STATISTICAL CERTIFICATION OF NEURAL BARRIER FUNCTIONS FOR INVARIANCE AND REACH-AVOIDANCE IN LEARNING-ENABLED SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Certifying safety and reach-avoidance for learning-enabled systems with unknown dynamics is challenging, as exact verification is intractable and finite data provide limited evidence. We propose a joint training-and-certification framework that combines multi-step statistical certification of neural barrier functions with reinforcement learning of control policies. Within this framework, PID-barrier feedback jointly trains the policy and barrier to improve safety and performance. After training, we certify probably approximately correct (PAC) multi-step safe invariant and reach-avoid sets using scenario approaches and the trained barrier function. Because the PAC statement bounds the portion of the certified set that may violate the property, rather than certifying individual states, the guarantees are reusable online as long as the system remains within the certified set and the dynamics and controller remain unchanged. The method attains $O(1/\alpha)$ sample complexity, improving on $O(1/\alpha^2)$ Monte Carlo bounds. Experiments on an inverted pendulum, Van der Pol oscillator, and 10-D Hopfield system demonstrated improved return-safety trade-offs and valid PAC certificates.

1 INTRODUCTION

Learning-enabled control systems are increasingly deployed in safety-critical settings, where reliable safety and reach-avoidance guarantees are essential. However, model-free reinforcement learning (RL) poses fundamental challenges for such guarantees: when the system dynamics are black-box, exact guarantees are generally intractable, while finite trajectory data provide only limited empirical evidence of safety and reach-avoidance. We address this gap by developing PAC certificates for multi-step invariance and reach-avoidance, together with a learning framework that explicitly reduces certification violations during policy optimization.

Barrier functions offer a principled framework for addressing these challenges by reducing the certification of such guarantees to functional inequalities Prajna & Jadbabaie (2004); Prajna et al. (2007). Classical barrier-function-based methods, however, typically require explicit system models. Recently, Wu et al. (2026) combined barrier-function-based approaches with scenario methods Campi et al. (2009), which replace robust constraints with constraints enforced over a finite set of sampled scenarios, to certify one-step PAC controlled invariant sets for black-box system. It is a set in which a specified proportion of states can be controlled to remain within it at the next step with at least a user-specified confidence level. This set is quit useful for online monitoring and control: if the system is observed to be within the set at the next step, the PAC invariance and safety guarantees for the set can be reapplied to the system. More detailed explanations of this recursive application of the PAC guarantees in an online setting are provided in Section 2. However, Wu et al. (2026) focused only on one-step invariance, do not address controller synthesis, and impose restrictive convex parameterizations on barrier functions.

To address these limitations, we propose a joint training-and-certification framework that integrates multi-step statistical certification of neural barrier functions with RL-based control-policy learning. During learning, a stochastic policy and a neural barrier function are learned jointly from rollout data. Barrier losses encode invariant or reach-avoid conditions, while the policy is optimized using a barrier-aware proximal policy optimization (PPO) objective. To stabilize the joint training process,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

proportional-integral-derivative (PID) multipliers adaptively incorporate empirical barrier violations into the learning objective. After training, the learned policy and barrier function are frozen and certified using an independent dataset. We formulate certification as a scalar optimization problem with violation slack. Under a fixed reference measure, the resulting normalized violation measure yields (α, β)-PAC guarantees for multi-step safe invariant and reach-avoid sets. Crucially, these are set-level guarantees rather than pointwise state-level certificates: the PAC statement bounds the portion of the certified set that may violate the property, and does not assert that any individual queried state is safe or reaches the target. Consequently, the guarantee is attached to the certified set as a whole and can be reapplied online: as long as the observed state remains within the certified set and the system dynamics and controller remain unchanged, the same PAC guarantee holds. This leads to an $O(1/\alpha)$ sample complexity, improving over the $O(1/\alpha^2)$ dependence of standard Monte Carlo estimates. Finally, experiments on an inverted pendulum, the Van der Pol oscillator, and a 10-dimensional Hopfield system demonstrate improved safety-performance trade-offs and empirically validate the resulting PAC certificates.

The main contributions of this work are summarized as follows.

- **Reusable multi-step PAC certification.** We establish (α, β)-PAC guarantees for multi-step safe invariant sets and reach-avoid sets, characterizing the proportion of states within the certified set that satisfy the corresponding properties. Because the guarantee is attached to the fixed closed-loop system and certified set, it can be reused as long as the closed-loop system remain unchanged.
- **Flexible scalar scenario optimization for certification.** For a fixed controller and neural barrier function, we formulate certification as a scalar scenario optimization problem incorporating violation slack. A single independent certification dataset supports a family of nested admissible certification sets, eliminating the need for retraining across different thresholds.
- **Superior sample complexity.** We theoretically establish a certification sample complexity of $O(1/\alpha)$ for a fixed confidence level which offers better sample efficiency than the $O(1/\alpha^2)$ complexity associated with traditional Monte Carlo probability bounds.
- **PID-Barrier guided joint learning.** By using empirical barrier violations as feedback signals for PID multipliers, we jointly train the control policy and the neural barrier function to enhance policy performance and safety while reducing certification violations. We evaluate the framework's control performance, safety, and certification results using the inverted pendulum, Van der Pol oscillator, and a 10-dimensional Hopfield system.

Related work. Barrier-function-based methods, which are based on functional inequalities, have long been used to certify safety, invariance, and reach-avoidance properties of dynamical systems Kushner (1967); Prajna et al. (2007); Xue et al. (2023); Zhi et al. (2024); Chen et al. (2025). Classical optimization methods, such as sum-of-squares (SOS) optimization, and synthesis-based methods, such as counterexample-guided inductive synthesis (CEGIS), provide powerful tools for searching for functions in the form of polynomials or neural networks that satisfy the resulting functional inequalities Parrilo (2000); Papachristodoulou & Prajna (2002); Prajna & Jadbabaie (2004); Ahmed et al. (2020); Zhao et al. (2020); Mathiesen et al. (2022); Mazouz et al. (2022); Chatterjee et al. (2023); Zhao et al. (2024); Zhi et al. (2024); Edwards et al. (2024); Badings et al. (2025). However, these computational methods typically require explicit system models, limiting their applicability to black-box systems. Scenario approaches (Calafiore & Campi, 2006; Campi et al., 2009) have enabled barrier-function-based methods to be applied to black-box systems for which explicit models of the dynamics are unavailable Xue et al. (2019); Rickard et al. (2026); Wu et al. (2026). It has also been used for PAC reachablility analysis, e.g., (Xue et al., 2020; Devonport & Arcak, 2020; Dietrich et al., 2024). Given an initial set, Xue et al. (2019); Rickard et al. (2026) certified, in the PAC sense, the proportion of states in the set from which the system satisfies the safety or reachability properties of interest. Rather than requiring a pre-specified initial set, Wu et al. (2026) computed a one-step PAC controlled invariant set, which provides a step-by-step safety guarantee that can be applied online as the system evolves. Building on this ides, we extend this notion to multi-steps PAC safe invariant sets and reach-avoid sets. Furthermore, whereas Wu et al. (2026) does not synthesize controllers and is restricted to barrier functions that are convex in the unknown parameters, our method certifies multi-step PAC safe invariant sets and reach-avoid sets for learned controllers and barrier functions. Consequently, our approach is not limited to one-step invariance or convex barrier-function parameterizations and can accommodate learned controllers and barrier functions within flexible function class, such as neural networks.

Besides, constrained policy optimization commonly balances return against a cost criterion (Ray et al., 2019; Tessler et al., 2018). Model-free neural barrier learning instead uses transition-based barrier losses to guide policy improvement (Yang et al., 2023). PID-Lagrangian methods introduce proportional and derivative feedback to reduce oscillations in constraint multipliers (Stooke et al., 2020). In this work, we combine barrier-based feedback with a PID multiplier to improve the candidates used by the certification stage.

Notations. The following notations are used throughout this paper: $\mathbb{R}$ denotes the set of real values; $\mathbb{R}^n$ and $\mathbb{N}$ denote the set of n-dimensional real vectors and non-negative integers, respectively; for sets Δ_1 and Δ_2, $\Delta_1 \setminus \Delta_2$ denotes the set of all elements in Δ_1 that are not in Δ_2.

2 PRELIMINARIES

In this section we present the black-box discrete-time system, PAC multi-step invariant sets and reach-avoid sets, and the problem of interest in this paper.

2.1 BLACK-BOX DISCRETE-TIME SYSTEMS

We consider the black-box discrete-time system

$$\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \qquad \forall t \in \mathbb{N}, \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state, $\boldsymbol{u}(t) \in U \subseteq \mathbb{R}^s$ is the control input, U is compact, and the dynamics $\boldsymbol{f} : \mathbb{R}^n \times U \to \mathbb{R}^n$ are unknown.

Given a controller $\boldsymbol{u} : \mathbb{R}^n \to U$ and an initial state $\boldsymbol{x}_0 \in \mathbb{R}^n$, the induced **trajectory** of system (1) is the map $\boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}_0} : \mathbb{N} \to \mathbb{R}^n$ defined by $\boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}_0}(0) = \boldsymbol{x}_0$ and $\boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}_0}(l+1) = \boldsymbol{f}\big(\boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}_0}(l), \boldsymbol{u}\big(\boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}_0}(l)\big)\big), \forall l \in \mathbb{N}$.

Let $\mathcal{S} \subseteq \mathbb{R}^n$ denote the safe set and let $\mathcal{T} \subseteq \mathcal{S}$ denote the target set. We consider specifications relative to these sets and assume that the system can be simulated over an arbitrary horizon for any controller and any initial state in $\mathcal{S}$, so that the visited states can be recorded.

Assumption 1. *For any controller $\boldsymbol{u}$ and initial state $\boldsymbol{x} \in \mathcal{S}$, we simulate system (1) over an arbitrary horizon and record the visited states $\boldsymbol{y}_i := \boldsymbol{\phi}_u^{\boldsymbol{x}}(i)$, $i \in \mathbb{N}$.*

In safety- and performance-critical systems, safe invariant sets and reach-avoid sets play a central role: safe invariant sets characterize regions from which the system remains within safe indefinitely, while reach-avoid sets characterize regions from which the system can safely reach a target.

Definition 1. *For a controller $\boldsymbol{u}$, a set $\widetilde{\mathcal{S}} \subseteq \mathcal{S}$ is an **safe invariant set** if $\forall \boldsymbol{x} \in \widetilde{\mathcal{S}},\ \forall k \in \mathbb{N} : \boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}}(k) \in \widetilde{\mathcal{S}}$. It is a **reach-avoid set** with respect to $\mathcal{T}$ if $\forall \boldsymbol{x} \in \widetilde{\mathcal{S}},\ \exists l \in \mathbb{N} : \boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}}(l) \in \mathcal{T}, \boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}}(i) \in \widetilde{\mathcal{S}} \setminus \mathcal{T},\ \forall i < l$.*

Definition 1 gives the deterministic specification. With unknown dynamics and finite trajectory data, computing invariant or reach-avoid sets exactly is generally infeasible. We therefore certify multi-step invariant and reach-avoid sets in the PAC sense, as formalized in the subsection below.

2.2 PAC CERTIFICATION

In this subsection, we introduce the concepts of PAC multi-step invariant sets and reach-avoid sets in Definitions 3 and 4, respectively, and the certification problem of interest in this paper. We first introduce the reference measure.

Definition 2 (Reference measure $\mu_{\mathcal{S}}$)**.** *Let $(\mathcal{S}, \mathcal{F}_{\boldsymbol{x}})$ be a measurable space with $0 < \mathrm{vol}(\mathcal{S}) < \infty$, where $\mathrm{vol}(\mathcal{S})$ denotes the n-dimensional Lebesgue measure. The normalized reference measure $\mu_{\mathcal{S}}$ on $\mathcal{S}$ is defined by*

$$\mu_{\mathcal{S}}(A) = \frac{\mathrm{vol}(A \cap \mathcal{S})}{\mathrm{vol}(\mathcal{S})}, \qquad A \in \mathcal{F}_{\boldsymbol{x}}.$$

Thus, $\mu_{\mathcal{S}}$ measures the relative volume of subsets of $\mathcal{S}$ and is used exclusively as a geometric reference measure, independent of the state distribution of the system.

Definition 3 (PAC Multi-Step Safe Invariant Sets)**.** *Given a fixed controller $\boldsymbol{u}$, a safe set $\mathcal{S}$, a finite horizon $k_{\max} \in \mathbb{N}$, a probability threshold $\alpha \in (0,1)$, and a confidence level $1-\beta \in (0,1)$, a data-dependent set $\widetilde{\mathcal{S}}(D_N) \subseteq \mathcal{S}$, abbreviated as $\widetilde{\mathcal{S}}$, is called an (α, β)-PAC multi-step safe invariant set*

for the system (1) if

$$\mathbb{P}_{D_N}\left[\mu_{\mathcal{S}}(\widetilde{\mathcal{S}})>0 \Longrightarrow \mu_{\mathcal{S}}\left(\left\{\boldsymbol{x}\in\widetilde{\mathcal{S}}:\bigwedge_{i=1}^{k_{\max}}\left(\boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}}(i)\in\widetilde{\mathcal{S}}\right)\right\}\middle|\,\widetilde{\mathcal{S}}\right)\geq 1-\frac{\alpha}{\mu_{\mathcal{S}}(\widetilde{\mathcal{S}})}\right]\geq 1-\beta,$$

where $D_N=\{\boldsymbol{x}_i\}_{i=1}^N$ consists of i.i.d. samples drawn according to $\mu_{\mathcal{S}}$, and $\mathbb{P}_{D_N}$ denotes the probability measure induced over the random sample set D_N. That is, with confidence at least $1-\beta$ over the sample set D_N, if $\mu_{\mathcal{S}}(\widetilde{\mathcal{S}})>0$, then the relative $\mu_{\mathcal{S}}$-measure of states in $\widetilde{\mathcal{S}}$ from which the system (1) remains inside $\widetilde{\mathcal{S}}$ for the next $k_{\max}$ time steps is at least $1-\alpha/\mu_{\mathcal{S}}(\widetilde{\mathcal{S}})$.

