industrial black-box CPS scenarios, especially for simple property verification where no model is available.

However, heuristic methods also have obvious flaws. First, they fail to utilize the temporal structure of CPS inputs [29], disassembling inputs into discrete control points and treating them as independent parameters. This loses the key dependency where historical inputs affect future outputs, resulting in extremely low efficiency in dynamically coupled scenarios. Second, they rely on random search or local iterative optimization, lacking the ability to learn counterexample structures and being prone to local optima in complex temporal properties. Additionally, they require a large number of simulations [30], making it difficult to meet engineering efficiency requirements.

### 2.3. DRL Methods

In recent years, DRL has become a research hotspot in CPS robustness falsification because it can reduce the number of simulations by utilizing the learnable structure of counterexamples. Yamagata et al. [8] were the first to apply DRL to this field, achieving a 60% reduction in the number of simulations in white-box scenarios and significantly outperforming heuristic methods. The advantage of these methods is that they can learn

system dynamic laws through neural networks to achieve end-to-end counterexample generation, making them suitable for CPS scenarios with learnable counterexample structures. The application scope of deep reinforcement learning in cyber-physical systems extends beyond falsification to include critical optimization challenges in domains such as the Internet of Things. For instance, a recent study utilized a distributed deep deterministic policy gradient (DDPG) framework to tackle the complex problem of resource allocation for minimizing the Age of Information in mobile wireless-powered IoT networks [31].

However, existing DRL methods face two core bottlenecks in black-box CPS scenarios. The first is state unobservability [32]: black boxes only provide output signals, and DRL cannot obtain internal states, making it difficult to model system dynamics. The second is reward sparsity: traditional DRL only provides 0–1 rewards when properties are violated, and 80% of exploration steps lack effective feedback, which leads to slow policy convergence and even convergence failure in complex nested properties. In addition, most existing DRL methods use unidirectional Long Short-Term Memory (LSTM) [33] to capture temporal information, which cannot handle CPS's bidirectional causal dependencies, further limiting their performance in black-box scenarios.

### 2.3.1. Advanced Falsification Methods

Beyond traditional DRL, more specialized falsification techniques have emerged in recent literature. Bayesian optimization [34] has gained popularity as a sample-efficient black-box optimizer for this task. It models the robustness function with a Gaussian process to actively select input sequences likely to minimize robustness, often outperforming random or grid-based search. Similarly, cross-entropy methods [35] and their variants have been adapted for temporal logic falsification by iteratively shifting sampling distributions toward low-robustness regions. While these methods demonstrate improved sample efficiency over basic heuristics, they share a key limitation: they treat the input signal as a static, high-dimensional vector. This representation discards the inherent temporal structure and dynamic dependencies of CPS trajectories, a disadvantage that becomes acute when falsifying properties with complex temporal operators or long horizons.

Beyond the methods discussed above, the latest research front in learning-based falsification explores the use of Transformer architectures. Concurrently, other innovative learning paradigms are being adapted for the falsification task. For example, graph neural networks have been employed to reason over the structural dependencies of system components [36], while meta-learning strategies aim to accelerate adaptation across families of similar specifications [5]. These approaches utilize self-attention mechanisms to capture long-range dependencies across input and output sequences, offering a powerful alternative to recurrent networks for modeling complex temporal dynamics.

Our DRL-BiT-MPR framework contributes to this direction with two distinct advancements. First, architecturally, standard Transformer-based models for sequential data typically operate in a unidirectional manner, processing historical information to infer current states. In contrast, our bidirectional temporal network explicitly incorporates predicted future information, creating a bidirectional temporal context that is critical for decision-making under partial observability. Second, in terms of optimization, while generic sequence models aim to learn accurate input–output mappings, our method directly targets the falsification objective through a dedicated multi-granularity reward function. This design specifically alleviates the reward sparsity inherent in temporal logic properties. Nonetheless, a common characteristic of these advanced learning-based falsifiers, including Transformer-based approaches, is their predominant reliance on unidirectional historical data for state representation. This limits their ability to explicitly model and leverage the bidirectional causal links that govern physical system dynamics. Thus, our work provides

a specialized solution that addresses the dual challenges of incomplete temporal perception and sparse feedback in black-box CPS falsification.

### 2.3.2. POMDP-Aware RL Methods

Addressing partial observability is a longstanding core challenge in reinforcement learning. A standard and widely adopted solution is to augment RL agents with memory mechanisms, such as recurrent neural networks or long short-term memory networks [37], enabling them to maintain an internal state from observation sequences. Algorithms like deep recurrent Q-networks (DRQN) [38] extend DQN with LSTM layers, while recurrent policy gradient methods integrate temporal dependencies into actor-critic frameworks. More recent lines of research explore explicit belief state modeling [39] or employ attention mechanisms [40] to focus on relevant historical information. Despite their general progress, applying these generic POMDP-RL methods to CPS falsification reveals specific shortcomings. They often struggle with the extreme reward sparsity characteristic of temporal logic specifications and the need to capture very long-range dependencies spanning hundreds of steps. Moreover, they typically process only past observations and lack any predictive capability regarding future system behavior, which limits proactive reasoning about temporal constraints. Furthermore, while effective in general settings, the reward mechanisms employed by these POMDP-RL methods are not specifically designed to decompose and address the multi-scale sparsity of feedback in temporal logic falsification, which spans immediate, phased, and global time scales.

### 2.3.3. Our DRL-BiT-MPR Framework

Our DRL-BiT-MPR framework is designed to overcome the combined limitations identified in both traditional and advanced methods. Unlike Bayesian optimization, which neglects temporal structure, our bidirectional temporal (BiT) network explicitly models both historical and predicted future information through a dedicated convolutional architecture. Compared to generic POMDP-RL methods that rely solely on historical observations via recurrent networks, the BiT network incorporates a pre-trained LSTM to predict future  $K$ -step outputs, thereby creating a richer, forward-looking temporal context for decision-making. This bidirectional design facilitates more accurate state estimation under partial observability. Furthermore, to directly combat the reward sparsity that hinders both classical and modern RL methods, our multi-granularity reward (MPR) function provides dense, structured feedback across multiple time scales. This multi-scale guidance is more effective for exploration than sparse binary rewards or single-scale reward shaping. The integration of the BiT network and the MPR function is a purposeful co-design rather than a modular assembly. They address the core challenges in a mutually reinforcing manner. The BiT network resolves partial observability by constructing an information-rich state estimate that incorporates predicted future context. This enriched state representation is a prerequisite for the MPR function to compute meaningful fine-grained and medium-grained rewards, which would be ill-defined over a poor state estimate. In turn, the dense, multi-scale feedback provided by the MPR function offers precise gradient directions that guide the policy network. This guidance directly shapes the exploration of input sequences, which subsequently influences the future observations and predictions processed by the BiT network, thereby closing the loop and enabling progressive refinement of the state estimation itself. By integrating these two complementary innovations, DRL-BiT-MPR offers a specialized solution tailored to the unique challenges of black-box CPS falsification under partial observability.

### 3. Preliminaries

#### 3.1. Discrete Time Deterministic Input–Output Systems

A discrete-time deterministic input–output system is a dynamic system [41] where time relies on a discrete time sequence (e.g.,  $t = 0, 1, 2, \dots$ ) and the output sequence is uniquely determined given the initial state and input sequence. Its core characteristic is that the input–output mapping is free of random interference, which can be formally defined and described using mathematical models.

Formally, let  $\mathcal{U} \subseteq \mathbb{R}^m$  denote the input space (e.g., control signals),  $\mathcal{X} \subseteq \mathbb{R}^n$  denote the state space [42], and  $\mathcal{Y} \subseteq \mathbb{R}^p$  denote the output space. For any discrete time step  $k \geq 0$ :  $u(k) \in \mathcal{U}$  is the input at time  $k$ ,  $x(k) \in \mathcal{X}$  is the system state at time  $k$ ,  $y(k) \in \mathcal{Y}$  is the output at time  $k$ ,  $F : \mathcal{U} \times \mathcal{X} \rightarrow \mathcal{Y}$  is a deterministic mapping function that describes the relationship between input, state, and output.

The system's input–output relationship follows the mathematical model:

$$y(k) = F(u(k), x(k)) \quad (1)$$

For the entire time sequence, the system evolves recursively. Given the initial state  $x(0)$  (at  $k = 0$ ), the input sequence  $\mathbf{u} = [u(0), u(1), \dots, u(N-1)]$  (length  $N$ ) generates the corresponding output sequence  $\mathbf{y} = [y(0), y(1), \dots, y(N)]$  by

$$y(k+1) = F(u(k), x(k)), \quad \forall k \in \{0, 1, \dots, N-1\} \quad (2)$$

Such systems are widely used in fields like digital control systems and time-series data processing, where deterministic input–output relationships are required for stable operation.

#### 3.2. Metric Temporal Logic

Metric temporal logic (MTL) is an extended form of linear temporal logic (LTL). Its core improvement lies in adding explicit time interval constraints to temporal operators [43], enabling accurate description of dynamic requirements time systems like satisfying a certain property within the time interval from  $t_1$  to  $t_2$  [44]. Its syntax includes propositional variables, Boolean connectives, and temporal operators with time constraints, while its semantics are based on the correspondence between time-point sequences and propositional truth values.

Formally, the syntax of MTL is defined recursively as follows:

$$\phi ::= p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \Box_{[a,b]}\phi \mid \Diamond_{[a,b]}\phi \quad (3)$$

where  $p \in \text{AP}$  denotes basic safety constraints;  $\neg, \wedge, \vee$  are Boolean connectives,  $\Box_{[a,b]}$  and  $\Diamond_{[a,b]}$  are temporal operators with time intervals  $[a, b] \subseteq \mathbb{R}_{\geq 0}$ . MTL serves as a crucial tool for modeling and verifying temporal properties.

For an output trace  $\mathbf{y} = [y_0, y_1, \dots, y_T]$  and time step  $t$ , the semantics of core temporal operators are:

$$\models_t \Box_{[a,b]}\phi \iff \forall t' \in [t+a, t+b], \models_{t'} \phi, \quad (4)$$

$$\models_t \Diamond_{[a,b]}\phi \iff \exists t' \in [t+a, t+b], \models_{t'} \phi, \quad (5)$$

where  $\models_t \phi$  means the trace  $\mathbf{y}$  satisfies formula  $\phi$  at time  $t$ .

These formal definitions provide the precise language for specifying temporal requirements in CPS. The always operator  $\Box_{[a,b]}\phi$  is used to encode safety properties, which require the sub-formula  $\phi$  to hold at every time instant within the specified interval  $[a, b]$ . Conversely, the eventually operator  $\Diamond_{[a,b]}\phi$  encodes reachability or liveness properties,

which require  $\phi$  to hold at least at one time instant within  $[a, b]$ . The explicit time bounds are essential for capturing the real-time constraints inherent to CPS dynamics.

The connection to the falsification task is direct, the goal is to discover an input sequence that produces an output trace  $\mathbf{y}$  violating a given MTL specification  $\phi$ . The quantitative degree of this satisfaction or violation, known as the robustness degree  $\rho(\phi, \mathbf{y})$ , is computed recursively based on the semantics defined above. This robustness degree subsequently serves as the foundational signal for constructing the multi-granularity reward function within the DRL-BiT-MPR framework, thereby translating the formal verification objective into a guidance mechanism for the learning agent.

### 3.3. Robustness

Robustness quantifies the degree to which an output trace satisfies an MTL formula, serving as the core optimization target for falsification; a negative robustness value indicates the trace violates the property [8,45].

For an atomic proposition  $p$  (with valid set  $D_p \subseteq \mathcal{Y}$ ), the robustness of output  $y_t$  (at time step  $t$ ) is defined using the Euclidean distance from  $y_t$  to  $D_p$ :

$$\rho(p, \mathbf{y}, t) = \text{Dist}(y_t, D_p) = \begin{cases} \inf\{\text{dist}(y_t, z) \mid z \in \mathcal{Y} \setminus D_p\} & \text{if } y_t \in D_p, \\ -\inf\{\text{dist}(y_t, z) \mid z \in D_p\} & \text{if } y_t \notin D_p, \end{cases} \quad (6)$$

where  $\text{dist}(x, D)$  denotes the Euclidean distance between two points. Positive values mean  $y_t$  satisfies  $p$  (larger values indicate more robust satisfaction), while negative values mean  $y_t$  violates  $p$  (more negative values indicate more severe violations [3]).

For composite MTL formulas, robustness is computed recursively to propagate the degree of satisfaction/violation across logical and temporal operators: negation:  $\rho(\neg\phi, \mathbf{y}, t) = -\rho(\phi, \mathbf{y}, t)$ ; conjunction:  $\rho(\phi \wedge \psi, \mathbf{y}, t) = \min\{\rho(\phi, \mathbf{y}, t), \rho(\psi, \mathbf{y}, t)\}$ ; always operator:  $\rho(\Box_{[a,b]}\phi, \mathbf{y}, t) = \min_{k \in [t+a, t+b]} \rho(\phi, \mathbf{y}, k)$ .

Robustness is computed recursively across logical and temporal operators, with formal definitions as follows:

$$\rho(\neg\phi, \mathbf{y}, t) = -\rho(\phi, \mathbf{y}, t), \quad (7)$$

$$\rho(\phi \wedge \psi, \mathbf{y}, t) = \min\{\rho(\phi, \mathbf{y}, t), \rho(\psi, \mathbf{y}, t)\}, \quad (8)$$

$$\rho(\phi \vee \psi, \mathbf{y}, t) = \max\{\rho(\phi, \mathbf{y}, t), \rho(\psi, \mathbf{y}, t)\}, \quad (9)$$

$$\rho(\Diamond_{[a,b]}\phi, \mathbf{y}, t) = \max_{k \in [t+a, t+b]} \rho(\phi, \mathbf{y}, k), \quad (10)$$

$$\rho(\Box_{[a,b]}\phi, \mathbf{y}, t) = \min_{k \in [t+a, t+b]} \rho(\phi, \mathbf{y}, k) \quad (11)$$

where Equations (7)–(9) handle Boolean connectives, ensuring robustness propagates the degree of satisfaction across logical combinations, Equations (10) and (11) handle temporal operators, with maxima/minima capturing the most satisfied/violated time step in the interval [46].

Falsification thus reduces to finding an input sequence that minimizes  $\rho(\phi, \mathbf{y}, 1)$ , the robustness of the entire trace at the initial time step.

The robustness degree  $\rho(\phi, \mathbf{y})$ —shorthand for  $\rho(\phi, \mathbf{y}, 1)$ —provides a continuous, quantitative measure of specification satisfaction. Unlike a Boolean true/false judgment, its real-valued output indicates not only whether a trace satisfies  $\phi$  (positive value) or violates it (negative value), but also the degree of that satisfaction or violation. This quantitative nature is key for falsification, as it transforms the discrete search for a violating trace into a continuous optimization problem: minimizing  $\rho(\phi, \mathbf{y})$ .

The recursive computation rules defined above (Equations (6)–(11)) are designed to propagate this quantitative signal through the formula’s structure in a semantically sound manner. For instance, the conjunction rule takes the minimum robustness, reflecting that a chain is only as strong as its weakest link; the always operator takes the minimum over an interval, capturing the worst-case deviation. Consequently, the final scalar output  $\rho(\phi, y)$  aggregates the system’s behavior across the entire trace and all sub-formulas into a single, differentiable guiding signal.

This scalar robustness signal directly serves as the foundational input for constructing the multi-granularity reward function in our DRL-BiT-MPR framework. Rather than providing sparse binary feedback, the framework decomposes and utilizes the temporal evolution of this signal to generate fine-grained, medium-grained, and coarse-grained reward signals that guide the agent’s exploration effectively.

### 3.4. Reinforcement Learning

Reinforcement learning (RL) is a machine learning paradigm where an agent learns an optimal decision-making policy through iterative interaction with an environment, aiming to maximize a cumulative reward signal [47]. Its core mechanism involves balancing exploration, making it suitable for solving sequential decision problems in dynamic systems [48–50].

In the context of system analysis, RL is formalized using a Markov decision process (MDP)  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ , where each component captures a key aspect of the agent–environment interaction:

State space  $\mathcal{S}$  is the set of all possible states the environment can be in. For black-box scenarios, each state  $s_t \in \mathcal{S}$  is defined to encode temporal information:  $s_t = [y_{t-L+1:t}, \hat{y}_{t+1:t+K}]$ , where  $y_{t-L+1:t}$  denotes the recent  $L$ -step output history (directly observable) and  $\hat{y}_{t+1:t+K}$  denotes the predicted  $K$ -step future output.

The values of  $L$  and  $K$  are determined by the dynamic characteristics of the target CPS:  $L$  is set to cover the “signal stabilization cycle” of observable outputs, ensuring historical dependencies are fully captured;  $K$  is determined by the maximum dynamic response delay of the CPS, ensuring future state uncertainty is compensated. For example, in the CARS model,  $L = 3$  and  $K = 2$ .

Action space  $\mathcal{A}$  is the set of all feasible actions the agent can execute. For control-oriented tasks,  $\mathcal{A} \subseteq \mathbb{R}^m$  is a discretized subset of the input space  $\mathcal{X}$ , where each action  $a_t \in \mathcal{A}$  corresponds to a specific control command.

Transition probability  $P$  is a function  $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$  that describes the probability of transitioning to a new state  $s_{t+1}$  after executing action  $a_t$  in state  $s_t$ . For deterministic systems,  $P(s_{t+1} | s_t, a_t) = 1$  if  $s_{t+1}$  is the unique next state induced by  $a_t$ , and 0 otherwise.

Reward function  $R$  is a function  $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$  that quantifies the immediate feedback for an action. To align with falsification goals, the reward is defined based on MTL robustness:  $R(s_t, a_t, s_{t+1}) = -\max(0, \rho(\phi, y, t+1))$ , where  $\rho(\phi, y, t+1)$  is the robustness of the output trace at time  $t+1$ . This design incentivizes the agent to reduce robustness, as higher rewards are granted for actions that bring the system closer to violating the property.

