university of groningen / faculty of science and engineering / bernoulli institute

Lyapunov Dual-Policy Control: A Physics-Informed Framework for Provably Safe Reinforcement Learning

Bachelor's Project Thesis

Oscar Miró López-Feliu, s5209234, o.miro.i.lopez.feliu@student.rug.nl
Supervisors: J.D. Cárdenas-Cartagena

Abstract: Deep Reinforcement Learning (DRL) has achieved superhuman performance in diverse control tasks, yet its adoption in safety-critical physical systems remains limited by a lack of formal stability guarantees during training and inference. Classical controllers, while provably stable, are often conservative and model-bound. We introduce the Lyapunov Dual-Policy (LDP) controller, a framework that synthesizes three key concepts to bridge this gap: (i) A dual-policy structure, based on the work of Zoboli and Dibangoye, guarantees local asymptotic stability throughout training and deployment by blending a provably stable local Linear-Quadratic Regulator with a high-performance global DRL agent. (ii) The global policy is a physics-informed Lyapunov Actor-Critic (LAC), whose critic learns a maximal Lyapunov function by minimizing the residual of Zubov's Partial Differential Equation, thereby maximizing the verifiable domain of attraction. (iii) A Counter-Example Guided Abstraction Refinement (CEGAR) loop uses formal verification to iteratively correct the learned stability certificate and stabilize the training process. Experiments on a nonlinear inverted-pendulum benchmark show that the LDP framework achieves a 100% convergence rate, yielding a significantly higher reward, and faster convergence than competing classical control and DRL baselines.

1 Introduction

Recent developments in Deep Reinforcement Learning (DRL) have demonstrated superhuman performance in a variety of complex tasks, from mastering strategy games like Go and chess to achieving champion-level performance in drone racing (Silver, Hubert, Schrittwieser, Antonoglou, Lai, Guez, Lanctot, Sifre, Kumaran, Graepel, Lillicrap, Simonyan, and Hassabis, 2018; Kaufmann, Bauersfeld, Loquercio, Müller, Koltun, and Scaramuzza, 2023). However, when deploying DRL policies on physical systems, even minor sensor noise or environment perturbations can lead to chaotic, unstable behavior (Young and Pugeault, 2024). For instance, recent work by Young and Pugeault (2024) shows that standard DRL controllers for continuous control can be deterministically chaotic, where infinitesimal state perturbations lead to large, divergent future trajectories. This brittleness is unacceptable in safety-critical domains such as autonomous vehicles, medical robotics, and industrial automation, where stability and safety must be guaranteed.

To rigorously address the challenge of safe control, it is essential to establish a formal hierarchy of safety guarantees. Drawing from the literature on safe learning and control, we can classify safety constraints into three distinct levels (Brunke, Greeff, Hall, Yuan, Zhou, Panerati, and Schoellig, 2022):

- Level 1 (Soft Constraints / Encouragement) includes methods that encourage, but do not guarantee, safe behavior. This is typically achieved by modifying the reward function to penalize unsafe actions or by employing risk-sensitive optimization criteria. Most standard DRL algorithms operate at this level.
- Level 2 (Probabilistic Constraints) provides statistical safety guarantees, ensuring that constraints are satisfied with a certain high probability. These methods are common in learning-based control approaches that utilize probabilistic models, such as Gaussian Processes (GPs), to quantify uncertainty and define high-confidence safety boundaries. While stronger than soft constraints, these guarantees are not absolute and admit a non-zero probability of failure.
- Level 3 (Hard Constraints / Formal Guarantees) represents the highest level of safety, where methods provide provable, deterministic guarantees of stability or constraint satisfac-

The source code is available at `https://github.com/oscar-omlf/lyapunov-dual-policy`.

tion within a well-defined operational domain.

The work presented in this thesis aims to achieve Level 3 safety. Classical control theory offers powerful tools for this, such as Lyapunov stability theory, which enables the design of controllers with formal stability guarantees. The Linear-Quadratic Regulator (LQR) is an example of this, providing provable local stability for systems around an equilibrium point (Brunton and Kutz, 2022). However, LQR relies heavily on an accurate, linearized model of system dynamics, which is often unattainable for complex real-world systems (Khalil, 2002). This requirement, coupled with the inherently conservative performance resulting from linearization, severely restricts its practical applicability.