Definition 4 (PAC Multi-Step Reach-Avoid Sets)**.** *Given a controller $\boldsymbol{u}$, a safe set $\mathcal{S}$, a target set $\mathcal{T}\subseteq\mathcal{S}$, a finite horizon $k_{\max}\in\mathbb{N}$, a probability threshold $\alpha\in(0,1)$, and a confidence level $1-\beta\in(0,1)$, a data-dependent set $\widetilde{\mathcal{S}}(D_N)(abbr.,\widetilde{\mathcal{S}})\subseteq\mathcal{S}$ is called an (α,β)-PAC multi-step reach-avoid set for the system (1) if*

$$\mathbb{P}_{D_N}\left(\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}]>0 \implies \mu_{\mathcal{S}}\left[\begin{array}{l}\exists\, l\in\{0,\dots,k_{\max}\}\text{ s.t. }\boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}}(l)\in\mathcal{T},\\ \boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}}(i)\in\widetilde{\mathcal{S}}\setminus\mathcal{T},\ \forall i<l\end{array}\middle|\ \boldsymbol{x}\in\widetilde{\mathcal{S}}\right]\geq 1-\frac{\alpha}{\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}]}\right)\geq 1-\beta.$$

That is, with confidence at least $1-\beta$ over the random dataset D_N, if $\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}]>0$, then among all states in $\widetilde{\mathcal{S}}$, the fraction from which the system (1) reaches the target set $\mathcal{T}$ within $k_{\max}$ steps while remaining inside $\widetilde{\mathcal{S}}$ prior to the first hitting time is at least $1-\frac{\alpha}{\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}]}$.

The main advance over existing PAC notions Xue et al. (2019; 2020); Dietrich et al. (2024); Rickard et al. (2026) lies in our reusable nature. With confidence at least $1-\beta$ over the certification data, the absolute $\mu_{\mathcal{S}}$-measure of states in $\widetilde{\mathcal{S}}$ that violate the specified $k_{\max}$-step invariance or reach-avoid property is at most α. Consequently, whenever $\mu_{\mathcal{S}}(\widetilde{\mathcal{S}})>0$, conditioning on membership in $\widetilde{\mathcal{S}}$ yields the normalized lower bound $1-\alpha/\mu_{\mathcal{S}}(\widetilde{\mathcal{S}})$ on the proportion of states in $\widetilde{\mathcal{S}}$ that satisfy the property. This is a set-level guarantee: once $\widetilde{\mathcal{S}}$ has been certified, the same guarantee can be reused whenever the system operates within $\widetilde{\mathcal{S}}$, without requiring recertification of the set.

Remark 1. *In practical computations, the set $\widetilde{\mathcal{S}}$ may be complex, making it challenging to compute the probability measure $\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}]$ exactly. One approach to address this challenge is to use Monte Carlo techniques which provide an estimate.*

Problem. In this work, we aim to learn a controller $\boldsymbol{u}$ within a safe RL framework, described in Subsection 3.1.2, and to compute an associated set $\widetilde{\mathcal{S}}\subseteq\mathcal{S}$ that satisfies a prescribed (α,β)-PAC multi-step safe invariant and reach-avoid guarantee. The controller is trained to maximize a specified reward while incorporating safety constraints through a neural barrier function. After training, the learned controller and barrier function are fixed, and the scenario approach introduced in Appendix B is employed to certify the set $\widetilde{\mathcal{S}}$.

3 PAC Multi-Step Safety and Reach-Avoid Guarantees

We develop a training-and-certification framework for learning neural barrier functions and controllers and subsequently establishing PAC multi-step safety and reach-avoid guarantees. The framework consists of two stages: joint policy–barrier training from rollout data, followed by statistical certification of the frozen policy–barrier pair using an independent set of i.i.d. initial states and their multi-step trajectories. Throughout this section, m denotes the training iteration, t the rollout time step, and $k_{\max}$ the certification horizon; (ϕ,θ,w) denote the parameters of the barrier, policy, and value networks, respectively. The frozen policy–barrier parameters are denoted by (θ^{s*},ϕ^{s*}) for invariance and $(\theta^{ra*},\phi^{ra*})$ for reach-avoidance.

3.1 Joint Training of the Policy and Neural Barrier Function

We describe how candidate policy–barrier pairs are learned. We begin by recalling the barrier-like conditions underlying the two verification tasks, introduce the neural parameterization and training losses, and then describe the PPO-based joint training procedure with the PID-Barrier update.

3.1.1 Barrier Conditions and Neural Barrier Parameterization

We first recall the barrier-like conditions, which express invariance and reach-avoidance as functional inequalities that must hold along the closed-loop dynamics. These conditions are formalised in Propositions 1 and 2, respectively.

Proposition 1. *If there exist a barrier function $B : \mathbb{R}^n \to \mathbb{R}$ and a controller $u : \mathbb{R}^n \to U$ satisfying*

$$\begin{cases} B(\boldsymbol{x}) < 0, & \forall \boldsymbol{x} \in \mathbb{R}^n \setminus \mathcal{S}, \\ B(f(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}))) - \lambda B(\boldsymbol{x}) \geq 0, & \forall \boldsymbol{x} \in \mathcal{S}, \end{cases} \tag{2}$$

where $\lambda \in (0, 1)$ is a user-specified scalar value, then $\{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x}) \geq 0\} \neq \emptyset$ is a safe invariant set, i.e., a set of initial states from which the system (1), under the controller u, remains in $\{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x}) \geq 0\}$ for all time.

Proposition 2 (Zhao et al. (2022))**.** *If there exist a bounded barrier function $B : \mathbb{R}^n \to \mathbb{R}$ and a controller $u : \mathbb{R}^n \to U$ satisfying*

$$\begin{cases} B(\boldsymbol{x}) \leq 0, & \forall \boldsymbol{x} \in \mathbb{R}^n \setminus \mathcal{S}, \\ B(f(\boldsymbol{x}, u(\boldsymbol{x}))) - \lambda B(\boldsymbol{x}) \geq 0, & \forall \boldsymbol{x} \in \mathcal{S} \setminus \mathcal{T}, \end{cases} \tag{3}$$

where $\lambda > 1$ is a user-specified scalar. Then the set $\mathcal{R} = \{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x}) > 0\} \neq \emptyset$ is a reach-avoid set, i.e., the set of initial states from which the trajectories of system (1), under the controller u, will reach the target set $\mathcal{T}$ in finite time while remaining in the safe set $\mathcal{S}$ until the target is reached. Moreover, if $B(\boldsymbol{x}) \leq G$ for all $\boldsymbol{x} \in \mathcal{S}$, then $\mathcal{R}_k = \{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x}) > \frac{G}{\lambda^k}\} \neq \emptyset$ defines a k-step reach-avoid set, i.e., a set of initial states from which the trajectories of (1), under u, will reach the target set $\mathcal{T}$ within k steps while staying inside $\mathcal{S}$ until the first hitting time.

Instead of restricting B to structured function classes such as polynomials, we represent its behavior inside the safe set by a neural network and extend it outside S using a prescribed constant:

$$B(\boldsymbol{x}; \phi) := \mathbf{1}_{\mathcal{S}}(\boldsymbol{x})\, B'(\boldsymbol{x}; \phi) + \mathbf{1}_{\mathbb{R}^n \setminus \mathcal{S}}(\boldsymbol{x})\, C, \tag{4}$$

where $C < 0$ is a user-specified constant in invariant task, $C = 0$ in reach-avoid task. $B'(\boldsymbol{x}; \phi)$ is a feedforward neural network parameterized by ϕ.

Remark 2. *The above parameterization form in (4) provides a convenient interface between training and certification: the neural network only needs to represent the barrier behavior inside the safe region, while the prescribed constant C determines the barrier value outside the safe set.*

The barrier conditions above are global functional inequalities, whereas training is performed from a finite collection of sampled transitions. We therefore construct empirical loss functions that encourage the learned barrier to satisfy the required conditions on the observed trajectories.

Let $\mathcal{D}^{(m)}$ denote the dataset collected at training iteration m, containing states, actions, rewards, and next states. For the invariant task, we define $\mathcal{D}^{(m)}_{\text{safe}} = \mathcal{D}^{(m)} \cap \mathcal{S}$ and $\mathcal{D}^{(m)}_{\text{unsafe}} = \mathcal{D}^{(m)} \cap (\mathbb{R}^n \setminus \mathcal{S})$. For the reach-avoid task, $\mathcal{D}^{(m)}_{\text{safe}}$ is instead defined as $\mathcal{D}^{(m)} \cap (\mathcal{S} \setminus \mathcal{T})$. Using the data collected over successive training iterations, we optimize the following empirical barrier loss.

Barrier Loss Design. For the invariant problem, the barrier network is trained by minimizing

$$\mathcal{L}_B(\phi) = \alpha_1\, \mathcal{L}^{(m)}_{\text{unsafe}}(\phi) + \alpha_2\, \mathcal{L}^{(m)}_{\text{safe}}(\phi) + \alpha_3\, \mathcal{L}^{(m)}_{\text{vol}}(\phi), \tag{5}$$

where $\alpha_i > 0$ are user-specified weights, $\mathcal{L}^{(m)}_{\text{vol}}(\phi) = -\frac{1}{|\mathcal{D}^{(m)}_{\text{safe}}|} \sum_{(\boldsymbol{x}, \boldsymbol{a}, r, \boldsymbol{x}') \in \mathcal{D}^{(m)}_{\text{safe}}} B(\boldsymbol{x}; \phi)$, $\mathcal{L}^{(m)}_{\text{unsafe}}(\phi) = \frac{1}{|\mathcal{D}^{(m)}_{\text{unsafe}}|} \sum_{(\boldsymbol{x}, \boldsymbol{a}, r, \boldsymbol{x}') \in \mathcal{D}^{(m)}_{\text{unsafe}}} \max\{0, B'(\boldsymbol{x}; \phi) - C\}$, and $\mathcal{L}^{(m)}_{\text{safe}}(\phi) = \frac{1}{|\mathcal{D}^{(m)}_{\text{safe}}|} \sum_{(\boldsymbol{x}, \boldsymbol{a}, r, \boldsymbol{x}') \in \mathcal{D}^{(m)}_{\text{safe}}} \max\{0, \lambda B(\boldsymbol{x}; \phi) - B(\boldsymbol{x}'; \phi)\}$. Herein, $\boldsymbol{x}'$ denotes the next state under the current policy. The loss $\mathcal{L}^{(m)}_{\text{unsafe}}$ shapes the barrier values on sampled unsafe states relative to the prescribed threshold C, $\mathcal{L}^{(m)}_{\text{safe}}$ penalizes violations of the one-step barrier condition along sampled transitions, and $\mathcal{L}^{(m)}_{\text{vol}}$ encourages larger barrier values and therefore promote larger candidate safe invariant or reach-avoid sets. For reach-avoid task, $\lambda > 1$ is used in $\mathcal{L}^{(m)}_{\text{safe}}$, and the

barrier form (3) is adopted. To handle the upper bound G required by Proposition 2, we augment (5) with an additional target-set penalty term $\alpha_4 \mathcal{L}_{\mathcal{S}}^{(m)}(\phi) = \frac{1}{|\mathcal{D}_{\mathcal{S}}^{(m)}|} \sum_{\boldsymbol{x} \in \mathcal{D}_{\mathcal{S}}^{(m)}} \max\big(0,\ B(\boldsymbol{x};\phi) - G\big)$, which penalizes barrier values exceeding G on states sampled from $\mathcal{S}$. During verification, we further clip the network output according to $B(\boldsymbol{x};\phi) \leftarrow \min\big(B(\boldsymbol{x};\phi),\ G\big)$ so that $B(\boldsymbol{x}) \leq G$ holds by construction without requiring the computation of a tight global upper bound.

3.1.2 Joint Policy–Barrier Training with PID Feedback

We next describe how exploration policy and candidate barrier are jointly trained. We adopt a PPO-based, actor-critic RL method adapted from model-free neural-barrier safe RL Yang et al. (2023). Let ρ_0 denote an initialstate distribution supported on $\mathcal{S}$, $r : \mathbb{R}^n \times U \to \mathbb{R}$ be a bounded reward function, and let $\gamma \in (0, 1)$. During training, a stochastic exploration policy $\pi_\theta(\cdot \mid \boldsymbol{x}) \in \Delta(U)$ (e.g. a squashed Gaussian) generates actions and maximizes the following discounted objective:

$$J(\theta) = \mathbb{E}_{\boldsymbol{x}(0)\sim\rho_0,\ \boldsymbol{a}(t)\sim\pi_\theta(\cdot|\boldsymbol{x}(t))}\Big[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{x}(t), \boldsymbol{a}(t))\Big]. \tag{6}$$

The training architecture consists of three neural networks. The **actor** $\pi_\theta(\cdot \mid \boldsymbol{x})$ generates stochastic actions. The **critic** $V_w(\boldsymbol{x})$ estimates the value function. In parallel, the **barrier network** $B_\phi(\boldsymbol{x})$ is optimized using the empirical barrier loss defined in Subsection 3.1.1. Thus, policy improvement and barrier learning are performed jointly from the same rollout data.

Following the method in Yang et al. (2023), we optimize the policy under a barrier-based constraint via the dual method, where the PPO importance ratio ρ_θ is introduced to correct the mismatch between rollout data collected by $\pi_{\theta_{\text{old}}}$ and updates to π_θ. The resulting optimization is given by

$$\max_\theta L_\pi^{\text{PPO}}(\theta) \quad \text{s.t.} \quad \rho_\theta\, \mathcal{L}_{\text{safe}}^{(m)}(\phi) \leq 0, \tag{7}$$

Using a Lagrangian relaxation gives the barrier-aware PPO objective

$$L_\pi(\theta) = L_\pi^{\text{PPO}}(\theta) - \nu^{(m+1)} \rho_\theta \mathcal{L}_{\text{safe}}^{(m)}(\phi) \tag{8}$$

where $\nu^{(m+1)} \geq 0$ is an adaptive multiplier. Unlike the conventional Lagrangian update, we update the multiplier using a PID-Barrier feedback rule. Let $e_m := \rho_\theta\, \mathcal{L}_{\text{safe}}^{(m)}(\phi)$. The integral and derivative states are $I_m := \max\{0, I_{m-1} + e_m\}$, $D_m := \max\{0, e_m - e_{m-1}\}$. The multiplier is updated as

$$\nu^{(m+1)} := \max\{0, K_P e_m + K_I I_m + K_D D_m\}, \tag{9}$$

where $K_P, K_I, K_D \geq 0$ are proportional, integral, and derivative gains.

The PID-Barrier multiplier is motivated by both RL and feedback control. In PPO, the importance factor and clipping mechanism partially mitigate the distribution shift between the rollout policy $\pi_{\theta_{\text{old}}}$ and the updated policy π_θ. Nevertheless, the barrier loss remains a finite-sample and nonstationary constraint signal. From a feedback control perspective, the multiplier ν acts as a control signal that regulates the constraint error e_m toward zero. **The proportional term** responds to the current constraint violation, the integral term accumulates persistent violations, and **the derivative term** responds to rapid changes in the violation signal. Compared with an integral-only Lagrangian update, the PID-Barrier rule can reduce oscillations in the multiplier during training. When $K_P = K_D = 0$, the update reduces to an integral-only Lagrangian-style multiplier update. The complete training procedure is summarized in Algorithm 1.