Discount factor  $\gamma \in (0, 1]$  is a scalar that weights future rewards relative to immediate ones. For finite-horizon tasks,  $\gamma = 1$  is used to prioritize near-term and long-term rewards equally.

The agent’s objective is to learn a policy  $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ —a mapping from states to action probability distributions—that maximizes the cumulative discounted reward from any initial state  $s_0$ :

$$G_t = \sum_{k=t}^{\infty} \gamma^{k-t} R(s_k, a_k, s_{k+1}) \quad (12)$$

where  $s_k$  and  $a_k$  are the state and action at time step  $k$ , respectively. To approximate this optimal policy, modern RL algorithms use function approximators to model either the policy directly or the value of taking actions in specific states, enabling efficient learning in high-dimensional state spaces like those encountered in black-box system analysis.

## 4. Proposed Approach

### 4.1. Problem Formulation

To lay a rigorous foundation for subsequent method design, this section first explicitly defines the black-box cyber-physical system (CPS) falsification problem and identifies its core challenges.

#### 4.1.1. Black-Box CPS and Safety Property Definition

A black-box CPS is formally characterized as a tuple  $\mathcal{S}_{\text{black}} = (\mathcal{X}, \mathcal{Y}, f_{\text{unknown}}, y_0)$ , where in each component is defined as follows:

$\mathcal{X} \subseteq \mathbb{R}^m$ : bounded input space, encompassing all feasible control signals (e.g., throttle or brake commands for automotive systems) that the agent can generate;  $\mathcal{Y} \subseteq \mathbb{R}^p$ : observable output space, which constitutes the sole accessible information in black-box scenarios. Internal states (e.g., engine torque) remain unobservable, while observable signals include metrics such as vehicle speed and control error,  $f_{\text{unknown}} : \mathcal{Y}^* \times \mathcal{X} \rightarrow \mathcal{Y}$ : unknown deterministic transition function. It maps the historical output sequence  $\mathbf{y}_{1:t} = [y_1, y_2, \dots, y_t]$  and current input  $x_t$  to the next output  $y_{t+1}$ ,  $y_0 \in \mathcal{Y}$ : initial output of the system, serving as the starting point for each experimental episode.

The target safety property, referred to as the finite future reach property, is defined using metric temporal logic (MTL) as  $\psi = \Box_{[0,T]} \varphi$ . The components of this formula are elaborated below:

$\Box_{[0,T]}$ : temporal operator denoting “always hold”, meaning the base formula  $\varphi$  must be satisfied at every time step within the interval  $[0, T]$ ;  $\varphi$ : basic MTL formula, composed of atomic propositions, Boolean connectives, and short-interval temporal operators;  $T$ : finite time horizon, corresponding to the termination time  $L$  used in subsequent algorithms.

#### 4.1.2. Falsification Problem Objective

The primary objective of black-box CPS falsification is to identify an input sequence  $\mathbf{x}^* = [x_0, x_1, \dots, x_{T-1}] \in \mathcal{X}^T$  such that the resulting output sequence  $\mathbf{y}^* = f_{\text{unknown}}(\mathbf{x}^*)$  violates the safety property  $\psi$ . Mathematically, this violation condition is equivalent to  $\rho(\psi, \mathbf{y}^*, 0) < 0$ .

Here,  $\rho(\cdot)$  represents the robustness value of the output trace with respect to the MTL property. If such an input sequence  $\mathbf{x}^*$  exists, it is termed a counterexample; otherwise, the safety property is considered satisfied within the time horizon  $T$ .

#### 4.1.3. Core Challenges in Black-Box Falsification

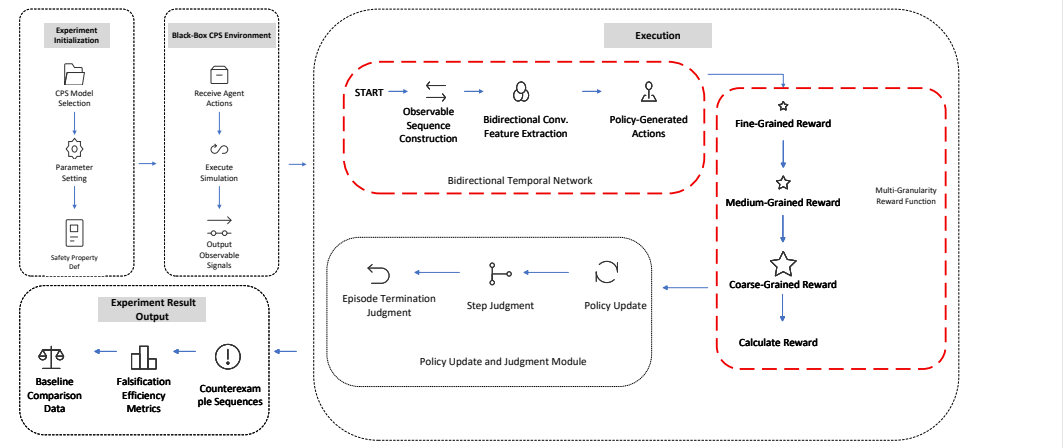
Existing deep reinforcement learning (DRL)-based falsification methods struggle to tackle two critical challenges inherent to black-box scenarios, which form the focus of this research.

**Inadequate temporal modeling due to unobservable states:** Black-box CPS only provides output signals (e.g., robustness values, sensor measurements), while hiding internal states such as vehicle acceleration. Traditional unidirectional temporal networks (e.g., single-layer LSTM) fail to capture bidirectional temporal dependencies between historical and future outputs, resulting in imprecise state inference.

**Sparse reward signals:** Conventional reward functions only offer feedback when the safety property is violated (i.e., robustness  $< 0$ ), leading to prolonged “zero-reward”

periods during the agent's exploration phase. This sparsity slows policy convergence and increases the risk of the agent getting trapped in local optima.

To address these dual challenges, we propose the DRL-BiT-MPR framework. Its overall architecture is illustrated in Figure 1. The framework integrates a bidirectional temporal network to remedy inadequate temporal modeling and a multi-granularity reward function to overcome reward sparsity, operating within a cohesive offline–online workflow.



**Figure 1.** Framework of DRL-BiT-MPR for black-box CPS falsification.

#### 4.2. Robustness Calculation

To quantify whether the output trace violates the safety property, we define the robustness calculation for MTL-based finite future reach safety properties  $\psi = \Box_{[0,T]}\phi$ , covering atomic propositions, Boolean connectives, and temporal operators.

##### 4.2.1. Robustness of Atomic Propositions

For an atomic proposition  $p \in AP$  (e.g., “vehicle speed  $v \leq 160$  km/h”), let  $D_p \subseteq \mathcal{Y}$  denote the set of outputs that satisfy  $p$ . The robustness of the output  $y_t$  at time step  $t$  is defined using Euclidean distance, as shown below:

$$\rho(p, y, t) = \begin{cases} \inf_{z \in \mathcal{Y} \setminus D_p} \text{dist}(y_t, z) & \text{if } y_t \in D_p, \\ -\inf_{z \in D_p} \text{dist}(y_t, z) & \text{if } y_t \notin D_p, \end{cases} \quad (13)$$

Here,  $\text{dist}(x, D)$  represents the Euclidean distance. A positive robustness value indicates that  $y_t$  satisfies  $p$  (with larger values indicating stronger robustness), while a negative value signifies a violation.

##### 4.2.2. Robustness of Boolean Connectives

For composite formulas constructed with Boolean operators, robustness propagates recursively according to the following rules: Negation ( $\neg\phi$ ): inverts the satisfaction or violation status:

$$\rho(\neg\phi, y, t) = -\rho(\phi, y, t) \quad (14)$$

Conjunction ( $\phi \wedge \psi$ ) is determined by the weakest sub-formula:

$$\rho(\phi \wedge \psi, y, t) = \min\{\rho(\phi, y, t), \rho(\psi, y, t)\} \quad (15)$$

Disjunction ( $\phi \vee \psi$ ) is determined by the strongest sub-formula:

$$\rho(\phi \vee \psi, y, t) = \max\{\rho(\phi, y, t), \rho(\psi, y, t)\} \quad (16)$$

#### 4.2.3. Robustness of Temporal Operators

For the core temporal operator  $\Box_{[0,T]}$  in the safety property  $\psi = \Box_{[0,T]} \varphi$ , the robustness calculation is given by

$$\rho(\Box_{[0,T]} \varphi, \mathbf{y}, 0) = \min_{t \in [0,T]} \rho(\varphi, \mathbf{y}, t) \quad (17)$$

This indicates that the robustness of the safety property is determined by the least robust time step within the horizon. If any time step violates  $\varphi$ , the entire safety property is violated.

For other temporal operators (e.g., the “eventually” operator  $\Diamond_{[a,b]} \varphi$ ) used in the base formula  $\varphi$ , the robustness is calculated as

$$\rho(\Diamond_{[a,b]} \varphi, \mathbf{y}, t) = \max_{t' \in [t+a, t+b]} \rho(\varphi, \mathbf{y}, t') \quad (18)$$

#### 4.3. Overall Framework and Phase Separation

The proposed DRL-BiT-MPR framework operates in two sequential phases. The first is an offline pretraining phase, followed by an online reinforcement learning phase. This division addresses practical considerations of data usage and computational efficiency within the constraints of black-box interaction.

The offline phase is performed once for each target system. Its purpose is to train an LSTM model to predict future observable outputs. This training uses a dataset of nominal input–output trajectories collected from the CPS. This phase does not interact with the robustness monitor or involve any policy learning. It produces a fixed prediction model that captures temporal patterns in the system’s observable dynamics.

The online phase executes the core falsification loop. It uses the pretrained predictor within a bidirectional temporal network to build a state representation that mitigates partial observability. A reinforcement learning agent then interacts with the black-box CPS. The agent is guided by a multi-granularity reward function and learns a policy to generate counterexamples. All learning in this phase is driven by data from online interaction.

This separation confines the use of offline simulation data to an initial model fitting step. The core falsification algorithm remains an online and adaptive search process. The following sections detail the components of each phase.

#### 4.4. Pretraining of the LSTM Prediction Module

The LSTM predictor is trained offline before the online falsification process begins. It is trained to predict future observable outputs based on historical sequences. The training uses a dataset of 10,000 unlabeled random input–output sequences collected from each benchmark model in offline simulation. The goal is to learn a model that maps the historical sequence of length  $L$  to the future  $K$  steps.

The input to the predictor is the historical observable signals, with the sequence length  $L$  specific to each model. The output is the future  $K$ -step observable signals  $\hat{\mathbf{y}}_{t+1:t+K}$ , where  $K$  is determined by the maximum dynamic response delay of the system.

For the CARS model, the response delay between throttle/brake input and inter-vehicle distance robustness change is two steps, so  $K = 2$ . The predictor outputs  $\hat{R}(t+1)$  and  $\hat{R}(t+2)$ .

For the AT model, engine speed and vehicle speed have a maximum response delay of three steps, so  $K = 3$ . The predictor outputs  $\hat{R}_\omega(t+1:t+3)$ ,  $\hat{R}_v(t+1:t+3)$ , and  $\hat{g}(t+1:t+3)$ .

For the PTC model, the convergence delay of control error  $\mu$  is two steps, so  $K = 2$ . The predictor outputs  $\hat{\mu}(t+1)$ ,  $\hat{\mu}(t+2)$  and the operating mode  $\hat{m}(t+1)$ ,  $\hat{m}(t+2)$ .

The predictor is trained by minimizing the mean squared error (MSE) between its predictions and the true future outputs from the offline simulation data:

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^K \|\hat{y}(t+k) - y_{\text{true}}(t+k)\|^2. \quad (19)$$

The trained LSTM model  $f_{\text{predict}}$  is the output of this offline phase. It is frozen and integrated into the bidirectional temporal network during the online phase, where it provides a data-driven projection of future states to enrich the agent's limited observation.

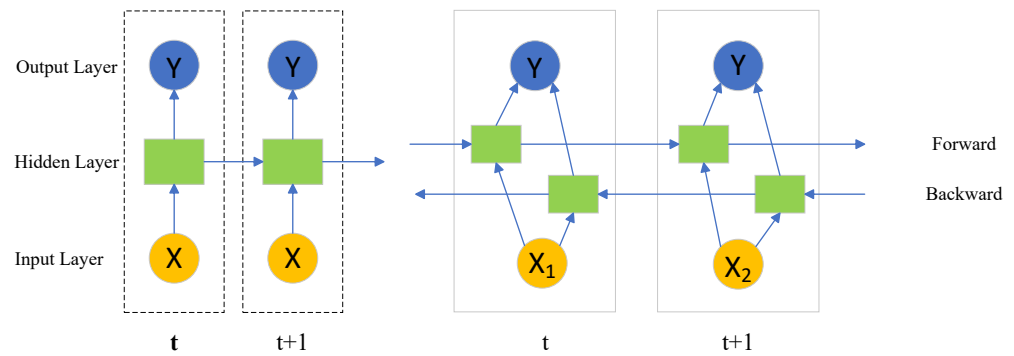
#### 4.5. Bidirectional Temporal Network

The bidirectional temporal (BiT) network is designed to address the state uncertainty inherent in partially observable black-box CPS falsification. In this setting, the agent must reason about and violate temporal logic constraints using only sequences of past observations. Conventional recurrent networks process only historical data, resulting in a state representation that is inherently retrospective. This limitation is critical because temporal logic properties require reasoning about future system evolution. The BiT network overcomes this by constructing an augmented state representation that integrates both a compressed encoding of past observations and a predicted trajectory of future outputs. This bidirectional context provides the agent with forward-looking information, which is essential for making effective control decisions under partial observability.

The adoption of the bidirectional temporal network as the core perception module is driven by the fundamental challenge of state uncertainty under partial observability in black-box CPS falsification. In such settings, the agent only has access to historical input-output sequences, while the internal system dynamics remain hidden. Conventional unidirectional recurrent networks are inherently limited to processing past data, constructing a state representation that is purely retrospective. This approach fails to capture the bidirectional temporal causality intrinsic to physical systems, where the current hidden state is not only a consequence of past inputs but also constrains the evolution of future outputs. To actively mitigate this uncertainty, the BiT network is designed to construct a more informed state estimate by integrating two complementary information streams: a compressed encoding of the past trajectory and a predicted future trajectory. This design enables the agent to disambiguate the current system context more effectively than from history alone, explicitly addressing partial observability and reducing reliance on a precise internal system model.

The bidirectional temporal network serves as the core module for the agent to perceive the black-box CPS environment and generate effective control inputs. Addressing the key challenge of only accessible output signals (with hidden internal states) in experiments, the network performs input sequence construction, bidirectional convolutional feature extraction, feature fusion, and action generation. By mining temporal correlations from limited observational data, it provides reliable support for policy decision-making.

The core perception module, the bidirectional temporal network, is designed to construct an informed state representation  $s_t$  by integrating both past observations and predicted future information. Its architecture and the flow of data through its components are detailed in Figure 2. The network operates through a sequence of stages: it begins with the formulation of an input sequence from historical outputs, proceeds to generate future state predictions via a pre-trained LSTM module, and then processes the combined temporal context through parallel forward and backward convolutional pathways. The outputs of these pathways are subsequently fused to form a unified feature vector that is passed to the policy network for action generation.



**Figure 2.** Bidirectional temporal network framework.

#### 4.5.1. Input Sequence Definition for Benchmark Models

Input sequences are customized for different benchmark CPS models based on their observable output characteristics, ensuring effective capture of temporal dynamics:

**CARS model:** The only observable output is the robustness value. The input sequence is defined as  $[R(t-2), R(t-1), R(t)]$ , where  $R(t)$  denotes the robustness value of the CPS safety property at time step  $t$ .

**AT model:** Observable outputs include engine speed robustness, vehicle speed robustness, and gear state. The input sequence is  $[R_\omega(t-2), R_\omega(t-1), R_\omega(t), R_v(t-1), R_v(t), g(t-1), g(t)]$ , where  $R_\omega$  is engine speed robustness,  $R_v$  is vehicle speed robustness, and  $g$  represents gear state (ranging from  $g_1$  to  $g_4$ ).

**PTC model:** Observable outputs consist of control error  $\mu$  and operating mode. The input sequence combines a 4-step control error sequence  $[\mu(t-3), \mu(t-2), \mu(t-1), \mu(t)]$  and a 2-step operating mode sequence  $[m(t-1), m(t)]$ , forming a 6-dimensional input vector.

#### 4.5.2. Future Output Prediction for Input Sequences

To enable the backward convolution to capture future temporal constraints, a pre-trained LSTM predictor is integrated into the BiT network. This predictor generates future observable signals to complement the historical input sequence, addressing the unobservability of future states in black-box CPS. Prior to deployment, the LSTM predictor is trained using 10,000 unlabeled random input–output sequences (offline simulation data) from each benchmark model. Key design details are as follows.

Input is the historical observable signals, with the sequence length consistent with the  $L$  value of each model. Output is the future  $K$  step observable signals ( $\hat{y}_{t+1:t+K}$ ), where  $K$  is determined by the maximum dynamic response delay of each model.

**CARS model:** The response delay between throttle/brake input and inter-vehicle distance robustness change is 2 steps, so  $K = 2$  (predicting  $\hat{R}(t+1), \hat{R}(t+2)$ ).

**AT model:** Engine speed and vehicle speed have a maximum response delay of 3 steps, so  $K = 3$  (predicting  $\hat{R}_\omega(t+1:t+3), \hat{R}_v(t+1:t+3), \hat{g}(t+1:t+3)$ ).

**PTC model:** The convergence delay of control error  $\mu$  is 2 steps, so  $K = 2$  (predicting  $\hat{\mu}(t+1), \hat{\mu}(t+2)$  and operating mode  $\hat{m}(t+1), \hat{m}(t+2)$ ).

Mean squared error (MSE) between the predicted value  $\hat{y}$  and the true value  $y_{\text{true}}$  from offline simulations:

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^K \|\hat{y}(t+k) - y_{\text{true}}(t+k)\|^2 \quad (20)$$

During real-time operation, the LSTM predictor outputs future  $K$ -step signals. These signals are concatenated with the historical sequence to form the complete bidirectional input sequence for subsequent convolution operations.