Conversely, modern DRL algorithms, such as the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, are entirely data-driven (Fujimoto, van Hoof, and Meger, 2018). They learn highly effective policies through direct interaction with the environment, obviating the need for a prior model. This flexibility allows them to achieve superior performance on complex, nonlinear tasks. Yet, this performance comes at the cost of safety; these methods offer no formal guarantees and operate at Level 1, making them unsuitable for safety-critical deployment. This creates a fundamental dichotomy: classical controllers like the LQR are optimal and high-performing under their strict operational assumptions (e.g., a known, linear system), but these assumptions often fail to capture the complex, nonlinear dynamics of real-world systems. In contrast, DRL methods excel in these complex environments but lack the formal guarantees required for safety-critical deployment.

Several recent works have begun to close the gap between high-performance DRL and formal stability guarantees. Chang, Roohi, and Gao (2019) were among the first to use neural networks to learn Lyapunov functions, which are then formally verified using a Satisfiability Modulo Theories (SMT) solver. Building on this, Wang and Fazlyab (2024) introduce a physics-informed actor-critic architecture where the critic's objective is to maximize the verifiable Domain of Attraction (DoA) by learning a maximal Lyapunov certificate from the residual of Zubov's Partial Differential Equation (PDE), while the actor is trained to select actions that maximally decrease the value of this certificate, thereby driving the system towards stability. Concurrently, Zoboli and Dibangoye (n.d) propose a dual-policy structure that blends a trusted, locally-stable LQR with a high-performance global DRL policy. This guarantees local stability while leveraging the learned policy's performance across the wider state space.

This thesis introduces the Lyapunov Dual-Policy (LDP) controller, a novel framework that synthesizes and advances these state-of-the-art approaches. Our key contributions are:

1. We design the LDP controller, which combines the guaranteed local stability of a dual-policy framework (Zoboli and Dibangoye, n.d) with a physics-informed global policy trained to explicitly maximize the verifiable DoA (Wang and Fazlyab, 2024). We introduce a key innovation by adapting the blended Q-function from Zoboli and Dibangoye (n.d) to inject the known, local Lyapunov structure directly into the global critic's learning process, providing a robust and direct learning signal.

2. We stabilize the training process by integrating a Counter-Example Guided Abstraction Refinement (CEGAR) loop. This loop uses formal verification not only to certify the final DoA but also to identify and add hard counter-examples to the training data, stabilizing the learning of the Lyapunov certificate.

3. We provide a comprehensive benchmark on a nonlinear control task, demonstrating that the LDP controller significantly outperforms classical LQR, standard DRL (TD3), and the constituent state-of-the-art methods (LAC, LAS-TD3) across metrics of performance, convergence rate, and speed of convergence.

2 Theoretical Framework

This section establishes the mathematical groundwork necessary to understand our proposed method. We begin with the fundamental concept of nonlinear control and Lyapunov theory, then we discuss the state-of-the-art learning-based approaches that form the predecessors of our work, and finally, we introduce the formal verification tools that enable provable guarantees.

2.1 Preliminaries

We consider the problem of stabilizing a continuous-time, controlled, dynamical system whose evolution is governed by an ordinary differential equation (ODE) of the form:

$$\dot{x} = f(x, u) \tag{2.1}$$

where $x \in \mathcal{D} \subseteq \mathbb{R}^n$ is the state vector residing in a constrained state-space $\mathcal{D}$, and $u \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input vector from a constrained action-space $\mathcal{U}$. The function $f : \mathcal{D} \times \mathcal{U} \to \mathbb{R}^n$ represents the system's vector field, which is assumed to be Lipschitz-continuous in its arguments to ensure the existence and uniqueness of solutions for any initial condition $x(0) = x_0$.

The central problem of setpoint stabilization is to design a feedback control policy $\pi : \mathcal{D} \to \mathcal{U}$, which maps a system state x to a control input $u = \pi(x)$. The goal is to render a desired equilibrium point, which we assume without a loss of generality to be the origin $x^* = 0$, asymptotically stable for the resulting closed-loop system:

$$\dot{x} = f\big(x,\, \pi(x)\big) \triangleq f_\pi(x) \tag{2.2}$$

An equilibrium point is a constant solution to the ODE in Eq. (2.2), i.e., a state x^* such that $f(x^*, \pi(x^*)) = 0$.