3.2 PAC Multi-Step Certification

The empirical barrier losses used during training enforce the barrier conditions only on sampled transitions and therefore do not by themselves provide guarantees over unseen initial states and trajectories. We address this gap through an independent post-training certification stage. Specifically, we freeze the learned policy–barrier pair and use scenario optimization to construct PAC multi-step safe invariant and reach-avoid sets from an independent set of i.i.d. initial states and their trajectories. For invariance, we use the fixed controller $\boldsymbol{u}(\cdot;\theta^{s*})$ and barrier $B(\cdot;\phi^{s*})$; for reach-avoidance, we use $\boldsymbol{u}(\cdot;\theta^{ra*})$ and $B(\cdot;\phi^{ra*})$.

(1) PAC multi-step safe invariant certification. First, we draw i.i.d. states $D_N = \{\boldsymbol{x}_i\}_{i=1}^N$ according to $\mu_{\mathcal{S}}$. For each initial state $\boldsymbol{x}_i$, we simulate system (1) under $\boldsymbol{u}(\boldsymbol{x}; \theta^{s*})$for at most $k_{\max}$ steps and set $\boldsymbol{y}_{i,0} := \boldsymbol{x}_i$, $\boldsymbol{y}_{i,j} := \boldsymbol{\phi}_u^{\boldsymbol{x}_i}(j)$. Let $k_i \le k_{\max}$ be the first time at which the trajectory exits $\mathcal{S}$, capped by the horizon $k_{\max}$. If the trajectory exits before the horizon, then $\boldsymbol{y}_{i,k_i} \notin \mathcal{S}$, otherwise $k_i = k_{\max}$. Finally, we solve the following linear program over ξ:

$$\min_{\xi} \xi \ \text{ s.t.} \begin{cases} B(\boldsymbol{y}_{i,j}; \phi^{s*}) - \lambda B(\boldsymbol{y}_{i,j-1}; \phi^{s*}) \ge -\xi, \\ \xi \in [-U_\xi; U_\xi], \\ i = 1, \dots, N; j = 1, \dots, k_i \end{cases} \tag{10}$$

where $U_\xi > 0$ is a user-defined scalar value satisfying $U_\xi \ge \max(2M, |C| + M)$, $M = \sup_{\boldsymbol{x} \in \mathcal{S}} B'(\boldsymbol{x}; \phi^{s*})$.

Theorem 1. *Let ξ^{s*} be the optimal solution to equation 10. Suppose that $\widetilde{\mathcal{S}}_\zeta := \left\{ \boldsymbol{x} \in \mathcal{S} \;\middle|\; B(\boldsymbol{x}; \phi^{s*}) \ge \frac{\zeta}{\lambda - 1} \right\}$, where $\xi^{s*} \le \zeta < (\lambda - 1)C$. If $N \ge \frac{2}{\alpha}\left(\ln \frac{1}{\beta} + 1\right)$, then $\widetilde{\mathcal{S}}_\zeta$ is an (α, β)-PAC multi-step safe invariant set in the Definition 3.*

(2) PAC multi-step reach-avoid certification. First, we draw i.i.d. states $D_N = \{\boldsymbol{x}_i\}_{i=1}^N$ according to $\mu_{\mathcal{S}}$. For each initial state $\boldsymbol{x}_i$, we simulate system (1) under $\boldsymbol{u}(\boldsymbol{x}; \theta^{ra*})$ for at most $k_{\max}$ steps and set $\boldsymbol{y}_{i,0} := \boldsymbol{x}_i$, $\boldsymbol{y}_{i,j} := \boldsymbol{\phi}_u^{\boldsymbol{x}_i}(j)$. Let $k_i \le k_{\max}$ be the first time at which the trajectory either exits $\mathcal{S}$ or reaches $\mathcal{T}$, capped by the horizon $k_{\max}$. If the trajectory stops before the horizon, then $\boldsymbol{y}_{i,k_i} \in \mathcal{T}$ or $\boldsymbol{y}_{i,k_i} \notin \mathcal{S}$, otherwise $k_i = k_{\max}$. We solve the following linear program over ξ:

$$\min_{\xi} \xi \ \text{ s.t.} \begin{cases} B(\boldsymbol{y}_{i,j}; \phi^{ra*}) - \lambda B(\boldsymbol{y}_{i,j-1}; \phi^{ra*}) \ge -\xi, \\ \xi \in [-U_\xi, U_\xi], \\ i = 1, \dots, N; j = 1, \dots, k_i \end{cases} \tag{11}$$

where $U_\xi > 0$ is a user-defined scalar value satisfying $U_\xi \ge (1 + \lambda) \sup_{\boldsymbol{x} \in \mathbb{S}} B'(\boldsymbol{x}; \phi^{ra*})$.

Theorem 2. *Let ξ^{ra*} be the optimal solution to equation 11. Suppose that $\widetilde{\mathcal{S}}_\zeta := \left\{ \boldsymbol{x} \in \mathcal{S} \;\middle|\; B(\boldsymbol{x}; \phi^{ra*}) > \frac{G}{\lambda^{k_{\max}}} + \zeta \frac{1 - \lambda^{-k_{\max}}}{\lambda - 1} \right\}$, where $\max\left(\xi^{ra*}, -\frac{G}{\lambda^{k_{\max}}}(\lambda - 1)\right) \le \zeta \le G(\lambda - 1)$. If $N \ge \frac{2}{\alpha}\left(\ln \frac{1}{\beta} + 1\right)$, then $\widetilde{\mathcal{S}}_\zeta$ is an (α, β)-PAC multi-step reach-avoid set in the Definition 4.*

The proofs of Theorem 1 and 2 are shown in Appendix C. The same calibration result supports every invariance threshold ζ satisfying $\xi^{s*} \le \zeta < (\lambda - 1)C$. Likewise, the reach-avoid calibration supports thresholds in the admissible range specified by Theorem 2. Thus, a single certification dataset supports a family of certified sets without retraining the policy or barrier. This also applies to the reach-avoid certification. Moreover, to the best of our knowledge, our framework permits a positive violation ζ, resulting in a more relaxed requirement compared with prior studies, which generally impose more stringent constraints without any counterexample guided approach.

4 Experiments

In this section, we evaluate from three perspectives: **(Q1)** Does the proposed PID-Barrier method achieve a better trade-off between control performance and safety? **(Q2)** Can our PAC framework provide meaningful multi-step formal guarantees for both invariant and reach-avoid tasks? **(Q3)** Is the proposed certification procedure more sample-efficient than Monte Carlo certification?

4.1 Experimental Setup

Environments and Tasks. We evaluate our framework on three benchmarks: an inverted pendulum, a Van der Pol, and a 10-dimensional Hopfield neural network. For each of these systems, we look at invariant and reach-avoid tasks. Further details are provided on the experiments in Appendix.

Compared Methods. We compare our method with: (1) Non-barrier baselines: PPO Schulman et al. (2017), Lag-Cost Ray et al. (2019), and RCPO Tessler et al. (2018). (2) Barrier-learning baselines : MF-NBC Yang et al. (2023) and Lag-Barrier (special $K_P = K_D = 0$ for ours); these methods learn a barrier during training, but they do not natively provide the PAC certification. Finally, (3) our method learns the policy and barrier together, using a PID-Barrier training method.

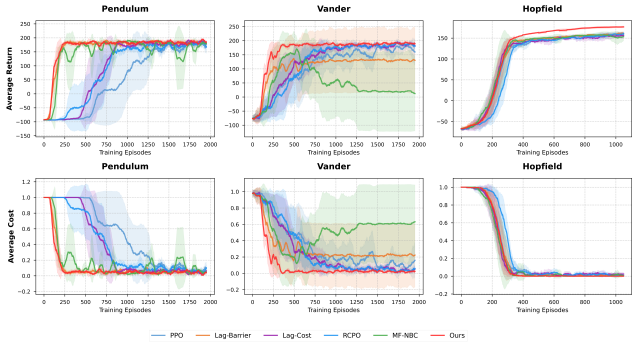

Figure 1: Average episode cost and return in the invariant task. The solid lines indicate the mean and the shaded regions indicate the 95% confidence interval across five runs.

Evaluation protocol. We evaluate each method in two stages. First, we compare learning performance using episode return and safety cost. Second, we freeze the learned policy and apply the same PAC certification procedure to every method. For policy-only baselines, we pair the learned policy with after trained barrier function; for barrier-learning baselines, we use the learned barriers directly. We also discussed how the PID parameters, horizon $k_{\max}$, sample size N, and verification bound ξ affect the training and PAC certification. The relevant results can be found in Appendix.

4.2 Control Performance and Safety Satisfaction

Fig. 1 shows that PID-Barrier achieves near-zero cost while maintaining high return on all invariance tasks. Non-barrier baselines, especially Lag-Cost and RCPO, respond more slowly because their safety feedback is based on cost rather than direct barrier preservation. Barrier-learning baselines reduce this mismatch but still have limitations: Lag-Barrier only accumulates barrier violations, and MF-NBC becomes unstable on the Van der Pol system, where sensitive nonlinear dynamics increase policy update changes. In contrast, PID-Barrier's integral term maintains pressure on persistent violations, while its derivative term dampens rapid policy updates, yielding smoother cost reductions and more stable returns. Similar trends for reach-avoid tasks are reported in the Appendix.

4.3 PAC Certification for Invariance and Reach-Avoidance

(1) Overall PAC certification performance. We now evaluate the learned fixed policies based on the PAC certification framework. Our main comparison of methods occurs at thresholds $\zeta = \xi^{s*}$ in invariant task and $\zeta' = \max\big(\xi^{ra*}, -\frac{G}{\lambda^{k_{\max}}}(\lambda - 1)\big)$ in reach-avoid. Table 1 presents the results over five random seeds on the inverted pendulum task. In the inverted pendulum task, multiple methods obtain nontrivial certificates. However, PID-Barrier achieves a competitive certified volume and consistently produces valid reach-avoid certificates across all random seeds.

(2) Threshold sweep after training. We then vary the certification threshold ζ across its admissible range in the invariant task. Importantly, it yields a family of certified sets and corresponding probability lower bounds from the same trained model without no retraining . For the reach-avoid task, when $\zeta = G(\lambda - 1)$, our candidate set is $\widetilde{\mathcal{S}}_\zeta := \{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x}; \phi^{ra*}) > G\}$, which is empty. Therefore, we do not discuss the effect of the range on the reach-avoid task here. In all three cases, our framework provides a family of certified sets, allowing the certified region and the associated probability bound to be adapted to practical needs.

(3) Verification capabilities of black box systems and high-dimensional systems. To demonstrate that our data-driven barrier functions are capable of learning valid control barrier functions, we use the linear bound propagation as in Vertovec et al. (2026) to check the continuous time CBF conditions on the true continuous time dynamics. For both the pendulum and Van der Pol systems, formal verification confirms that the learned CBFs are valid. The 10-dimensional Hopfield system is beyond the scope of formal verification, due to its high dimensionality Vertovec et al. (2026). This highlights the inherent limitations of exact formal verification for high-dimensional systems and underpins the value of taking a data-driven approach.

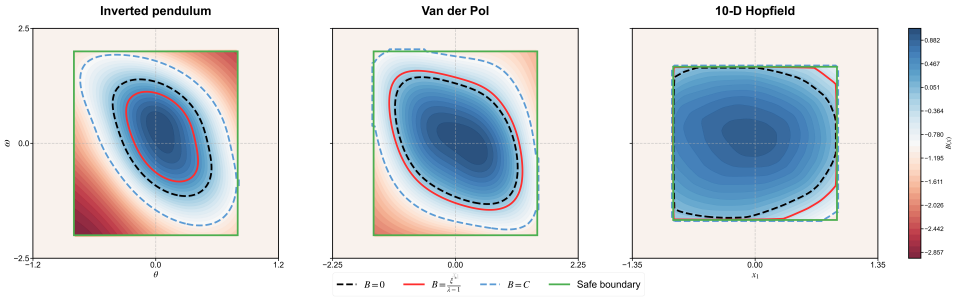

Figure 2: Comparison of verification ζ in the invariant task.

Table 1: PAC verification results on the invariance and reach-avoidance in the inverted pendulum.

	Invariant task					Reach-avoid task				
Method	$\zeta \downarrow$	Valid	$P \uparrow$	Theo.$\uparrow$	Safe$\uparrow$	$\zeta' \downarrow$	Valid	$P \uparrow$	Theo.$\uparrow$	Safe$\uparrow$
PPO	$0.0023[-0.0167,\ 0.0862]$	**5/5**	0.426	0.963	**1.000**	$0.0004[-0.0089,\ 0.0097]$	4/5	0.000	–	0.000
Lag-C	$-0.0081[-0.0288,\ -0.0037]$	**5/5**	0.471	0.966	**1.000**	$\mathbf{-0.0042}[\mathbf{-0.0079},\ \mathbf{0.0123}]$	4/5	0.609	0.974	**1.000**
RCPO	$\mathbf{-0.0155}[\mathbf{-0.021},\ \mathbf{0.4242}]$	4/5	0.457	0.965	**1.000**	$-0.0021[-0.0171,\ 0.0005]$	4/5	**0.619**	**0.974**	**1.000**
MF-NBC	$0.0135[-0.0286,\ 0.0382]$	**5/5**	**0.551**	**0.971**	**1.000**	$0.0000[-0.0071,\ 0.0049]$	1/5	0.000	–	0.000
Lag-Barrier	$-0.0049[-0.0093,\ 0.0076]$	**5/5**	0.458	0.965	**1.000**	$-0.0031[-0.0092,\ -0.0020]$	**5/5**	0.488	0.967	**1.000**
Ours	$-0.0031[-0.0107,\ 0.0061]$	**5/5**	0.461	0.965	**1.000**	$-0.0020[-0.0037,\ 0.0001]$	**5/5**	0.405	0.961	**1.000**

Valid: for the invariant task, the number of seeds with PAC certificates; for the reach-avoid task, whose estimated reach-avoid set measure is positive. The entry in each ζ or ζ' column is reported as median $[\min, \max]$ over 5 seeds. Here $P := \mu_{\mathcal{S}}(\widehat{\mathcal{S}}_{\zeta*})$, Theo. $:= 1 - \frac{\alpha}{\mu_{\mathcal{S}}(\widehat{\mathcal{S}}_{\zeta*})}$, and Safe denotes the empirical probability in the median results.

