Certifying Stability of Reinforcement Learning Policies using Generalized Lyapunov Functions

Kehan Long Jorge Cortés Nikolay Atanasov
Contextual Robotics Institute
University of California San Diego
{k3long, cortes, natanasov}@ucsd.edu

Abstract

Establishing stability certificates for closed-loop systems under reinforcement learning (RL) policies is essential to move beyond empirical performance and offer guarantees of system behavior. Classical Lyapunov methods require a strict stepwise decrease in the Lyapunov function but such certificates are difficult to construct for learned policies. The RL value function is a natural candidate but it is not well understood how it can be adapted for this purpose. To gain intuition, we first study the linear quadratic regulator (LQR) problem and make two key observations. First, a Lyapunov function can be obtained from the value function of an LQR policy by augmenting it with a residual term related to the system dynamics and stage cost. Second, the classical Lyapunov decrease requirement can be relaxed to a generalized Lyapunov condition requiring only decrease on average over multiple time steps. Using this intuition, we consider the nonlinear setting and formulate an approach to learn generalized Lyapunov functions by augmenting RL value functions with neural network residual terms. Our approach successfully certifies the stability of RL policies trained on Gymnasium and DeepMind Control benchmarks. We also extend our method to jointly train neural controllers and stability certificates using a multi-step Lyapunov loss, resulting in larger certified inner approximations of the region of attraction compared to the classical Lyapunov approach. Overall, our formulation enables stability certification for a broad class of systems with learned policies by making certificates easier to construct, thereby bridging classical control theory and modern learning-based methods.

1 Introduction

Designing and certifying stable control policies is a core challenge in both control theory and reinforcement learning (RL). Although RL methods are capable of learning policies that optimize long-term performance, their stability guarantees are often lacking. The classical Lyapunov theory [Lyapunov, 1992] offers a principled framework for certifying stability but constructing a function that satisfies the required stepwise decrease condition is often challenging in practice. While value functions from optimal control or RL encode long-term cumulative cost, they do not naturally satisfy the Lyapunov stability conditions.

We propose certifying stability using a generalized notion of Lyapunov function [Fürnsinn et al., 2023, 2025] that replaces the stepwise decrease condition with a multi-step, weighted decrease criterion, allowing temporary increases over a finite horizon. As a result, constructing a generalized Lyapunov stability certificate is easier, enabling stability verification for a broader class of policies and systems.

For linear systems with quadratic cost under discounted optimal control, the value function typically fails to satisfy classical Lyapunov conditions due to the influence of the discount factor. Postoyan et al. [2017] show that augmenting the value function with a quadratic residual term allows certifying

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

stability via a linear matrix inequality (LMI), although the guarantees are conservative with respect to the discount factor. Leveraging this insight, we show that augmenting the value function with a quadratic residual leads to a valid generalized Lyapunov function, which certifies stability for a boarder range of discount factors.

Building on the success of our approach in the linear setting, we extend it to nonlinear systems, where classical Lyapunov analysis is substantially more difficult. We augment value functions learned by RL algorithms (e.g., PPO [Schulman et al., 2017], SAC [Haarnoja et al., 2018]) with a neural residual term to form generalized Lyapunov functions, and jointly learn state-dependent multi-step weights to satisfy the generalized decrease condition. This joint learning scheme enables certification of nonlinear RL policies in settings where classical Lyapunov functions are challenging to construct by learning generalized certificates that tolerate non-monotonic behavior along trajectories.

In addition to showing stability of fixed policies, we extend our approach to support the joint synthesis of stable neural controllers and generalized Lyapunov certificates. Prior works [Chang et al., 2019, Wu et al., 2023, Yang et al., 2024] frame this problem as learning a controller–certificate pair that maximizes the volume of a certifiable region, while iteratively training on counter-examples where Lyapunov conditions fail. We incorporate our generalized multi-step condition into this formulation and show that it enables larger formally verifiable inner approximations of the region of attraction compared to standard one-step Lyapunov training.

Contributions. In summary, we make the following contributions:

- We introduce an approach for certifying stability of reinforcement learning policies by constructing generalized Lyapunov functions, composed of the policy's value function and a learned neural residual. Empirically, our formulation certifies policies on Gymnasium [Towers et al., 2024] and DeepMind Control Suite [Tassa et al., 2018] benchmarks where classical single-step Lyapunov methods fail.
- We show analytically that, for the linear quadratic regulator (LQR) problem, the validity of the proposed generalized Lyapunov function can be certified by a set of LMIs.
- We extend the formulation to jointly learn a control policy and a stability certificate using a generalized multi-step Lyapunov loss, resulting in larger certified regions of attraction compared to classical Lyapunov approaches.
- We provide an open-source implementation of our method at `https://github.com/ExistentialRobotics/Generalized_Policy_Stability`.

Related Work. Optimal control [Bertsekas, 2012] and reinforcement learning (RL) [Sutton et al., 1998] offer a variety of principled techniques for designing control policies to optimize the performance of intelligent agents and dynamical systems. In the special case of linear systems with quadratic rewards, the optimality and stability of LQR policies [Anderson and Moore, 1990] are well understood. However, despite a variety of methods for synthesizing policies (e.g., PPO [Schulman et al., 2017], SAC [Haarnoja et al., 2018], and DDPG [Lillicrap et al., 2015] are among the most widely used) for nonlinear systems with nonlinear costs, very few techniques are available for certifying the stability of the resulting closed-loop system.

Several works have explored connections between Lyapunov stability [Lyapunov, 1992] and RL, mainly in the context of safe RL. Perkins and Barto [2002] proposed a safe RL formulation by learning to switch among hand-designed Lyapunov-stable controllers. Berkenkamp et al. [2017] developed a model-based RL algorithm that uses Gaussian process dynamics and Lyapunov verification to safely improve policies while guaranteeing stability with high probability. Chow et al. [2018] formulate a Lyapunov-based method for safe RL in constrained Markov decision processes, using linear constraints to ensure safety during training. Hejase and Ozguner [2023] regularized policy training by penalizing violations of a learned Lyapunov condition to encourage stability in autonomous driving. Mittal et al. [2020] learn a Lyapunov function from suboptimal demonstrations and incorporate it as a terminal cost in a one-step model predictive control formulation to improve stability.

In parallel, there has been growing interest in jointly learning policies and Lyapunov functions from data to obtain stability guarantees for nonlinear systems. Chang et al. [2019] trained Lyapunov certificates with formal verification via satisfiability modulo theories. Other works constrain neural architectures to satisfy Lyapunov conditions structurally [Gaby et al., 2022, Boffi et al., 2021], or apply mixed-integer programming [Dai et al., 2021]. Long et al. [2023] studied neural distributionally robust Lyapunov functions under additive uncertainty. Wu et al. [2023] learn provably stable controllers for discrete-time nonlinear systems by jointly training a policy and Lyapunov function, with formal verification via mixed-integer linear programs. Yang et al. [2024] use gradient-guided falsification for training a controller-certificate pair, and apply formal verification via α, β-CROWN [Wang et al., 2021] to certify stability post training. See also Dawson et al. [2023] for a comprehensive survey.

2 Problem Formulation

Consider the discrete-time system:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_k \in \mathcal{X} \subseteq \mathbb{R}^n, \quad \mathbf{u}_k \in \mathcal{U} \subseteq \mathbb{R}^m, \tag{1}$$

where $\mathcal{X}$ is open and $\mathbf{f} : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is locally Lipschitz. A control policy $\boldsymbol{\pi} : \mathcal{X} \to \mathcal{U}$ for the system can be obtained by solving an infinite-horizon discounted optimal control problem:

$$\begin{aligned} J_\gamma^*(\mathbf{x}_0) = \min_{\boldsymbol{\pi}} J_\gamma^{\boldsymbol{\pi}}(\mathbf{x}_0) &:= \sum_{k=0}^{\infty} \gamma^k \, \ell(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)), \\ \text{s.t.} \;\; \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)), &\quad \mathbf{x}_k \in \mathcal{X}, \quad \boldsymbol{\pi}(\mathbf{x}_k) \in \mathcal{U}, \end{aligned} \tag{2}$$

where $\gamma \in (0, 1)$ is a discount factor and $\ell(\mathbf{x}, \mathbf{u})$ is a stage cost (or reward in the case of maximization), specifying the performance criterion.

Optimal control and reinforcement learning methods usually solve a problem like (2) to obtain a policy $\boldsymbol{\pi}$ and associated value function $J_\gamma^{\boldsymbol{\pi}}$. We consider a deterministic problem in our theoretical development for simplicity but apply our approach to stochastic problems in the evaluation (Section 5). Note that, while reinforcement learning methods work with stochastic control policies during training, only the mode of the final trained policy is used at test time, allowing us to treat it as deterministic.

Given a policy $\boldsymbol{\pi}$, we are interested in certifying whether it stabilizes the closed-loop system:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)). \tag{3}$$

We assume $\boldsymbol{\pi}(\mathbf{0}_n) = \mathbf{0}_m$, $\mathbf{f}(\mathbf{0}_n, \mathbf{0}_m) = \mathbf{0}_n$, $\ell(\mathbf{0}_n, \mathbf{0}_m) = 0$, and $\mathbf{0}_n \in \mathcal{X}$, so that the origin is an equilibrium point of the system. Stability can be certified by identifying a Lyapunov function.

Definition 2.1 (Lyapunov Function). Consider the closed-loop system (3). A continuous function $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is a *Lyapunov function* if it satisfies:

$$V(\mathbf{0}_n) = 0, \quad V(\mathbf{x}) > 0 \quad \forall \mathbf{x} \in \mathcal{X} \setminus \{\mathbf{0}_n\}, \tag{4a}$$

$$V\big(\mathbf{f}(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x}))\big) - V(\mathbf{x}) < 0 \quad \forall \mathbf{x} \in \mathcal{X} \setminus \{\mathbf{0}_n\}. \tag{4b}$$

Lyapunov functions certify the asymptotic stability of the system, as stated in the following result.

Theorem 2.2 (Asymptotic Stability via a Lyapunov Function [Khalil, 1996, Theorem 3.3]). *If there exists a Lyapunov function V as in Definition 2.1, then the origin $\mathbf{x} = \mathbf{0}_n$ is an asymptotically stable equilibrium of the system* (3).