2.2 Lyapunov Theory

Aleksandr Lyapunov's seminal work provides a powerful tool for analysing the stability of dynamical systems without having to explicitly solve their governing differential equations. This method is the theoretical foundation of modern control theory and our proposed approach.

2.2.1 Stability in the Sense of Lyapunov

Before introducing Lyapunov's main theorem, we must formally define the different notions of stability for an equilibrium point (Vidyasagar, 2002; Fridman, 2014).

Definition 2.1: Stability. An equilibrium point x^* is said to be stable (or Lyapunov stable) if, for any $\epsilon > 0$, there exists a $\delta > 0$ such that if $||x(0)|| < \delta$, then $||x(t)|| < \epsilon$ for all $t \geq 0$. Intuitively, this means that for any arbitrarily small region $\epsilon > 0$ around the equilibrium, there exists a region of size $\delta > 0$ such that any trajectory starting within the δ-region will remain within the ϵ-region for all future time. This definition implies that trajectories do not diverge, but does not require them to converge to the origin.

Definition 2.2: Asymptotic Stability. An equilibrium point x^* is asymptotically stable if it is stable, and and there exists a $\delta > 0$ such that if $||x(0)|| < \delta$, then $\lim_{t\to\infty} x(t) = 0$. In other words, it is stable, and it is locally attractive. Local attractivity means that all trajectories beginning sufficiently close to the origin will eventually converge to the origin as time approaches infinity. For setpoint control problems, asymptotic stability is the desired property, as it guarantees both boundedness and convergence.

2.2.2 Lyapunov's Direct Method for Asymptotic Stability

Lyapunov's direct method provides a sufficient condition for proving asymptotic stability by constructing a scalar, energy-like function, known as a Lyapunov function, whose value continuously decreases along the system's trajectories.

Theorem 1: Lyapunov Stability Theorem (Khalil, 2002). Consider the controlled system in (2.2), with an equilibrium at the origin. If there exists a continuously differentiable scalar function $V : \mathcal{D} \to \mathbb{R}$, called a Lyapunov function, that satisfies the following three conditions, then the origin is asymptotically stable:

$$V(0) = 0, \tag{2.3}$$

$$V(x) > 0, \qquad \forall x \in \mathcal{D} \setminus \{0\} \tag{2.4}$$

$$\nabla V(x)^\top f_\pi(x) < 0 \qquad \forall x \in \mathcal{D}\setminus\{0\} \tag{2.5}$$

The intuition is that $V(x)$ acts as a generalized energy of the system. The conditions require that this energy is always positive (except at the equilibrium) and is always decreasing. A system whose energy is constantly decreasing must eventually settle at its minimum energy state, which is the stable point of equilibrium.

2.2.3 The Domain of Attraction and its Conservatism

A key definition associated with asymptotic stability is the Domain of Attraction (DoA), also known as the region of attraction, or basin of attraction. The DoA, denoted $\mathcal{A}$, is the set of all initial states x_0 from which the system's trajectories are guaranteed to converge to the stable equilibrium point at the origin (Han, El-Guindy, and Althoff, 2016).

More formally,

$$\mathcal{A} = \big\{x_0 \in \mathcal{D} \mid \lim_{t\to\infty} x(t; x_0) = 0\big\} \tag{2.6}$$

The size of the DoA is an important performance metric for any stabilizing controller, as it represents the robustness of the closed-loop system to perturbations and disturbances. A larger DoA implies the system can recover from a wider range of initial states or disturbances.

A key property of a Lyapunov function is that any of its sub-level sets contained within the domain $\mathcal{D}$ constitutes a certified region of attraction. A sub-level set is defined as $\mathcal{A}_c = \big\{x \in \mathcal{D} \mid V(x) < c\big\}$, for some $c > 0$. However, a significant limitation of Lyapunov theory is that it provides no guidance on how to find a $V(x)$ that yields the largest possible DoA. While many methods exist for estimating the DoA, such as using invariant sets or Sum-of-Squares programming, precisely identifying the boundary of the largest certified sub-level set remains a challenge. This boundary identification can be approached with numerical techniques using a binary search algorithm and an Satisfiability Modulo Theories (SMT) solver.