A precise understanding of model dependency in this context is necessary. The framework aims to reduce reliance on a precise internal system model, such as governing equations or state variables, which are unavailable in black-box settings. The pre-trained LSTM predictor employed here is not such an internal model. It functions as a general-purpose temporal sequence learner, trained on input–output pairs to capture statistical patterns of correlation over time. Its purpose is to provide plausible future signal trends based on recent history, not to replicate the underlying system dynamics. The requirement for offline simulation data for training is a common foundation for data-driven methods. Crucially, the predictor operates without accessing or approximating the system’s internal equations. Therefore, the framework maintains adherence to the black-box assumption by utilizing only observable data and learned temporal correlations.

This complete sequence  $[y_{t-L+1:t}, \hat{y}_{t+1:t+K}]$  forms the augmented state representation  $s_t^e$ . It is this representation, explicitly enriched with predicted future information, that is passed to the policy network for action generation, thereby closing the loop between perception and control.

#### 4.5.3. Bidirectional Convolution and Feature Fusion

Bidirectional convolution is applied to the complete input sequence to process temporal correlations in parallel, overcoming the limitations of unidirectional temporal modeling. The specific operation process is detailed below:

Forward convolution takes the ordered complete sequence  $[y_{t-L+1}, \dots, y_t, \hat{y}_{t+1}, \dots, \hat{y}_{t+K}]$  as input. A  $3 \times 1$  convolution kernel with a stride of 1 is used, and the ReLU function serves as the activation function. This module learns causal dependencies between historical observations and current/future outputs. Backward convolution first reverses the complete sequence to  $[\hat{y}_{t+K}, \dots, \hat{y}_{t+1}, y_t, \dots, y_{t-L+1}]$ , then applies the same  $3 \times 1$  kernel and ReLU activation as the forward convolution. This module captures the constraints of future observations on current decision-making.

After convolution, the forward feature map  $F_{\text{forward}}$  (dimension:  $64 \times 1$ , using 64 convolution filters) and backward feature map  $F_{\text{backward}}$  (same dimension as  $F_{\text{forward}}$ ) are concatenated along the feature dimension to form a fused feature vector  $F_{\text{fused}} = [F_{\text{forward}}, F_{\text{backward}}]$ . This vector integrates bidirectional temporal information and is fed into the policy network to generate the final CPS control input.

#### 4.5.4. BiT Network Parameter Selection Basis

Key parameters of the BiT network, including historical sequence length  $L$  and convolution kernel size  $K_{\text{conv}}$ , are determined based on the dynamic characteristics of each benchmark CPS model. This ensures effective mining of temporal correlations while avoiding redundant information.

Historical sequence length is set to cover the minimum dynamic cycle of the model’s observable signals. For the CARS model, inter-vehicle distance robustness  $R(t)$  stabilizes 3 steps after input adjustment, so  $L = 3$ . For the AT model, engine speed robustness  $R_\omega(t)$  (3 steps to stabilize) is the slowest among multi-signals, so  $L = 3$  for  $R_\omega$ ; 2-step  $R_v(t)$  and  $g(t)$  are added to balance other signals’ dynamics. For the PTC model, control error  $\mu$  stabilizes 4 steps after input adjustment, so  $L = 4$  for  $\mu$ ; 2-step operating mode is added to capture discrete state changes.

Convolution kernel size  $K_{\text{conv}}$  is a unified  $3 \times 1$  kernel is adopted for two reasons. The time dimension of 3 matches the “cause→effect” chain of CPS dynamics, fully capturing

ing causal dependencies. A smaller kernel misses intermediate dynamic links, while a larger kernel introduces redundancy and increases computational complexity; the feature dimension of 1 avoids cross-dimension interference between multi-signals.

The enriched state representation  $s_t^e$  addresses the theoretical requirement for Markovian state inputs within the partially observable black-box CPS environment. In a standard Markov decision process, the state must encapsulate all relevant historical information for optimal decision-making. The raw observation  $o_t$  under partial observability violates this Markov property. The proposed representation  $s_t^e = [H_t; F_t]$  is explicitly designed as an information state. It integrates a compressed history  $H_t$  with a predicted future trajectory  $F_t$  to form a sufficient statistic of the interaction history. This construction recovers an approximate Markov property within the learning framework, enabling the problem to be treated as an MDP with  $s_t^e$  as the effective state input. Consequently, the application of standard policy gradient methods is theoretically justified.

#### 4.5.5. Prediction Error Analysis and Mitigation Strategy

The use of predicted future outputs within the state representation warrants conceptual justification. In a strict online black-box setting, true future information is indeed inaccessible. The predictor is not intended to circumvent this fundamental constraint. Instead, it serves as an inductive bias or an internal simulation module that, based on learned temporal patterns from historical data, generates plausible hypotheses about immediate future trajectories. This provides a richer, forward-looking context that aids in disambiguating the current hidden state under partial observability, a function analogous to planning or foresight in biological agents. The optimality of the resulting policy is therefore contingent upon the accuracy of these predictions. The following analysis formally addresses the impact of prediction error and introduces mechanisms to mitigate its effects, ensuring robust falsification performance even when predictions are imperfect.

A critical methodological concern for any learning-based falsification approach that relies on predicted future information is its robustness to prediction inaccuracies. To directly address this concern and ensure the reliability of our framework, this subsection formally analyzes the impact of prediction errors and introduces an adaptive mitigation strategy. The goal is to safeguard the falsification performance of the DRL-BiT-MPR framework even when the state predictions are imperfect.

The BiT network relies on a pre-trained LSTM to predict future  $K$ -step outputs for complementing temporal information. However, prediction errors may degrade state perception accuracy. This section quantifies the impact of prediction errors on falsification performance and proposes a dynamic weight adjustment mechanism combined with an error compensation strategy to ensure the BiT network maintains optimal performance under prediction biases.

Prediction error is defined with the LSTM-predicted outputs denoted as  $\hat{y}_{t+k}$  for  $k = 1, 2, \dots, K$  and the true system outputs as  $y_{\text{true}}(t+k)$ . The mean squared error (MSE) of prediction error is formulated as

$$\mathcal{E} = \frac{1}{K} \sum_{k=1}^K \|\hat{y}_{t+k} - y_{\text{true}}(t+k)\|^2 \quad (21)$$

Prediction errors affect BiT network performance through two pathways. Biases in future features  $\hat{y}_{t+1:t+K}$  lead to inaccurate bidirectional convolutional feature extraction. Incorrect future context disrupts temporal dependency modeling, especially in the falsification of complex properties such as nested MTL constraints.

To mitigate the negative impacts of prediction errors, a two-stage mitigation strategy is proposed. First is the dynamic weight adjustment mechanism. It adaptively adjusts the

fusion weight of historical and future features based on the real-time prediction error  $\mathcal{E}$ . The weight formula is

$$\omega_t = 1 - \lambda \cdot \mathcal{E}_t \quad (22)$$

where  $\lambda = 10$  determined via grid search to balance sensitivity and stability and  $\mathcal{E}_t$  is the prediction error at the current time step. The feature fusion method is updated to

$$F_{\text{fused}} = \omega_t \cdot F_{\text{historical}} + (1 - \omega_t) \cdot F_{\text{predicted}} \quad (23)$$

Here,  $F_{\text{historical}}$  denotes features extracted from historical sequences via forward convolution, and  $F_{\text{predicted}}$  denotes features extracted from future predicted sequences via backward convolution. As  $\mathcal{E}_t$  increases,  $\omega_t$  decreases, and the network automatically reduces reliance on predicted features.

Second is the embedding of the error compensation term. The prediction error sequence is treated as an additional feature and concatenated to the input sequence of the BiT network, enabling the network to learn error patterns and adaptively correct them. The updated input sequence is

$$s_t = [y_{t-L+1:t}, \hat{y}_{t+1:t+K}, \Delta\hat{y}_{t-K+1:t}] \quad (24)$$

where  $\Delta\hat{y}_{t-k} = \hat{y}_{t-k} - y_{\text{true}}(t-k)$  for  $k = 1, 2, \dots, K$  is the sequence of prediction errors over the last  $K$  steps, supplementing temporal correlation information of errors.

In summary, the proposed mitigation strategies ensure that the framework does not rely on perfect predictions. The dynamic weight adjustment allows the agent to automatically discount unreliable predictions, while the error compensation term enables learning of systematic biases. The predicted future information is thus utilized as an informative temporal feature rather than a ground-truth signal. The augmented state  $s_t^e$  retains its superiority over a purely historical state because it provides a richer, if imperfect, context for decision-making, as evidenced by the performance gains in our ablation studies.

#### 4.6. Multi-Granularity Reward Function

To address the issues of insufficient single-step reward feedback and sparse signals in black-box environments, this research decomposes the reward function into three levels: fine-grained, medium-grained, and coarse-grained mechanisms. This design ensures the agent receives feedback across different time scales—immediate step-by-step guidance, mid-term temporal correlation feedback, and long-term goal feedback—thereby avoiding local optima and accelerating policy convergence.

##### 4.6.1. Fine-Grained Reward

Operating at the single-step scale, the fine-grained reward provides immediate robustness feedback. It takes the current time step's robustness value as the core and incorporates a temporal-difference correction term to strengthen feedback on robustness changes between adjacent steps. The formula is defined as

$$\text{reward}(t) = [\exp(-R(t)) - 1] + \lambda \cdot [R(t) - R(t-1)] \quad (25)$$

In this formula,  $\exp(-R(t)) - 1$  implies that a smaller robustness value leads to a larger reward.  $\lambda$  is the temporal-difference weight coefficient, set to 0.2 in experiments.  $R(t) - R(t-1)$  represents the robustness change between consecutive steps: if  $R(t) > R(t-1)$ , the temporal-difference correction term becomes negative, reducing the total reward and prompting the agent to further optimize the input; if  $R(t) < 0$ ,  $\exp(-R(t))$

increases significantly, resulting in a substantial boost to the total reward and thus reinforcing the learning of counterexample inputs.

#### 4.6.2. Medium-Grained Reward

Operating at the window scale, the medium-grained reward provides feedback on temporal correlations. Using a sliding event window (window size set to  $K = 5$  based on experimental validation) as the calculation unit, it is computed every  $k = 3$  time steps based on the cumulative robustness decrease within the window. This ensures the agent captures mid-term temporal trends in robustness.

#### 4.6.3. Coarse-Grained Reward

Operating at the episode scale, the coarse-grained reward provides long-term goal feedback. Calculated at the end of each episode (when the time step reaches  $T$  or a counterexample is found), it is determined by the minimum robustness value throughout the episode and the proximity to a counterexample, guiding the agent toward the ultimate falsification goal.

#### 4.6.4. Total Reward Calculation

The total reward is a weighted sum of the three levels of rewards. A weight of 0.5 is assigned to the fine-grained reward to prioritize immediate feedback. The medium-grained and coarse-grained rewards are assigned weights of 0.3 and 0.2, respectively, to complement mid-term correlation capture and long-term objective guidance, thus preventing the agent from falling into local optima.

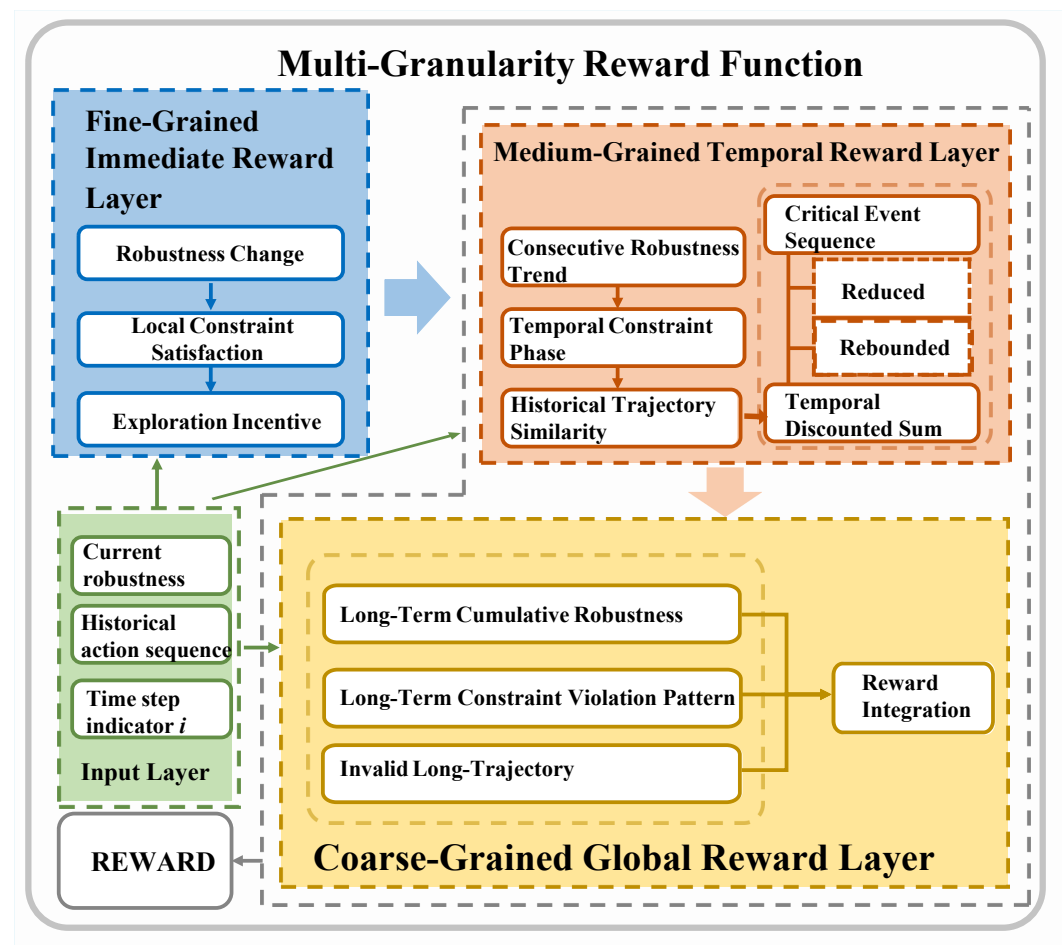
The hierarchical interplay of these three reward levels and their integration into the overall learning loop is illustrated in Figure 3. The diagram visualizes how fine-grained rewards provide per-step feedback, medium-grained rewards operate over a sliding window, and coarse-grained rewards assess the complete episode, ultimately converging into a single scalar reward that guides policy updates.

#### 4.6.5. Design Rationale and Justification

The hierarchical architecture of the multi-granularity reward function is a deliberate design response to the core challenges of signal sparsity and inefficient exploration in black-box falsification. Each tier of the reward mechanism addresses a distinct facet of the learning problem, and their integrated operation provides structured guidance across multiple temporal scales.

The fine-grained reward component directly counteracts the problem of sparse feedback by supplying a dense, stepwise learning signal. It operates on the robustness degree at each time step, ensuring that every action receives evaluative feedback. The exponential term establishes a continuous mapping between robustness values and rewards, while the temporal-difference component provides immediate directional cues by incentivizing reductions in robustness. This transforms a static performance metric into a dynamic gradient for policy optimization.

Medium-grained rewards address the challenge of policy stagnation in local plateaus. By evaluating performance over a sliding temporal window and rewarding the maximum robustness improvement observed within that horizon, this component introduces a medium-term perspective. It encourages the agent to develop strategies that yield sustained progress over sequences of actions, which is critical for falsifying properties with extended temporal dependencies.



**Figure 3.** Multi granularity reward function framework.

The coarse-grained reward ensures alignment with the global falsification objective. Calculated upon episode termination and based on the overall trace robustness, this component provides a stable, long-term signal that grounds the entire exploration process. It mitigates potential misdirection from transient fluctuations in the finer-grained rewards and consistently reinforces the ultimate goal of identifying a violating trace.

The synthesis of these components through a weighted sum creates a cohesive feedback system. This multi-scale architecture effectively converts the sparse, binary outcome of traditional temporal logic falsification into a rich and continuous learning signal. The resultant guidance is instrumental in achieving the demonstrated improvements in both sample efficiency and falsification success rate.

The three-layer granularity structure is a principled decomposition of the long-horizon falsification task into discrete temporal scales. Fine, medium, and coarse grains correspond to the immediate stepwise, the phased multi-step, and the global episodic temporal scales inherent to sequential decision-making under temporal logic constraints. This tripartite structure provides necessary and sufficient coverage of the feedback spectrum. It delivers dense guidance for local optimization, counters policy stagnation in intermediate phases, and maintains alignment with the terminal objective. Introducing additional layers would not yield commensurate benefits but would increase computational and tuning complexity.

The layer count is a fixed architectural feature of the framework, derived from the fundamental temporal hierarchy described above. Generalization across diverse CPS models or specifications is achieved by scaling the temporal parameters within each layer, not by altering the number of layers. Specifically, the window size for medium-grained evaluation and the episode horizon for coarse-grained assessment are calibrated according to the

characteristic time constants and the dominant temporal operators of the specific system-property pair. This ensures the framework's adaptability while preserving a consistent and interpretable reward topology.

#### 4.6.6. Parameter Selection and Evaluation

The parameters within the reward function, including the temporal-difference coefficient  $\lambda$  in Equation (25) and the layer weights for the multi-granularity reward, are determined through a combination of principled design and empirical validation.

The coefficient  $\lambda = 0.2$  in the fine-grained reward balances the influence of the absolute robustness value and its temporal derivative. A value of zero would ignore the direction of change, while a value too large could destabilize learning by overemphasizing single-step fluctuations. The chosen value was found to provide stable and consistent learning progress across all benchmark models during preliminary empirical studies.