When $\boldsymbol{\pi}$ arises from (2), the corresponding value function $J_\gamma^{\boldsymbol{\pi}}$ is also available. We are interested in whether $J_\gamma^{\boldsymbol{\pi}}$ can itself certify the stability of (3) or assist in constructing a certificate.

3 Lyapunov Stability Analysis for Linear Quadratic Problems

To understand the relationship between a discounted value function $J_\gamma^{\boldsymbol{\pi}}$ obtained from (2) and a Lyapunov function V certifying stability of the closed-loop system (3), we first study a simple setting with a linear system and quadratic stage cost.

Consider the discrete-time linear system: $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$, where $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{u}_k \in \mathbb{R}^m$, and $\mathbf{A}$, $\mathbf{B}$ are known constant matrices.

Choosing the stage cost in (2) as $\ell(\mathbf{x}, \mathbf{u}) = \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u}$ with $\mathbf{Q} = \mathbf{C}^\top \mathbf{C} \succeq 0$ and $\mathbf{R} \succ 0$ leads to the discounted LQR problem [Anderson and Moore, 1990]. The optimal value function $J_\gamma^*(\mathbf{x}) = \mathbf{x}^\top \mathbf{P}_\gamma \mathbf{x}$ is quadratic, where $\mathbf{P}_\gamma$ solves the discounted Algebraic Riccati Equation (ARE):

$$\mathbf{P}_\gamma = \mathbf{A}^\top \big(\mathbf{P}_\gamma - \mathbf{P}_\gamma \mathbf{B} \big(\gamma \mathbf{B}^\top \mathbf{P}_\gamma \mathbf{B} + \mathbf{R}\big)^{-1} \mathbf{B}^\top \mathbf{P}_\gamma\big) \mathbf{A} + \mathbf{Q}. \tag{5}$$

The corresponding optimal feedback gain is $\mathbf{K}_\gamma^\star = -(\gamma \mathbf{B}^\top \mathbf{P}_\gamma \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\top \mathbf{P}_\gamma \mathbf{A}$, which yields the policy $\boldsymbol{\pi}_\gamma^*(\mathbf{x}_k) = \mathbf{K}_\gamma^\star \mathbf{x}_k$. Under this policy, the closed-loop system evolves as

$$\mathbf{x}_{k+1} = \mathbf{F}_\gamma^* \mathbf{x}_k, \quad \text{where} \quad \mathbf{F}_\gamma^* = \mathbf{A} + \mathbf{B}\mathbf{K}_\gamma^\star. \tag{6}$$

Although J_γ^* satisfies the discounted Bellman equation, it is well understood that it does not guarantee Lyapunov decrease due to $\gamma \in (0, 1)$ [De Farias and Van Roy, 2003]. Moreover, verifying stability by directly analyzing the eigenvalues of $\mathbf{F}_\gamma^*$ and solving the discounted ARE (5) is nontrivial due to its nonlinear dependence on γ. This motivates the idea [Postoyan et al., 2017] of constructing valid Lyapunov functions by augmenting J_γ^* with a residual term. To formalize this, we first state standard assumptions that ensure the existence of a stabilizing policy and a well-defined value function.

Assumption 3.1. *The pair* $(\mathbf{A}, \mathbf{B})$ *is stabilizable and the pair* $(\mathbf{A}, \mathbf{C})$ *is detectable.*

The following result shows the existence of a Lyapunov certificate derived from the value function.

Theorem 3.2 (**Lyapunov Stability via LMIs for Discounted LQR** [Postoyan et al., 2017]). *Suppose Assumption 3.1 holds. Consider an optimal policy* $\boldsymbol{\pi}_\gamma^*$ *and value function* J_γ^* *obtained from the LQR problem with discount* γ. *Let* $\mathbf{P}$ *denote the solution to the undiscounted ARE in* (5) *(with* $\gamma = 1$*). If there exist symmetric positive definite matrices* $\mathbf{S}_0, \mathbf{S}_1 \succ 0$ *and scalars* $\varpi, \alpha > 0$ *such that:*

$$\begin{bmatrix} \mathbf{A}^\top \mathbf{S}_0 \mathbf{A} - \mathbf{S}_0 + \mathbf{S}_1 - \varpi \mathbf{Q} & \mathbf{A}^\top \mathbf{S}_0 \mathbf{B} \\ \mathbf{B}^\top \mathbf{S}_0 \mathbf{A} & \mathbf{B}^\top \mathbf{S}_0 \mathbf{B} - \varpi \mathbf{R} \end{bmatrix} \preceq 0, \quad \alpha \mathbf{P} \preceq \mathbf{S}_1, \tag{7}$$

then the function $V(\mathbf{x}) := J_\gamma^*(\mathbf{x}) + \frac{1}{\varpi} \mathbf{x}^\top \mathbf{S}_0 \mathbf{x}$ *is a Lyapunov function for the closed-loop system* (6), *and certifies global exponential stability for any discount factor satisfying* $\gamma > \frac{\varpi}{\varpi + \alpha}$.

Theorem 3.2 shows that while the discounted value function J_γ^* may not satisfy the Lyapunov condition on its own, it can be modified into a valid certificate when γ is sufficiently large. The required residual term and a computable lower bound on γ are obtained by solving a set of LMIs. However, this bound is often conservative compared to the true stability threshold γ^* obtained by analyzing the closed-loop dynamics via the discounted ARE, as illustrated in the following example.

Example 3.3 ([Postoyan et al., 2017, Example 1]). *Consider the scalar system* $x_{k+1} = 2x_k + u_k$ *with stage cost* $\ell(x, u) = x^2 + u^2$. *The optimal policy is* $u_k = K_\gamma^\star x_k$, *where*

$$K_\gamma^\star = -2\big(1 + 2\big(5\gamma - 1 + \sqrt{(5\gamma - 1)^2 + 4\gamma}\big)^{-1}\big)^{-1}.$$

The closed-loop multiplier is $F_\gamma^* = 2 + K_\gamma^\star$, *and the origin is globally exponentially stable if and only if* $|F_\gamma^*| < 1$, *which is equivalent to* $\gamma > \gamma^* = 1/3$. *However, applying the LMIs from Theorem 3.2 yields a feasible solution for* $\gamma > 0.8090$, *which is significantly more conservative.*

Example 3.3 highlights the limitation of classical Lyapunov analysis and motivates the development of an alternative formulation with less conservative conditions for certifying stability.

4 Generalized Lyapunov Stability and Applications to Linear Systems

Building on the observation from Example 3.3, we introduce a generalized notion of Lyapunov stability [Fürnsinn et al., 2023, Definition 2.2] that relaxes the classical pointwise decrease condition.

Definition 4.1 (**Generalized Lyapunov Function**). Consider the closed-loop system in (3). A continuous function $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is a *generalized Lyapunov function* if it satisfies (4a) and there exist an integer $M \in \mathbb{N}_{>0}$ and state-dependent non-negative weights $\sigma_1(\mathbf{x}), \ldots, \sigma_M(\mathbf{x})$ such that

$$\frac{1}{M} \sum_{i=1}^{M} \sigma_i(\mathbf{x}) \geq 1, \tag{8}$$

and, for any $\mathbf{x} \in \mathcal{X} \setminus \{\mathbf{0}_n\}$, the following generalized decrease condition holds:

$$\frac{1}{M} \left(\sum_{i=1}^{M} \sigma_i(\mathbf{x})\, V(\mathbf{x}_i) \right) - V(\mathbf{x}) < 0, \tag{9}$$

where $\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \boldsymbol{\pi}(\mathbf{x}_i))$ with $\mathbf{x}_0 = \mathbf{x}$.

Condition (9) generalizes the decrease requirement in (4b) by allowing temporary increases in V over individual steps, provided that its weighted average decreases over a finite horizon of length M. We state the stability guarantee provided by a generalized Lyapunov function in the next theorem.

Theorem 4.2 (**Asymptotic Stability via a Generalized Lyapunov Function**). *Consider the closed-loop system in* (3)*, and assume the policy $\boldsymbol{\pi}$ is Lipschitz on $\mathcal{X}$. If there exists a generalized Lyapunov function $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ as in Definition 4.1, then $\mathbf{x} = \mathbf{0}_n$ is an asymptotically stable equilibrium.*

See Appendix A for the proof (adapted from [Fürnsinn et al., 2023, Theorem 2]).

Remark 4.3 (**Relation to k-Inductive Verification**). *The generalized Lyapunov condition in* (9) *can be viewed through the lens of k-step inductive verification [Brain et al., 2015, Anand et al., 2022, Wooding and Lavaei, 2024]. In particular, by selecting weights $\sigma_1 = \cdots = \sigma_{k-1} = 0$ and $\sigma_k = M$, condition* (9) *reduces to $V(x_{t+k}) \leq (1-\alpha)V(x_t)$, which is analogous to a k-step Lyapunov decrease condition used in k-inductive reasoning. Classical k-induction requires a strict decrease at the final step while permitting bounded increases in intermediate steps. In contrast, Definition 4.1 enforces a decrease in a weighted average across multiple steps, with the weights σ_i chosen adaptively or learned during training. This distinction provides additional flexibility, enabling certificates that may be easier to construct for nonlinear systems with learned controllers.*

Returning to the LQR setting in Section 3, we consider certifying the stability of (6) via a generalized Lyapunov function. Building on Theorem 3.2, we augment J^*_γ with a quadratic residual and require the composite function to satisfy (9), leading to a new set of LMIs for stability certification.