4.4 Sample Efficiency: Scenario Optimization vs. Monte Carlo

Compared with the Monte Carlo certificate based on Hoeffding's inequality, our scenario-based certificate employs a scalar optimization variable and attains a sample complexity of $\mathcal{O}(\frac{1}{\alpha})$, whereas Monte Carlo requires $\mathcal{O}(\frac{1}{\alpha^2})$ samples to achieve comparable additive accuracy α. Both the theoretical analysis and the experimental results in Figure 3 show that the proposed method is more sample-efficient. Meanwhile, as N increases, the value of α decreases, and accordingly the corresponding lower bound becomes larger, which is consistent with Definition 3. The method for computing the Monte Carlo probability lower bound is provided in Appendix D.

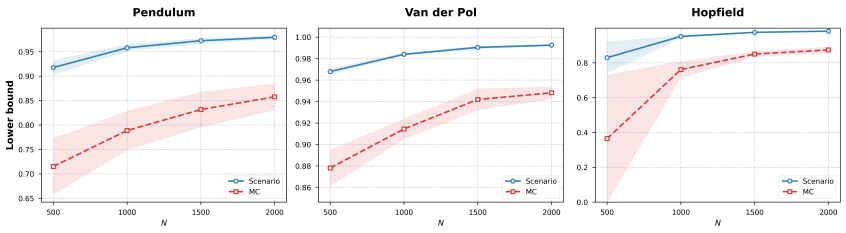

Figure 3: Comparison of scenario optimization and Monte Carlo methods in the invariant task.

5 Conclusion

In this paper, we proposed a training-and-certification framework that provides PAC multi-step guarantees for safety and reach-avoidance in black-box systems. The integration of a PID-barrier multiplier update stabilizes the joint training of policies and neural barrier functions, improving both the safety and performance of the learned policy. Furthermore, our scenario-based certification yields PAC multi-step invariant and reach-avoid sets for the system under the learned policy, while reducing sample complexity to scale linearly with the inverse of the desired accuracy rather than quadratically as in Monte Carlo bounds, thereby enabling the efficient derivation of flexible, certified, trustworthy operating regions for complex systems where traditional verification is intractable.

AI USE STATEMENT

In this work, generative AI tools were used solely to assist with language polishing and improving the clarity of written expressions. No generative AI tools were used for developing the research ideas, deriving the theoretical results, or conducting the experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All assumptions required for the theoretical results are explicitly stated in the main text, and complete proofs of the corresponding theoretical results are provided in the appendix. To facilitate reproducibility of the experimental results, the implementation code and brief instructions are publicly available in an anonymous repository .

REFERENCES

Daniele Ahmed, Andrea Peruffo, and Alessandro Abate. Automated and sound synthesis of lyapunov functions with smt solvers. In International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pp. 97–114. Springer, 2020.

Thom Badings, Wietze Koops, Sebastian Junges, and Nils Jansen. Policy verification in stochastic dynamical systems using logarithmic neural certificates. In International Conference on Computer Aided Verification, pp. 349–375. Springer, 2025.

Giuseppe Carlo Calafiore and Marco C Campi. The scenario approach to robust control design. IEEE Transactions on automatic control, 51(5):742–753, 2006.

Marco C Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. Annual Reviews in Control, 33(2):149–157, 2009.

Krishnendu Chatterjee, Thomas A Henzinger, Mathias Lechner, and Đorđe Žikelić. A learner-verifier framework for neural network controllers and certificates of stochastic systems. In International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pp. 3–25. Springer, 2023.

Yu Chen, Shaoyuan Li, and Xiang Yin. On the construction of barrier certificate: A dynamic programming perspective. arXiv preprint arXiv:2507.17222, 2025.

Alex Devonport and Murat Arcak. Estimating reachable sets with scenario optimization. In Learning for dynamics and control, pp. 75–84. PMLR, 2020.

Elizabeth Dietrich, Alex Devonport, and Murat Arcak. Nonconvex scenario optimization for data-driven reachability. In 6th Annual Learning for Dynamics & Control Conference, pp. 514–527. PMLR, 2024.

Alec Edwards, Andrea Peruffo, and Alessandro Abate. Fossil 2.0: Formal certificate synthesis for the verification and control of dynamical models. In Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control, pp. 1–10, 2024.

Ze Gong, Akshat Kumar, and Pradeep Varakantham. Offline safe reinforcement learning using trajectory classification. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pp. 16880–16887, 2025.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. Journal of the American statistical association, 58(301):13–30, 1963.

Harold Joseph Kushner. Stochastic stability and control. 1967.

Frederik Baymler Mathiesen, Simeon C Calvert, and Luca Laurenti. Safety certification for stochastic systems via neural barrier functions. IEEE Control Systems Letters, 7:973–978, 2022.

Rayan Mazouz, Karan Muvvala, Akash Ratheesh Babu, Luca Laurenti, and Morteza Lahijanian. Safety guarantees for neural network dynamic systems via stochastic barrier functions. Advances in Neural Information Processing Systems, 35:9672–9686, 2022.

Antonis Papachristodoulou and Stephen Prajna. On the construction of lyapunov functions using the sum of squares decomposition. In Proceedings of the 41st IEEE Conference on Decision and Control, 2002., volume 3, pp. 3482–3487. IEEE, 2002.

Pablo A Parrilo. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. California Institute of Technology, 2000.

Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In International workshop on hybrid systems: Computation and control, pp. 477–492. Springer, 2004.

Stephen Prajna, Ali Jadbabaie, and George J Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. IEEE Transactions on Automatic Control, 52(8): 1415–1428, 2007.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. arXiv preprint arXiv 1910.01708, 7(1):2, 2019.

Luke Rickard, Alessandro Abate, and Kostas Margellos. Data-driven certificate synthesis. Automatica, 185:112798, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In International conference on machine learning, pp. 9133–9143. PMLR, 2020.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. arXiv preprint arXiv:1805.11074, 2018.

Nikolaus Vertovec, Frederik Baymler Mathiesen, Thom Badings, Luca Laurenti, and Alessandro Abate. Scalable verification of neural control barrier functions using linear bound propagation, 2026. URL https://arxiv.org/abs/2511.06341.

Taoran Wu, Yiling Xue, Jingduo Pan, Dejin Ren, Arvind Easwaran, and Bai Xue. Convex computations for controlled safety invariant sets of black-box discrete-time dynamical systems. To appear in IFAC26, 2026.

Bai Xue, Martin Fränzle, Hengjun Zhao, Naijun Zhan, and Arvind Easwaran. Probably approximate safety verification of hybrid dynamical systems. In International Conference on Formal Engineering Methods, pp. 236–252. Springer, 2019.

Bai Xue, Miaomiao Zhang, Arvind Easwaran, and Qin Li. Pac model checking of black-box continuous-time dynamical systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(11):3944–3955, 2020.

Bai Xue, Naijun Zhan, and Martin Fränzle. Reach-avoid analysis for polynomial stochastic differential equations. IEEE Transactions on Automatic Control, 69(3):1882–1889, 2023.

Yujie Yang, Yuxuan Jiang, Yichen Liu, Jianyu Chen, and Shengbo Eben Li. Model-free safe reinforcement learning through neural barrier certificate. IEEE Robotics and Automation Letters, 8 (3):1295–1302, 2023.

Changyuan Zhao, Shuyuan Zhang, Lei Wang, and Bai Xue. Inner approximating robust reach-avoid sets for discrete-time polynomial dynamical systems. IEEE Transactions on Automatic Control, 68(8):4682–4694, 2022.

Hanrui Zhao, Banglong Liu, Lydia Dehbi, Huijiao Xie, Zhengfeng Yang, and Haifeng Qian. Polynomial neural barrier certificate synthesis of hybrid systems via counterexample guidance. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 43(11):3756–3767, 2024.

Hengjun Zhao, Xia Zeng, Taolue Chen, and Zhiming Liu. Synthesizing barrier certificates using neural networks. In Proceedings of the 23rd international conference on hybrid systems: Computation and control, pp. 1–11, 2020.

Dapeng Zhi, Peixin Wang, Si Liu, C-H Luke Ong, and Min Zhang. Unifying qualitative and quantitative safety verification of dnn-controlled systems. In International Conference on Computer Aided Verification, pp. 401–426. Springer, 2024.

A Limitation and Future work

While our PAC-based method is able to find safe invariant and reach-avoid sets for all three tasks, there remains room to further improve the size and practical usefulness of the certified set. An interesting direction for future work is to refine the learned controller or neural barrier-like functions, using the already collected data Gong et al. (2025), to enlarge the set.

B Scenario Optimization as a Certification Tool

To convert finitely many certification rollouts into the setlevel measure guarantee required by Definitions 3 and 4, we use the scenario approach. Consider a robust convex program with an uncertain parameter $\boldsymbol{\delta} \in \Delta$ distributed according to a probability measure $\mathbb{P}_\delta$:

$$\min_{\boldsymbol{z} \in D \subset \mathbb{R}^m} \ \boldsymbol{c}^\top \boldsymbol{z} \quad \text{s.t.} \quad \boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\delta}) \leq 0, \ \forall \boldsymbol{\delta} \in \Delta. \tag{ROP}$$

Given N i.i.d. samples $\{\boldsymbol{\delta}_i\}_{i=1}^N$ drawn from $\mathbb{P}_\delta$, the corresponding scenario program is

$$\min_{\boldsymbol{z} \in D} \ \boldsymbol{c}^\top \boldsymbol{z} \quad \text{s.t.} \quad \boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\delta}_i) \leq 0, \ i = 1, \ldots, N. \tag{ROP-N}$$

For a solution $\boldsymbol{z}$, define the violation set $F(\boldsymbol{z}) = \{\boldsymbol{\delta} \in \Delta \mid \boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\delta}) \not\leq 0\}$ and the violation probability $\mathcal{V}(\boldsymbol{z}) = \mathbb{P}_\delta[\boldsymbol{\delta} \in F(\boldsymbol{z})]$.

Assumption 2 (Convexity)**.** *$D \subseteq \mathbb{R}^d$ is convex and closed, $\Delta \subseteq \mathbb{R}^{n_\delta}$, and $\boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\delta})$ is continuous on $D \times \Delta$ and convex in $\boldsymbol{z}$ for every fixed $\boldsymbol{\delta} \in \Delta$.*

Assumption 3 (Unique optimum)**.** *For almost every set of N samples $\{\boldsymbol{\delta}_i\}_{i=1}^N$, program equation ROP-N admits a unique optimal solution $\boldsymbol{z}_N^*$.*

Proposition 3 (Scenario PAC bound Campi et al. (2009))**.** *Under Assumptions 2–3, $\mathbb{P}_\delta^N\big[\mathcal{V}(\boldsymbol{z}_N^*) \leq \alpha\big] \geq 1 - \beta$, provided $\alpha \geq \frac{2}{N}\big(\ln \frac{1}{\beta} + d\big)$, where $\mathbb{P}_\delta^N$ is the product measure induced by the N i.i.d. samples.*

C Proof of Theorem 1 and Theorem 2

Theorem 1 Let ξ^{s*} be the optimal solution to equation 10. Suppose that $\widetilde{\mathcal{S}}_\zeta := \left\{\boldsymbol{x} \in \mathcal{S} \;\middle|\; B(\boldsymbol{x}; \phi^{s*}) \geq \frac{\zeta}{\lambda - 1}\right\}$, where $\xi^{s*} \leq \zeta < (\lambda - 1)C$. If $N \geq \frac{2}{\alpha}\left(\ln \frac{1}{\beta} + 1\right)$, then $\widetilde{\mathcal{S}}_\zeta$ is an (α, β)-PAC multi-step safe invariant set for equation 1 in Definition 3.

Proof. Let's consider the following robust optimization:

$$\begin{aligned} &\min_{\xi} \xi \\ &\text{s.t.} \begin{cases} \text{MINB}(\boldsymbol{x}; \phi^{s*}) \geq -\xi, \forall \boldsymbol{x} \in \mathcal{S}, \\ \xi \in [-U_\xi, U_\xi], \end{cases} \end{aligned} \tag{12}$$

where $\text{MINB}(\boldsymbol{x}) = \min_{i=1}^{\min\{k_{\boldsymbol{x}}, k_{\max}\}} [B(\boldsymbol{\phi}_u^{\boldsymbol{x}}(i); \phi^{s*}) - \lambda B(\boldsymbol{\phi}_u^{\boldsymbol{x}}(i-1); \phi^{s*})]$ with $k_{\boldsymbol{x}} \leq k_{\max}$ being the smallest value such that $\boldsymbol{\phi}_u^{\boldsymbol{x}}(i) \notin \mathcal{S}$ ($k_{\boldsymbol{x}}$ is infinity when $\boldsymbol{\phi}_u^{\boldsymbol{x}}(i) \in \mathcal{S}$ for all $i \in \mathbb{N}$), and $\lambda \in (0, 1)$ and $U_\xi > 0$ are the values in equation 10.

It is straightforward to verify that the robust optimization problem equation 12 satisfies Assumptions 2 and 3. Moreover, since the optimization problem equation 10 can be regarded as the scenario counterpart of the robust formulation equation 12 with scenarios $\{\boldsymbol{x}_i\}_{i=1}^N$, we have the conclusion that with confidence at least $1 - \beta$,

$$\mathbb{P}_{\boldsymbol{x}}[\text{MINB}(\boldsymbol{x}; \phi^{s*}) \geq -\xi^{s*}] \geq 1 - \alpha,$$

which follows directly from Proposition 3. This further implies that with confidence at least $1 - \beta$,

$$\mathbb{P}_{\boldsymbol{x}}[\text{MINB}(\boldsymbol{x}; \phi^{s*}) \geq -\zeta] \geq 1 - \alpha,$$

In addition, when $\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}_\zeta] > 0$, we have

$$\mu_{\mathcal{S}}[\text{MINB}(\boldsymbol{x};\phi^{s*}) \geq -\zeta \mid \boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta] = 1 - \mu_{\mathcal{S}}[\text{MINB}(\boldsymbol{x};\phi^{s*}) < -\zeta \mid \boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]$$
$$= 1 - \frac{\mu_{\mathcal{S}}[\text{MINB}(\boldsymbol{x};\phi^{s*}) < -\zeta \wedge \boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]}{\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}_\zeta]} \geq 1 - \frac{\mu_{\mathcal{S}}[\text{MINB}(\boldsymbol{x};\phi^{s*}) < -\xi^{s*}]}{\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}_\zeta]} \geq 1 - \frac{\alpha}{\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}_\zeta]}.$$

On the other hand, assume $\boldsymbol{x}_0 \in \{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x};\phi^{s*}) \geq -\frac{\zeta}{1-\lambda}\}$ and $\text{MINB}(\boldsymbol{x}_0;\phi^{s*}) \geq -\zeta$. We can obtain

$$\min_{i=1}^{\min\{k_{\boldsymbol{x}_0}, k_{\max}\}} \left[B(\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i);\phi^{s*}) + \frac{\zeta}{1-\lambda} - \lambda(B(\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i-1);\phi^{s*}) + \frac{\zeta}{1-\lambda})\right] \geq 0.$$

Since $B(\boldsymbol{x}_0;\phi^{s*}) + \frac{\zeta}{1-\lambda} \geq 0$, we have $B(\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i);\phi^{s*}) + \frac{\zeta}{1-\lambda} \geq 0$ for $i \leq \min\{k_{\boldsymbol{x}_0}, k_{\max}\}$.