2.2.4 Maximal Lyapunov Functions and Zubov's Equation

Beyond the classical Lyapunov theorem, one can characterize the maximal domain of attraction $\mathcal{A}$ exactly by introducing two stronger notions: a maximal Lyapunov function V_m and a bounded Zubov function W.

Theorem 2: Maximal Lyapunov Functions (Vannelli and Vidyasagar, 1985; Vidyasagar, 2002). A continuously differentiable $V_m : \mathcal{A} \to [0, \infty)$ is called a maximal Lyapunov function if it satisfies:

$$V_m(0) = 0, \tag{2.7}$$
$$V_m(x) > 0, \qquad \forall x \in \mathcal{A} \setminus \{0\} \tag{2.8}$$
$$\nabla V_m(x)^\top f(x) = -\Phi, \qquad \forall\, x \in \mathcal{A} \tag{2.9}$$
$$V_m(x) = \infty \qquad \text{as } x \to \partial\mathcal{A} \text{ or } ||x|| \to \infty \tag{2.10}$$

where Φ is a user-chosen positive-definite function that sets the desired rate of energy dissipation and thereby shapes the Lyapunov landscape, and $\partial\mathcal{A}$ denotes the boundary of the DoA $\mathcal{A}$.

Under these conditions, every sub-level set $\mathcal{A}_c$ is an inner-approximation of $\mathcal{A}$, and as $c \to \infty$ they approach $\mathcal{A}$ exactly. Vannelli and Vidyasagar (1985) show that V_m always exists for any Lipschitz f on its true DoA, and that it can be constructed by solving a suitable first-order PDE and extending any local Lyapunov candidate globally.

While a maximal Lyapunov function perfectly characterizes the DoA, its value must go to infinity at the boundary, which can be difficult to approximate. Zubov's Theorem provides a clever alternative by constructing a bounded, energy-like function whose shape is explicitly designed to map the entire DoA into the finite range $[0, 1)$.

Theorem 3: Zubov's Theorem (Zubov, 1964; Vidyasagar, 2002). An alternative is to construct a bounded certificate $W : \mathcal{A} \to [0, 1)$, such that $W(x) = \tanh(\alpha V(x))$, whose 1-level set coincides with $\partial\mathcal{A}$:

$$W(0) = 0, \tag{2.11}$$
$$0 < W(x) < 1, \qquad \forall x \in \mathcal{A} \setminus \{0\} \tag{2.12}$$
$$W(x) \to 1 \qquad \text{as } x \to \partial\mathcal{A} \text{ or } ||x|| \to \infty \tag{2.13}$$
$$\nabla W(x)^\top f(x) = -\Psi(x)\big(1 - W(x)\big), \quad \forall\, x \in \mathcal{A} \tag{2.14}$$

The intuition is that, like a standard Lyapunov function, $W(x)$ represents a form of system energy that must always decrease outside the origin. However, the innovation is the $(1 - W(x))$ term in the decrease condition (Eq. 2.14). This term dynamically adjusts the required rate of energy decrease. Near the origin, $W(x)$ is small, so $(1 - W(x))$ is close to 1, and the function behaves like a normal Lyapunov function. But as the system state approaches the boundary of the DoA, $W(x)$ approaches 1, causing the $(1 - W(x))$ term to shrink to zero. This cleverly forces the energy decrease to become shallower, allowing the function to flatten out and form a "wall" at an energy level of 1, which precisely defines the boundary of the DoA.

Solving this Zubov PDE exactly yields

$$\mathcal{A} = \big\{x : W(x) < 1\big\} \tag{2.15}$$

so that W provides both, a certificate of stability and the precise DoA boundary. Learning-based methods can now be seen as attempts to approximate either V_m or W with neural networks, while retaining formal guarantees via counter-example search or residual minimization.

2.3 State-of-the-Art in Learning-Based Safe Control

With the theoretical foundations established, we now review the contemporary literature on learning-based control, focusing on the works that motivate our proposed LDP framework.