Theorem 4.4 (**Generalized Lyapunov Stability for Discounted LQR via LMIs**). *Suppose Assumption 3.1 holds. Consider an optimal policy $\boldsymbol{\pi}^*_\gamma$ and value function J^*_γ obtained from the LQR problem with discount γ. Let $\mathbf{P}$ denote the solution to the undiscounted ARE in* (5) *(with $\gamma = 1$). If there exist symmetric positive definite matrices $\mathbf{S}_0, \mathbf{S}_1, \dots, \mathbf{S}_M$, scalars $\varpi, \alpha_1, \dots, \alpha_M > 0$, and weights $\sigma_1, \dots, \sigma_M \geq 0$ satisfying $\sum_{i=1}^M \sigma_i \geq M$, such that:*

$$\begin{bmatrix} \frac{\sigma_1}{M}\mathbf{A}^\top\mathbf{S}_0\mathbf{A} - \mathbf{S}_0 + \mathbf{S}_1 - \varpi\mathbf{Q} & \frac{\sigma_1}{M}\mathbf{A}^\top\mathbf{S}_0\mathbf{B} \\ \frac{\sigma_1}{M}\mathbf{B}^\top\mathbf{S}_0\mathbf{A} & \frac{\sigma_1}{M}\mathbf{B}^\top\mathbf{S}_0\mathbf{B} - \varpi\mathbf{R} \end{bmatrix} \preceq 0, \tag{10a}$$

$$\begin{bmatrix} \frac{\sigma_{i+1}}{M}\mathbf{A}^\top\mathbf{S}_0\mathbf{A} - \mathbf{S}_{i+1} - \varpi\mathbf{Q} & \frac{\sigma_{i+1}}{M}\mathbf{A}^\top\mathbf{S}_0\mathbf{B} \\ \frac{\sigma_{i+1}}{M}\mathbf{B}^\top\mathbf{S}_0\mathbf{A} & \frac{\sigma_{i+1}}{M}\mathbf{B}^\top\mathbf{S}_0\mathbf{B} - \varpi\mathbf{R} \end{bmatrix} \preceq 0, \quad \forall i = 1, \dots, M-1, \tag{10b}$$

$$\alpha_i\mathbf{P} \preceq \mathbf{S}_i, \quad \forall i = 1, \dots, M, \tag{10c}$$

then the function

$$V(\mathbf{x}) := J^*_\gamma(\mathbf{x}) + \frac{1}{\varpi}\mathbf{x}^\top\mathbf{S}_0\mathbf{x} \tag{11}$$

is a generalized Lyapunov function in the sense of Definition 4.1 (with constant weights), and certifies that the origin is globally exponentially stable for the closed-loop system (6)*, provided that:*

$$\gamma > \max\left(\frac{\sigma_M\varpi}{M\alpha_M}, \dots, \frac{\sigma_2\varpi}{M\alpha_2}, \frac{\sigma_1\varpi}{M(\varpi + \alpha_1)}\right). \tag{12}$$

The proof is provided in Appendix B.

Remark 4.5 (**Feasibility of Multi-Step LMIs**). *Note that when $\sigma_1 = M$ and $\sigma_i = 0$ for all $2 \leq i \leq M$, Theorem 4.4 recovers the classical Lyapunov result in Theorem 3.2. Therefore, if the original one-step LMIs* (7) *are feasible, then feasible solutions for* (10) *also exist. Empirically, the multi-step formulation can certify stability in additional cases where the one-step condition is infeasible, effectively enlarging the set of stabilizable systems. Formal characterization of this weaker feasibility property is left for future work.*

Remark 4.6 (**Choice of σ_i Weights**). *While the multi-step formulation enables optimizing the weights $\sigma_1, \dots, \sigma_M$ to minimize the certified lower bound on γ in* (12)*, finding globally optimal weights is challenging due to its non-convexity nature. In practice, heuristics such as grid search for small M and random sampling with local refinements for larger M are sufficient to achieve noticeable improvements over the one-step baseline.*

Example 4.7 (**Example 3.3 Revisited**). *We illustrate the benefits of the generalized multi-step LMIs from Theorem 4.4 in Example 3.3. We first focus on the case $M = 2$. Figure 1 shows how the certified* *bound on γ varies with the choice of weights σ_1 and σ_2, constrained by $\sigma_1 + \sigma_2 = 2$. Since the undiscounted value function ($\gamma = 1$) satisfies the Lyapunov condition, this bound is capped at $\gamma = 1$. The classical Lyapunov setting $(\sigma_1, \sigma_2) = (2, 0)$ yields a conservative bound of $\gamma > 0.8090$, whereas optimizing over σ_1 and σ_2 improves the bound to $\gamma > 0.6229$ at $(\sigma_1, \sigma_2) = (1.54, 0.46)$. Another important observation is that increasing M significantly reduces the certified lower bound on γ, progressively approaching the true stability threshold $\gamma^\star = 1/3$, as shown in Figure 2.*

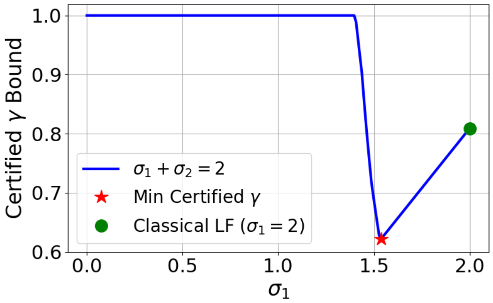

Figure 1: Certified γ bound for $M = 2$.

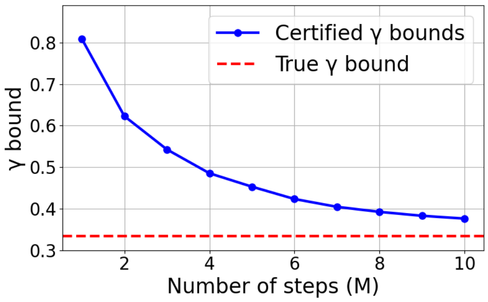

Figure 2: Certified γ bound versus M.

5 Nonlinear Systems: Stability Certification of RL Policies

Inspired by the treatment of linear systems described in Section 4, we first formulate a post-hoc approach to certify stability of policies for nonlinear systems obtained by RL using a generalized Lyapunov function. The key observation from Theorem 4.4 is that augmenting the optimal value function with a residual term can result in a valid generalized Lyapunov function, as in (11). Here, we use the same idea to form generalized Lyapunov functions for nonlinear systems with unknown dynamics, which is a problem considered by RL. The value function and policy obtained by an RL algorithm are typically parameterized by neural networks. Let $\boldsymbol{\pi}_{\mathrm{RL}}$ be a pre-trained RL policy (e.g., obtained by [Schulman et al., 2017, Haarnoja et al., 2018, Hansen et al., 2022]), and let $J_\gamma^{\boldsymbol{\pi}_{\mathrm{RL}}}$ denote its corresponding learned value function. We consider a generalized Lyapunov candidate as:

$$V(\mathbf{x}; \boldsymbol{\theta}_1) = J_\gamma^{\boldsymbol{\pi}_{\mathrm{RL}}}(\mathbf{x}) + \varphi(\mathbf{x}; \boldsymbol{\theta}_1), \tag{13}$$

where $\varphi(\mathbf{x}; \boldsymbol{\theta}_1)$ is a neural residual correction. To allow more flexibility in the generalized Lyapunov condition (9), we introduce a step-weighting network $\sigma(\mathbf{x}; \boldsymbol{\theta}_2) \in \mathbb{R}^M_{\geq 0}$ that outputs non-negative weights over a rollout horizon of length M. The weights are required to satisfy $\sum_{i=1}^M \sigma_i(\mathbf{x}; \boldsymbol{\theta}_2) \geq M$. These weights are then used to construct the generalized Lyapunov decrease condition:

$$F(\mathbf{x}_k) := \frac{1}{M}\left(\sum_{i=1}^{M} \sigma_i(\mathbf{x}_k; \boldsymbol{\theta}_2)\, V(\mathbf{x}_{k+i}; \boldsymbol{\theta}_1)\right) - (1 - \bar{\alpha})\, V(\mathbf{x}_k; \boldsymbol{\theta}_1), \tag{14}$$

where $\bar{\alpha} \in (0, 1)$ is a user-specified decay parameter. We train the networks $\varphi(\cdot; \boldsymbol{\theta}_1)$ and $\sigma(\cdot; \boldsymbol{\theta}_2)$ jointly by minimizing a loss that penalizes violations of the generalized Lyapunov condition:

$$\mathcal{L}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) := \frac{1}{N}\sum_{i=1}^{N} \mathrm{ReLU}\big(F(\mathbf{x}_k^{(i)})\big), \tag{15}$$

where $\{\mathbf{x}_k^{(i)}\}_{i=1}^N$ are sampled initial states.

Remark 5.1 (**Network Architecture**). *In practice, the learned value function $J_\gamma^{\boldsymbol{\pi}_{RL}}$ may not be optimal, hence, may not satisfy $J_\gamma^{\boldsymbol{\pi}_{RL}}(\mathbf{0}_n) = 0$. To ensure that the generalized Lyapunov candidate satisfies $V(\mathbf{0}_n; \boldsymbol{\theta}_1) = 0$ and is positive definite, we modify (13) as:*

$$V(\mathbf{x}; \boldsymbol{\theta}_1) := \left|J_\gamma^{\boldsymbol{\pi}_{RL}}(\mathbf{x}) - J_\gamma^{\boldsymbol{\pi}_{RL}}(\mathbf{0}_n)\right| + |\varphi(\mathbf{x}; \boldsymbol{\theta}_1) - \varphi(\mathbf{0}_n; \boldsymbol{\theta}_1)| + \beta\|\mathbf{x}\|^2, \tag{16}$$

where $\beta > 0$ is a small constant (the term $\beta\|\mathbf{x}\|^2$ enforces strict positive definiteness for $\mathbf{x} \neq \mathbf{0}_n$ and improves numerical stability near the origin.) The step-weighting network $\sigma(\mathbf{x}; \boldsymbol{\theta}_2) \in \mathbb{R}^M_{\geq 0}$ ends with a softmax layer scaled by M, ensuring the output weights satisfy $\frac{1}{M}\sum_{i=1}^M \sigma_i(\mathbf{x}; \boldsymbol{\theta}_2) = 1$.