In addition, since $B(\boldsymbol{x};\phi^{s*}) + \frac{\zeta}{1-\lambda} = C + \frac{\zeta}{1-\lambda} < 0$ for $\boldsymbol{x} \notin \mathcal{S}$, we conclude

$$B(\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i);\phi^{s*}) + \frac{\zeta}{1-\lambda} \geq 0, \forall i \leq k_{\max}$$

and consequently, $\wedge_{i=0}^{k_{\max}} \boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i) \in \widetilde{\mathcal{S}}_\zeta$ holds.

Combining the above discussions, we have that, with confidence at least $1-\beta$, if $\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}] > 0$, $\mu_{\mathcal{S}}[\wedge_{i=0}^{k_{\max}} \boldsymbol{\phi}_u^{\boldsymbol{x}}(i) \in \widetilde{\mathcal{S}}_\zeta \mid \boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta] \geq 1 - \frac{1}{\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}_\zeta]}\alpha$. □

Theorem 2 Let ξ^{ra*} be the optimal solution to equation 11. Suppose that $\widetilde{\mathcal{S}}_\zeta := \left\{\boldsymbol{x} \in \mathcal{S} \;\middle|\; B(\boldsymbol{x};\phi^{ra*}) > \frac{G}{\lambda^{k_{\max}}} + \zeta\frac{1-\lambda^{-k_{\max}}}{\lambda-1}\right\}$, where $\max\left(\xi^{ra*}, -\frac{G}{\lambda^{k_{\max}}}(\lambda-1)\right) \leq \zeta \leq G(\lambda-1)$. If $N \geq \frac{2}{\alpha}\left(\ln\frac{1}{\beta}+1\right)$, then $\widetilde{\mathcal{S}}_\zeta$ is an (α,β)-PAC multi-step reach-avoid set for equation 1 in Definition 4.

Proof. Let's consider the following robust optimization:

$$\begin{aligned} &\min_{\xi} \xi \\ &\text{s.t.} \begin{cases} \text{MINB}(\boldsymbol{x}) \geq -\xi, \forall \boldsymbol{x} \in \mathcal{S}, \\ \xi \in [-U_\xi, U_\xi], \end{cases} \end{aligned} \tag{13}$$

where $\text{MINB}(\boldsymbol{x}) = \min_{i=1}^{\min\{k_{\boldsymbol{x}}, k_{\max}\}}[B(\boldsymbol{\phi}_u^{\boldsymbol{x}}(i);\phi^{ra*}) - \lambda B(\boldsymbol{\phi}_u^{\boldsymbol{x}}(i-1);\phi^{ra*})]$ with the convention that the constraint is vacuous if $\min\{k_{\boldsymbol{x}}, k_{\max}\} = 0$. Here $k_{\boldsymbol{x}}$ is the smallest value such that $\boldsymbol{\phi}_u^{\boldsymbol{x}}(i) \in (\mathbb{R}^n \setminus \mathcal{S}) \cup \mathcal{T}$ ($k_{\boldsymbol{x}}$ is infinity when $\boldsymbol{\phi}_u^{\boldsymbol{x}}(i) \in \mathcal{S} \setminus \mathcal{T}$ for all $i \in \mathbb{N}$), and $\lambda \in (1,\infty)$ and $U_\xi > 0$ are the same values in equation 11.

This robust optimization equation 13 satisfies Assumptions 2 and 3. Also, the optimization problem equation 11 can be regarded as the scenario counterpart of the robust formulation equation 13 with scenarios $\{\boldsymbol{x}_i\}_{i=1}^N$. Therefore, with confidence at least $1-\beta$,

$$\mu_{\mathcal{S}}[\text{MINB}(\boldsymbol{x}) \geq -\xi^{ra*}] \geq 1-\alpha,$$

which follows directly from Proposition 3. Since $\zeta \geq \xi^{ra*}$, we further have $\mathbb{P}_{\boldsymbol{x}}[\text{MINB}(\boldsymbol{x}) \geq -\zeta] \geq 1-\alpha$. Similar to the arguments in the proof of Theorem 1, when $\mathbb{P}_{\boldsymbol{x}}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta] > 0$, we can obtain

$$\mu_{\mathcal{S}}[\text{MINB}(\boldsymbol{x}) \geq -\zeta \mid \boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta] \geq 1 - \frac{\alpha}{\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}_\zeta]}.$$

It remains to show that every $\boldsymbol{x}_0 \in \widetilde{\mathcal{S}}_\zeta$ satisfying $\text{MINB}(\boldsymbol{x}_0) \geq -\zeta$ is a k-step reach-avoid initial state. Let $h_\zeta := \frac{G}{\lambda^{k_{\max}}} + \zeta\frac{1-\lambda^{-k_{\max}}}{\lambda-1}$, $B_i := B(\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i);\phi^{ra*})$, and $m := \min\{k_{\boldsymbol{x}_0}, k_{\max}\}$. If $\boldsymbol{x}_0 \in \mathcal{T}$, the claim is trivial. Hence, we assume $\boldsymbol{x}_0 \in \widetilde{\mathcal{S}}_\zeta \setminus \mathcal{T}$. Since $\boldsymbol{x}_0 \in \widetilde{\mathcal{S}}_\zeta$, we have $B_0 > h_\zeta$. Moreover, from $\zeta \geq -\frac{G}{\lambda^{k_{\max}}}(\lambda-1)$, we have $h_\zeta \geq \frac{G}{\lambda^{2k_{\max}}} > 0$, and from $\zeta \leq G(\lambda-1)$, we have

$$h_\zeta - \frac{\zeta}{\lambda-1} = \frac{1}{\lambda^k}\left(G - \frac{\zeta}{\lambda-1}\right) \geq 0.$$

(1)We first prove that the trajectory remains in $\widetilde{\mathcal{S}}_\zeta$ before reaching $\mathcal{T}$. Since $\widetilde{\mathcal{S}}_\zeta$ is defined by both $\boldsymbol{x} \in \mathcal{S}$ and $B(\boldsymbol{x}; \phi^{ra*}) > h_\zeta$, we need to verify both conditions along the trajectory. For any $0 \le i < m$, suppose $\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i) \in \widetilde{\mathcal{S}}_\zeta \setminus \mathcal{T}$. Then $B_i > h_\zeta$. Since $\text{MINB}(\boldsymbol{x}_0) \ge -\zeta$, we have $B_{i+1} - \lambda B_i \ge -\zeta$. Together with the condition $h_\zeta - \frac{\zeta}{\lambda - 1} \ge 0$, we have

$$B_{i+1} - B_i \ge (\lambda - 1)B_i - \zeta > (\lambda - 1)h_\zeta - \zeta \ge 0.$$

Therefore, $B_{i+1} > B_i > h_\zeta$.

Since $C = 0$ in the reach-avoid task, by the construction of B in equation 4, we have $B(\boldsymbol{x}; \phi^{ra*}) = 0$ for $\boldsymbol{x} \notin \mathcal{S}$. Since $h_\zeta > 0$, the inequality $B_{i+1} > h_\zeta$ implies that $\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i+1)$ cannot be outside $\mathcal{S}$. Thus, if $\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i+1) \notin \mathcal{T}$, then $\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i+1) \in \mathcal{S}$ and $B_{i+1} > h_\zeta$, namely $\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i+1) \in \widetilde{\mathcal{S}}_\zeta \setminus \mathcal{T}$. By induction, the trajectory remains in $\widetilde{\mathcal{S}}_\zeta \setminus \mathcal{T}$ until it reaches $\mathcal{T}$.

(2)Next, we prove that the trajectory must reach $\mathcal{T}$ within $k_{\max}$ steps. By contradiction, that $\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i) \notin \mathcal{T}$ for all $i = 0, \dots, k_{\max}$. From the previous paragraph, the trajectory stays in $\widetilde{\mathcal{S}}_\zeta \setminus \mathcal{T}$ up to time k, and thus $m = k$. For all $1 \le i \le k_{\max}$, the inequality $B_i - \lambda B_{i-1} \ge -\zeta$ implies

$$B_i \ge \lambda^i B_0 - \zeta \frac{\lambda^i - 1}{\lambda - 1}.$$

Taking $i = k_{\max}$ and using $B_0 > h_\zeta$ gives

$$B_{k_{\max}} > \lambda^{k_{\max}} \left(\frac{G}{\lambda^{k_{\max}}} + \zeta \frac{1 - \lambda^{-k_{\max}}}{\lambda - 1} \right) - \zeta \frac{\lambda^{k_{\max}} - 1}{\lambda - 1} = G.$$

However, by the reach-avoid barrier construction, $B(\boldsymbol{x}; \phi^{ra*}) \le G$ for all $\boldsymbol{x} \in \mathcal{S} \setminus \mathcal{T}$. This contradicts $\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(k_{\max}) \in \widetilde{\mathcal{S}}_\zeta \setminus \mathcal{T} \subseteq \mathcal{S} \setminus \mathcal{T}$. Therefore, there exists some $l \le k_{\max}$ such that $\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(l) \in \mathcal{T}$, and $\boldsymbol{\phi}_u^{\boldsymbol{x}_0}(i) \in \widetilde{\mathcal{S}}_\zeta \setminus \mathcal{T}$ for all $i = 0, \dots, l-1$.

Combining the above deterministic implication with the conditional PAC bound, we have that

$$\mathbb{P}_{D_N} \left(\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}] > 0 \implies \begin{array}{l} \mu_{\mathcal{S}} \left[\exists\, l \in \{0, \dots, k_{\max}\} \text{ s.t. } \boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}}(l) \in \mathcal{T}, \right. \\ \left. \boldsymbol{\phi}_{\boldsymbol{u}}^{\boldsymbol{x}}(i) \in \widetilde{\mathcal{S}} \setminus \mathcal{T},\ \forall i < l \;\middle|\; \boldsymbol{x} \in \widetilde{\mathcal{S}} \right] \\ \ge 1 - \dfrac{\alpha}{\mu_{\mathcal{S}}[\widetilde{\mathcal{S}}]} \end{array} \right) \ge 1 - \beta.$$

Therefore, $\widetilde{\mathcal{S}}_\zeta$ is an (α, β)-PAC multi-step reach-avoid set for equation 1 in Definition 4. □

D MONTE CARLO COMPARISON DERIVATION

In this section, we will introduce the proposed method with the well-known Monte Carlo method, together with the well-known Hoeffding's inequality. Another comparable one is Clopper-Pearson (CP) method, for which the sample complexity is also $O(1/\alpha^2)$ generally.

Given a level value ζ, Monte Carlo methods, together with the well-known Hoeffding's inequality Hoeffding (1963), can be used to provide a PAC characterization of the sets $\widetilde{\mathcal{S}}_\zeta = \{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x}; \phi^{s*}) \ge \frac{\zeta}{\lambda - 1}\}$ and $\widetilde{\mathcal{S}}_\zeta = \{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x}; \phi^{ra*}) > \frac{G}{\lambda^{k_{\max}}} + \zeta \frac{1 - \lambda^{-k_{\max}}}{\lambda - 1}\}$ on invariance and reach-avoidance guarantees, respectively. There are two primary ways to perform this estimation. Here, we illustrate the procedure for the case of safety guarantees; the same approach applies to reach-avoidance guarantees.

1) The first approach is based on evaluating the barrier inequality $\text{MINB}(\boldsymbol{x}) \ge -\xi$ for a family of N i.i.d. states $\{\boldsymbol{x}_i\}_{i=1}^N$ from the uniform probability space $(\mathcal{S}, \mathcal{F}_{\boldsymbol{x}}, \mathbb{P})$, together with their resulting state trajectory $\{\boldsymbol{y}_{i,j} := \boldsymbol{\phi}_u^{\boldsymbol{x}_i}(j)\}_{j=1}^{k_i}$, where $1 \le j \le k_i \le k_{\max}$, $\boldsymbol{y}_{i,j} \in \mathcal{S}$ for all $0 \le j < k_i$, and $\boldsymbol{y}_{i,k_i} \notin \mathcal{S}$. For each sample $\boldsymbol{x}_i$, define the indicator variable

$$Z_i = \begin{cases} 1, & \text{if } \text{MINB}(\boldsymbol{x}_i) \ge -\xi, \\ 0, & \text{otherwise.} \end{cases}$$

The empirical satisfaction probability over the sample set is $\hat{p}_N := \frac{1}{N}\sum_{i=1}^{N} Z_i$, and the true probability of satisfaction is $p := \mathbb{P}\Big[\text{MINB}(\boldsymbol{x}) \geq -\xi\Big]$. Since the Z_i are independent and bounded in $[0,1]$, Hoeffding's inequality implies that, for any $\epsilon > 0$, $\mathbb{P}^N\left[p \geq \hat{p}_N - \epsilon\right] \leq \exp(-2N\epsilon^2)$ holds.

Consequently, with confidence at least $1-\delta$, $p \geq \hat{p}_N - \epsilon$ holds, where $N \geq \frac{1}{2\epsilon^2}\ln\frac{1}{\delta}$. Following the deduction on the conditional probability in Theorem 1, we can obtain that with confidence at least $1-\delta$,

$$\mu_{\mathcal{S}}[\wedge_{i=0}^{k_{\max}} \boldsymbol{\phi}_u^{\boldsymbol{x}}(i) \in \widetilde{\mathcal{S}}_\zeta | \boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta] \geq 1 - \frac{1-(\hat{p}_N-\epsilon)}{\mu_{\mathcal{S}}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]},$$

Let $\beta = \delta$ and $\alpha = \epsilon$, the guarantee in Theorem 1 is tighter and requires only $O(\frac{1}{\alpha})$ samples, whereas the Monte Carlo bound is more conservative and demands $O(\frac{1}{\alpha^2})$ samples. Indeed,

$$1 - \frac{1-(1-\epsilon)}{\mu_{\mathcal{S}}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]} = 1 - \frac{\alpha}{\mu_{\mathcal{S}}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]} \geq 1 - \frac{1-(\hat{p}_N-\epsilon)}{\mu_{\mathcal{S}}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]},$$

showing that the Monte Carlo characterization yields a weaker bound. Therefore, our approach not only provides a sharper guarantee but also achieves it with significantly fewer samples, thereby outperforming the Monte Carlo method.