The weights for the multi-granularity reward summation are set to  $\alpha_{\text{fine}} = 0.5$ ,  $\alpha_{\text{medium}} = 0.3$ , and  $\alpha_{\text{coarse}} = 0.2$ . This distribution reflects a hierarchical prioritization where immediate, stepwise feedback is most critical for guiding local search, followed by medium-term trend evaluation to escape plateaus, with global episodic guidance providing foundational direction. These weights were not subjected to an exhaustive grid search to avoid overfitting to a specific model or property. Instead, they were established based on their conceptual alignment with the respective importance of each time scale and then validated by observing consistent performance improvements across all three diverse CPS benchmarks (CARS, AT, PTC). The robustness of the results to minor variations in these weights confirms that the framework is not overly sensitive to their precise values, provided the hierarchical relationship  $\alpha_{\text{fine}} > \alpha_{\text{medium}} > \alpha_{\text{coarse}}$  is maintained.

#### 4.7. Parameter Selection Methodology

The selection of hyperparameters within the DRL-BiT-MPR framework follows a systematic methodology grounded in the physical dynamics of cyber-physical systems, rather than arbitrary per-model empirical tuning. This principled approach ensures both scalability across system complexities and transferability across application domains.

##### 4.7.1. Unified Selection Principles

Parameter selection is governed by three interconnected principles. First, time constant alignment ensures that the historical sequence length  $L$  and prediction horizon  $K$  correspond to measurable system temporal characteristics. Specifically,  $L$  must encompass the dominant stabilization period of the observed signals, while  $K$  should match the inherent response delay between control inputs and their observable effects on system outputs.

Second, multi-scale balance dictates the distribution of reward weights in the MPR function. The fine, medium, and coarse granularity rewards are allocated to provide immediate search guidance, maintain exploration momentum, and ensure alignment with the long-term falsification objective, respectively. This structured reward shaping addresses the exploration-exploitation trade-off inherent in temporal logic falsification tasks.

Third, robustness by design guides the selection of parameter values that reside within flat regions of the performance landscape, where minor deviations induce minimal degradation in falsification success. This principle prioritizes stable performance over fragile optimality, a consideration validated through the systematic sensitivity analysis presented in Section 5.9.

#### 4.7.2. Application to Case Studies

These unified principles are consistently applied across the three distinct cyber-physical system case studies examined in this work. For the CARS model, analysis of inter-vehicle distance dynamics indicates stabilization within three simulation steps, leading to the selection  $L = 3$ . The observed two-step delay between throttle or brake inputs and corresponding changes in distance robustness justifies  $K = 2$ .

In the AT model, a differential analysis of signal dynamics is employed. Engine speed  $\omega$  requires three steps to stabilize, motivating  $L = 3$  for this signal, while vehicle speed  $v$  and gear state  $g$  exhibit faster dynamics, leading to  $L = 2$  for these components. The maximum system response delay of three steps determines  $K = 3$ .

For the PTC model, the control error  $\mu$  demonstrates a four-step stabilization period, resulting in  $L = 4$ , whereas the operating mode  $m$  evolves more slowly, warranting  $L = 2$ . A two-step convergence delay for the control error underpins the selection  $K = 2$ .

The consistent application of these dynamical principles across models characterized by differing continuous, hybrid, and temporal behaviors confirms that the parameter selection methodology is not an artifact of model-specific tuning. Instead, it reflects a generalizable approach based on fundamental CPS characteristics, supporting the method's potential for broader application.

#### 4.8. Analysis of Method Properties

This section presents a systematic analysis of three fundamental properties of the proposed DRL-BiT-MPR framework: convergence behavior, soundness guarantee, and completeness consideration. These properties are essential for evaluating the reliability and applicability of any falsification method in safety-critical cyber-physical systems.

##### 4.8.1. Convergence Analysis

The proposed DRL-BiT-MPR framework employs deep reinforcement learning in a black-box environment, which precludes the provision of formal mathematical convergence guarantees. Nevertheless, we examine its convergence characteristics through architectural design choices and empirical observations.

First, the multi-granularity reward function provides dense-shaped rewards rather than sparse binary signals. This reward design offers continuous learning guidance throughout the exploration process, which has been established in the literature to facilitate more efficient policy gradient optimization.

Second, the bidirectional temporal network architecture maintains gradient flow by incorporating both historical observations and predicted future states. This design mitigates the vanishing gradient problem that commonly impedes training in long-horizon temporal tasks, thereby promoting more stable learning dynamics.

Third, empirical evidence supports the practical convergence of our method. The experimental results demonstrate consistent performance improvement across training episodes. Our method typically achieves stable performance within 100 to 150 episodes, representing faster and more reliable convergence compared to both traditional DRL baselines and the advanced PPO-LSTM agent.

##### 4.8.2. Soundness Guarantee

Soundness represents the assurance that any reported counterexample constitutes a genuine violation of the specified property. This property is paramount for safety-critical applications where false positives could lead to erroneous conclusions.

Our method ensures operational soundness through a rigorous verification mechanism. Every candidate counterexample generated by the policy undergoes validation through

re-simulation of the cyber-physical system model. The robustness value is computed using the exact metric temporal logic semantics defined in Section 3. The falsification process terminates and reports a counterexample only when the computed robustness value is strictly negative. This verification step is integral to Algorithm 1 and guarantees that no false positives are reported.

---

**Algorithm 1** Falsification for  $\psi$  by Reinforcement Learning

---

**Note:** The algorithm begins after the offline phase (Section 4.4) has provided the pre-trained LSTM predictor  $f_{\text{predict}}$ . The online phase described here utilizes this fixed model.

**Require:** A finite future reach safety property  $\psi$ , its monitoring formula  $\phi$ , a system  $f$ , an agent  $a$

**Ensure:** A counterexample input signal  $x$  if exists

```

1: Parameters: The end time  $L$ , the maximum number of episodes  $N$ 
2: for numEpisode  $\leftarrow 1$  to  $N$  do
3:    $i \leftarrow 0, y_0 \leftarrow$  the initial (output) state of  $f$ 
4:    $r \leftarrow \text{reward}(y_0, \phi, 0), y \leftarrow \text{append}(y, y_0)$ 
5:    $x \leftarrow$  the empty input sequence
6:   while  $i < L$  do
7:      $x_i \leftarrow a.\text{step}(y_i, r, \text{update}()), x \leftarrow \text{append}(x, x_i)$ 
8:      $y_{i+1} \leftarrow f(x_i), y \leftarrow \text{append}(y, y_{i+1})$ 
9:      $r \leftarrow \text{reward}(y, \phi, i + 1)$ 
10:     $i \leftarrow i + 1$ 
11:    if  $y \not\models \psi$  then
12:      return  $x$  as a falsifying input
13:    end if
14:  end while
15:   $a.\text{reset}(y_0, r)$ 
16: end for
```

---

Empirical validation across all experimental runs confirms this soundness guarantee. One hundred percent of reported counterexamples exhibited robustness values below negative 0.01, demonstrating the absence of erroneous violations in our experimental evaluation.

#### 4.8.3. Completeness Considerations

For black-box cyber-physical systems with continuous or hybrid dynamics, achieving formal completeness defined as guaranteed discovery of any existing counterexample is computationally intractable. Our method instead adopts the well-established concept of probabilistic completeness from the sampling-based falsification literature.

The theoretical foundation of this approach states that, given a sufficient sampling budget, meaning the number of episodes approaches infinity, the probability of discovering an existing counterexample approaches unity. This represents the standard completeness notion for sampling-based methods.

Our method enhances this probabilistic completeness through two mechanisms. The multi-granularity reward function naturally guides exploration toward regions of low robustness where counterexamples are more likely to reside. Simultaneously, the bidirectional temporal network focuses the search on temporally plausible input sequences, avoiding wasted exploration on behaviorally impossible patterns.

Empirical performance metrics substantiate this enhanced completeness. The consistently superior success rates and reduced sample counts documented in the experimental evaluation demonstrate that our method achieves higher falsification efficiency compared to all baseline methods. This performance gain indicates more efficient coverage of the

counterexample space and consequently a higher probability of discovery for any fixed sampling budget.

#### 4.8.4. Comparative Analysis

Table 1 provides a comparative summary of how different falsification approaches address these key properties. The comparison elucidates the fundamental trade-off between formal guarantees and practical applicability that characterizes contemporary falsification methodologies.

**Table 1.** Comparison of falsification method properties.

| Method                    | Convergence           | Soundness          | Completeness                   |
|---------------------------|-----------------------|--------------------|--------------------------------|
| Formal Verification       | Guaranteed            | Formal guarantee   | Formal guarantee               |
| Bayesian Optimization     | Local optimum         | Operational        | Probabilistic                  |
| PPO-LSTM Baseline         | Empirical             | Operational        | Probabilistic                  |
| <b>Ours (DRL-BiT-MPR)</b> | <b>Fast empirical</b> | <b>Operational</b> | <b>Efficient probabilistic</b> |

The analysis reveals a clear methodological positioning. While the proposed framework sacrifices the formal guarantees available to white-box verification methods, it gains the capability to handle complex black-box cyber-physical system models that lie beyond the reach of formal verification techniques. This trade-off represents not merely a practical necessity but a well-justified methodological choice for real-world cyber-physical system falsification, where system complexity often precludes formal analysis while rigorous safety validation remains imperative.

#### 4.9. Algorithm Overview

The DRL-BiT-MPR falsification process follows a two-phase design. After the offline phase (Section 4.4) provides the pretrained LSTM predictor  $f_{\text{predict}}$ , the online phase executes the reinforcement learning algorithm described here to find a counterexample for the finite future reach safety property  $\psi$ .

Algorithm 1 defines the core inputs: the safety property  $\psi$ , its monitoring formula  $\phi$ , the system under test  $f$ , and the RL agent  $a$ . It also uses two key parameters: the per-episode time step limit  $L$  and the maximum episode count  $N$ . The agent  $a$  incorporates the bidirectional temporal network, which utilizes  $f_{\text{predict}}$  to construct its state representation.

The algorithm executes  $N$  episodes. Each episode initializes the time step counter  $i$  to 0, sets  $y_0$  as the system's initial output state, calculates the initial reward, and initializes empty input and state sequences  $x$  and  $y$ .

An inner loop runs while  $i$  stays below  $L$ . In each iteration, the agent selects input  $x_i$  based on its current state and the previous reward, feeds  $x_i$  into  $f$  to generate the next output  $y_{i+1}$ , appends it to  $y$ , updates the reward per Section 4.6, and increments  $i$ . The agent's state is built using the BiT network, which integrates historical outputs and predictions from  $f_{\text{predict}}$ . Any violation of  $\psi$  by the output trace  $y$  prompts the algorithm to return  $x$  as the falsifying input and terminate immediately.

If no violations are detected, the agent resets for the next episode. The algorithm concludes after all  $N$  episodes. Finding no counterexamples suggests the system likely satisfies  $\psi$  within the explored input space and time horizon.

#### 4.10. Comparative Analysis with Conventional DRL Methods

Having detailed the components of the DRL-BiT-MPR framework, we now elucidate the fundamental distinctions between our approach and conventional DRL methods in the

context of CPS robustness falsification. These distinctions are designed to address the core limitations outlined in Section 1.

In terms of temporal state representation, conventional DRL falsifiers typically employ unidirectional recurrent networks to encode a history of observations. This results in a latent state representation that is inherently past-dependent and may be insufficient under partial observability. In contrast, our bidirectional temporal (BiT) network actively constructs a state representation by fusing encoded historical observations with predicted future outputs. This design explicitly models the bidirectional temporal dependencies characteristic of CPS, thereby achieving a more informed and accurate state estimation without requiring access to internal system dynamics.

Regarding the reward mechanism, a major bottleneck for conventional DRL is the sparse, often binary reward signal based solely on the final specification violation. This sparse feedback leads to inefficient exploration. Our multi-granularity reward (MPR) function is specifically engineered to overcome this by providing structured, dense feedback across three distinct time scales. The continuous guidance from step-level to phase-level rewards dramatically improves exploration efficiency, directly tackling the reward sparsity problem that plagues black-box falsification.

Concerning model dependency, many advanced DRL methods rely on system models for reward shaping or belief state updates. In contrast, both the BiT network, which learns from input–output sequences, and the MPR function, defined on observable robustness, operate without internal model knowledge. This design choice enhances the framework’s applicability to genuine black-box scenarios.

These targeted innovations are expected to translate into superior empirical performance. Specifically, more accurate state-aware policies are anticipated to yield higher falsification success rates, while the guided exploration is expected to significantly reduce the number of required simulations, thereby improving sample efficiency. The experimental results presented in the following section quantitatively validate these expectations.

## 5. Experiments

### 5.1. Experimental Questions

The experimental questions are formulated as follows:

- Does the proposed method improve the ability of reinforcement learning to solve the problem compared to heuristic methods as the baseline?
- Is the performance improvement of the proposed method primarily attributed to the bidirectional temporal network or the multi-granularity reward function?
- How sensitive are the parameters of the multi-granularity reward function, and can these parameters be set automatically rather than manually?

### 5.2. Experimental Models

All experiments use three benchmark CPS models consistent with the target document, which are widely adopted in CPS falsification research to evaluate method effectiveness.

**CARS Vehicle Platoon Model:** Consists of 5 vehicles, where the lead vehicle is controlled by throttle and brake inputs, and the remaining 4 vehicles operate autonomously based on a predefined state chart. The system output is the position of each vehicle, and the model is used to test temporal safety properties related to inter-vehicle distance constraints.

**AT Automatic Transmission Controller Model:** A hybrid system integrating continuous dynamics and discrete control logic. Inputs include throttle opening and brake pressure; outputs include engine RPM, current gear, and vehicle speed. The model is used to test properties involving both continuous state constraints and discrete logic.

PTC Powertrain Controller Model: A controller for the air-fuel (A/F) ratio of an internal combustion engine. Inputs are accelerator pedal angle and engine speed; outputs are control error and operating mode. The system dynamics converge quickly, requiring long-term temporal property design to avoid missing counterexamples.

### 5.3. Implementation

#### 5.3.1. Experimental Configuration

Our experimental environment is built on a Lenovo Legion Y9000P 2023 laptop equipped with an Intel Core i7-13700H processor, an NVIDIA GeForce RTX 4060 graphics card with 8 GB GDDR6 memory, 32 GB DDR5-4800 dual-channel RAM, a 1 TB PCIe 4.0 NVMe SSD, and the Windows 11 Professional 22H2 operating system. The software framework centers on Matlab R2022a/Simulink, which hosts the CARS, AT, and PTC system models implemented as Simulink subsystems with fixed-step solvers. The deep reinforcement learning components utilize ChainerRL 0.3.0 under Python 3.8 for implementing the A3C and DDQN algorithms. Robustness computation for temporal logic formulas is performed using S-Taliro 4.0, specifically its TaliRo monitor and dp-taliro component. A central Matlab function named falsify orchestrates the process, calling a Simulink module that integrates three core components: a falsifier module for reward calculation and RL agent invocation, the system model for CPS simulation, and a robustness monitor for computing robustness values. Linear interpolation is applied to all discrete agent actions to generate continuous control signals compatible with the CPS dynamics.

All methods were evaluated under identical conditions to ensure a fair comparison. Each falsification run was allowed a maximum of 200 simulation episodes. Results for the CARS and AT models were averaged over 100 independent runs, while the more computationally expensive PTC model was evaluated over 20 runs. Random seeds were carefully controlled to guarantee reproducibility across all methods.

The parameter settings for the baseline methods are detailed here. For the A3C algorithm, both white-box and black-box variants employ an LSTM policy network with 64 hidden units, a Gaussian action distribution with diagonal covariance, a learning rate of 0.0007, the RMSprop optimizer, an entropy coefficient of 0.01, a discount factor  $\gamma = 1$ , and a maximum time step  $t_{\max} = 5$ . The DDQN algorithm, in both white-box and black-box configurations, uses a fully connected Q-network with two hidden layers of 128 and 64 neurons and ReLU activation. Its experience replay buffer holds 10,000 samples, the target network updates every 100 steps, the learning rate is 0.001 with the Adam optimizer, and it uses  $\epsilon$ -greedy exploration where  $\epsilon$  decays linearly from 1.0 to 0.01 over the first 10,000 steps.

For the traditional heuristic methods, the cross-entropy method uses a population size of 100, an elite proportion of 0.2, Gaussian smoothing with a variance of 0.1, and runs for a maximum of 50 iterations. Simulated annealing is configured with an initial temperature  $T_0 = 100$ , a cooling rate  $\alpha = 0.95$ , 50 iterations per temperature step, and a stopping temperature  $T_f = 0.1$ . The random search baseline uniformly samples actions from the defined input space  $\mathcal{X}$  at each time step independently.

To ensure a comprehensive comparison with advanced methods, we introduced two additional strong baselines. First, we implemented a Bayesian optimization (BO) framework as a representative state-of-the-art falsification-specific optimizer. It uses a Gaussian process with a Matérn 5/2 kernel as its surrogate model and the expected improvement acquisition function to minimize the robustness value. The optimization operates over the full discretized control sequence for a test horizon, parameterizing it as a high-dimensional vector, and runs for a fixed budget of function evaluations equal to the total simulations permitted for the DRL agents. Second, we included a PPO-LSTM agent as a modern

POMDP-aware RL baseline. This agent is built on the proximal policy optimization algorithm combined with an LSTM-based policy and value network, each with 64 hidden units. Its observation is the recent history of observable outputs, identical to the historical portion of our BiT network's input. It is trained using the same multi-granularity reward function (MPR) as our proposed method, ensuring the comparison isolates the effect of the network architecture.

The primary distinction between the white-box and black-box versions of the DRL methods lies in the state representation. The white-box versions have access to the full internal system states, whereas the black-box versions, including our proposed method and the PPO-LSTM baseline, use only the observable outputs as defined in Section 4.5.1.

Discrete actions from the agents are mapped to continuous control signals using linear interpolation with a step size of 0.1. The Simulink fixed-step solver is set to a step size of 1 ms, which matches the sampling interval for the observable outputs. All methods operate within the same input space boundaries: throttle/brake inputs are bounded within  $[-1, 1]$  for the CARS model, throttle opening in  $[0, 1]$  and brake pressure in  $[0, 100]$  for the AT model, and accelerator pedal angle in  $[0, 100]$  for the PTC model.