Remark 5.2 (**Equilibrium Behavior**)**.** *In practice, RL policies often converge to a neighborhood near the origin rather than the origin itself. Thus, we train and verify the generalized Lyapunov condition over the set $\mathcal{X} \setminus \mathcal{B}(\mathbf{0}_n; \delta)$, where $\mathcal{B}(\mathbf{0}_n; \delta)$ denotes a ball of radius δ around the origin.*

Training and Evaluation Setup. We evaluate our method on two standard RL control benchmarks from Gymnasium [Towers et al., 2024] and the DeepMind Control Suite [Tassa et al., 2018]. RL policies $\boldsymbol{\pi}_{\text{RL}}$ and their corresponding value functions $J_\gamma^{\boldsymbol{\pi}_{\text{RL}}}$ are trained using implementations from Raffin et al. [2021], Hansen et al. [2022]. We collect a total of N rollout trajectories by simulating the closed-loop system under $\boldsymbol{\pi}_{\text{RL}}$, with each trajectory $\tau^{(i)} = \{\mathbf{x}_k^{(i)}\}_{k=0}^M$ starting from a randomly sampled initial state $\mathbf{x}_0^{(i)}$. Additional training and architecture details are provided in Appendix C.

Inverted Pendulum Swingup. We consider the inverted pendulum environment from Towers et al. [2024] with parameters $m = 1$, $l = 1$, $g = 10$, and control limits $|u| \leq \frac{mgl}{5} = 2$. The state space is $[-\pi, \pi) \times [-8, 8]$. Due to tight torque limits, the pendulum must swing back and forth to build momentum before reaching the upright position. This makes it challenging to synthesize a policy with a Lyapunov certificate valid over the entire state space. Prior work [Yang et al., 2024, Long et al., 2024] has shown that regions where stability can be verified are typically restricted to small neighborhoods near the upright position, failing to cover the full swing-up trajectories. In contrast, modern RL policies can discover effective swing-up behaviors but lack stability guarantees. Our method bridges this gap: we take a trained SAC policy Haarnoja et al. [2018] and apply our generalized Lyapunov function training with $M = 15$. Figure 3 shows the generalized Lyapunov function values along several trajectories from different initial states. As expected, the function exhibits non-monotonic behavior but shows an overall decline over the planning horizon. Figure 4 visualizes the Lyapunov function across the state space. Figure 5 plots the residual $F(\mathbf{x}_k)$ defined in (14), verifying that the generalized decrease condition is satisfied throughout the domain.

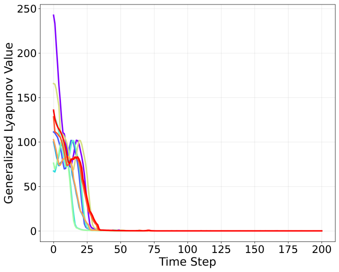

Figure 3: Generalized Lyapunov function values along trajectories.

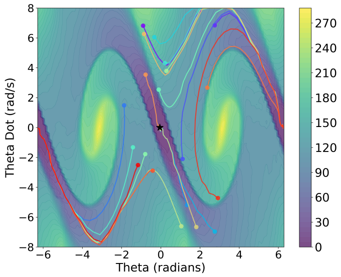

Figure 4: Generalized Lyapunov function value over the state space.

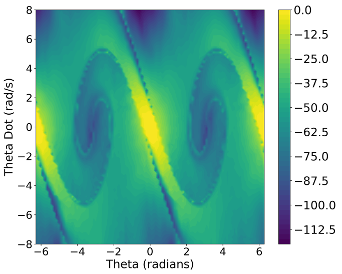

Figure 5: Generalized Lyapunov decrease condition.

Cartpole Swingup. We consider the cartpole swing-up task from Tassa et al. [2018]. The system state is represented as $[x, \cos(\theta), \sin(\theta), \dot{x}, \dot{\theta}]$, where x is the cart position, θ is the pole angle, and the remaining terms are their velocities. The goal is to swing the pole upright $\theta = 0$ and stabilize the position at $x = 0$. We use a TD-MPC policy [Hansen et al., 2022] and apply our certificate training (15) with $M = 20$. Figure 6 visualize the generalized Lyapunov condition across three representative 2D slices of the state space, with the remaining two states fixed at zero. The generalized decrease condition is satisfied throughout these slices, suggesting asymptotic stability of the RL policy.

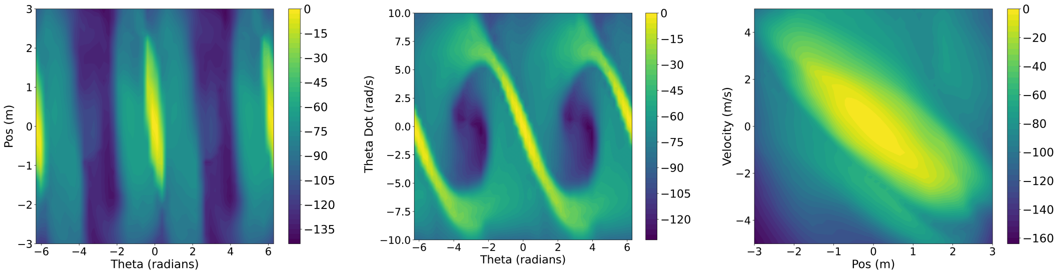

Figure 6: Generalized Lyapunov decrease condition for the cartpole swing-up using a TD-MPC policy.

In addition to qualitative visualizations, we quantitatively evaluate our learned certificates by sampling N_{test} states from the full state space and checking whether (14) is satisfied. For each environment, we train certificates for multiple RL policies (the PPO policy fails to stabilize the cartpole from some initial states). As shown in Table 1, the condition holds for all test states in both environments.

Table 1: Percentage of sampled test states satisfying $F(\mathbf{x}_k) \leq 0$ under different RL policies.

Environment	**RL Methods**	M	N_{test}	**% Satisfying** $F(\mathbf{x}_k) \leq 0$
Inverted Pendulum	PPO, SAC, TD-MPC	15	10,000	100%
Cartpole	SAC, TD-MPC	20	1,000,000	100%

Concentration of Learned Step-Weights. To further analyze the learned generalized Lyapunov function, we examine the distribution of the learned step-weights across the horizon. For each trained policy, the step-weight network was evaluated over $10,000$ uniformly sampled states from the state space, and the normalized weights were averaged across samples. The weights were then grouped into five equal bins corresponding to different segments of the Lyapunov horizon. Table 2 summarizes the fraction of the total weight assigned to each bin.

Table 2: Average fraction of total step-weight assigned to each portion of the rollout horizon. Results are averaged over 10,000 test states for each policy.

Environment + Policy	Steps 0–20%	20–40%	40–60%	60–80%	80–100%
Pendulum + PPO	12.3	11.4	16.7	21.8	**37.8**
Pendulum + SAC	10.2	13.3	17.5	22.6	**36.4**
Cartpole + SAC	10.8	10.5	16.2	25.6	**36.1**
Cartpole + TD-MPC	14.1	13.9	16.8	24.7	**30.5**

As shown in Table 2, the consistent concentration of weights toward the end of the horizon reveals a key strategy learned by our method. By assigning lower weights to initial steps and higher weights to later ones, the network effectively learns to tolerate initial non-monotonic transients in the Lyapunov candidate, focusing instead on ensuring a decisive, weighted decrease over the full horizon. This behavior empirically realizes the central benefit of the generalized Lyapunov condition: providing a flexible yet formal basis for certifying complex neural policies.

6 Joint Synthesis of Stable Neural Policies and Certificates

So far, we addressed the certification of stability for fixed pre-trained control policies. In this section, we consider *joint synthesis* of neural controllers and Lyapunov certificates and employ formal verification inspired by Chang et al. [2019], Dai et al. [2021], Wu et al. [2023], Yang et al. [2024]. Compared to prior methods, our generalized Lyapunov framework naturally extends this setting by replacing the standard one-step decrease condition with the proposed multi-step, weighted formulation in (9).

Definition 6.1 (Region of Attraction). The *region of attraction (ROA)* $\mathcal{R} \subseteq \mathbb{R}^n$ of the (locally asymptotically stable) origin for the discrete-time system (3) is the set of all points from which the system trajectory converges to the origin, i.e., $\mathbf{x}_0 \in \mathcal{R}$ implies $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{0}_n$.

In general, precise characterizations of $\mathcal{R}$ are challenging to obtain and one instead looks for suitable approximations. Here, we seek to jointly learn a policy $\boldsymbol{\pi}(\mathbf{x}; \boldsymbol{\phi})$ and a certificate function $V(\mathbf{x}; \boldsymbol{\theta}_1)$ such that the closed-loop system $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\pi}_\phi(\mathbf{x}_k))$ is provably asymptotically stable and obtain at the same time an inner approximation $\mathcal{S}$ of the ROA $\mathcal{R}$. Yang et al. [2024] formalized it as a constrained optimization that maximizes the volume of a Lyapunov sublevel set $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : V(\mathbf{x}) \leq \rho\}$ under Lyapunov constraints

$$\max_{\boldsymbol{\theta}_1, \boldsymbol{\phi}} \quad \mathrm{Vol}(\mathcal{S}) \tag{17a}$$

$$\text{s.t.} \quad V(\mathbf{0}_n; \boldsymbol{\theta}_1) = 0, \quad V(\mathbf{x}; \boldsymbol{\theta}_1) > 0 \quad \forall \mathbf{x} \in \mathcal{S} \setminus \{\mathbf{0}_n\}, \tag{17b}$$

$$V(\mathbf{x}_{k+1}; \boldsymbol{\theta}_1) - (1 - \bar{\alpha}) V(\mathbf{x}_k; \boldsymbol{\theta}_1) \leq 0 \quad \forall \mathbf{x}_k \in \mathcal{S}, \tag{17c}$$

where $\bar{\alpha} \in (0, 1)$ is a user-specified decay parameter. To utilize a generalized Lyapunov function, we replace (17c) with the generalized M-step condition for each $\mathbf{x}_k \in \mathcal{S}$, with $\sum_{i=1}^{M} \sigma_i(\mathbf{x}_k) \geq M$:

$$F(\mathbf{x}_k) := \frac{1}{M}\left(\sum_{i=1}^{M} \sigma_i(\mathbf{x}_k) V(\mathbf{x}_{k+i}; \boldsymbol{\theta}_1)\right) - (1-\bar{\alpha}) V(\mathbf{x}_k; \boldsymbol{\theta}_1) \leq 0, \quad \sigma_i(\mathbf{x}_k) \geq \underline{\sigma} > 0, \tag{18}$$

where $\underline{\sigma} \in \mathbb{R}_{>0}$ is a uniform lower bound on the weights. The condition (18) enables learning certificates that tolerate non-monotonic behavior along a trajectory, but it no longer guarantees that $\mathcal{S}$ is *forward invariant*. However, as Theorem 6.2 shows, $\mathcal{S}$ still defines a valid inner approximation of the ROA and guarantees asymptotic stability of the origin.