2) The second one is based on evaluating the set $\widetilde{\mathcal{S}}_\zeta = \{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x}; \phi^{s*}) \geq \frac{\zeta}{\lambda-1}\}$ directly for a family of N i.i.d. states $\{\boldsymbol{x}_i\}_{i=1}^N$ from the uniform probability space $(\widetilde{\mathcal{S}}_\zeta, \mathcal{F}'_{\boldsymbol{x}})$, together with their resulting state trajectory $\{\boldsymbol{y}_{i,j} := \boldsymbol{\phi}_u^{\boldsymbol{x}_i}(j)\}_{j=1}^{k_i}$, where $1 \leq j \leq k_i \leq k_{\max}$, $\boldsymbol{y}_{i,j} \in \mathcal{S}$ for all $0 \leq j < k_i$, and $\boldsymbol{y}_{i,k_i} \notin \mathcal{S}$. For each sample $\boldsymbol{x}_i$, define the indicator variable

$$Z_i = \begin{cases} 1, & \text{if } \wedge_{j=1}^{k} \boldsymbol{y}_{i,j} \in \widetilde{\mathcal{S}}_\zeta, \\ 0, & \text{otherwise.} \end{cases}$$

The empirical satisfaction probability over the sample set is $\hat{p}_N := \frac{1}{N}\sum_{i=1}^{N} Z_i$, and the true probability of satisfaction is $p := \mathbb{P}'[\wedge_{j=0}^{k_{\max}} \boldsymbol{\phi}_u^{\boldsymbol{x}_i}(j) \in \widetilde{\mathcal{S}}]$. Since the Z_i are independent and bounded in $[0,1]$, Hoeffding's inequality implies that, for any $\epsilon > 0$, $\mathbb{P}'^N_{\boldsymbol{x}}\left[\hat{p}_N - p \geq \epsilon\right] \leq \exp(-2N\epsilon^2)$ holds. Consequently, with confidence at least $1-\delta$, $p \geq \hat{p}_N - \epsilon$ holds, where $N \geq \frac{1}{2\epsilon^2}\ln\frac{1}{\delta}$. Following the deduction on the conditional probability in Theorem 1, we can obtain that with confidence at least $1-\delta$, $\mu'_{\mathcal{S}}[\wedge_{i=0}^{k_{\max}} \boldsymbol{\phi}_u^{\boldsymbol{x}}(i) \in \widetilde{\mathcal{S}}_\zeta] \geq \hat{p}_N - \epsilon$ holds.

Although $\mathbb{P}'_{\boldsymbol{x}}[\wedge_{i=0}^{k_{\max}} \boldsymbol{\phi}_u^{\boldsymbol{x}}(l) \in \widetilde{\mathcal{S}}_\zeta] = \mathbb{P}[\wedge_{i=0}^{k_{\max}} \boldsymbol{\phi}_u^{\boldsymbol{x}}(l) \in \widetilde{\mathcal{S}}_\zeta \mid \boldsymbol{x} \in \widetilde{\mathcal{S}}]$, the associated confidence levels in Theorem 1 and in the above PAC statement correspond to different probability spaces. Here, we compare only their numerical values. Setting $\beta = \delta$ and $\alpha = \epsilon\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]$, when $\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]$ is large, i.e., $\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta] > \frac{2\epsilon(\ln\frac{1}{\delta}+1)}{\ln\frac{1}{\delta}}$, we reach the same conclusion as before: our approach not only provides a sharper guarantee but also achieves it with fewer samples, thereby outperforming the Monte Carlo method.

In addition, the probability measure $\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]$, a byproduct of our method, serves as an important indicator for evaluating the quality of the learned controller and its associated neural barrier-like function. As discussed in Remark 1, this measure is positively correlated with the quantity $1 - \frac{\alpha}{\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]}$. A larger value of $\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]$ implies a higher lower bound on the probability measure of states in $\widetilde{\mathcal{S}}_\zeta$ that satisfy k-step invariance, indicating that the system is safer when operating within $\widetilde{\mathcal{S}}_\zeta$. Moreover, a larger value also corresponds to a larger high-likelihood k-step safe invariant region, which in turn enlarges the effective safe-operating region and allows for more robust control over a broader set of states.

Therefore, an interesting direction for future work is to refine the learned controller or NBF, using the already collected data Gong et al. (2025), to enlarge the set $\widetilde{\mathcal{S}}_\zeta$particularly in scenarios where $\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]$ is small.

Although Chernoff's inequality can sometimes provide a less conservative bound than Hoeffding's inequality, we do not use it for comparisons in this paper. One reason is that the required number of sampled data often depends on the unknown probability p.

E Environment details and hyperparameters

E.1 Environment details

Inverted Pendulum. The dynamics is given by

$$\begin{aligned}\dot{x}_1 &= x_2,\\ \dot{x}_2 &= \frac{3g}{2l}\sin(x_1) + \frac{3}{ml^2}u,\end{aligned}$$

where x_1 denotes the angle and x_2 denotes the angular velocity, u is the control input. The discrete-time update with time step $\Delta t = 0.05$ is:

$$\begin{aligned}x_1(t+1) &= x_1(t) + x_2(t)\cdot\Delta t,\\ x_2(t+1) &= x_2(t) + \left(\frac{3g}{2l}\sin(x_1) + \frac{3}{ml^2}u\right)\cdot\Delta t.\end{aligned}$$

The system parameters are: $g = 9.81$, $l = 1.0$ and $m = 1.0$. The state constraints $\mathcal{S}$ are $|x_1| \le 0.8$ and $|x_2| \le 2.0$. The control limit is $|u| < 10$. In the reach-avoid task, the goal set is $|x_1| \le 0.2$ and $|x_2| \le 0.2$. We use 10^4 points to estimate the volume of the candidate set.

Van der Pol Oscillator. The dynamics is given by

$$\begin{aligned}\dot{x}_1 &= x_2,\\ \dot{x}_2 &= \mu(1-x_1^2)x_2 - x_1 + u,\end{aligned}$$

where x_1 denotes the position, x_2 denotes the velocity, u is the control input, and $\mu = 2.0$ is the damping parameter. The discrete-time update with time step $\Delta t = 0.05$ is:

$$\begin{aligned}x_1(t+1) &= x_1(t) + x_2(t)\cdot\Delta t,\\ x_2(t+1) &= x_2(t) + \big(\mu(1-x_1^2(t))x_2(t) - x_1(t) + u(t)\big)\cdot\Delta t.\end{aligned}$$

The state constraints $\mathcal{S}$ are $|x_1| \le 1.5$ and $|x_2| \le 1.5$. The control limit is $|u| < 5$. In the reach-avoid task, the goal set is $|x_1| \le 0.25$ and $|x_2| \le 0.25$. We use 10^4 points to estimate the volume of the candidate set.

Hopfield Neural Network. The dynamics is given by

$$\dot{x} = \lambda(1-x^2)\odot\left[-\frac{\operatorname{arctanh}(x)}{\tau\lambda} + Wx + u\right],$$

where $x \in \mathbb{R}^{10}$ is the neural state vector, $u \in \mathbb{R}^{10}$ is the control input, $\lambda = 1.0$ is the output gain, $\tau = 1.0$ is the time constant, $\odot$ denotes element-wise multiplication, and $\operatorname{arctanh}$ is applied element-wise. The synaptic weight matrix W is a 10×10 symmetric matrix generated as:

$$\tilde{W} \sim \text{Uniform}(-1.0, 1.0)^{10\times 10}, \quad W = \frac{\tilde{W} + \tilde{W}^T}{2},$$

with random seed 44. The discrete-time update with time step $\Delta t = 0.05$ is:

$$x(t+1) = x(t) + \lambda(1-x^2)\odot\left[-\frac{\operatorname{arctanh}(x)}{\tau\lambda} + Wx + u\right]\cdot\Delta t,$$

The state constraints $\mathcal{S}$ are $|x_i| \le 0.9$ for all $i = 1,\dots,10$. The control limit is $|u_i| \le 2.0$ for all i. In the reach-avoid task, the goal set is $|x_i| \le 0.4 (\forall i = 1, 2, \dots, 10)$. The goal set accounts for only $\left(\frac{0.4}{0.9}\right)^{10} = \left(\frac{4}{9}\right)^{10} \approx 0.03\%$ of the safe state space volume. We use 10^6 points to estimate the volume of the candidate set because of the high dimension.

E.2 HYPERPARAMETERS

The reward function is designed to account for both task performance and control energy. For the invariant task, the reward is defined as

$$r(\boldsymbol{x}, u) = \begin{cases} 1 - c_u \|u\|_2^2, & \text{if } f(\boldsymbol{x}, u) \in \mathcal{S}, \\ -100, & \text{if } f(\boldsymbol{x}, u) \notin \mathcal{S}, \end{cases}$$

where $c_u > 0$ is the control-energy coefficient. For the reach-avoid task, the reward is defined as

$$r(\boldsymbol{x}, u) = \begin{cases} 1 - c_u \|u\|_2^2, & \text{if } f(\boldsymbol{x}, u) \in \mathcal{S} \setminus \mathcal{T}, \\ 1, & \text{if } f(\boldsymbol{x}, u) \in \mathcal{T}, \\ -100, & \text{if } f(\boldsymbol{x}, u) \notin \mathcal{S}. \end{cases}$$

We set $c_u = 0.01$ for the inverted pendulum and Van der Pol systems, and $c_u = 0.05$ for the 10-dimensional Hopfield system. The coefficient is fixed for each system and shared by all methods to ensure a fair comparison.

Table 2: Hyperparameters used in all experiments.

Category	**Hyperparameter**	**Value**
Shared	Training episodes	2000/ 1000 for Hopfied
	Max steps per episode	200
	Discount factor γ	0.99
	GAE parameter λ	0.95
	PPO clip ϵ	0.2
	Actor learning rate	3e-4
	Actor network hidden sizes	(128, 128, 128)
	Actor activation function	Tanh
	Critic learning rate	1e-3
	Critic network hidden sizes	(128, 128)
	Critic activation function	Tanh
	Barrier learning rate	1e-3
	Barrier network hidden sizes	(128, 128)
	Barrier activation function	ReLU
	Optimizer	Adam
RCPO	Multiplier learning rate	1e-4
	Multiplier upper bound	10.0
Lagrangian	Initial multiplier	0.0
	Multiplier step size K_p	1.0
MF-NBC	Initial multiplier	0.0
	Multiplier step size K_p	1.0
	Barrier lookahead steps	5
Ours	Pendulum K_p / K_i / K_d	0.5 / 0.05 / 0.05
	Van der Pol K_p / K_i / K_d	0.5 / 0.05 / 0.1
	Hopfield K_p / K_i / K_d	0.5 / 0.05 / 0.5
Verification steps	Invariant task	200
	Reach-avoid task	200

F PID-BARRIER RL ALGORITHM AND VERIFICATION ALGORITHM

F.1 PPO ACTOR AND VALUE LOSSES

Let w and $\bar{x}$ denote the parameters of the value network and target value network, respectively. The definition of the advantage function $A(\boldsymbol{x}, u)$ can be found in Schulman et al. (2017). The actor loss

function $L_\pi^{\mathrm{PPO}}(\theta)$ and the value function loss $\mathcal{L}_V(w)$ used in Subsection 3.1.2 are

$$L_\pi^{\mathrm{PPO}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{x},u \in \mathcal{D}} \min\left(\rho_\theta A^{\pi_{\text{old}}}(\boldsymbol{x},u), \text{clip}(\rho_\theta, 1-\epsilon, 1+\epsilon) A^{\pi_{\text{old}}}(\boldsymbol{x},u)\right) \tag{14}$$

$$\mathcal{L}_V(w) = \frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{x},u \in \mathcal{D}} \left(V(\boldsymbol{x};w) - (A^{\pi_{\text{old}}}(\boldsymbol{x},u) + V_{\bar{x}})\right)^2 \tag{15}$$

F.2 HYPERPARAMETER SETTING

In the hyperparameter table below, most categories are consistent. For the training episodes, we chose a lower value of 1000 for the Hopfield system, because this was already sufficient to achieve convergence. We use progressively larger K_D across the three benchmarks ($0.05 \to 0.1 \to 0.5$ for the inverted pendulum, Van der Pol, and 10-D Hopfield system, respectively). The inverted pendulum is a low-dimensional benchmark and produces relatively smooth barrier-violation signals, so a small derivative gain is sufficient. As the Van der Pol dynamics become more nonlinear, the violation signal can change more sharply during policy updates, which suggests using a higher derivative gain. For the high-dimensional Hopfield system, finite rollout sampling provides sparser coverage of the state space and leads to noisier barrier violation estimates; therefore, a stronger derivative feedback term is used to improve responsiveness to rapid multiplier changes.