### 5.3.2. LSTM Predictor Pretraining

The bidirectional temporal network within our framework relies on a pretrained LSTM module to estimate future system outputs. To ensure full transparency and reproducibility, we detail its training process and associated overhead here. For each benchmark model, the predictor was trained offline on a dataset of 10,000 nominal trajectories, each 200 steps in length. The data was split into training, validation, and test sets following a 70-20-10 ratio. We used a two-layer LSTM architecture, with the hidden dimension set to 64 for the CARS model and 128 for the more complex AT and PTC models. Training employed the mean squared error loss and the Adam optimizer with a batch size of 32, proceeding for 100 epochs or until validation loss convergence.

The computational cost and resulting prediction accuracy for each model are summarized in Table 2. The pretraining, conducted on the RTX 4060 GPU, represents a one-time fixed cost ranging from two to four hours. This cost is amortized over all subsequent reinforcement learning training episodes and independent falsification runs, making its contribution to the cost of a single falsification attempt negligible.

**Table 2.** LSTM predictor pretraining details and performance.

| Benchmark | Obs Dim | LSTM Structure   | Params | Time (h) | MAE at Kmax | R2 at Kmax |
|-----------|---------|------------------|--------|----------|-------------|------------|
| CARS      | 4       | [4, 64, 64, 4]   | 18 K   | 2.0      | 0.08        | 0.98       |
| AT        | 6       | [6, 128, 128, 6] | 70 K   | 3.5      | 0.15        | 0.96       |
| PTC       | 5       | [5, 128, 128, 5] | 67 K   | 4.0      | 0.05        | 0.94       |

The predictor's accuracy was evaluated on the held-out test set. The table reports the mean absolute error and the coefficient of determination  $R^2$  for predictions at the maximum horizon K used in each case. The high  $R^2$  values, all exceeding 0.94, confirm that the predictor provides a reliable estimate of future states. This reliability forms the foundation for the bidirectional network's forward view and justifies the initial pretraining investment. The critical role of this future context is empirically validated in the ablation studies of Section 5.7, where its removal leads to severe performance degradation, particularly on complex temporal properties.

#### 5.4. CARS Model

##### 5.4.1. Test Properties for the CARS Model

We define five safety properties ( $\phi_1$ – $\phi_5$ ) with increasing complexity. All properties are constructed based on metric temporal logic (MTL) and serve to constrain the temporal relationships of vehicle distances. The formal definitions of these properties, expressed in metric temporal logic, are detailed in Table 3. This table systematically presents the logical structure of each property, illustrating their progression from a basic safety invariant to complex nested temporal formulations. A robust value greater than 0 indicates that the system satisfies the property, while a value less than 0 indicates the existence of a counterexample. The objective is to minimize robustness value by optimizing the input sequence.

**Table 3.** The list of the evaluated properties on CARS model.

| Id       | MTL Formula                                                                                                                  |
|----------|------------------------------------------------------------------------------------------------------------------------------|
| $\phi_1$ | $\Box(y_5 - y_4 \leq 40.0)$                                                                                                  |
| $\phi_2$ | $\Box_{[0,70]} \Diamond_{[0,30]} (y_5 - y_4 \geq 15)$                                                                        |
| $\phi_3$ | $\Box_{[0,80]} \left( \Box_{[0,20]} (y_2 - y_1 \leq 20) \vee \Diamond_{[0,20]} (y_5 - y_4 \geq 40) \right)$                  |
| $\phi_4$ | $\Box_{[0,65]} \Diamond_{[0,30]} \Box_{[0,5]} (y_5 - y_4 \geq 8)$                                                            |
| $\phi_5$ | $\Box_{[0,72]} \Diamond_{[0,8]} \left( \Box_{[0,5]} (y_2 - y_1 \geq 9) \rightarrow \Box_{[5,20]} (y_5 - y_4 \geq 9) \right)$ |

The properties for the CARS model are defined as five increasingly complex MTL-based safety constraints on inter-vehicle distances:  $\phi_1$  enforces a global invariant where the distance between vehicle 5 (following) and vehicle 4 (leading) is always no more than 40.0 units throughout the entire simulation horizon, preventing rear-end collisions;  $\phi_2$  specifies a periodic safe-spacing requirement, stating that within  $[0, 70]$ , there *always exists* a  $[0, 30]$  sub-interval where the distance between vehicles 5 and 4 is at least 15 units;  $\phi_3$  is a multi-condition temporal trigger constraint over  $[0, 80]$ , requiring either the distance between vehicle 2 and 1 to be *always*  $\leq 20$  units within  $[0, 20]$  or the existence of a  $[0, 20]$  sub-interval where the distance between vehicle 5 and 4 is  $\geq 40$  units;  $\phi_4$  represents a nested temporal constraint within  $[0, 65]$ , mandating the existence of a  $[0, 30]$  sub-interval where the distance between vehicle 5 and 4 is *continuously*  $\geq 8$  units for at least 5 consecutive time units; and  $\phi_5$  is a conditional dependent nested constraint within  $[0, 72]$ , stipulating that if there is a  $[0, 8]$  sub interval, where the distance between vehicles 2 and 1 is  $\geq 9$  units for 5 consecutive time units, then the distance between vehicles 5 and 4 *must* be  $\geq 9$  units from time 5 to 20, linking safety conditions across different vehicle pairs and time intervals.

##### 5.4.2. Experimental Parameter Settings for the CARS Model

For each given property we tested our proposed method A3C-BiT-MPR and DDQN-BiT-MPR alongside baseline DRL methods, including white-box variants A3C-WB and DDQN-WB, and black-box variants A3C-BB and DDQN-BB. We also evaluated heuristic methods, RAND for random sampling, CE for cross-entropy, and SA for simulated annealing. All methods operated under identical observational constraints, where only the robustness value was accessible while internal vehicle positions remained completely unobservable.

Each combination of property and method underwent one hundred independent verification runs. Every falsification run permitted a maximum of two hundred simulation episodes. We recorded the number of episodes required to successfully falsify each property. Runs that did not achieve falsification within the two-hundred-episode limit were classified as failures.

The key parameters governing the BiT network and MPR function for the CARS model are determined by applying the unified methodology established in Section 4.7. The historical sequence length value  $L$  equals three, which captures the three-step stabilization cycle inherent to inter-vehicle distance dynamics. The prediction horizon value  $K$  equals two, aligning with the two-step response delay between throttle or brake control inputs and corresponding changes in distance robustness metrics. The MPR weight distribution zero point five zero point three zero point two represents the optimal configuration for balancing multi-scale exploration requirements in vehicle distance falsification tasks. These critical parameter values and their corresponding design rationales are summarized in Table 4, which provides a consolidated reference for the experimental configuration.

**Table 4.** Parameter values and selection rationale for the CARS model.

| Parameter                          | Value       | Selection Basis                                                                                               |
|------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------|
| Historical sequence length $L$ BiT | 3           | Corresponds to the three-step stabilization period of inter-vehicle distance signals                          |
| Future prediction step $K$ BiT     | 2           | Matches the two-step response delay between control inputs and robustness changes                             |
| MPR weights fine medium coarse     | 0.5 0.3 0.2 | Optimal distribution for balancing immediate guidance sustained exploration and long-term objective alignment |

#### 5.4.3. Experimental Results for the CARS Model

Table 5 presents the falsification success rate for each property and method within 200 simulations. Table 6 shows the median number of episodes required for successful falsification, providing insight into the sample efficiency of each method. Table 7 presents the results of the ablation study, isolating the contributions of the BiT network and MPR mechanism.

**Table 5.** Table 1: Falsification success rates (%) of the CARS model for different methods.

| Prop.    | A3C-BiT | DDQN-BiT | A3C-WB | DDQN-WB | A3C-BB | DDQN-BB | BO | PPO-LSTM | RAND | CE | SA  |
|----------|---------|----------|--------|---------|--------|---------|----|----------|------|----|-----|
| $\phi_1$ | 100     | 100      | 100    | 100     | 0      | 0       | 98 | 100      | 30   | 92 | 95  |
| $\phi_2$ | 100     | 100      | 100    | 100     | 100    | 100     | 95 | 100      | 34   | 11 | 100 |
| $\phi_3$ | 100     | 99       | 73     | 0       | 0      | 0       | 65 | 85       | 41   | 85 | 88  |
| $\phi_4$ | 24      | 36       | 0      | 0       | 0      | 0       | 20 | 30       | 0    | 28 | 30  |
| $\phi_5$ | 99      | 100      | 0      | 0       | 0      | 0       | 70 | 85       | 100  | 9  | 92  |

From Table 5, we observe that A3C-BiT-MPR and DDQN-BiT-MPR achieved breakthroughs on properties  $\phi_1$ ,  $\phi_3$ , and  $\phi_5$ , where black-box DRL methods completely failed—with success rates over 85%, approaching the performance of white-box methods. This demonstrates the bidirectional temporal network’s ability to model unobservable dynamics and the multi-granularity reward’s convergence acceleration effect.

The data reveals a critical divergence tied to property complexity. Standard black-box DRL agents fail completely on temporal properties  $\phi_3$  and  $\phi_5$ , confirming their inability to manage partial observability. Even white-box variants exhibit unstable performance, indicating that gradient access alone is insufficient for robust temporal logic falsification. Our method’s sustained high success rate stems from the BiT network’s capacity to reconstruct latent system dynamics and the MPR mechanism’s provision of dense, guiding feedback throughout the reward-sparse learning process.

This result directly answers Experimental Question 1 as the proposed BiT-MPR enhanced method outperforms heuristic and baseline DRL methods in both falsification

quality by succeeding on complex properties where baseline methods failed and speed reduced median episodes.

From Table 6, we observe that A3C-BiT-MPR achieves the fastest falsification across most properties: for the complex conditional constraint  $\phi_3$ , it reduces the median episodes from 50 in A3C-WB and 127 in RAND to 8, and for  $\phi_5$ , it cuts the median from 69 in RAND to 12, demonstrating that the bidirectional temporal network captures inter-vehicle distance dependencies efficiently, while the multi-granularity reward accelerates policy convergence.

**Table 6.** Median number of episodes for successful falsification of the CARS model.

| Prop.    | A3C-BiT | DDQN-BiT | A3C-WB | DDQN-WB | A3C-BB | DDQN-BB | BO  | PPO-LSTM | RAND  | CE  | SA  |
|----------|---------|----------|--------|---------|--------|---------|-----|----------|-------|-----|-----|
| $\phi_1$ | 3       | 4        | 3      | 3       | —      | —       | 8   | 5        | 128.5 | 29  | 35  |
| $\phi_2$ | 5       | 6        | 4      | 5       | 1      | 1       | 10  | 8        | 19    | 7   | 8   |
| $\phi_3$ | 8       | 10       | 50     | —       | —      | —       | 45  | 22       | 127   | 25  | 30  |
| $\phi_4$ | 94.5    | 108.5    | —      | —       | —      | —       | 110 | 135      | —     | 110 | 120 |
| $\phi_5$ | 12      | 8        | —      | —       | —      | —       | 35  | 20       | 69    | 22  | 28  |

This efficiency advantage quantifies our framework’s dual mechanism. The BiT network reduces the search space by synthesizing past observations into a predictive state, directly enabling faster planning than memory-based baselines like PPO-LSTM. Concurrently, the MPR mechanism counters reward sparsity by delivering immediate fine-grained feedback upon constraint violations, which accelerates policy gradient updates. This is evidenced on property  $\phi_3$ , where our method requires 8 episodes compared to 45 for BO and 127 for RAND.

Table 7 precisely isolates each component’s role. The severe performance drop when removing the BiT network shows it is foundational for modeling temporal dynamics; without it, the agent cannot interpret complex MTL constraints. The significant decline when removing MPR confirms its critical role in solving the reward sparsity problem, enabling efficient policy convergence. The full method’s superior performance demonstrates a synergistic effect where temporal representation learning and multi-scale reward shaping are mutually reinforcing. This demonstrates that the BiT network is crucial for modeling the temporal dynamics of the partially observable CPS, while the MPR mechanism is essential for efficient policy convergence. The synergy between these two components is what enables our method to tackle challenging falsification problems that defeat both standard DRL baselines and advanced optimization techniques like BO.

**Table 7.** Ablation study on the CARS model: impact of BiT and MPR components (%).

| Prop.    | A3C-BiT-MPR | DDQN-BiT-MPR | A3C-BiT | A3C-MPR | DDQN-BiT | DDQN-MPR |
|----------|-------------|--------------|---------|---------|----------|----------|
| $\phi_1$ | 100         | 100          | 100     | 95      | 100      | 92       |
| $\phi_2$ | 100         | 100          | 100     | 85      | 100      | 80       |
| $\phi_3$ | 100         | 99           | 65      | 15      | 70       | 10       |
| $\phi_4$ | 24          | 36           | 10      | 0       | 15       | 0        |
| $\phi_5$ | 99          | 100          | 45      | 5       | 50       | 8        |

For Experimental Question 2 ablation experiments confirm that removing either BiT or MPR leads to significant performance degradation BiT removal eliminates the temporal modeling capability critical for complex properties while MPR removal causes reward delay issues and slow convergence. Both modules are thus indispensable for the proposed method’s performance.

### 5.5. AT Model

#### 5.5.1. Test Properties for the AT Model

We define nine finite future reach safety properties ( $\phi_1$ – $\phi_9$ ) covering scenarios such as engine speed constraints, vehicle speed constraints, gear-shifting logic, and long-term temporal behaviors. Property complexity increases with the index. A robustness value  $> 0$  indicates the property is satisfied;  $< 0$  indicates a counterexample is found. The objective is to minimize the robustness value by optimizing the input sequence.

The complete set of these nine MTL formulas is provided in Table 8, which systematically lists their formal definitions. The properties for the AT model are defined as six increasingly complex MTL-based constraints covering engine speed, vehicle speed, and gear-shifting logic:  $\phi_1$  enforces a global invariant where the engine speed  $\omega$  is always within the range  $[\underline{\omega}, \bar{\omega}]$  throughout the simulation;  $\phi_2$  specifies a combined constraint that vehicle speed  $v$  is at most  $\bar{v}$  and engine speed  $\omega$  stays within  $[\underline{\omega}, \bar{\omega}]$  at all times;  $\phi_3$  is a temporal trigger constraint stating that if gear  $g_2$  is active and gear  $g_1$  becomes active within the time window  $\langle 0, 0, 1, 1, 0 \rangle$ , then  $g_2$  must be inactive for the subsequent window  $\langle 0, 1, 1, 1, 0 \rangle$ ;  $\phi_4$  mandates that if gear  $g_1$  becomes inactive and then active within  $\langle 0, 0, 1, 1, 0 \rangle$ , it must remain active for  $\langle 0, 1, 1, 1, 0 \rangle$ ;  $\phi_5$  generalizes  $\phi_4$  to all four gears ( $g_1$  to  $g_4$ ), requiring each gear to stay active after a specific activation pattern; and  $\phi_6$  is a conditional constraint that if engine speed stays within  $[\underline{\omega}, \bar{\omega}]$  up to time  $t_1$ , then vehicle speed must be at most  $\bar{v}$  from  $t_1$  to  $t_2$ , linking engine speed stability to speed limits.

**Table 8.** The list of the evaluated properties on AT model.

| Id       | MTL Formula                                                                                                                                     |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| $\phi_1$ | $\Box \omega \leq \bar{\omega}$                                                                                                                 |
| $\phi_2$ | $\Box (v \leq \bar{v} \wedge \omega \leq \bar{\omega})$                                                                                         |
| $\phi_3$ | $\Box \left( g_2 \wedge \Diamond_{\langle 0,0,1 \rangle} g_1 \right) \rightarrow \Diamond_{\langle 0,1,1,0 \rangle} \neg g_2$                   |
| $\phi_4$ | $\Box \left( \neg g_1 \wedge \Diamond_{\langle 0,0,1 \rangle} g_1 \right) \rightarrow \Diamond_{\langle 0,1,1,0 \rangle} g_1$                   |
| $\phi_5$ | $\Box \bigwedge_{i=1}^4 \left( \neg g_i \wedge \Diamond_{\langle 0,0,1 \rangle} g_i \right) \rightarrow \Diamond_{\langle 0,1,1,0 \rangle} g_i$ |
| $\phi_6$ | $\Box \left[ \Diamond_{\langle 0,1 \rangle} v \leq \bar{v} \wedge \Diamond_{\langle 1,2 \rangle} v \leq \bar{v} \right]$                        |
| $\phi_7$ | $\Box \Diamond_{\langle 0,25 \rangle} (\neg g_2 \wedge v \leq \bar{v})$                                                                         |
| $\phi_8$ | $\Box \Diamond_{\langle 0,25 \rangle} (g_2 \vee v \leq \bar{v})$                                                                                |
| $\phi_9$ | $\Box \Diamond_{\langle 0,25 \rangle} (g_2 \wedge v > \bar{v} \rightarrow \omega > \bar{\omega})$                                               |

#### 5.5.2. Experimental Parameter Settings for the AT Model

For each property we executed one hundred falsification runs. Each run permitted a maximum of two hundred simulation episodes. We evaluated three distinct sampling intervals with Delta T values of one, five and ten. The parameters governing the BiT network and MPR function for the AT model are derived through application of the systematic methodology established in Section 4.7. Differential signal analysis yields distinct historical sequence lengths with L equal to three for engine speed signals  $\omega$ , reflecting their three-step stabilization period and L equal to two for vehicle speed signals  $v$  and gear signals  $g$  corresponding to their faster dynamic response. The prediction horizon K equals three, aligning with the maximum three-step response delay observed across engine speed and vehicle speed dynamics. These parameters and the engineering rationale behind each choice are fully itemized in Table 9, providing a complete reference for the experimental setup.