Theorem 6.2 (**Asymptotic Stability Relative to** $\mathcal{S}$). *If the optimization problem* (17) *is solved with the generalized condition* (18) *instead of* (17c)*, then* $\mathcal{S} \subset \mathcal{R}$.

See Appendix D for the proof.

Training Formulation. Following [Yang et al., 2024], we reformulate the problem (17) into a learning objective by sampling states $\mathbf{x}_k \in \mathcal{X}$ and penalizing violations of the Lyapunov conditions using soft constraints. For each $\mathbf{x}_k \in \mathcal{D}$, we simulate M forward steps under the closed-loop dynamics and compute the generalized Lyapunov residual $F(\mathbf{x}_k)$ as defined in (18).

To enforce stability and domain constraints, we define the stability loss:

$$\mathcal{L}_{\text{stab}}(\mathbf{x}_k) := \text{ReLU}\left(\min\left\{\text{ReLU}(F(\mathbf{x}_k)) + c_0 \mathcal{H}(\mathbf{x}_k), \rho - V(\mathbf{x}_k, \boldsymbol{\theta}_1)\right\}\right), \tag{19}$$

where $\mathcal{H}(\mathbf{x}_k) := \sum_{i=1}^{M} \|\text{ReLU}(\mathbf{x}_{k+i} - \mathbf{x}_{\text{up}}) + \text{ReLU}(\mathbf{x}_{\text{lo}} - \mathbf{x}_{k+i})\|_1$ penalizes violations of the bounded domain $\mathcal{X} = \{\mathbf{x} \mid \mathbf{x}_{\text{lo}} \leq \mathbf{x} \leq \mathbf{x}_{\text{up}}\}$. The inner min ensures that the generalized Lyapunov decrease condition is only enforced inside the certified region $\mathcal{S}$.

To expand the certifiable region, Yang et al. [2024] proposed a surrogate region loss $\mathcal{L}_{\text{region}} := \sum_{j=1}^{N} \text{ReLU}\left(V(\mathbf{x}_j; \boldsymbol{\theta}_1)/\rho - 1\right)$, where the candidate states $\mathbf{x}_j$ are obtained via random boundary sampling or projected gradient descent (PGD) to minimize $V(\cdot; \boldsymbol{\theta}_1)$. PGD is also used for falsification by maximizing stability violations, generating additional training states for $\mathcal{D}$.

The final training objective is $\mathcal{L}(\boldsymbol{\theta}_1, \boldsymbol{\phi}) := \sum_{\mathbf{x}_k \in \mathcal{D}} \mathcal{L}_{\text{stab}}(\mathbf{x}_k) + c_1 \mathcal{L}_{\text{region}} + c_2 \|\boldsymbol{\theta}_1, \boldsymbol{\phi}\|_1$, where $\mathcal{D}$ contains states sampled randomly and generated by falsification.

Remark 6.3. *Although the theoretical framework (Theorem 4.2) supports trainable weights* $\sigma_i(\mathbf{x}_k)$*, including neural network parameterizations (see Section 5), certifying stability under jointly learned controllers, Lyapunov functions, and weights remains computationally challenging with existing tools [Wang et al., 2021, Gao et al., 2013]. In practice, we fix* σ_i *during training. For small values of* M*, we select the best-performing weight configuration via a simple grid search.*

Verification Formulation. After training, we verify that the generalized decrease condition holds within $\mathcal{X}$ using the α-β-CROWN verifier [Wang et al., 2021]. We formally certify Lyapunov stability of the origin if the following holds for all $\mathbf{x}_k \in \mathcal{X}$:

$$\left(-F(\mathbf{x}_k) \geq 0 \;\wedge\; \bigwedge_{i=1}^{M} \mathbf{x}_{k+i} \in \mathcal{X}\right) \;\vee\; \left(V(\mathbf{x}_k) \geq \rho\right). \tag{20}$$

As Theorem 6.2 implies, if (20) holds for all $\mathbf{x}_k \in \mathcal{X}$, then $\mathcal{S}$ is an inner approximation of $\mathcal{R}$.

Evaluation. We evaluate our generalized Lyapunov synthesis approach on three systems presented in [Yang et al., 2024]: inverted pendulum, path tracking, and 2D quadrotor (with a 6D state space). Detailed system dynamics and neural network architectures used are provided in Appendix E.

Figures 7 and 8 show the certified stability regions $\mathcal{S}$ for different horizon lengths M. We use fixed step weights $(\sigma_1, \sigma_2, \dots)$ selected via grid search: for the inverted pendulum, $(0.4, 1.6)$ for M=2 and $(0.3, 1.5, 1.2)$ for M=3; for path tracking, $(0.4, 1.6)$ and $(1.2, 1.2, 0.6)$, respectively. As shown in the figures, multi-step training and verification yields consistently larger certifiable ROAs.

Table 3 reports the quantitative results, including the volume $(\mathcal{S})$ and the verification time. The volume is estimated via Monte Carlo integration: we sample $n = 10^6$ points in $\mathcal{X}$ and compute the fraction satisfying $V(\mathbf{x}) \leq \rho$, averaged across trials and scaled by the total domain volume. To assess

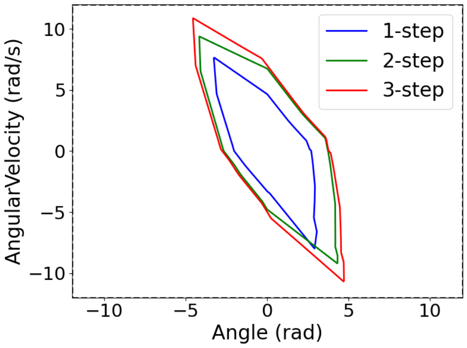

Figure 7: Certified ROAs for inverted pendulum.

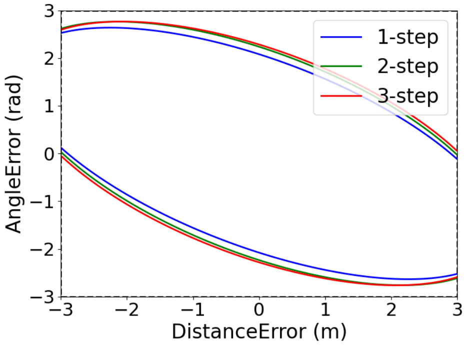

Figure 8: Certified ROAs for path-tracking.

robustness, we repeat each experiment 10 times with different random seeds and report the mean and standard deviation. The results confirm that our generalized stability formulation consistently certifies larger ROA volumes compared to the classical one-step Lyapunov approach. This improvement comes at the cost of longer verification times, since bounding multiple intermediate states and their weighted combinations increases the complexity of bound propagation and branching decisions.

Table 3: Certified region volume (left) and verification time (right) under different M-step Lyapunov training. Abbreviations: I.P. = Inverted Pendulum, P.T. = Path Tracking, 2.Q. = 2D Quadrotor.

System	$M = 1$	$M = 2$	$M = 3$	$M = 1$	$M = 2$	$M = 3$
I.P.	42.9 ± 1.2	76.7 ± 1.3	89.2 ± 1.2	11.7 ± 0.5	21.5 ± 0.8	39.2 ± 1.3
P.T.	21.8 ± 0.6	23.6 ± 0.5	23.9 ± 0.5	8.6 ± 0.5	19.5 ± 1.0	36.7 ± 1.3
2.Q.	103.5 ± 1.8	109.1 ± 2.0	113.7 ± 2.0	2209.5 ± 72.6	3858.7 ± 112.4	5628.6 ± 185.9

(a) Certified ROA volume ($\mathcal{S}$).

(b) Verification time (seconds).

7 Conclusion, Limitations, and Future Work

We presented a new approach for constructing stability certificates for closed-loop systems under policies obtained from optimal control and RL. We also showed that our method can be used for joint synthesis of neural policies and stability certificates. Rather than learning certificates from scratch, a key insight is that the value function of a considered policy can be augmented with a residual term to obtain a certificate candidate. We also replaced the classical Lyapunov decrease condition with a generalized multi-step Lyapunov formulation, which provides asymptotical stability guarantees while improving learning flexibility. Our theoretical and empirical results show that this perspective enables construction of more effective stability certificates compared to traditional Lyapunov approaches.

Our approach has several limitations. First, the certification horizon M is fixed during training, and it remains unclear how to determine the minimal viable M for a given system. Second, the method has not yet been applied to high-dimensional systems such as humanoids or dexterous manipulators. Third, formally verifying neural policies (e.g., from RL), along with learned state-dependent weights and certificate functions, remains computationally challenging with existing tools.

This work opens up several promising directions for future research. One key avenue is to develop partial-state stability certification, providing guarantees for task-relevant components such as torso height or velocity in humanoid locomotion rather than for the full system state, which can improve both scalability and interpretability. A second direction is to extend the generalized Lyapunov formulation to certify robust stability under bounded perturbations and stochastic disturbances, taking inspiration from input-to-state and robust Lyapunov theory. Another important direction is to relax the requirement of knowing the equilibrium state, allowing stability analysis when the equilibrium is unknown or non-isolated, as often occurs in reinforcement learning, by inferring equilibria from data or from the reward structure. Finally, investigating the connection between generalized Lyapunov conditions and optimality objectives may yield new insights for reward design and policy refinement in reinforcement learning.

Acknowledgments

We gratefully acknowledge support from ONR Award N00014-23-1-2353 and NSF Award CCF-2112665 (TILOS).

References

Mahathi Anand, Vishnu Murali, Ashutosh Trivedi, and Majid Zamani. K-inductive barrier certificates for stochastic systems. In *ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2022.