2.3.1 DRL Baseline: Actor-Critic and TD3

The actor-critic (AC) paradigm is a dominant approach in DRL for continuous control problems (Konda and Tsitsiklis, 1999). AC methods maintain two neural networks: the actor, parametrized by θ, which represents the policy $\pi^\theta(x)$ and is responsible for selecting actions, and the critic, parametrized by ϕ, which evaluates these actions through a value function $Q^\phi(x, u)$ or a Zubov function $W^\phi(x)$.

For our DRL baseline, we use the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, a state-of-the-art model-free AC method that significantly improves upon its predecessors through three key innovations: clipped double Q-learning, delayed policy updates, and target policy smoothing (Fujimoto et al., 2018). Despite its high empirical performance, the TD3 offers no formal stability or safety guarantees, making it a powerful but "unsafe" baseline.

2.3.2 Review of the Core Literature

Our work builds directly upon three foundational papers that represent the state-of-the-art of safe and stable learning-based control.

Chang et al. (2019): Neural Lyapunov Control Chang et al. were among the first to learn a neural Lyapunov function $V^\phi(x)$ by directly minimizing the empirical risk of violating the Lyapunov conditions. Here, ϕ represents the learnable parameters of the neural network. Concretely, given

N samples $\{x_i\}$ drawn from the domain $\mathcal{D}$, they optimize:

$$L_N(\phi) = \frac{1}{N}\sum_{i=1}^{N}\Big[\max\big(0,\, -V^\phi(x_i)\big) + \max\big(0,\, \nabla V^\phi(x_i)^\top f(x_i, \pi(x_i))\big)\Big] + V^\phi(0)^2, \tag{2.16}$$

where the first term penalizes any violations of positive definiteness ($V > 0$ for $x \neq 0$), Eq. (2.4), the second term enforces the negative lie derivative ($\nabla V^\top f < 0$), Eq. (2.5), and the last term imposes $V(0) = 0$ (Eq. (2.3)).

After this empirical risk converges, they switch to a Counter-Example Guided Abstraction Refinement (CEGAR) loop using the δ-complete SMT solver dReal (Gao, Kong, and Clarke, 2013). They encode the negation of the Lyapunov conditions as a satisfiability query up to precision δ:

$$\Phi_\varepsilon(x) \;:=\; \Big(\sum_{i=1}^{n} x_i^2 \geq \varepsilon\Big) \;\wedge\; \Big(V^\phi(x) \leq 0 \;\vee\; \nabla V^\phi(x)^\top f(x, \pi(x)) \geq 0\Big). \tag{2.17}$$

If dReal returns a counterexample x^*, it is added to the training set and the network is re-trained on the augmented data; otherwise, no counterexample exists and V^ϕ is certified to satisfy Eqs. (2.3) – (2.5) over $\mathcal{D}$. Due to numerical challenges near the origin, the verification procedure typically excludes a small ball around the equilibrium, $||x|| < \epsilon$, in their work $\epsilon = 0.25$. This yields a formally verified Lyapunov certificate and a provable inner approximation of the system's DoA.

Wang and Fazlyab (2024): Actor-Critic Physics-Informed Neural Lyapunov Control Wang and Fazlyab propose an actor-critic framework for stabilizing neural controllers trained alongside a Lyapunov critic. Their goal is to explicitly maximize the closed-loop DoA under input constraints. The critic network is trained to approximate a Zubov function $W^\phi(x) = \tanh(\alpha V(x))$, solving Zubov's PDE (Eq. (2.14)), while the actor π^θ is trained to reduce $W^\phi(x)$ along trajectories, effectively driving the system to stability.