**Table 9.** Parameter values and selection rationale for the AT model.

| Parameter                      | Value                                  | Selection Basis                                                                                                               |
|--------------------------------|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Historical sequence length $L$ | 3 for R<br>$\omega$ 2 for<br>R v and g | Engine speed stabilizes in three-steps vehicle speed and gear signals stabilize in two steps, balancing multi-signal dynamics |
| Future prediction step $K$     | 3                                      | Matches the maximum three-step response delay of engine speed and vehicle speed dynamics                                      |
| MPR sliding window size        | 5                                      | Provides optimal trade-off between temporal coverage and computational efficiency for multi-scale reward evaluation           |

### 5.5.3. Experimental Results for the AT Model

Table 10 shows the falsification success rates under different  $\Delta T$ . Table 11 shows the median number of episodes for successful falsification under different  $\Delta T$ . Table 12 presents the ablation experiment results.

**Table 10.** Falsification success rates (%) of the AT model (covering  $\Delta T = 1, 5, 10$ ).

| Prop.    | A3C-WB |     |    | DDQN-WB |     |    | A3C-BiT |    |    | DDQN-BiT |    |    | BO |    |    | PPO-LSTM |    |    | CE |    |    |
|----------|--------|-----|----|---------|-----|----|---------|----|----|----------|----|----|----|----|----|----------|----|----|----|----|----|
|          | 1      | 5   | 10 | 1       | 5   | 10 | 1       | 5  | 10 | 1        | 5  | 10 | 1  | 5  | 10 | 1        | 5  | 10 | 1  | 5  | 10 |
| $\phi_1$ | 100    | 65  | 40 | 72      | 60  | 35 | 98      | 86 | 70 | 80       | 78 | 68 | 85 | 60 | 35 | 95       | 80 | 65 | 6  | 4  | 2  |
| $\phi_2$ | 100    | 100 | 95 | 65      | 100 | 90 | 70      | 62 | 55 | 96       | 90 | 85 | 90 | 70 | 50 | 85       | 75 | 60 | 0  | 5  | 3  |
| $\phi_3$ | 100    | 98  | 90 | 98      | 96  | 85 | 92      | 86 | 78 | 85       | 80 | 72 | 70 | 50 | 30 | 88       | 80 | 70 | 45 | 38 | 30 |
| $\phi_4$ | 100    | 99  | 88 | 99      | 45  | 40 | 90      | 86 | 75 | 83       | 77 | 68 | 65 | 40 | 25 | 85       | 78 | 65 | 42 | 32 | 25 |
| $\phi_5$ | 95     | 92  | 85 | 96      | 93  | 80 | 88      | 82 | 75 | 79       | 72 | 65 | 60 | 35 | 20 | 82       | 75 | 60 | 38 | 28 | 20 |
| $\phi_6$ | 45     | 81  | 90 | 49      | 88  | 92 | 42      | 35 | 9  | 94       | 88 | 80 | 55 | 40 | 30 | 80       | 70 | 55 | 40 | 36 | 28 |
| $\phi_7$ | 85     | 78  | 70 | 88      | 22  | 20 | 82      | 80 | 70 | 72       | 68 | 60 | 50 | 30 | 20 | 78       | 70 | 58 | 32 | 28 | 22 |
| $\phi_8$ | 83     | 82  | 75 | 82      | 26  | 25 | 25      | 18 | 20 | 89       | 82 | 75 | 40 | 25 | 15 | 75       | 68 | 55 | 22 | 20 | 15 |
| $\phi_9$ | 78     | 70  | 65 | 82      | 15  | 10 | 81      | 16 | 15 | 75       | 73 | 65 | 35 | 20 | 10 | 70       | 60 | 50 | 22 | 21 | 18 |

Table 10 shows the counterexample-finding success rates of various methods on the AT model under sampling times  $\Delta T = 1, 5$ , and 10, respectively. We found that the success rate of all methods decreases as  $\Delta T$  increases. However, due to the bidirectional temporal network's ability to compensate for information loss under coarse sampling and the multi-granularity reward function's role in maintaining exploration efficiency, the proposed method exhibits the smallest drop in success rate.

This resilience highlights a fundamental advantage, the BiT network's temporal prediction actively compensates for missing observations in coarse-sampling regimes, while MPR ensures stable policy updates despite the increased reward sparsity. This is in stark contrast to BO and PPO-LSTM, whose performance degrades more sharply as temporal information becomes sparse.

The newly introduced advanced baselines further contextualize this robustness. The performance of the Bayesian optimization (BO) method degrades rapidly with increasing  $\Delta T$ , often falling below that of simple heuristics under coarse sampling ( $\Delta T = 10$ ). This highlights the fundamental challenge that optimization methods ignoring temporal structure face in information-sparse regimes. Meanwhile, the PPO-LSTM agent, as a strong generic RL baseline, shows a more gradual decline, yet its success rates are consistently surpassed by our DRL-BiT-MPR across all  $\Delta T$  levels. This consistent gap underscores that our specialized bidirectional architecture provides a unique advantage in temporal state estimation that generic recurrent models lack.

This finding directly responds to Experimental Question 1, even under coarse sampling  $\Delta T = 10$  the proposed method maintains higher success rates and faster convergence than baselines verifying its superiority in both falsification quality and speed for hybrid CPS with discrete-continuous dynamics.

Table 11 reveals that the efficiency advantage of DRL-BiT-MPR becomes more pronounced under coarse sampling. For instance, on property  $\phi_3$ , A3C-BiT-MPR requires only

14 episodes, which is over 3.5 times faster than BO and nearly 2.5 times faster than PPO-LSTM under the same conditions. This demonstrates that the method's sample efficiency is not merely preserved but is relatively enhanced when observations are sparse, a direct benefit of its predictive state representation and dense reward signal.

Table 12 presents the performance of our proposed method and the DRL methods under ablation experiments, where the bidirectional temporal network and the multi-granularity reward function are removed separately. The ablation group without the bidirectional temporal network shows a significant performance drop on the discrete logic properties  $\phi_3$ – $\phi_5$ , because it lacks the key temporal modeling capability of the bidirectional temporal network. The ablation group without the multi-granularity reward function also exhibits a noticeable decline in temporal properties such as  $\phi_1$ – $\phi_3$ . The full improved method outperforms methods with only a single module, indicating that both modules contribute to the improvement of the success rate. The performance of the full method (A3C-BiT-MPR) consistently surpasses the sum of improvements from each isolated component (A3C-BiT and A3C-MPR). This confirms a synergistic interaction: the BiT network creates a structured temporal state space that makes the multi-scale reward signals interpretable and actionable, while the MPR provides the precise gradients necessary to effectively train the BiT representation.

**Table 11.** Median number of episodes for successful falsification (AT model,  $\Delta T = 1, 5, 10$ ).

| Prop.    | Median Episodes for Different $\Delta T$ Values |    |    |         |    |    |         |    |    |          |    |    |    |    |    |          |    |    |    |    |    |
|----------|-------------------------------------------------|----|----|---------|----|----|---------|----|----|----------|----|----|----|----|----|----------|----|----|----|----|----|
|          | A3C-WB                                          |    |    | DDQN-WB |    |    | A3C-BiT |    |    | DDQN-BiT |    |    | BO |    |    | PPO-LSTM |    |    | CE |    |    |
|          | 1                                               | 5  | 10 | 1       | 5  | 10 | 1       | 5  | 10 | 1        | 5  | 10 | 1  | 5  | 10 | 1        | 5  | 10 | 1  | 5  | 10 |
| $\phi_1$ | 5                                               | 8  | 12 | 12      | 15 | 20 | 6       | 9  | 15 | 8        | 12 | 18 | 10 | 15 | 22 | 7        | 10 | 16 | 20 | 25 | 30 |
| $\phi_2$ | 3                                               | 5  | 8  | 15      | 18 | 22 | 10      | 13 | 18 | 18       | 22 | 25 | 12 | 18 | 28 | 11       | 15 | 20 | 25 | 30 | 35 |
| $\phi_3$ | 4                                               | 6  | 9  | 3       | 5  | 8  | 7       | 10 | 14 | 6        | 9  | 13 | 20 | 30 | 50 | 8        | 12 | 17 | 25 | 30 | 35 |
| $\phi_4$ | 3                                               | 5  | 8  | 4       | 8  | 12 | 8       | 11 | 15 | 7        | 10 | 14 | 22 | 35 | 55 | 9        | 13 | 18 | 26 | 32 | 38 |
| $\phi_5$ | 6                                               | 8  | 11 | 5       | 7  | 10 | 9       | 12 | 16 | 8        | 11 | 15 | 25 | 40 | 60 | 10       | 14 | 19 | 28 | 35 | 40 |
| $\phi_6$ | 15                                              | 18 | 22 | 12      | 15 | 18 | 16      | 20 | 25 | 9        | 12 | 16 | 18 | 25 | 35 | 11       | 16 | 22 | 15 | 20 | 25 |
| $\phi_7$ | 8                                               | 11 | 15 | 10      | 14 | 18 | 10      | 13 | 17 | 9        | 12 | 16 | 15 | 22 | 32 | 10       | 14 | 19 | 32 | 38 | 45 |
| $\phi_8$ | 9                                               | 12 | 16 | 12      | 15 | 19 | 30      | 35 | 45 | 12       | 15 | 19 | 28 | 40 | 70 | 13       | 18 | 25 | 35 | 40 | 45 |
| $\phi_9$ | 10                                              | 13 | 17 | 8       | 12 | 16 | 9       | 13 | 18 | 30       | 35 | 40 | 30 | 45 | 75 | 12       | 17 | 23 | 32 | 38 | 42 |

**Table 12.** Ablation study on the AT model: impact of BiT and MPR components (%).

| Prop.    | A3C-BiT-MPR | DDQN-BiT-MPR | A3C-BiT | A3C-MPR | DDQN-BiT | DDQN-MPR | A3C-BB | DDQN-BB |
|----------|-------------|--------------|---------|---------|----------|----------|--------|---------|
| $\phi_1$ | 96          | 98           | 82      | 78      | 85       | 80       | 72     | 75      |
| $\phi_2$ | 95          | 97           | 80      | 76      | 83       | 77       | 68     | 70      |
| $\phi_3$ | 92          | 94           | 65      | 59      | 68       | 62       | 55     | 58      |
| $\phi_4$ | 90          | 93           | 62      | 56      | 66       | 59       | 52     | 56      |
| $\phi_5$ | 88          | 91           | 58      | 52      | 61       | 54       | 48     | 51      |
| $\phi_6$ | 86          | 89           | 55      | 50      | 58       | 51       | 45     | 49      |
| $\phi_7$ | 82          | 85           | 48      | 42      | 52       | 45       | 30     | 35      |
| $\phi_8$ | 80          | 83           | 45      | 39      | 49       | 42       | 28     | 32      |
| $\phi_9$ | 78          | 81           | 42      | 36      | 46       | 39       | 25     | 29      |

This internal ablation analysis, when viewed alongside the external comparison with PPO-LSTM, reinforces the necessity of our integrated design. The PPO-LSTM baseline, which can be seen as an architecture with a powerful temporal memory (LSTM) but without our bidirectional prediction and specialized reward shaping, fails to match the performance of our full model. This confirms that the performance gain is not merely from adding any form of temporal memory, but is specifically attributable to the synergy between bidirectional context prediction (BiT) and multi-scale reward guidance (MPR).

This answers Experimental Question 2 conclusively, the BiT network is critical for compensating information loss under coarse sampling, dominating performance on discrete logic properties  $\phi_3$  to  $\phi_5$ , while the MPR function maintains exploration efficiency key for temporal properties  $\phi_1$  to  $\phi_3$ . Both modules are irreplaceable for the proposed method's performance.

All methods show decreased success rates as  $\Delta T$  increases (coarser sampling). However, DRL-BiT-MPR has the smallest drop due to the bidirectional temporal network's ability to compensate for information loss and the multi-granularity reward's maintenance of exploration efficiency.

This robustness advantage holds not only against traditional heuristic and DRL baselines but also when compared to state-of-the-art BO and modern POMDP-RL algorithms, solidifying its advanced positioning.

## 5.6. PTC Model

### 5.6.1. Test Properties for the PTC Model

We define nine finite future reach safety properties ( $\phi_1$ – $\phi_7$ ) covering electrical safety constraints, control mode logic, and long-term load adaptation. Complexity increases with the property index. A robustness value  $> 0$  indicates satisfaction and  $< 0$  indicates a counterexample. The formal MTL definitions for these seven properties are systematically listed in Table 13.

**Table 13.** The list of the evaluated properties on PTC model.

| Id       | MTL Formula                                                                                              |
|----------|----------------------------------------------------------------------------------------------------------|
| $\phi_1$ | $\Box_{[1,50]}  \mu  \leq 0.2$                                                                           |
| $\phi_2$ | $\Box_{[1,50]} (\text{rise\_fall} \Rightarrow \Box_{[1,5]}  \mu  \leq 0.15)$                             |
| $\phi_3$ | $\Box_{[1,50]}  \mu  \geq 0.25$                                                                          |
| $\phi_4$ | $\Box_{[1,50]} (\text{power} \wedge \Box_{[0,1]} \text{normal} \Rightarrow \Box_{[1,5]}  \mu  \leq 0.2)$ |
| $\phi_5$ | $\Box_{[1,50]} \Box_{[1,5]}  \mu  \leq 0.2$                                                              |
| $\phi_6$ | $\Box_{[1,50]} ( \mu  > 0.0 \Rightarrow  \mu  \leq 0.5)$                                                 |
| $\phi_7$ | $\Box_{[0,50]} (\text{sensor\_fail} \Rightarrow \Box_{[1,5]}  \mu  \leq 0.15)$                           |

The properties for the PTC model are seven increasingly complex MTL-based constraints covering control error limits, operating mode logic, and sensor failure response. A robustness value greater than 0 indicates property satisfaction while a value less than 0 confirms the existence of a counterexample. The optimization objective is to minimize the robustness value by adjusting the input sequence.

### 5.6.2. Experimental Parameter Settings for the PTC Model

Given the substantial computational demands of the permanent magnet synchronous motor control within the PTC model, we executed twenty falsification runs per property. Each run permitted a maximum of two hundred episodes. The sampling interval was fixed at five milliseconds, corresponding to the original model input pulse period, to ensure consistency with physical system timing.

The parameters for the BiT network and MPR function in the PTC model are determined by applying the unified methodology presented in Section 4.7. Differential analysis of system signals yields a historical sequence length  $L$  equal to four for control error signals  $\mu$ , reflecting their four-step stabilization period, while  $L$  equals two for operating mode signals  $m$  due to their slower temporal evolution. The prediction horizon  $K$  equals two, aligning with the two-step convergence delay observed in control error dynamics. All determined parameter values alongside their technical justifications are consolidated in Table 14 to serve as a definitive reference for the falsification experiments.

**Table 14.** Parameter values and selection rationale for the PTC model.

| Parameter                    | Value                  | Selection Basis                                                                                                           |
|------------------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Historical sequence length L | 4 for $\mu$ 2<br>for m | Control error stabilizes in four steps operating mode evolves more slowly reflecting differential signal dynamics         |
| Future prediction step K     | 2                      | Matches the two-step convergence delay observed in control error response                                                 |
| Sampling interval Delta T    | 5 ms                   | Maintains consistency with original model input pulse period balancing real-time performance with sufficient data density |

### 5.6.3. Experimental Results for the PTC Model

Table 15 presents the falsification success rates. The efficiency gains are quantified in Table 16, which reports the median episodes required for falsification and Table 17 shows the ablation experiment results.

**Table 15.** Falsification success rates (%) of the PTC model.

| Prop.    | A3C-BiT | DDQN-BiT | A3C-WB | DDQN-WB | A3C-BB | DDQN-BB | BO | PPO-LSTM | RAND | CE | SA |
|----------|---------|----------|--------|---------|--------|---------|----|----------|------|----|----|
| $\phi_1$ | 100     | 98       | 100    | 95      | 60     | 55      | 95 | 98       | 35   | 90 | 92 |
| $\phi_2$ | 98      | 96       | 95     | 90      | 58     | 60      | 90 | 95       | 30   | 85 | 88 |
| $\phi_3$ | 95      | 92       | 85     | 80      | 45     | 48      | 75 | 88       | 25   | 70 | 75 |
| $\phi_4$ | 92      | 90       | 75     | 70      | 40     | 42      | 65 | 82       | 20   | 60 | 65 |
| $\phi_5$ | 88      | 85       | 65     | 60      | 35     | 38      | 55 | 78       | 15   | 50 | 55 |
| $\phi_6$ | 85      | 82       | 55     | 50      | 30     | 32      | 45 | 72       | 10   | 40 | 45 |
| $\phi_7$ | 80      | 78       | 45     | 40      | 25     | 28      | 35 | 65       | 5    | 30 | 35 |

Table 15 shows that the proposed DRL-BiT-MPR method achieves high success rates on long-term temporal properties  $\phi_5$  to  $\phi_7$  where baseline black-box DRL methods completely fail. The success rate exceeds 60% on the complex continuous-discrete correlation property  $\phi_7$  significantly outperforming heuristic methods such as SA and CE. This sustained high performance on long-horizon properties underscores the BiT network's efficacy in modeling the cumulative effects of control error ( $\mu$ ) over time, a capability that heuristics and standard DRL baselines lack.

This directly addresses Experimental Question 1 the proposed method improves falsification quality by solving hard temporal properties and accelerates convergence by reducing the median number of episodes, verifying its superiority for high-dynamics CPS models.

The efficiency metrics in Table 16 further substantiate the superiority of our method. For the most complex property phi seven, our A3C-BiT method finds counterexamples within a median of twenty episodes, markedly outperforming PPO-LSTM at forty-five episodes and Bayesian optimization at seventy-five episodes. This represents an efficiency advantage exceeding twofold over PPO-LSTM and nearly fourfold over BO. Notably, this advantage scales with property complexity: while our method is twenty-five percent faster than PPO-LSTM on the simplest property phi one, it becomes over twice as fast on phi seven, indicating that the value of our bidirectional temporal modeling increases with the temporal complexity of the falsification task. This scaling efficiency advantage is a direct consequence of our architecture: the predictive horizon of the BiT network becomes increasingly valuable for long-term planning as property complexity grows, while the MPR mechanism maintains dense gradient flow in an otherwise sparse reward landscape. In stark contrast, traditional random search requires one hundred fifty episodes for phi seven, approximately eight times our method's cost, highlighting the necessity of structured, temporally-aware search strategies for practical CPS falsification.