Brian D. O. Anderson and John B. Moore. *Optimal control: linear quadratic methods*. Prentice-Hall Inc., 1990.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, volume 30, page 908–919, 2017.

Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2012.

Nicholas Boffi, Stephen Tu, Nikolai Matni, Jean-Jacques Slotine, and Vikas Sindhwani. Learning stability certificates from data. In *Conference on Robot Learning*, pages 1341–1350. PMLR, 2021.

Martin Brain, Saurabh Joshi, Daniel Kroening, and Peter Schrammel. Safety verification and refutation by k-invariants and k-induction. In *International Static Analysis Symposium*, pages 145–161. Springer, 2015.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural Lyapunov control. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A Lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Hongkai Dai, Benoit Landry, Lujie Yang, Marco Pavone, and Russ Tedrake. Lyapunov-stable neural-network control. In *Robotics: Science and Systems*, Virtual, July 2021.

Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 39(3):1749–1767, 2023. doi: 10.1109/TRO.2022.3232542.

Daniela Pucci De Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865, 2003.

Annika Fürnsinn, Christian Ebenbauer, and Bahman Gharesifard. Relaxed feasibility and stability criteria for flexible-step MPC. *IEEE Control Systems Letters*, 7:2851–2856, 2023. doi: 10.1109/LCSYS.2023.3289150.

Annika Fürnsinn, Christian Ebenbauer, and Bahman Gharesifard. Flexible-step model predictive control based on generalized Lyapunov functions. *Automatica*, 175:112215, 2025. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2025.112215. URL https://www.sciencedirect.com/science/article/pii/S0005109825001074.

Nathan Gaby, Fumin Zhang, and Xiaojing Ye. Lyapunov-net: A deep neural network architecture for Lyapunov function approximation. In *IEEE Conference on Decision and Control*, pages 2091–2096, 2022. doi: 10.1109/CDC51059.2022.9993006.

Sicun Gao, Soonho Kong, and Edmund M Clarke. dreal: An smt solver for nonlinear theories over the reals. In *International Conference on Automated Deduction*, pages 208–214. Springer, 2013.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, pages 8387–8406. PMLR, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

Bilal Hejase and Umit Ozguner. Lyapunov stability regulation of deep reinforcement learning control with application to automated driving. In *American Control Conference*, pages 4437–4442, 2023.

Hassan Khalil. *Nonlinear Systems*. Prentice Hall, 1996.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Kehan Long, Yinzhuang Yi, Jorge Cortes, and Nikolay Atanasov. Distributionally robust Lyapunov function search under uncertainty. In *Learning for Dynamics and Control Conference*, pages 864–877. PMLR, 2023.

Kehan Long, Jorge Cortés, and Nikolay Atanasov. Distributionally robust policy and lyapunov-certificate learning. *IEEE Open Journal of Control Systems*, 3:375–388, 2024. doi: 10.1109/OJCSYS.2024.3440051.

Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.

Mayank Mittal, Marco Gallieri, Alessio Quaglino, Seyed Sina Mirrazavi Salehian, and Jan Koutník. Neural Lyapunov model predictive control: Learning safe global controllers from sub-optimal examples. *arXiv preprint arXiv:2002.10451*, 2020.

Theodore J Perkins and Andrew G Barto. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3(Dec):803–832, 2002.

Romain Postoyan, Lucian Buşoniu, Dragan Nešić, and Jamal Daafouz. Stability analysis of discrete-time infinite-horizon optimal control with discounted cost. *IEEE Transactions on Automatic Control*, 62(6):2736–2749, 2017. doi: 10.1109/TAC.2016.2616644.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *Advances in Neural Information Processing Systems*, 34, 2021.

Ben Wooding and Abolfazl Lavaei. Learning k-inductive control barrier certificates for unknown nonlinear dynamics beyond polynomials. *arXiv preprint arXiv:2412.07232*, 2024.

Junlin Wu, Andrew Clark, Yiannis Kantaros, and Yevgeniy Vorobeychik. Neural Lyapunov control for discrete-time systems. In *Advances in Neural Information Processing Systems*, volume 36, pages 2939–2955, 2023.

Lujie Yang, Hongkai Dai, Zhouxing Shi, Cho-Jui Hsieh, Russ Tedrake, and Huan Zhang. Lyapunov-stable neural control for state and output feedback: A novel formulation. In *International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=3xPMW9JURD.

A Proof of Theorem 4.2

Proof. Attractivity. We first show attractivity. Fix any initial condition $\mathbf{x}_0 \in \mathcal{X} \setminus \{\mathbf{0}\}$ and let $(\mathbf{x}_k)_{k\geq 0}$ be the corresponding trajectory. For each $k \geq 0$ choose an index

$$\ell(k) \in \{1, \ldots, M\} \quad \text{such that} \quad V\big(\mathbf{x}_{k+\ell(k)}\big) < V(\mathbf{x}_k), \tag{21}$$

whose existence is guaranteed by (9). Define the strictly increasing sequence of *return indices*

$$k_0 := 0, \qquad k_{j+1} := k_j + \ell\big(k_j\big), \quad j = 0, 1, \ldots, \tag{22}$$

for which, by construction,

$$0 < k_{j+1} - k_j \leq M, \qquad V\big(\mathbf{x}_{k_{j+1}}\big) < V\big(\mathbf{x}_{k_j}\big), \quad j = 0, 1, \ldots. \tag{23}$$

Due to the positive-definiteness of V, applying standard Lyapunov arguments on $\{V(x_{k_j})\}$, we conclude that $\lim_{k_j\to\infty} V(\mathbf{x}_{k_j}) = 0$.

Because the gaps satisfy $k_{j+1} - k_j \leq M$ and V is continuous, we have

$$\lim_{k\to\infty} V(\mathbf{x}_k) = 0. \tag{24}$$

Finally, the positive-definiteness of V yields $\mathbf{x}_k \to \mathbf{0}_n$ as $k \to \infty$. Thus the origin is *attractive*.

Stability. Since both the system dynamics $\mathbf{f}$ and the policy $\boldsymbol{\pi}$ are assumed to be Lipschitz, the closed-loop map $\mathbf{x} \mapsto \mathbf{f}(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x}))$ is also Lipschitz on the forward-invariant set $\mathcal{X}$. Let $L > 0$ denote the Lipschitz constant. Then, for all $0 \leq i < M$ and $k \geq 0$,

$$\|\mathbf{x}_{k+i}\| \;\leq\; L^i \, \|\mathbf{x}_k\|. \tag{25}$$

Fix an arbitrary $\varepsilon > 0$ and choose

$$\delta \;:=\; \frac{\varepsilon}{L^{M-1}}. \tag{26}$$

Consider any initial state with $\|\mathbf{x}_0\| < \delta$. By construction of the return indices k_j in (22) we have $\|\mathbf{x}_{k_{j+1}}\| \leq \|\mathbf{x}_{k_j}\|$ for all j, so in particular

$$\|\mathbf{x}_{k_j}\| \;<\; \delta, \qquad j = 0, 1, \ldots. \tag{27}$$

For an arbitrary time index k pick j such that $k_j \leq k < k_j + M$. Applying (25) with $i = k - k_j < M$ and (27) yields

$$\|\mathbf{x}_k\| \;\leq\; L^{M-1} \, \|\mathbf{x}_{k_j}\| \;<\; L^{M-1} \, \delta \;=\; \varepsilon.$$

Because ε was arbitrary, the ε–δ condition holds and the origin is Lyapunov stable.

In conclusion, the origin is both attractive and Lyapunov–stable on $\mathcal{X}$; therefore it is asymptotically stable relative to $\mathcal{X}$. □

Remark A.1 (Exponential Stability). *Suppose the function $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ satisfies the positive definiteness condition in (4a), and there exist constants $\beta_1, \beta_2, p > 0$ such that*

$$\beta_1 \|\mathbf{x}\|^p \leq V(\mathbf{x}) \leq \beta_2 \|\mathbf{x}\|^p, \quad \forall \mathbf{x} \in \mathcal{X}. \tag{28}$$

Suppose further that V satisfies the strict generalized decrease condition for all $\mathbf{x}_k \in \mathcal{X}$,

$$\frac{1}{M} \sum_{i=1}^{M} \sigma_i(\mathbf{x}_k) V(\mathbf{x}_{k+i}) - V(\mathbf{x}_k) \leq -\alpha(V(\mathbf{x}_k)), \tag{29}$$

where $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class-$\mathcal{K}$ function. Then the origin $\mathbf{x} = \mathbf{0}_n$ is exponentially stable.

B Proof of Theorem 4.4

For clarity, we present the detailed proof for the case $M = 2$. The general M-step case follows analogously by extending the argument to multiple future steps.

Proof. We consider the generalized Lyapunov decrease condition over two steps:

$$\frac{1}{2}\left(\sigma_2 V(\mathbf{x}_{k+2}) + \sigma_1 V(\mathbf{x}_{k+1})\right) - V(\mathbf{x}_k) < 0, \tag{30}$$

where $\sigma_1, \sigma_2 \geq 0$ and $\sigma_1 + \sigma_2 \geq 2$.

Using the discounted Bellman equation for the optimal value function V_γ,

$$V_\gamma(\mathbf{x}_{k+1}) - V_\gamma(\mathbf{x}_k) = -\gamma^{-1}\ell(\mathbf{x}_k, \mathbf{u}_k^*) + (1-\gamma)\gamma^{-1}V_\gamma(\mathbf{x}_k), \tag{31}$$

we can recursively expand the two-step generalized decrease condition for V_γ. Direct computation yields:

$$\begin{aligned}
&\frac{1}{2}\left(\sigma_2 V_\gamma(\mathbf{x}_{k+2}) + \sigma_1 V_\gamma(\mathbf{x}_{k+1})\right) - V_\gamma(\mathbf{x}_k)\\
&= \frac{\sigma_2}{2}\gamma^{-1}V_\gamma(\mathbf{x}_{k+1}) + \left(\frac{\sigma_1}{2}\gamma^{-1} - 1\right)V_\gamma(\mathbf{x}_k) - \frac{\sigma_1}{2}\gamma^{-1}\ell(\mathbf{x}_k, \mathbf{u}_k^*) - \frac{\sigma_2}{2}\gamma^{-1}\ell(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}^*)\\
&\leq \begin{bmatrix}\mathbf{x}_{k+1} & \mathbf{x}_k\end{bmatrix}\begin{bmatrix}\frac{\sigma_2}{2}\gamma^{-1}\mathbf{P} & \mathbf{0}\\ \mathbf{0} & (\frac{\sigma_1}{2}\gamma^{-1} - 1)\mathbf{P}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{k+1}\\ \mathbf{x}_k\end{bmatrix} - \frac{\sigma_2}{2}\ell(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}^*) - \frac{\sigma_1}{2}\ell(\mathbf{x}_k, \mathbf{u}_k^*)
\end{aligned} \tag{32}$$

However, due to the presence of the discount factor $\gamma \in (0, 1)$, the value function V_γ does not naturally satisfy the required decrease condition.