Algorithm 1 PPO-Based Safe RL with Barrier Certificate and PID-Barrier

Initialize: barrier parameters ϕ, policy parameters θ, value parameters w, multiplier $\nu^{(1)} \leftarrow 0$, integral state $\mathcal{I}_0 \leftarrow 0$, previous error $e_0 \leftarrow 0$, PID gains $K_P, K_I, K_D \geq 0$, learning rates η_B, η_V, η_π.
for $m = 1, 2, \ldots$ **(each training iteration) do**
  Initialize rollout buffer $\mathcal{D}^{(m)} \leftarrow \emptyset$, sample $\boldsymbol{x}(0) \sim \mathcal{S}$.
  for each environment step t **do**
    Observe $\boldsymbol{x}(t)$ and sample control $\boldsymbol{u}(t) = \pi(\boldsymbol{x}(t);\theta)$.
    Apply $\boldsymbol{u}(t)$; observe $\boldsymbol{x}(t+1)$ and reward $r(t)$.
    $\mathcal{D}^{(m)} \leftarrow \mathcal{D}^{(m)} \cup \{\boldsymbol{x}(t), \boldsymbol{u}(t), r(t), \boldsymbol{x}(t+1)\}$.
  end for
  Compute the barrier violation: $\hat{J}_C^{(m)} \leftarrow \frac{1}{|\boldsymbol{x} \in \mathcal{D}_{\text{safe}}^{(m)}|} \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{safe}}^{(m)}} \max(0,\ \lambda B(\boldsymbol{x};\phi) - B(\boldsymbol{x}';\phi))$.
  Compute current error: $e_m \leftarrow \hat{J}_C^{(m)}$.
  Update PID states: $\mathcal{I}_m \leftarrow \max(0, \mathcal{I}_{m-1} + e_m),\ \mathcal{D}_m \leftarrow \max(0, e_m - e_{m-1})$.
  Update multiplier: $\nu^{(m+1)} \leftarrow \max(0,\ K_P e_m + K_I \mathcal{I}_m + K_D \mathcal{D}_m)$.
  for each gradient step do
    Update barrier network: $\phi \leftarrow \phi - \eta_B \nabla_\phi \mathcal{L}_B(\phi)$.
    Update value network: $w \leftarrow w - \eta_V \nabla_w \mathcal{L}_V(w)$.
    Update policy network: $\theta \leftarrow \theta + \eta_\pi \nabla_\theta \mathcal{L}_\pi(\theta)$.
  end for
  $e_{m-1} \leftarrow e_m$.
end for

Algorithm 2 PAC Multiple-Step Certification

Input: Trained barrier network ϕ^*, trained policy network θ^*, sample size N, confidence parameter $\beta \in (0, 1)$, horizon k, safe set $\mathcal{S}$, target set $\mathcal{T}$ (set $\mathcal{T} = \emptyset$ for invariance certification), verify rate λ ($\lambda \in (0, 1)$ for invariance, $\lambda > 1$ for reach-avoidance), bounding scalar U_ξ.
Output: Optimal LP solution ξ^*, PAC error bound α, and the certified probability lower bound P_{lower}.
Calculate PAC error parameter: $\alpha \leftarrow \frac{2}{N}\left(\ln \frac{1}{\beta} + 1\right)$.
Initialize trajectory transition dataset $\mathcal{D}_{\text{cert}} \leftarrow \emptyset$ and initial state set $\mathcal{X}_{\text{init}} \leftarrow \emptyset$.
for $i = 1, \ldots, N$ **do**
  Sample initial state $\boldsymbol{x}_i \sim \mathbb{P}$ uniformly from $\mathcal{S}$.
  $\mathcal{X}_{\text{init}} \leftarrow \mathcal{X}_{\text{init}} \cup \{\boldsymbol{x}_i\}$.
  Initialize state $\boldsymbol{y}_{i,0} \leftarrow \boldsymbol{x}_i$, and step counter $j \leftarrow 0$.
  while $j < k$ **and** $\boldsymbol{y}_{i,j} \in \mathcal{S}$ **do**
    if $\boldsymbol{y}_{i,j} \in \mathcal{T}$ **then**
      break {Terminate early if target is reached}
    end if
    Compute control action $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{y}_{i,j}; \theta^*)$.
    Simulate one step to obtain next state $\boldsymbol{y}_{i,j+1}$.
    $\mathcal{D}_{\text{cert}} \leftarrow \mathcal{D}_{\text{cert}} \cup \{(\boldsymbol{y}_{i,j}, \boldsymbol{y}_{i,j+1})\}$.
    $j \leftarrow j + 1$.
  end while
end for
Solve the corresponding linear program equation 10 or equation 11 over ξ subject to the transitions in $\mathcal{D}_{\text{cert}}$ and bound U_ξ to obtain optimal solution ξ^*.
Select a valid slack threshold ζ based on ξ^* according to Theorem 1 (for invariance) or Theorem 2 (for reach-avoidance).
Estimate the measure of the certified initial set $\hat{p}_\zeta \approx \mu_{\mathcal{S}}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]$ by calculating the proportion of samples in $\mathcal{X}_{\text{init}}$ that satisfy the barrier condition for $\widetilde{\mathcal{S}}_\zeta$.
Compute the probabilistic safety/reach-avoid lower bound: $P_{\text{lower}} \leftarrow \max\left(0, 1 - \frac{\alpha}{\hat{p}_\zeta}\right)$.
Return ξ^*, α, and P_{lower}.

G ADDITIONAL BASELINES FOR RL AND PAC CERTIFICATION

G.1 REACH-AVOID RETURN AND COST

A similar trend is observed in the reach-avoid tasks. Our method improves safe target-reaching behavior by reducing dangerous action while keeping task performance competitive. The benefit is especially clear in environments where the controller needs to balance moving toward the target with long-term safety rules. Overall, the experiments support Q1: the proposed method offers a better return-safety balance than standard RL baselines.

We further clarify that PPO, Lag-Cost, and RCPO are not trained with barrier constraints. For these baselines, the barrier certificate is constructed after policy training, whereas MF-NBC, Lag-Barrier, and our method (PID-Barrier) jointly optimize the policy and barrier-related objectives. Therefore, the resulting ζ values are not directly comparable as indicators of learning quality. We mainly compare the barrier-learning methods in terms of the trade-off among task reward, control cost, and certification performance.

Table 3: Learning-performance and certification trade-offs among barrier-learning methods.

System	Task	Method	Return ↑	Cost ↓	ζ/ζ' ↓	Valid ↑	P ↑
Inverted pendulum	Invariant	MF-NBC	100.47 ± 109.98	0.250 ± 0.420	$0.0135[-0.0286,\ 0.0382]$	$\mathbf{5/5}$	$\mathbf{0.551}$
		Lag-Barrier	174.69 ± 8.66	0.054 ± 0.026	$\mathbf{-0.0049}[-0.0093,\ 0.0076]$	$\mathbf{5/5}$	0.458
		Ours	$\mathbf{179.41 \pm 10.40}$	$\mathbf{0.042 \pm 0.034}$	$-0.0031[-0.0107,\ 0.0061]$	$\mathbf{5/5}$	0.461
	Reach-avoid	MF-NBC	192.87 ± 169.71	0.230 ± 0.431	$0.0000[-0.0071,\ 0.0049]$	$1/5$	0.000
		Lag-Barrier	348.37 ± 6.25	0.064 ± 0.011	$\mathbf{-0.0031}[-0.0092,\ -0.0020]$	$\mathbf{5/5}$	$\mathbf{0.488}$
		Ours	$\mathbf{361.49 \pm 10.66}$	$\mathbf{0.040 \pm 0.023}$	$-0.0020[-0.0037,\ 0.0001]$	$\mathbf{5/5}$	0.405
Van der Pol	Invariant	MF-NBC	16.87 ± 154.78	0.616 ± 0.526	$1.2503[-0.0119,\ 1.6168]$	$2/5$	0.186
		Lag-Barrier	128.23 ± 127.12	0.230 ± 0.431	$0.0213[-0.0122,\ 1.6843]$	$4/5$	0.588
		Ours	$\mathbf{189.79 \pm 3.93}$	$\mathbf{0.024 \pm 0.013}$	$\mathbf{0.0005}[-0.0079,\ 0.0418]$	$\mathbf{5/5}$	$\mathbf{0.624}$
	Reach-avoid	MF-NBC	104.46 ± 127.93	0.354 ± 0.412	$0.1350[0.0017,\ 0.5995]$	$1/5$	0.000
		Lag-Barrier	$\mathbf{352.03 \pm 9.11}$	$\mathbf{0.022 \pm 0.019}$	$-0.0001[-0.0019,\ 0.0250]$	$\mathbf{5/5}$	0.365
		Ours	347.41 ± 6.86	0.028 ± 0.018	$\mathbf{-0.0008}[-0.0021,\ 0.0002]$	$\mathbf{5/5}$	$\mathbf{0.586}$
Hopfield	Invariant	MF-NBC	190.88 ± 2.52	$\mathbf{0.002 \pm 0.004}$	$\mathbf{0.1111}[0.0760,\ 0.1783]$	$1/5$	0.056
		Lag-Barrier	186.29 ± 5.22	0.012 ± 0.022	$0.1156[0.0547,\ 0.2284]$	$\mathbf{2/5}$	$\mathbf{0.089}$
		Ours	$\mathbf{192.00 \pm 1.32}$	0.004 ± 0.005	$0.1165[0.0934,\ 0.1705]$	$\mathbf{2/5}$	0.031
	Reach-avoid	MF-NBC	326.74 ± 74.30	0.016 ± 0.036	$0.0291[0.0033,\ 0.0642]$	$3/5$	0.001
		Lag-Barrier	377.21 ± 1.73	$\mathbf{0.000 \pm 0.000}$	$\mathbf{0.0022}[0.0013,\ 0.0038]$	$\mathbf{5/5}$	$\mathbf{1.000}$
		Ours	$\mathbf{380.97 \pm 1.53}$	$\mathbf{0.000 \pm 0.000}$	$0.0041[0.0010,\ 0.0048]$	$\mathbf{5/5}$	$\mathbf{1.000}$

Return and Cost are reported as mean $\pm$ standard deviation over five random seeds. Each seed is summarized using its final 100 training episodes. The certification slack is reported as median $[\min, \max]$ over five seeds. Valid denotes the number of seeds yielding a valid certificate. P is the certified-set measure associated with the median-slack result. Boldface denotes the best result within each system-task group;

G.2 VERIFICATION RESULTS

In the invariant task, we set $\lambda = 0.9$ and $C = -1$, so the verification upper bound is 0.1. In reach avoid task, we set $\lambda = 1.02$ and $G = 5$, so the verification bound is 0.1. The sample size N is chosen as 500 and the horizon k is chosen as 200 in every table. We use confidence level $1 - \beta = 95\%$, i.e., $\beta = 0.05$. We next introduce the verification results about Van der Pol and Hopfield.

Results analysis. In the Van der Pol system, the policy trained with PID-Barrier achieves a 100% validation success rate. Although its safe invariant set or reach-avoid set is smaller than those of some competing methods, the reward curves help explain this result. At the end of training, PID-Barrier obtains a relatively high return while maintaining the cost near zero, whereas most other methods show higher cost values (PPO in reach-avoid task). This indicates that other methods rely on greater control effort to achieve larger safe invariant or reach-avoid sets.

Results analysis. In the hopfield system, invariant task and reach-avoid task give us a consideration about training. This is because the definition of reach-avoid already incorporates a safety requirement. In our reach-avoid results, we obtain a certified volume of 1, whereas for the safe invariant set, the certified volume is 0.031. This discrepancy may be attributed to differences in the reward design during RL training, as well as differences in the barrier function parameters.

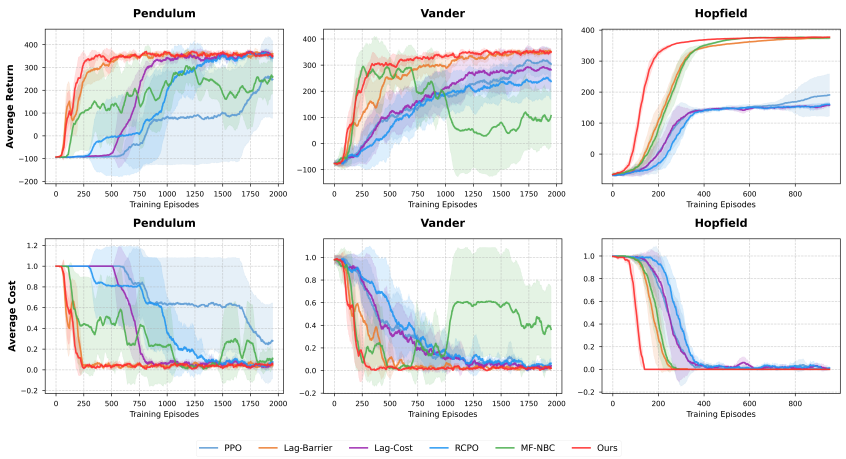

Figure 4: Average episode cost and return in the reach-avoid. The solid lines indicate the mean and the shaded regions indicate the 95% confidence interval across five runs.

Table 4: PAC verification results on the invariance and reach-avoidance in the Van der Pol.

	Invariant task					Reach-avoid task				
Method	$\zeta \downarrow$	Valid	$P \uparrow$	Theo.↑	Safe↑	$\zeta' \downarrow$	Valid	$P \uparrow$	Theo.↑	Safe↑
PPO†	$0.2822[0.0247, 1.4421]$	2/5	**0.684**	**0.977**	**1.000**	$0.0104[0.0059, 0.0197]$	**5/5**	0.138	0.884	**1.000**
Lag-C†	$0.0198[0.0065, 0.6608]$	3/5	0.629	0.975	**1.000**	$0.0019[-0.0011, 0.1023]$	3/5	0.360	0.956	**1.000**
RCPO†	$0.0387[0.0191, 1.1661]$	3/5	0.615	0.974	**1.000**	$0.0202[-0.0013, 0.0981]$	2/5	-	-	-
MF-NBC‡	$1.2503[-0.0119, 1.6168]$	2/5	0.186	0.914	**1.000**	$0.1350[0.0017, 0.5995]$	1/5	-	-	-
Lag-Barrier†	$0.0213[-0.0122, 1.6843]$	4/5	0.588	0.973	**1.000**	$-0.0001[-0.0019, 0.0250]$	**5/5**	0.365	0.956	**1.000**
Ours	$\mathbf{0.0005}[\mathbf{-0.0079}, \mathbf{0.0418}]$	**5/5**	0.624	0.974	**1.000**	$\mathbf{-0.0008}[\mathbf{-0.0021}, \mathbf{0.0002}]$	**5/5**	**0.586**	**0.973**	**1.000**

† Policy-only baselines require auxiliary barrier fitting before certification. ‡ Barrier-learning baselines do not provide our PAC-style formal certification. *Valid*: for the invariant task, the number of seeds with PAC certificates; for the reach-avoid task, whose estimated reach-avoid set measure is positive. The entry in each ζ or ζ' column is reported as median $[\min, \max]$ over 5 seeds. Here $P := \mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_{\zeta^*}]$, $\text{Theo.} := 1 - \frac{\alpha}{\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_{\zeta^*}]}$, and Safe denotes the empirical probability in the median results.

Overall, the experiments on high-dimensional systems demonstrate that our PAC verification method can provide valuable formal guarantees.

H Comparison of Scenario Optimization and Monte Carlo Methods in Reach-Avoid Tasks

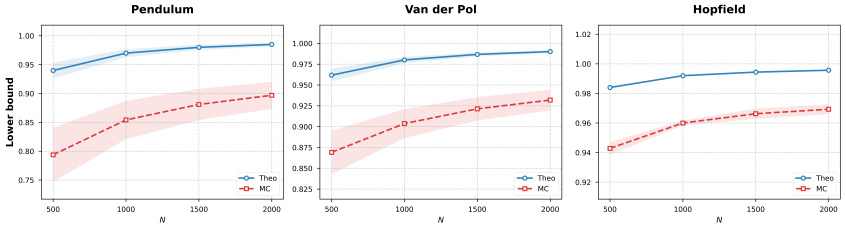

Figure 5: Comparison of scenario optimization and Monte Carlo methods in the reach-avoid task.