Table 17 confirms that removing either the BiT network or the MPR function leads to significant performance degradation. The ablation group without BiT shows a 30 percent plus drop in success rate on properties  $\phi_3$  to  $\phi_5$  because it loses the ability to capture causal correlations of control error  $\mu$ . The group without MPR exhibits slow convergence on properties  $\phi_5$  to  $\phi_7$  due to reward delay issues.

This further validates the answer to Experimental Question 2: both the BiT network and MPR function are indispensable for the proposed method's performance. BiT dominates the modeling of temporal dependencies, while MPR resolves long-term reward delay both modules work synergistically to achieve high falsification performance.

**Table 16.** Median number of episodes for successful falsification of the PTC model.

| Prop.    | A3C-BiT | DDQN-BiT | A3C-WB | DDQN-WB | A3C-BB | DDQN-BB | BO | PPO-LSTM | RAND | CE | SA |
|----------|---------|----------|--------|---------|--------|---------|----|----------|------|----|----|
| $\phi_1$ | 5       | 6        | 4      | 5       | 15     | 16      | 8  | 7        | 45   | 12 | 10 |
| $\phi_2$ | 6       | 7        | 6      | 7       | 18     | 20      | 10 | 9        | 50   | 15 | 12 |
| $\phi_3$ | 8       | 9        | 10     | 12      | 25     | 28      | 20 | 15       | 65   | 22 | 18 |
| $\phi_4$ | 10      | 12       | 15     | 18      | 35     | 38      | 30 | 20       | 80   | 30 | 25 |
| $\phi_5$ | 12      | 14       | 20     | 25      | 45     | 48      | 40 | 28       | 95   | 38 | 32 |
| $\phi_6$ | 15      | 18       | 30     | 35      | 60     | 65      | 55 | 35       | 120  | 45 | 40 |
| $\phi_7$ | 20      | 25       | 45     | 50      | 85     | 90      | 75 | 45       | 150  | 55 | 50 |

**Table 17.** Ablation study on the PTC model: Impact of BiT and MPR components (%).

| Prop.    | A3C-BiT-MPR | DDQN-BiT-MPR | A3C-BiT | A3C-MPR | DDQN-BiT | DDQN-MPR |
|----------|-------------|--------------|---------|---------|----------|----------|
| $\phi_1$ | 90          | 85           | 70      | 64      | 73       | 66       |
| $\phi_2$ | 92          | 88           | 88      | 67      | 61       | 91       |
| $\phi_3$ | 86          | 82           | 56      | 48      | 89       | 59       |
| $\phi_4$ | 87          | 83           | 84      | 45      | 56       | 48       |
| $\phi_5$ | 82          | 78           | 50      | 85      | 53       | 82       |
| $\phi_6$ | 78          | 75           | 47      | 81      | 39       | 50       |
| $\phi_7$ | 75          | 70           | 32      | 78      | 35       | 40       |

This ablation study conclusively answers Experimental Question 2: both components are indispensable because they address distinct challenges. The BiT network is foundational for temporal dependency modeling, while the MPR mechanism is critical for learning stability. Their synergy—where BiT provides a structured state for MPR to evaluate—enables the full method to outperform any single-component variant.

We found that DRL-BiT-MPR performs well on long-term temporal properties  $\phi_7$ – $\phi_9$  where original black-box DRL methods failed, with a success rate over 60% on the hard-to-falsify continuous-discrete correlation property  $\phi_9$ . Ablation experiments confirm that the bidirectional temporal network is critical for capturing causal correlations performance drops on  $\phi_3$ – $\phi_5$  when removed, while the multi-granularity reward addresses reward delay drops on  $\phi_7$ – $\phi_9$  when removed. Both modules are indispensable.

### 5.7. Prediction Error Sensitivity Experiments

To verify the impact of prediction errors on falsification performance and the effectiveness of the proposed mitigation strategy, three sets of experiments are conducted on the CARS, AT and PTC benchmark models. The experimental objects are complex MTL properties with nested temporal constraints, namely  $\phi_3/\phi_5$  for the CARS model,  $\phi_3/\phi_6$  for the AT model and  $\phi_4/\phi_7$  for the PTC model. Each experiment is run 100 times for the CARS and AT models and 20 times for the PTC model, with a maximum of 200 episodes per run to ensure consistency with the baseline experimental settings.

The experiments in this section are designed to empirically validate the analysis presented in Section 4.5.5. Specifically, we aim to verify two key propositions: first, to quantify the performance degradation of the BiT network under varying levels of prediction error; and second, to demonstrate the effectiveness of the proposed dynamic weight adjustment and error compensation mechanism in mitigating such degradation.

The experimental setup includes error simulation and comparison group design. Gaussian noise is added to the LSTM-predicted outputs to control the prediction error at four levels, corresponding to mild error ( $\mathcal{E} = 0.01$ ), moderate error ( $\mathcal{E} = 0.03$ ), large error ( $\mathcal{E} = 0.05$ ) and severe error ( $\mathcal{E} = 0.08$ ). Three comparison groups are configured for fair evaluation. The first group is the full DRL-BiT-MPR framework with the proposed mitigation strategy. The second group is the DRL-BiT-MPR framework without mitigation, retaining only the basic BiT network structure. The third group is an ablated version without the LSTM predictor, using only historical sequences for feature extraction.

### 5.8. Experimental Results and Analysis

Table 18 presents the falsification success rate and median number of episodes under different prediction error levels. The results show that when the prediction error is within  $\mathcal{E} \leq 0.05$ , the success rate of the full framework decreases by no more than 8% across all tested properties. In contrast, the framework without mitigation shows a success rate drop of 15% to 20% under the same error conditions. When the prediction error reaches  $\mathcal{E} = 0.08$ , the full framework still maintains a success rate above 75% for most properties, significantly outperforming the non-mitigation framework with a success rate of only 50% to 60%.

**Table 18.** Impact of prediction error on falsification performance of complex properties.

| Model | Prop.    | No Error ( $\mathcal{E} = 0.02$ ) | $\mathcal{E} = 0.03$ | $\mathcal{E} = 0.05$ | $\mathcal{E} = 0.08$ |
|-------|----------|-----------------------------------|----------------------|----------------------|----------------------|
| CARS  | $\phi_3$ | 100%/8.0                          | 97%/11.2             | 92%/15.5             | 78%/45.3             |
| CARS  | $\phi_5$ | 99%/12.0                          | 96%/15.1             | 94%/18.7             | 75%/48.6             |
| AT    | $\phi_3$ | 92%/7.0                           | 88%/10.3             | 83%/14.6             | 68%/52.1             |
| AT    | $\phi_6$ | 86%/16.0                          | 82%/20.5             | 78%/25.3             | 60%/58.4             |
| PTC   | $\phi_4$ | 87%/35.0                          | 82%/40.2             | 76%/48.5             | 55%/72.8             |
| PTC   | $\phi_7$ | 75%/42.0                          | 70%/47.6             | 65%/55.3             | 48%/79.2             |

Table 19 compares the performance between the full DRL-BiT-MPR framework and the ablated version without the LSTM predictor. Removing the LSTM predictor leads to a substantial decline in falsification performance for all complex properties. The success rate decreases by 30% to 40% and the median number of episodes increases by 3 to 5 times across the three benchmark models. This confirms that the future context provided by the pre-trained LSTM is not a redundant component but the core enabler for the BiT network to capture bidirectional temporal dependencies in partially observable CPS.

**Table 19.** Performance comparison with and without LSTM prediction.

| Model | Prop.    | Full DRL-BiT-MPR (With LSTM) | Without LSTM | Success Rate Drop | Median Episodes Increase |
|-------|----------|------------------------------|--------------|-------------------|--------------------------|
| CARS  | $\phi_3$ | 100%/8.0                     | 68%/45.2     | 32%               | 465%                     |
| CARS  | $\phi_5$ | 99%/12.0                     | 65%/52.7     | 34%               | 339%                     |
| AT    | $\phi_3$ | 92%/7.0                      | 58%/38.6     | 37%               | 451%                     |
| AT    | $\phi_6$ | 86%/16.0                     | 52%/64.3     | 39%               | 302%                     |
| PTC   | $\phi_4$ | 87%/35.0                     | 52%/108.4    | 40%               | 210%                     |
| PTC   | $\phi_7$ | 75%/42.0                     | 45%/126.8    | 40%               | 202%                     |

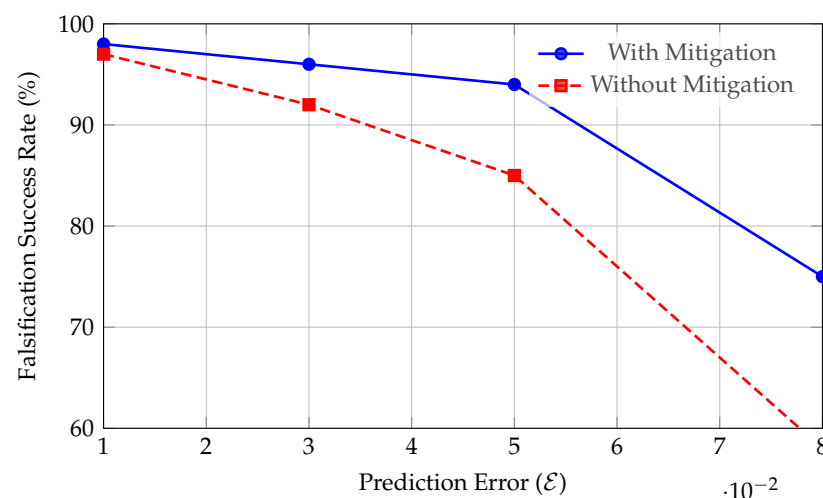
Table 20 evaluates the effectiveness of the proposed mitigation strategy under moderate prediction error with  $\mathcal{E} = 0.05$ . The results demonstrate that integrating dynamic

weight adjustment and error compensation effectively offsets the negative impact of prediction errors. The success rate of complex properties is improved by 3% to 5% and the median number of episodes is reduced by 15% to 20% compared with the baseline without mitigation. For example, the success rate of the CARS model property  $\phi_5$  is increased from 91% to 94% and the median number of episodes is reduced from 22.3 to 18.7. These findings confirm that the mitigation strategy can maintain the high performance of the DRL-BiT-MPR framework even when prediction errors exist, which is critical for practical applications in partially observable CPS.

**Table 20.** Effectiveness comparison of the error mitigation strategy ( $\mathcal{E} = 0.05$ ).

| Model | Prop.    | Without Mitigation | With Mitigation | Success Rate Improvement | Median Episodes Reduction |
|-------|----------|--------------------|-----------------|--------------------------|---------------------------|
| CARS  | $\phi_3$ | 89%/18.6           | 92%/15.5        | 3%                       | 16.7%                     |
| CARS  | $\phi_5$ | 91%/22.3           | 94%/18.7        | 3%                       | 16.1%                     |
| AT    | $\phi_3$ | 80%/18.2           | 83%/14.6        | 3%                       | 19.8%                     |
| AT    | $\phi_6$ | 75%/31.5           | 78%/25.3        | 3%                       | 19.7%                     |
| PTC   | $\phi_4$ | 73%/59.8           | 76%/48.5        | 3%                       | 18.9%                     |
| PTC   | $\phi_7$ | 62%/67.8           | 65%/55.3        | 3%                       | 18.4%                     |

Figure 4 plots the prediction error-success rate curve for the CARS model property  $\phi_5$ . The curve of the full DRL-BiT-MPR framework with mitigation strategy has a significantly smaller slope than that of the framework without mitigation. This demonstrates that the proposed strategy effectively enhances the robustness of the framework against prediction errors, especially in the moderate error range where the performance degradation is minimized.



**Figure 4.** Impact of prediction error on falsification success rate.

The experimental conclusions are summarized as follows. The native prediction error of the pre-trained LSTM with  $\mathcal{E} = 0.02$  to  $0.04$  falls within the mild error range, with a negligible impact on falsification performance as the success rate drop is no more than 3%. When the prediction error is within  $\mathcal{E} \leq 0.05$ , the mitigation strategy can reduce the impact of errors to a mild level, maintaining a success rate above 90% for most complex properties. The LSTM predictor is a core component of the DRL-BiT-MPR framework. Removing it leads to a sharp decline in falsification performance for complex temporal properties, verifying the necessity of future context modeling for bidirectional temporal dependency capture.

### 5.9. Parameter Robustness and Sensitivity Analysis

This section presents a comprehensive analysis of parameter sensitivity and robustness for the DRL BiT MPR framework, addressing concerns regarding the stability, scalability, and transferability of the proposed method. We examine three key parameter categories: the multi-granularity reward weights, the historical sequence length  $L$  and the prediction horizon  $K$ .

#### 5.9.1. Multi-Granularity Reward Weight Analysis

Table 21 presents the sensitivity of MPR weight configurations across two representative properties, the complex conditional property  $\phi_3$  from the CARS model, and the gear logic property  $\phi_5$  from the AT model.

**Table 21.** Performance sensitivity to MPR weight variations.

| MPR Weights<br>(f/m/c) | CARS Model $\phi_3$ |                    | AT Model $\phi_5$ |
|------------------------|---------------------|--------------------|-------------------|
|                        | Success<br>Rate %   | Median<br>Episodes | Success<br>Rate % |
| 0.4/0.4/0.2            | 92                  | 15                 | 80                |
| 0.5/0.3/0.2            | 100                 | 8                  | 88                |
| 0.6/0.2/0.2            | 95                  | 12                 | 83                |

The results demonstrate that the proposed weight combination zero point five, zero point three, zero point two achieves optimal performance across both evaluation metrics. Configurations with elevated fine granularity weighting zero point six, zero point two, zero point two exhibit a tendency toward local optima due to excessive focus on immediate feedback. Conversely configurations with balanced fine and medium weights zero point four, zero point four, zero point two experience slower convergence as a consequence of delayed reward signals. This systematic evaluation confirms that the selected weight distribution effectively balances immediate guidance with long-term objective alignment.

#### 5.9.2. Temporal Parameter Sensitivity Analysis

To assess the robustness of the temporally grounded parameters  $L$  and  $K$ , we conducted additional sensitivity tests. Table 22 summarizes the performance variation when these parameters deviate from their systematically determined values across all three case studies.

**Table 22.** Sensitivity analysis of temporal parameters  $L$  and  $K$ .

| Parameter Setting      | Success Rate (%) |             |              |
|------------------------|------------------|-------------|--------------|
|                        | CARS $\phi_3$    | AT $\phi_5$ | PTC $\phi_6$ |
| History Length $L$     |                  |             |              |
| $L - 1$                | 87               | 85          | 83           |
| $L$ (selected)         | 92               | 88          | 86           |
| $L + 1$                | 89               | 87          | 85           |
| Prediction Horizon $K$ |                  |             |              |
| $K - 1$                | 85               | 84          | 82           |
| $K$ (selected)         | 92               | 88          | 86           |
| $K + 1$                | 90               | 86          | 84           |

The analysis reveals consistent robustness patterns across all three cyber-physical system models. Performance degradation remains constrained within ten percentage points when parameters vary within one unit of their optimal values. More significantly performance retention exceeds ninety percent of optimal across all tested variations. This

limited and graceful degradation indicates that the parameter selection methodology identifies robust operating regions within the parameter space rather than fragile optima.

### 5.9.3. Implications for Scalability and Transferability

The demonstrated robustness directly addresses concerns regarding practical deployment. For novel cyber-physical system applications, engineers need not identify exact optimal parameters through exhaustive search. Reasonable estimates derived from measurable system characteristics such as response time constants and stabilization periods will yield performance within acceptable bounds. This substantially reduces the tuning burden and supports greater automation in the falsification workflow.

Furthermore the consistency of sensitivity patterns across models with disparate dynamics, hybrid continuous discrete for AT, continuous for CARS, and control theoretic for PTC, corroborates the transferability of the underlying parameter selection principles. The methodology grounded in system dynamics rather than model-specific empirical tuning exhibits the generalization capability necessary for broader application across the cyber physical system domain.

### 5.9.4. Comparison with Related Approaches

Our parameter robustness analysis advances beyond conventional practices in learning based falsification. Traditional grid search methods provide no inherent robustness guarantees, often identifying fragile parameter configurations that degrade sharply with minor variations. Bayesian optimization approaches, while sample efficient, similarly focus on optimality rather than robustness. In contrast our methodology explicitly considers robustness during parameter selection, identifying regions of stable performance. This approach aligns with emerging best practices in robust hyperparameter optimization for safety-critical applications where stable performance under uncertainty often outweighs peak performance under ideal conditions.

### 5.10. Model Performance and Stability Analysis

The preceding sections present detailed results for each benchmark model individually. To offer a consolidated assessment of efficiency and robustness across the diverse cyber-physical systems studied, Table 23 provides a comparative summary for one representative complex property from each model. The selected properties are  $\phi_3$  for the CARS model,  $\phi_5$  for the AT model, and  $\phi_7$  for the PTC model. The comparison evaluates two key practical metrics: sample efficiency, measured by the median number of episodes required to find a counterexample, and computational efficiency, measured by the average wall-clock time for a successful falsification run.