Following the approach in [Postoyan et al., 2017], we introduce an auxiliary quadratic function:

$$W(\mathbf{x}) = \frac{1}{\varpi}\mathbf{x}^\top \mathbf{S}_0 \mathbf{x}, \tag{33}$$

where $\mathbf{S}_0 \succ 0$ and $\varpi > 0$ are design parameters.

Define the composite function:

$$Y_\gamma(\mathbf{x}) := V_\gamma(\mathbf{x}) + W(\mathbf{x}). \tag{34}$$

Our goal is to ensure that Y_γ satisfies the generalized Lyapunov decrease condition in (30).

We compute the change in the auxiliary function $W(\mathbf{x})$ over two steps:

$$\begin{aligned}
&\frac{1}{2}\left(\sigma_2 W(\mathbf{x}_{k+2}) + \sigma_1 W(\mathbf{x}_{k+1})\right) - W(\mathbf{x}_k)\\
&= \frac{1}{\varpi}\left(\frac{\sigma_2}{2}\mathbf{x}_{k+2}^\top \mathbf{S}_0 \mathbf{x}_{k+2} + \frac{\sigma_1}{2}\mathbf{x}_{k+1}^\top \mathbf{S}_0 \mathbf{x}_{k+1} - \mathbf{x}_k^\top \mathbf{S}_0 \mathbf{x}_k\right).
\end{aligned} \tag{35}$$

Expanding the system dynamics

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \quad \mathbf{x}_{k+2} = \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{u}_{k+1}, \tag{36}$$

and substituting into (35), we express the change in W as

$$\begin{aligned}
&\frac{1}{2}\left(\sigma_2 W(\mathbf{x}_{k+2}) + \sigma_1 W(\mathbf{x}_{k+1})\right) - W(\mathbf{x}_k)\\
&= \frac{\sigma_1}{2\varpi}\left(\mathbf{x}_k^\top(\mathbf{A}^\top \mathbf{S}_0 \mathbf{A} - \mathbf{S}_0)\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{B}^\top \mathbf{S}_0 \mathbf{B}\mathbf{u}_k + 2\mathbf{x}_k^\top \mathbf{A}^\top \mathbf{S}_0 \mathbf{B}\mathbf{u}_k\right)\\
&\quad + \frac{\sigma_2}{2\varpi}\left(\mathbf{x}_{k+1}^\top \mathbf{A}^\top \mathbf{S}_0 \mathbf{A}\mathbf{x}_{k+1} + \mathbf{u}_{k+1}^\top \mathbf{B}^\top \mathbf{S}_0 \mathbf{B}\mathbf{u}_{k+1} + 2\mathbf{x}_{k+1}^\top \mathbf{A}^\top \mathbf{S}_0 \mathbf{B}\mathbf{u}_{k+1}\right).
\end{aligned} \tag{37}$$

To upper bound this change, we introduce positive definite matrices $\mathbf{S}_1, \mathbf{S}_2 \succ 0$ and impose:

$$\begin{aligned}
&\frac{1}{2}\left(\sigma_2 W(\mathbf{x}_{k+2}) + \sigma_1 W(\mathbf{x}_{k+1})\right) - W(\mathbf{x}_k)\\
&\leq -\frac{1}{\varpi}\mathbf{x}_{k+1}^\top \mathbf{S}_2 \mathbf{x}_{k+1} - \frac{1}{\varpi}\mathbf{x}_k^\top \mathbf{S}_1 \mathbf{x}_k + \frac{\sigma_1}{2}\ell(\mathbf{x}_k, \mathbf{u}_k^*) + \frac{\sigma_2}{2}\ell(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}^*).
\end{aligned} \tag{38}$$

Combining the changes in V_γ and W, we obtain:

$$\begin{aligned}
&\frac{1}{2}\left(\sigma_2 Y_\gamma(\mathbf{x}_{k+2}) + \sigma_1 Y_\gamma(\mathbf{x}_{k+1})\right) - Y_\gamma(\mathbf{x}_k) \\
&\leq -\mathbf{x}_{k+1}^\top \left(\frac{1}{\varpi}\mathbf{S}_2 - \frac{\sigma_2}{2}\gamma^{-1}\mathbf{P}\right)\mathbf{x}_{k+1} - \mathbf{x}_k^\top \left(\frac{1}{\varpi}\mathbf{S}_1 - \left(\frac{\sigma_1}{2}\gamma^{-1} - 1\right)\mathbf{P}\right)\mathbf{x}_k.
\end{aligned} \tag{39}$$

Thus, to ensure strict decay of Y_γ, it suffices to require:

$$\frac{1}{\varpi}\mathbf{S}_2 - \frac{\sigma_2}{2}\gamma^{-1}\mathbf{P} \succ 0, \tag{40}$$

$$\frac{1}{\varpi}\mathbf{S}_1 - \left(\frac{\sigma_1}{2}\gamma^{-1} - 1\right)\mathbf{P} \succ 0. \tag{41}$$

This leads to the requirement that γ must satisfy:

$$\gamma > \max\left(\frac{\sigma_2\varpi}{2\alpha_2}, \quad \frac{\sigma_1\varpi}{2(\varpi + \alpha_1)}\right), \tag{42}$$

where $\alpha_1, \alpha_2 > 0$ satisfy the comparison inequalities:

$$\alpha_1\mathbf{P} \preceq \mathbf{S}_1, \quad \alpha_2\mathbf{P} \preceq \mathbf{S}_2. \tag{43}$$

Furthermore, by comparing (37) and (38), the auxiliary matrices must satisfy the following LMIs:

$$\begin{bmatrix} \frac{\sigma_1}{2}\mathbf{A}^\top\mathbf{S}_0\mathbf{A} - \mathbf{S}_0 + \mathbf{S}_1 - \varpi\mathbf{Q} & \frac{\sigma_1}{2}\mathbf{A}^\top\mathbf{S}_0\mathbf{B} \\ \frac{\sigma_1}{2}\mathbf{B}^\top\mathbf{S}_0\mathbf{A} & \frac{\sigma_1}{2}\mathbf{B}^\top\mathbf{S}_0\mathbf{B} - \varpi\mathbf{R} \end{bmatrix} \preceq 0, \tag{44}$$

$$\begin{bmatrix} \frac{\sigma_2}{2}\mathbf{A}^\top\mathbf{S}_0\mathbf{A} - \mathbf{S}_2 - \varpi\mathbf{Q} & \frac{\sigma_2}{2}\mathbf{A}^\top\mathbf{S}_0\mathbf{B} \\ \frac{\sigma_2}{2}\mathbf{B}^\top\mathbf{S}_0\mathbf{A} & \frac{\sigma_2}{2}\mathbf{B}^\top\mathbf{S}_0\mathbf{B} - \varpi\mathbf{R} \end{bmatrix} \preceq 0. \tag{45}$$

This concludes the proof. □

C RL Certificates Training Details

C.1 Network Architectures

We use feedforward neural networks with LeakyReLU activations for both the residual network $\varphi(\cdot, \boldsymbol{\theta}_1)$ and step-weighting network $\sigma(\cdot, \boldsymbol{\theta}_2)$. The architectures are summarized in Table 4. The residual network is initialized using Kaiming initialization [He et al., 2015], and all biases are initialized to zero.

Table 4: Neural network architectures for the residual and step-weighting networks.

Component	Architecture	Output
Residual Network	3 layers, width 64	Scalar value
Step-weighting Network	3 layers, width 64	M-dimensional weights

C.2 Training Configuration

We train all networks using the Adam optimizer with an initial learning rate of 5×10^{-4}, a ReduceLROnPlateau scheduler (factor 0.5, patience 500), and a batch size of 256 for 1000 epochs. Gradients are clipped at 5.0. We set the decay parameter to $\bar{\alpha} = 0.02$ and the slack to $\beta = 0.01$. As discussed in Remark 5.2, we exclude a small ball around the origin during both training and evaluation. Specifically, we set $\delta = 0.05$ for the inverted pendulum and $\delta = 0.5$ for the cartpole.

C.3 Compute Resources

All experiments were run on a single workstation with an NVIDIA RTX 4090 GPU, AMD Ryzen 9 7950X CPU, and 64 GB RAM. Training the Lyapunov certificate for the inverted pendulum takes approximately 40 minutes, and for the cartpole environment around 4 hours. These durations include all rollouts and training steps for both the residual and step-weight networks.

D Proof of Theorem 6.2

For clarity, we prove the result for $M = 2$ and assume σ_1, σ_2 are constants.

Proof. We restate the assumptions. Let the closed-loop system be:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\pi}_\phi(\mathbf{x}_k)),$$

and $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a continuous function satisfying

$$V(\mathbf{0}_n) = 0, \quad V(\mathbf{x}_k) > 0 \quad \forall \mathbf{x}_k \in \mathcal{S} \setminus \{\mathbf{0}_n\}, \tag{46}$$

$$\frac{1}{2}(\sigma_1 V(\mathbf{x}_{k+1}) + \sigma_2 V(\mathbf{x}_{k+2})) \leq (1 - \bar{\alpha}) V(\mathbf{x}_k), \quad \forall \mathbf{x}_k \in \mathcal{S}, \tag{47}$$

with $\sigma_1, \sigma_2 > \underline{\sigma} > 0$ and $\sigma_1 + \sigma_2 \geq 2$.