Results analysis. Consequently, the estimated verification lower bound shows greater variation under different random seeds. Given that the lower bound comparison $1 - \frac{\alpha}{\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]} \geq 1 - \frac{1-(\hat{p}_N - \epsilon)}{\mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_\zeta]}$, the MC method relies on two uncertainty parameters : $\hat{p}_N$ and ϵ. Consequently, the estimated verification lower bound shows greater variation under different random seeds.

Table 5: PAC verification results on the invariance and reach-avoidance in the Hopfield.

	Invariant task					Reach-avoid task				
Method	$\zeta \downarrow$	Valid	$P \uparrow$	Theo.↑	Safe↑	$\zeta' \downarrow$	Valid	$P \uparrow$	Theo.↑	Safe↑
PPO†	$0.1311[0.0879, 0.2023]$	2/5	**0.102**	**0.843**	**1.000**	$0.0412[0.0238, 0.0798]$	3/5	1e-4	0.000	**1.000**
Lag-C†	$0.1698[0.1288, 0.1873]$	0/5	-	-	-	$0.0231[0.0212, 0.0332]$	**5/5**	0.006	0.000	**1.000**
RCPO†	$0.1507[0.1084, 0.2156]$	0/5	-	-	-	$0.0298[0.0228, 0.0417]$	**5/5**	0.002	0.000	**1.000**
MF-NBC‡	$\mathbf{0.1111[0.076, 0.1783]}$	1/5	0.056	0.714	**1.000**	$0.0291[0.0033, 0.0642]$	3/5	0.001	0.000	**1.000**
Lag-Barrier†	$0.1156[0.0547, 0.2284]$	2/5	0.089	0.820	**1.000**	$\mathbf{0.0022[0.0013, 0.0038]}$	**5/5**	**1.000**	**0.984**	**1.000**
Ours	$0.1165[0.0934, 0.1705]$	2/5	0.031	0.483	**1.000**	$0.0041[0.0010, 0.0048]$	**5/5**	**1.000**	**0.984**	**1.000**

† Policy-only baselines require auxiliary barrier fitting before certification. ‡ Barrier-learning baselines do not provide our PAC-style formal certification. *Valid*: for the invariant task, the number of seeds with PAC certificates; for the reach-avoid task, whose estimated reach-avoid set measure is positive. The entry in each ζ or ζ' column is reported as median $[\min, \max]$ over 5 seeds. Here $P := \mathbb{P}[\boldsymbol{x} \in \widetilde{\mathcal{S}}_{\zeta^*}]$, Theo. $:= 1 - \frac{\alpha}{\mathbb{P}[\boldsymbol{x} \in \mathcal{S}_{\zeta^*}]}$, and Safe denotes the empirical probability in the median results.

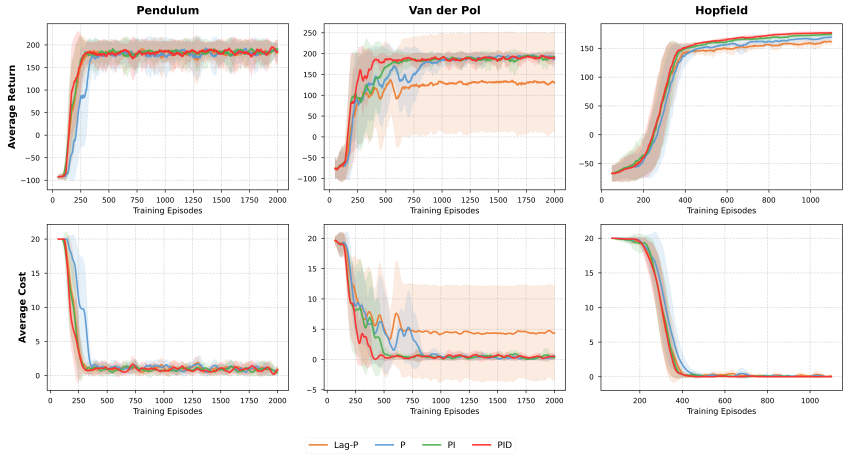

Figure 6: Comparison of PID parameters in the invariant task

I Ablation and Sensitivity Studies

We conduct a series of ablation and sensitivity studies to understand which design choices influence the performance of our method. Specifically, we look at three factors: the multiplier update rule during training, the trajectory step length used in verification, and the post-training certification threshold.

(1) Effect of PID-style multiplier updates. We mainly compare the return-cost trends and their impact on verification errors during validation in the setting Lag-P($K_p = 1$), P, PI, and PID-Barrier.

Results analysis. Return-cost: In these two tasks, PID-Barrier shows the most significant advantage for the Van der Pol system. This is because the original Van der Pol system does not have a stable equilibrium point, which leads to large trajectory fluctuations and makes policy updates more susceptible to disturbance. In contrast, PID-Barrier can quickly recover from short-term failures through its dynamic adjustment of the weighting coefficients.

Also, we evaluate the verification error in the invariant and reach-avoid task.

Results analysis. Verification error: PID-Barrier demonstrates stable results, all of which remain within the verification bound.

(2) Effect of verification step length $k_{\max}$ and sample size N. We investigate how the trajectory horizon $k_{\max}$ and sample size N affect the invariant and reach-avoid verification procedure. This parameter is important in practice and distinguishes our framework from several existing methods. We evaluate the same trained model under different values of $k_{\max}$ without changing training. We set $N \in 500, 1000, 1500, 2000$ for both the invariant and reach-avoid tasks. The horizon for the

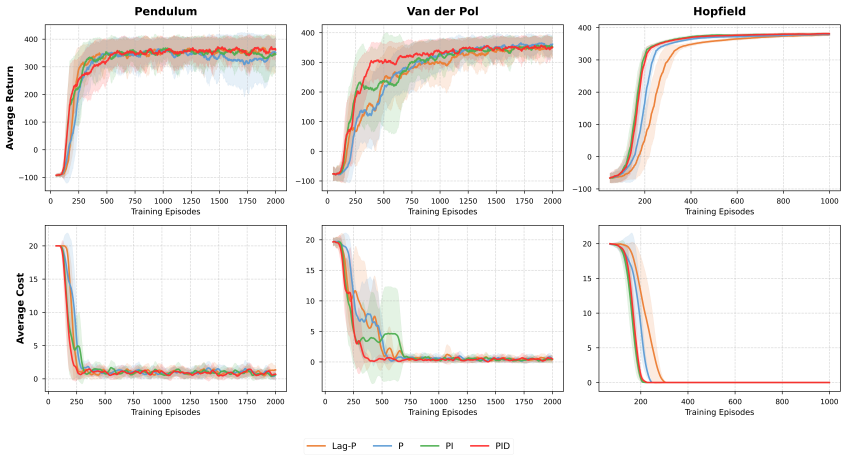

Figure 7: Comparison of PID parameters in the reach-avoid task

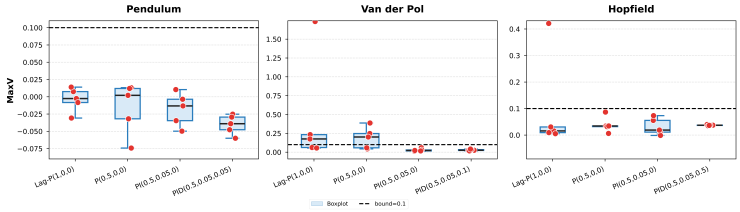

Figure 8: Comparison ξ^{s*} of PID parameters in the invariant task

invariant task was $k_{\max} \in 1, 5, 200$, because this choice helps us compare the results with the previously obtained one-step safe invariant set. For the reach-avoid task, the horizon was chosen as $k_{\max} \in 100, 150, 200$ because it is impossible for all points in $\widetilde{\mathcal{S}}_\zeta$ to enter the target set in a single step. This clarifies how the verification horizon influences certification quality: ξ^* and the probability lower bound.

Results analysis. Invariant task: A larger horizon length $k_{\max}$ slightly increases the violation value ξ^*, but it remains within the verification bound. This indicates that a larger horizon does not reduce the verification success rate; instead, it can provide stronger recursive properties. In the Van der Pol and Hopfield systems, a larger sample size N may cause a seed that previously passed verification to fail direct verification, because higher sampling is more likely to include boundary points or points that are difficult to train on. In addition, according to the lower-bound results, choosing a larger N can yield a better lower bound, although it is also necessary to consider the possibility of verification failure. This demonstrates the flexibility of our method.

Results analysis. Reach-avoid task: in the reach-avoid task, the results share both similarities with and differences from those of the invariant-set task. First, varying the horizon does not cause a substantial change in ξ^*. Second, increasing N likewise does not result in a substantial change in ξ^*. Third, in terms of the probabilistic lower bound, we observe that a larger horizon k yields a better bound, as suggested by the definition of $\widetilde{\mathcal{S}}_\zeta := \left\{ \boldsymbol{x} \in \mathcal{S} \;\middle|\; B(\boldsymbol{x}; \phi^{ra*}) > \frac{G}{\lambda^k} + \zeta \frac{1-\lambda^{-k}}{\lambda-1} \right\}$. When xi remains unchanged, an increase in k leads to a larger volume of $\widetilde{\mathcal{S}}_\zeta$, which in turn increases the lower bound of the probability.

(3) Threshold sweep after training. We then fix a trained policy-barrier pair and vary the certification threshold ζ across its admissible range in the invariant task. Importantly, this analysis requires no retraining. Instead, it yields a family of certified sets and corresponding probability lower bounds from the same trained model. For the reach-avoid task, when $\zeta = G(\lambda - 1)$, our candidate set is $\widetilde{\mathcal{S}}_\zeta := \{\boldsymbol{x} \in \mathcal{S} \mid B(\boldsymbol{x}; \phi^{ra*}) > G\}$, which is empty. Therefore, we do not discuss the effect of the range on the reach-avoid task here.

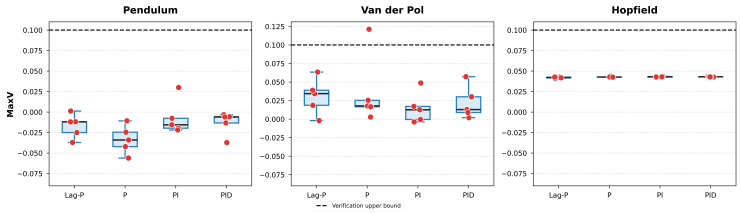

Figure 9: Comparison ξ^{sa*} of PID parameters in the reach-avoid task

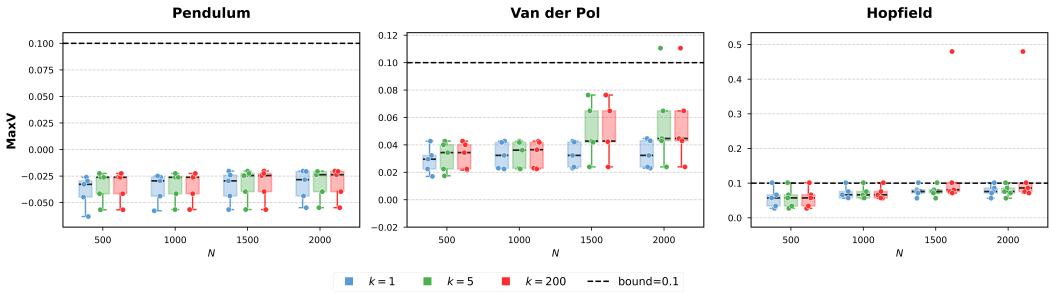

Figure 10: Comparison ξ^{s*} of horizon k and sample size N in the invariant task.

We can find that our method provides optional validation sets for all three cases, which indicates that the validation scope and the corresponding lower probability bounds can be flexibly determined according to practical needs.

I.1 AI USAGE

We used AI to generate some of the code in the experiments section (GPT 6-Astra; Grok 4.6). however, we carefully reviewed itparticularly the formal verification componentsto ensure compliance with the verification algorithms defined in the paper. The authors take full responsibility for the entire content of the paper.

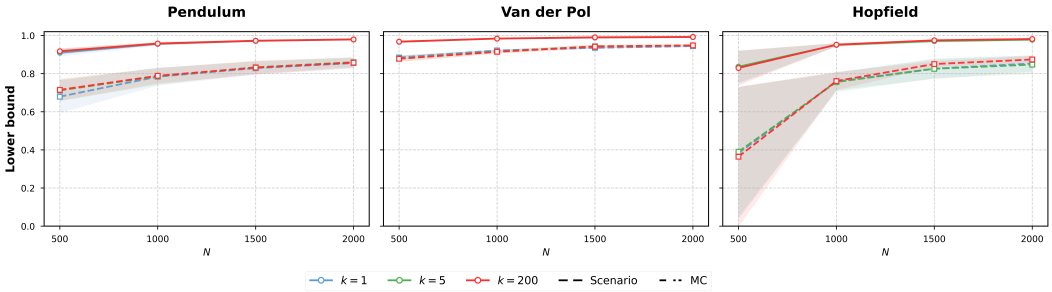

Figure 11: Comparison verification lower bound of horizon k and sample size N in the invariant task.

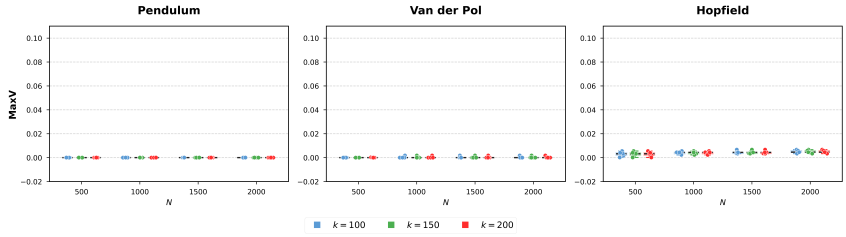

Figure 12: Comparison ξ^{sa*} of horizon $k_{\max}$ and sample size N in the reach-avoid task.

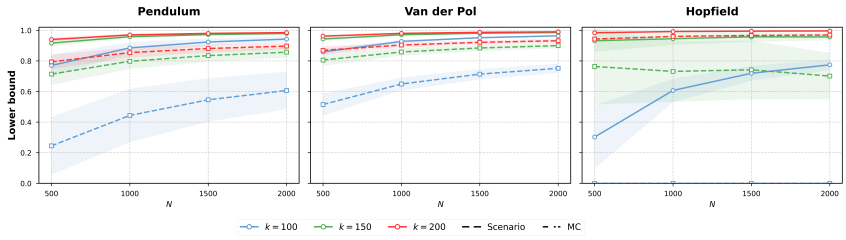

Figure 13: Comparison verification lower bound of horizon k and sample size N in the reach-avoid task.