To achieve this, their training objective includes a physics-informed loss function composed of five components. The first component, L_z, enforces the condition that the Lyapunov function must be zero at the origin ($W(0) = 0$), as per Eq. (2.11).

$$L_z = W^\phi(0)^2 \tag{2.18}$$

The second loss component, L_r encourages the critic network $W^\phi(x)$ to conform to the structure of a Zubov function, which can be defined in relation to an underlying (and unknown) standard Lyapunov function $V(x)$.

$$L_r = \mathbb{E}\Big[\big(W^\phi(x) - \tanh(\alpha\, V(x))\big)^2\Big] \tag{2.19}$$

In practice, the true underlying Lyapunov function $V(x)$ is unknown. For training purposes, it is estimated by simulating the system's trajectory from an initial state x_0 and calculating the integral of the state norm over time. This integral serves as a practical estimate for a candidate Lyapunov function where the rate of energy dissipation is chosen as $\Phi(x) = ||x||$.

$$V(x_0) \;=\; \begin{cases} \displaystyle\int_0^\infty \big\|x\big(t; x_0\big)\big\|\, dt, & \text{if convergent,} \\ \infty, & \text{otherwise.} \end{cases} \tag{2.20}$$

The core physics-informed component, L_p, penalizes the residual of Zubov's partial differential equation (Eq. 2.14). This drives the learned critic W^ϕ to satisfy the properties of a true Zubov function for the system dynamics. Here, $\Phi(x)$ is a chosen positive definite function.

$$L_p = \mathbb{E}\Big[\big(\nabla W^\phi(x)^\top f(x, \pi^\theta(x)) + \alpha\,\big(1 - W^\phi(x)\big)\big(1 + W^\phi(x)\big)\Phi(x)\big)^2\Big] \tag{2.21}$$

The fourth term, L_c, is the actor improvement loss. Minimizing this term encourages the actor policy π^θ to select actions that cause the value of the Zubov function W^ϕ to decrease as rapidly as possible, thus driving the system towards the stable equilibrium.

$$L_c = \mathbb{E}[\nabla W^\phi(x)^\top f(x, \pi^\theta(x))] \tag{2.22}$$

Finally, the boundary loss, L_b, ensures that the Zubov function approaches its maximum value of 1 at the boundary of the estimated DoA. This condition is enforced on states x' sampled from the edge of the training region.

$$L_b = \mathbb{E}[|W^\phi(x') - 1|] \tag{2.23}$$

These components are combined into a single objective function:

$$L(\theta, \phi) = \lambda_z L_z + \lambda_r L_r + \lambda_p L_p + \lambda_c L_c + \lambda_b L_b \tag{2.24}$$

where the λ-values are hyperparameters that balance the contribution of each loss component, in this work being $\lambda_z = 5.0$, $\lambda_r = 1.0$, $\lambda_p = 1.0$, $\lambda_c = 0.5$, and $\lambda_b = 5.0$.

These losses are evaluated by sampling states from two distinct regions: $R_1 \subseteq \mathcal{D}$, the primary region where training states x are sampled, and

a larger region $R_2 = \{a\,x : x \in R_1\}$; its boundary is ∂R_2, from which the samples x' are drawn. Typically, the scaling factor is set to $a = 2$, doubling the size of the training region. By optimizing these losses simultaneously, the method effectively learns both a robust stabilizing policy and a Lyapunov certificate. After training, a binary search algorithm using an SMT solver is used to formally verify the learned Zubov function $W(x)$, similarly to Chang et al. (2019), using $\epsilon = 0.5$.

Algorithm (2.1) provides a high-level overview of the actor-critic training loop employed by Wang and Fazlyab, highlighting state sampling and gradient updates. Trajectories are simulated using the fourth-order Runge-Kutta (RK4) integrator, a numerical integration method commonly used for approximating solutions to ODEs (Butcher, 2003).

Algorithm 2.1 Physics-Informed Neural Lyapunov Control

Randomly initialize W^ϕ and π^θ
for $n = 1, \ldots, T$ **do**
  Randomly sample $x_1, \ldots, x_K$ from R_1
  Randomly sample $x'_1, \ldots, x'_K$ from ∂R_2
  Simulate trajectories $x(t, x_i; \pi^\theta)$ (with RK4 integrator)
  Estimate $V(x_i) \approx \int_t \|x(t, x_i; \pi^\theta)\| dt$ for $i = 1, \ldots, M$
  Take a gradient step to minimize $L(\theta, \phi)$ in Eq. (2.24)
end for