Attractivity. The attractivity proof follows as in Appendix A, we conclude $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{0}_n$.

Stability. We follow the classical ε–δ Lyapunov proof in [Khalil, 1996, Thm. 3.3]. Fix an arbitrary radius $\varepsilon > 0$ and define

$$m_\varepsilon \;:=\; \inf_{\|x\|=\varepsilon} V(x). \tag{48}$$

Because the ε-sphere $\mathcal{E} = \{x \in \mathbb{R}^n : \|x\| = \varepsilon\}$ is compact and V is continuous, the infimum is achieved on $\mathcal{E}$. Positive-definiteness of V further implies $m_\varepsilon > 0$.

Define

$$c \;:=\; \frac{2(1-\bar{\alpha})}{\underline{\sigma}} \;>\; 0.$$

Because V is continuous at the origin and $V(0) = 0$, for every threshold $\eta > 0$ there exists a radius $\delta(\eta) > 0$ such that

$$\|x\| < \delta(\eta) \implies V(x) < \eta.$$

Choosing $\eta = m_\varepsilon / c$ and setting $\delta = \delta(m_\varepsilon/c)$, gives

$$\|x\| < \delta \implies V(x) < \frac{m_\varepsilon}{c}. \tag{49}$$

Now, pick any index k with $x_k \in \mathcal{S}$ and $\|x_k\| < \delta$. From (49), we have the bound

$$V(x_k) \;<\; \frac{m_\varepsilon}{c}. \tag{50}$$

Using the two–step Lyapunov condition (47) and the fact that $\sigma_1, \sigma_2 \geq \underline{\sigma}$, we obtain

$$\underline{\sigma}(V(x_{k+1}) + V(x_{k+2})) \;\leq\; 2(1-\bar{\alpha})\, V(x_k) \implies V(x_{k+i}) \;\leq\; c\, V(x_k), \quad i = 1, 2. \tag{51}$$

Combining (51) with (50) gives

$$V(x_{k+1}) < m_\varepsilon, \qquad V(x_{k+2}) < m_\varepsilon.$$

By definition of m_ε, $V(x) < m_\varepsilon$ implies $\|x\| < \varepsilon$; hence

$$\|x_{k+1}\| < \varepsilon, \qquad \|x_{k+2}\| < \varepsilon. \tag{52}$$

Equation (52) shows that, starting from any state inside the δ-ball, the trajectory remains inside the prescribed ε-ball.

Moreover, due to the strict decrease enforced by (47), it is not possible for both $V(\mathbf{x}_{k+1}) \geq V(\mathbf{x}_k)$ and $V(\mathbf{x}_{k+2}) \geq V(\mathbf{x}_k)$ to hold simultaneously. Therefore, at least one of the two future states must satisfy $V(\mathbf{x}_{k+i}) < V(\mathbf{x}_k)$ for $i \in \{1, 2\}$, hence the trajectory enters a strictly smaller sublevel set of V within two steps, allowing the generalized Lyapunov condition to be re-applied.

We now distinguish two cases to formalize the re-application logic:

Case 1: $\mathbf{x}_{k+1} \in \mathcal{S}$. Then the generalized condition can be re-applied at $\mathbf{x}_{k+1}$, and since $V(\mathbf{x}_{k+1}) < \epsilon$, the next steps $\mathbf{x}_{k+2}, \mathbf{x}_{k+3}$ will also lie within the ϵ-ball by repeating the same argument.

Case 2: $\mathbf{x}_{k+1} \notin \mathcal{S}$. Then $\mathbf{x}_{k+2} \in \mathcal{S}$, and again $V(\mathbf{x}_{k+2}) < \epsilon \Rightarrow \|\mathbf{x}_{k+2}\| < \epsilon$, allowing the condition to be re-applied at step $k + 2$.

Conclusion. In both cases, the trajectory remains within the ϵ-ball for all time, and the generalized decrease condition continues to hold along the sequence. Therefore, for all $\epsilon > 0$, there exists $\delta > 0$ such that:

$$\|\mathbf{x}_0\| < \delta \quad \Rightarrow \quad \|\mathbf{x}_k\| < \epsilon \quad \forall k \geq 0, \tag{53}$$

proving Lyapunov stability of the origin relative to $\mathcal{S}$.

Therefore, we conclude that $\mathcal{S}$ is a valid inner approximation of the ROA. □

E System Dynamics for Formal Verification

We describe below the continuous-time dynamics, control constraints, and verification domain $\mathcal{X}$ for each benchmark system used in our evaluations. All systems are discretized using explicit Euler integration with a time step of $\Delta t = 0.05$ s.

E.1 Inverted Pendulum

The inverted pendulum is modeled as a single-link rigid body rotating about its base, subject to bounded torque control. The continuous-time dynamics are:

$$\dot{\theta} = \omega, \tag{54}$$

$$\dot{\omega} = \frac{g}{\ell}\sin(\theta) - \frac{\beta}{m\ell^2}\omega + \frac{u}{m\ell^2}, \tag{55}$$

where θ is the pendulum angle (upright at $\theta = 0$), ω is the angular velocity, and u is the control input. We use the following parameters: gravity $g = 9.81$ m/s^2, mass $m = 0.15$ kg, length $\ell = 0.5$ m, and damping coefficient $\beta = 0.1$.

The control input is saturated to $u \in [-6, 6]$ Nm. The state vector is $\mathbf{x} = [\theta, \omega]^\top$, and the verification region is defined as:

$$\mathcal{X}_{\text{pendulum}} := \left\{ \mathbf{x} \in \mathbb{R}^2 \;\middle|\; \theta \in [-12, 12],\; \omega \in [-12, 12] \right\}.$$

E.2 Path Tracking

We consider a kinematic model for a ground vehicle tracking a circular path at constant forward speed v. The system state is $\mathbf{x} = [d_e, \theta_e]^\top$, where d_e is the lateral distance to the reference path and θ_e is the heading error. The control input u is the steering angle. The continuous-time dynamics are:

$$\dot{d}_e = v\sin(\theta_e), \tag{56}$$

$$\dot{\theta}_e = \frac{v}{L}u - \frac{\cos(\theta_e)}{R - d_e}, \tag{57}$$

where $L = 1$ m is the wheelbase length, $R = 10$ m is the radius of the reference path, and $v = 2$ m/s is the constant forward speed. The control input is bounded by $|u| \leq 0.84$.

The verification domain is defined as:

$$\mathcal{X}_{\text{path}} := \left\{ \mathbf{x} \in \mathbb{R}^2 \;\middle|\; d_e \in [-3, 3],\; \theta_e \in [-3, 3] \right\}.$$

E.3 2D Quadrotor

The 2D quadrotor models a rigid body with six states: horizontal and vertical positions x, z, pitch angle θ, and their corresponding velocities $\dot{x}, \dot{z}, \dot{\theta}$. The system is controlled by two rotor thrusts u_1, u_2, and the continuous-time dynamics are:

$$\ddot{x} = -\frac{1}{m}\sin(\theta)(u_1 + u_2), \tag{58}$$

$$\ddot{z} = \frac{1}{m}\cos(\theta)(u_1 + u_2) - g, \tag{59}$$

$$\ddot{\theta} = \frac{\ell}{I}(u_1 - u_2), \tag{60}$$

where $m = 0.486\,\text{kg}$, $\ell = 0.25\,\text{m}$, $I = 0.00383\,\text{kg} \cdot \text{m}^2$, and $g = 9.81\,\text{m/s}^2$. The state vector is $\mathbf{x} = [x, z, \theta, \dot{x}, \dot{z}, \dot{\theta}]^\top$. The control inputs are constrained to be within $[0, 2.5 \cdot u_{\text{eq}}]$, where $u_{\text{eq}} = \frac{mg}{2}$.

The verification domain is defined as:

$$\mathcal{X}_{\text{quadrotor}} := \left\{ \mathbf{x} \in \mathbb{R}^6 : \mathbf{x} \in [-0.75, 0.75]^2 \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times [-4, 4]^2 \times [-3, 3] \right\}.$$

E.4 Neural Network Architectures and Training Parameters

We adopt the same network structures used in Yang et al. [2024]. The configurations are summarized in Table 5. All networks use leaky ReLU activations.

Table 5: Neural network architecture for each system.

System	Controller Network	Lyapunov Network
Inverted Pendulum	4 layers, width 8	3 layers, widths [16, 16, 8]
Path Tracking	4 layers, width 8	Quadratic form
2D Quadrotor	2 layers, width 8	Quadratic form

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the paper's two main claims: (1) a stability certificate can be constructed by augmenting the value function of with a residual term, and (2) the classical decrease condition in Lyapunov theory can be relaxed to a generalized multi-step Lyapunov condition. These claims are supported by theoretical analysis and validated through examples and simulation results in Sections 4 to 6.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical results are stated with their full assumptions and are either proved in the appendix or directly adapted from cited references with proper attribution.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All experimental details needed to reproduce our results are provided in the paper and appendices. In particular, Appendix C details the training configuration for RL certificate experiments, and Appendix E specifies details for the joint synthesis part.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We open source our code and implementation details.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and evaluation details are provided in the Appendix C and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: Our main results are based on verification of stability conditions, not stochastic evaluations or statistical comparisons. Therefore, notions such as error bars or statistical significance do not apply.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide full compute resource details in Appendix C and in Section 6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This research does not involve human subjects, sensitive personal data, or potentially harmful applications. We have carefully reviewed the NeurIPS Code of Ethics and confirm that our work adheres to its guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

Justification: This work proposes a theoretical and algorithmic framework for stability certification of optimal control and reinforcement learning policies. It is foundational in nature and does not involve deployment, human interaction, or domain-specific applications that would raise immediate societal concerns.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper does not release any models or datasets with potential for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets used in this work are properly cited. This includes simulation environments from Gymnasium [Towers et al., 2024] and the DeepMind Control Suite [Tassa et al., 2018], as well as baselines and architectures from Yang et al. [2024]. For formal verification, we reference and use tools from Wang et al. [2021], with appropriate credit and citations.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: This work does not involve human subjects or crowdsourcing and therefore does not require IRB approval.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.