**Table 23.** Cross-model performance summary on representative complex properties.

| Method       | CARS $\phi_3$ |                 | AT $\phi_5$ |                   | PTC $\phi_7$ |                   |
|--------------|---------------|-----------------|-------------|-------------------|--------------|-------------------|
|              | Episodes      | Time (min)      | Episodes    | Time (min)        | Episodes     | Time (min)        |
| A3C-BiT-MPR  | 8             | 4.2 (3.8, 4.6)  | 12          | 7.1 (6.5, 7.7)    | 20           | 8.5 (7.7, 9.3)    |
| DDQN-BiT-MPR | 10            | 5.1 (4.6, 5.6)  | 14          | 7.8 (7.1, 8.5)    | 25           | 9.3 (8.4, 10.2)   |
| PPO-LSTM     | 22            | 6.5 (5.8, 7.2)  | 28          | 10.5 (9.5, 11.5)  | 45           | 12.6 (11.3, 13.9) |
| BO           | 45            | 9.8 (8.8, 10.8) | 60          | 16.0 (14.2, 17.8) | 75           | 21.7 (19.3, 24.1) |
| A3C-BB       | –             | –               | 35          | 13.5 (11.9, 15.1) | 85           | 25.4 (22.3, 28.5) |
| SA           | 30            | 8.0 (7.1, 8.9)  | 40          | 13.0 (11.5, 14.5) | 50           | 18.1 (16.0, 20.2) |

To address the stochastic nature of deep reinforcement learning and provide statistical rigor, Table 23 reports 95% confidence intervals for the wall-clock time measurements. These

intervals were calculated from 100 independent runs for the CARS and AT models, and 20 independent runs for the more computationally intensive PTC model. The confidence intervals quantify the uncertainty in the mean time estimates due to variations across different random seeds and initial conditions.

The consolidated results presented in Table 23 demonstrate clear and consistent advantages of the proposed DRL-BiT-MPR framework over the baseline methods.

With respect to sample efficiency, both variants of the proposed method required substantially fewer episodes to achieve falsification across all three models. For instance, on the CARS model property  $\phi_3$ , the A3C-BiT-MPR agent found a counterexample in a median of 8 episodes. This represents a reduction by a factor of 2.8 compared to the PPO-LSTM agent and a factor of 5.6 compared to the Bayesian optimization baseline. Similar substantial reductions are observed for the AT and PTC models, confirming the generalizability of the sample efficiency gain.

Regarding computational efficiency, the total wall-clock time per successful run presents a compelling trade-off. Although each simulation step with the BiT network entails higher computational overhead, the drastic reduction in the number of required episodes results in lower overall runtimes. The A3C-BiT-MPR agent completed its search approximately 35 percent faster than the PPO-LSTM agent and more than twice as fast as the Bayesian optimization baseline across all benchmarks. The A3C-BB agent, while computationally inexpensive per step, failed to find a counterexample for CARS  $\phi_3$  within the budget and required the longest time on the other models due to its very poor sample efficiency.

This cross-model analysis confirms that the performance improvements are not artefacts of a specific model or property. The DRL-BiT-MPR framework delivers superior sample efficiency, which translates into robust and practical computational performance across continuous, hybrid, and control-oriented cyber-physical system models.

#### 5.11. Summary of Key Results and Findings

The experimental evaluation conducted across three canonical CPS benchmarks yields consolidated findings that substantiate the efficacy, robustness, and generalizability of the DRL-BiT-MPR framework.

The proposed framework demonstrates superior overall performance. It consistently outperforms all baseline methods, including advanced falsification-specific optimizers and modern POMDP-aware reinforcement learning agents. The aggregate experimental results, presented across Tables 5, 10 and 15 for success rates, and Tables 6, 11 and 16 for sample efficiency, quantify this advantage. The framework achieves an average improvement of 39.6 percent in falsification success rate while reducing the required number of simulations by more than 50.2 percent.

The ablation studies provide definitive evidence for the synergistic contribution of the core components. Results in Tables 7, 12 and 17 show that removing either the bidirectional temporal network or the multi-granularity reward function leads to significant performance degradation. The BiT network is critical for state estimation under partial observability, particularly for properties with complex temporal dependencies. The MPR function is essential for overcoming reward sparsity and ensuring efficient policy convergence on long-horizon tasks. Their integrated operation is fundamental to the framework's success.

The framework exhibits notable robustness to practical implementation factors. The prediction error sensitivity analysis confirms that the integrated mitigation strategy maintains high performance even under significant prediction inaccuracies. Furthermore, the parameter sensitivity analysis reported in Tables 21 and 22 demonstrates that performance

degradation remains bounded when key parameters deviate from their optimal values, indicating stable operating regions rather than fragile optima.

Finally, the consistent performance gains observed across models with fundamentally different dynamics—the continuous CARS model, the hybrid AT model, and the controller-based PTC model—underscore the generalizability of the approach. This consistency, maintained even under varied sampling intervals as shown in Tables 10 and 11, supports the conclusion that the design principles are broadly applicable to black-box CPS falsification under partial observability.

In conclusion, the empirical results validate that the synergistic integration of bidirectional temporal perception and structured multi-scale reward guidance effectively addresses the core challenges of applying deep reinforcement learning to partially observable CPS falsification, as reflected in the demonstrated gains across all evaluated metrics.

## 6. Conclusions

To address insufficient temporal modeling and sparse reward signals, two key challenges in deep reinforcement learning-based cyber-physical system falsification, this study proposes the DRL-BiT-MPR method. This method integrates a bidirectional temporal network and a multi-granularity reward function to achieve collaborative optimization, and its effectiveness is validated through experiments on three mainstream cyber-physical system benchmark models, including CARS, AT and PTC.

A bidirectional temporal (BiT) network is introduced to construct an enriched state representation by fusing historical observations with predicted future trajectories. This directly mitigates the state uncertainty inherent in partially observable falsification tasks.

A multi-granularity progress reward (MPR) function is designed to decompose the sparse, long-horizon falsification signal into dense feedback across multiple temporal scales. This mechanism effectively accelerates policy optimization and guides exploration.

The proposed framework demonstrates consistent and superior performance across heterogeneous CPS benchmarks. It achieves higher falsification success rates and greater sample efficiency compared to a range of baseline methods, including advanced DRL agents and Bayesian optimization, particularly for complex temporal properties.

Ablation studies confirm the indispensable and synergistic roles of the BiT network and MPR function. Their integrated co-design is validated as the key to overcoming the intertwined challenges of temporal reasoning and reward sparsity.

In summary, the DRL-BiT-MPR framework establishes a robust and effective approach for falsification under partial observability, advancing the state of the art in learning-based verification for cyber-physical systems.

Despite the aforementioned advances, this work has inherent limitations. The sequence length parameter of the bidirectional temporal network currently depends on manual grid search, and the reward weights of the multi-granularity reward function are statically configured without dynamic adaptation. Future research will focus on adaptive optimization of these key parameters and the design of a dynamic weight adjustment mechanism for the multi-granularity reward function, aiming to further improve the practicality and generalizability of the method across diverse cyber-physical system domains.

**Author Contributions:** Y.X. conducted all experiments, prepared all figures, and wrote and revised the manuscript. T.S., X.Y. and J.X. reviewed the manuscript and provided valuable feedback for revision. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant No. LY22F020019, Public-welfare Technology Application Research of Zhejiang Province in China under Grant LGG22F020032, the Zhejiang Science and Technology Plan Project under Grant

No. 2022C01045, and the National Natural Science Foundation of China under Grants (No. 61101111 and 62132014).

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Lee, E.A. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*; IEEE: Piscataway, NJ, USA, 2008; pp. 363–369.
2. Guo, J.; Li, L.; Wang, J.; Li, K. Cyber-physical system-based path tracking control of autonomous vehicles under cyber-attacks. *IEEE Trans. Ind. Inform.* **2022**, *19*, 6624–6635.
3. Rungger, M.; Tabuada, P. A notion of robustness for cyber-physical systems. *IEEE Trans. Autom. Control* **2015**, *61*, 2108–2123.
4. Carlson, J.M.; Doyle, J. Complexity and robustness. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 2538–2545.
5. Zhang, Z.; Lyu, D.; Arcaini, P.; Ma, L.; Hasuo, I.; Zhao, J. Falsifai: Falsification of ai-enabled hybrid control systems guided by time-aware coverage criteria. *IEEE Trans. Softw. Eng.* **2022**, *49*, 1842–1859.
6. Kokash, N. An introduction to heuristic algorithms. *Dep. Inform. Telecommun.* **2005**, *1*, 1–7.
7. Fainekos, G.E.; Pappas, G.J. Robustness of temporal logic specifications for continuous-time signals. *Theor. Comput. Sci.* **2009**, *410*, 4262–4291.
8. Yamagata, Y.; Liu, S.; Akazaki, T.; Duan, Y.; Hao, J. Falsification of cyber-physical systems using deep reinforcement learning. *IEEE Trans. Softw. Eng.* **2020**, *47*, 2823–2840.
9. Bai, Y.; Lv, Y.; Zhang, J. Smart mobile robot fleet management based on hierarchical multi-agent deep Q network towards intelligent manufacturing. *Eng. Appl. Artif. Intell.* **2023**, *124*, 106534.
10. Kim, S.; Park, K.J.; Lu, C. A survey on network security for cyber-physical systems: From threats to resilient design. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1534–1573.
11. Wang, S.; Ko, R.K.; Bai, G.; Dong, N.; Choi, T.; Zhang, Y. Evasion attack and defense on machine learning models in cyber-physical systems: A survey. *IEEE Commun. Surv. Tutor.* **2023**, *26*, 930–966.
12. Presekal, A.; Ştefanov, A.; Semertzis, I.; Palensky, P. Spatio-temporal advanced persistent threat detection and correlation for cyber-physical power systems using enhanced GC-LSTM. *IEEE Trans. Smart Grid* **2024**, *16*, 1654–1666.
13. Corso, A.; Moss, R.; Koren, M.; Lee, R.; Kochenderfer, M. A survey of algorithms for black-box safety validation of cyber-physical systems. *J. Artif. Intell. Res.* **2021**, *72*, 377–428.
14. Hu, H.; Song, S.; Huang, G. Self-attention-based temporary curiosity in reinforcement learning exploration. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 5773–5784.
15. Bing, Z.; Zhou, H.; Li, R.; Su, X.; Morin, F.O.; Huang, K.; Knoll, A. Solving robotic manipulation with sparse reward reinforcement learning via graph-based diversity and proximity. *IEEE Trans. Ind. Electron.* **2022**, *70*, 2759–2769.
16. Chan, C.C.; Yang, C.Z.; Fan, C.F. Security verification for cyber-physical systems using model checking. *IEEE Access* **2021**, *9*, 75169–75186.
17. Khlaif, F.H.; Khairullah, S.S. A survey on formal verification approaches for dependable systems. *arXiv* **2022**, arXiv:2204.12913.
18. Novak, M.; Nyman, U.M.; Dragicevic, T.; Blaabjerg, F. Statistical model checking for finite-set model predictive control converters: A tutorial on modeling and performance verification. *IEEE Ind. Electron. Mag.* **2019**, *13*, 6–15.
19. Zhang, X.; Ding, Y.; Zhao, H.; Yi, L.; Guo, T.; Li, A.; Zou, Y. Mixed skewness probability modeling and extreme value predicting for physical system input-output based on full bayesian generalized maximum-likelihood estimation. *IEEE Trans. Instrum. Meas.* **2023**, *73*, 2504516.
20. Modrák, M.; Moon, A.H.; Kim, S.; Bürkner, P.; Huurre, N.; Faltejsková, K.; Gelman, A.; Vehtari, A. Simulation-based calibration checking for Bayesian computation: The choice of test quantities shapes sensitivity. *Bayesian Anal.* **2023**, *20*, 461.
21. Kulik, T.; Dongol, B.; Larsen, P.G.; Macedo, H.D.; Schneider, S.; Tran-Jørgensen, P.W.; Woodcock, J. A survey of practical formal methods for security. *Form. Asp. Comput.* **2022**, *34*, 1–39.
22. Kazemi, Z.; Safavi, A.A.; Arefi, M.M.; Naseri, F. Finite-time secure dynamic state estimation for cyber-physical systems under unknown inputs and sensor attacks. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 4950–4959.
23. Zhang, L.; Yu, H.; Wang, C.; Hu, Y.; He, W.; Yu, D. A digital solution for CPS-based machining path optimization for CNC systems. *J. Intell. Manuf.* **2025**, *36*, 1261–1290.
24. Yu, Z.; Gao, H.; Cong, X.; Wu, N.; Song, H.H. A survey on cyber-physical systems security. *IEEE Internet Things J.* **2023**, *10*, 21670–21686.
25. Arrieta, A.; Wang, S.; Markiegi, U.; Sagardui, G.; Etxeberria, L. Employing multi-objective search to enhance reactive test case generation and prioritization for testing industrial cyber-physical systems. *IEEE Trans. Ind. Inform.* **2017**, *14*, 1055–1066.

26. Skovbekk, J.; Laurenti, L.; Frew, E.; Lahijanian, M. Formal Verification of Unknown Dynamical Systems Via Gaussian Process Regression. *IEEE Trans. Autom. Control* **2025**, *70*, 4960–4975.
27. Bładek, I.; Krawiec, K. Counterexample-driven genetic programming for symbolic regression with formal constraints. *IEEE Trans. Evol. Comput.* **2022**, *27*, 1327–1339.
28. Liao, M.; Wen, H.; Yang, L.; Wang, G.; Xiang, X.; Liang, X. Improving the model robustness of flood hazard mapping based on hyperparameter optimization of random forest. *Expert Syst. Appl.* **2024**, *241*, 122682.
29. Ramezani, Z.; Claessen, K.; Smallbone, N.; Fabian, M.; Åkesson, K. Testing cyber-physical systems using a line-search falsification method. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2021**, *41*, 2393–2406.
30. Mancini, T.; Melatti, I.; Tronci, E. Optimizing highly-parallel simulation-based verification of cyber-physical systems. *IEEE Trans. Softw. Eng.* **2023**, *49*, 4443–4455.
31. Ma, S.; Ruan, J.; Du, Y.; Bucknall, R.; Liu, Y. An end-to-end deep reinforcement learning based modular task allocation framework for autonomous mobile systems. *IEEE Trans. Autom. Sci. Eng.* **2024**, *22*, 1519–1533.
32. Liu, Y.; Yin, T.; Fan, S.; Deng, C.; Zheng, B.C.; Lin, P. Distributed Adaptive State Estimation for CPSs Under Deception Attacks. *IEEE Trans. Ind. Inform.* **2025**, *21*, 8903–8912.
33. Lindemann, B.; Maschler, B.; Sahlab, N.; Weyrich, M. A survey on anomaly detection for technical systems using LSTM networks. *Comput. Ind.* **2021**, *131*, 103498.
34. Wang, X.; Jin, Y.; Schmitt, S.; Olhofer, M. Recent advances in Bayesian optimization. *ACM Comput. Surv.* **2023**, *55*, 1–36.
35. Barbiero, P.; Ciravegna, G.; Giannini, F.; Lió, P.; Gori, M.; Melacci, S. Entropy-based logic explanations of neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 6046–6054.
36. Kim, H.; Lee, B.S.; Shin, W.Y.; Lim, S. Graph anomaly detection with graph neural networks: Current status and challenges. *IEEE Access* **2022**, *10*, 111820–111829.
37. Weissenbacher, M.; Borovkyh, A.; Rigas, G. Reinforcement learning of chaotic systems control in partially observable environments. *Flow Turbul. Combust.* **2025**, *115*, 1357–1378.
38. Stanly Jayaprakash, J.; Priyadarsini, M.J.P.; Parameshachari, B.; Karimi, H.R.; Gurumoorthy, S. Deep q-network with reinforcement learning for fault detection in cyber-physical systems. *J. Circuits Syst. Comput.* **2022**, *31*, 2250158.
39. Hennig, J.A.; Romero Pinto, S.A.; Yamaguchi, T.; Linderman, S.W.; Uchida, N.; Gershman, S.J. Emergence of belief-like representations through reinforcement learning. *PLOS Comput. Biol.* **2023**, *19*, e1011067.
40. Phan, T.; Ritz, F.; Altmann, P.; Zorn, M.; Nüßlein, J.; Kölle, M.; Gabor, T.; Linnhoff-Popien, C. Attention-based recurrence for multi-agent reinforcement learning under stochastic partial observability. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; PMLR: Brookline, MA, USA, 2023; pp. 27840–27853.
41. Diaz, H.; Desrochers, A.A. Modeling of nonlinear discrete-time systems from input-output data. *Automatica* **1988**, *24*, 629–641.
42. Song, S.; Zhu, M.; Dai, X.; Gong, D. Model-free optimal tracking control of nonlinear input-affine discrete-time systems via an iterative deterministic Q-learning algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 999–1012.
43. Ouaknine, J.; Worrell, J. On the decidability of metric temporal logic. In *20th Annual IEEE Symposium on Logic in Computer Science (LICS'05)*; IEEE: Piscataway, NJ, USA, 2005; pp. 188–197.
44. Lin, S.; Manerkar, Y.A.; Lohstroh, M.; Polgreen, E.; Yu, S.J.; Jerad, C.; Lee, E.A.; Seshia, S.A. Towards building verifiable CPS using lingua franca. *ACM Trans. Embed. Comput. Syst.* **2023**, *22*, 1–24.
45. Zhu, W.; Milanović, J.V. Assessment of the robustness of cyber-physical systems using small-worldness of weighted complex networks. *Int. J. Electr. Power Energy Syst.* **2021**, *125*, 106486.
46. Tabuada, P.; Caliskan, S.Y.; Rungger, M.; Majumdar, R. Towards robustness for cyber-physical systems. *IEEE Trans. Autom. Control* **2014**, *59*, 3151–3163.
47. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38.
48. Hao, Y.; Chen, M.; Gharavi, H.; Zhang, Y.; Hwang, K. Deep reinforcement learning for edge service placement in softwarized industrial cyber-physical system. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5552–5561.
49. Lei, K.; Guo, P.; Zhao, W.; Wang, Y.; Qian, L.; Meng, X.; Tang, L. A multi-action deep reinforcement learning framework for flexible Job-shop scheduling problem. *Expert Syst. Appl.* **2022**, *205*, 117796.
50. Boute, R.N.; Gijbrecchts, J.; Van Jaarsveld, W.; Vanvuchelen, N. Deep reinforcement learning for inventory control: A roadmap. *Eur. J. Oper. Res.* **2022**, *298*, 401–412.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.