Zoboli and Dibangoye (n.d): Locally Asymptotically Stable Deep Actor-Critic Algorithms For Quadratic Cost Setpoint Optimal Control Problems Zoboli and Dibangoye (n.d); Zoboli, Andrieu, Astolfi, Casadei, Dibangoye, and Nadri (2021) address the challenge of maintaining local asymptotic stability (LAS) in deep RL controllers by introducing a dual-policy architecture. In their approach, a trusted local policy π_{loc} (e.g. an LQR regulator) handles the system near the setpoint, while a learned global policy π^θ_{glob} (e.g., a DRL controller) is responsible for optimal control over the wider state space. A continuously differentiable blending function $h(x) \in [0, 1]$ interpolates between the two policies. Specifically, the blended policy is defined as:

$$\begin{aligned} \pi^\theta(x) \;&=\; \pi_{\text{loc}}(x) \\ &\quad +\; h_1(x)\left(\pi^\theta_{\text{glob}}(x) - \pi_{\text{loc}}(x)\right) \end{aligned} \tag{2.25}$$

such that $h_1(x) = 0$ near the equilibrium (making $\pi^\theta \approx \pi_{\text{loc}}$) and $h_1(x) \to 1$ far from the equilibrium (so π^θ relies on the global policy). This guarantees that within the local controller's domain of attraction, the behavior defaults to the provably stable controller, ensuring stability, while outside that region the agent can exploit the more performant learned policy. An analogous blending is used for the critic:

$$\begin{aligned} \hat{Q}_\pi(x, u) \;&=\; Q_{\text{loc}}(x, u) \\ &\quad +\; h_2(x)\left(Q^\phi_{\text{glob}}(x, u) - Q_{\text{loc}}(x, u)\right) \end{aligned} \tag{2.26}$$

where $Q_{\text{loc}}(x, u)$ is a known Lyapunov-based cost-to-go for the local policy (e.g. the LQR's quadratic value function). By folding in the reliable local value estimate, the critic's predictions are accurate near the equilibrium and gradually transition to the learned critic Q^ϕ_{glob} away from it.

The blending function $h_2(x)$ is analogous to $h_1(x)$. By choosing $h_2(x)$ appropriately, the critic inherits local accuracy from Q_{loc} near the equilibrium (where Q^ϕ_{glob} may be unreliable), and gradually uses the learned critic in the outer region. As a result, the value estimates used for policy improvement are consistent with the true local Lyapunov value function in the inner zone, injecting information to the global policy learning process.

The functions $h_1(x)$ and $h_2(x)$ are designed to transition smoothly from 0 to 1 as the state moves outward. Zoboli and Dibangoye base this transition on a normalized Lyapunov function value. Let $V(x)$ be a known positive-definite Lyapunov function for the local policy (for an LQR, one can use the Lyapunov candidate $V(x) = x^\top P x$, where P is a unique, symmetric positive-definite solution to the continuous-time algebraic Riccati equation (CARE) associated with the LQR problem). If c^* is the maximum V within the local domain of attraction, they define $\nu(x) = \frac{V(x)}{c^*}$ so that $\nu(x) \in [0, 1]$ for states inside the local verified region. This normalized Lyapunov level $\nu(x)$ serves as the blending variable. The blending functions are defined as

$$h_1(x) = \tanh\Big(s\,\nu(x)\Big) \tag{2.27}$$

$$h_2(x) = \tanh\Big(s\,\nu(x)^{3/2}\Big) \tag{2.28}$$

where $s = \operatorname{arctanh}(\beta)$ and $0 < \beta < 1$. This form implies $h_1(x) = h_2(x) = 0$ (the local controller dominates at the origin). And as $\nu \to 1$ (approaching the edge of the local region), $h_1 \to \tanh(s) = \beta$ and $h_2 \to \tanh(s) = \beta$. In other words, at the boundary of the local domain ($V(x) = c^*$), the global policy/critic are weighted by β and the local ones by $1 - \beta$.

3 Methodology

This section details the experimental setup designed to evaluate the proposed Lyapunov Dual-Policy (LDP) controller. We first describe the benchmark control task, then outline the architectures of the baseline and state-of-the-art controllers used for comparison, followed by a detailed description of the LDP controller and its training routine. Finally, we specify the metrics used for evaluation.

3.1 Environment: The Inverted Pendulum

The benchmark used in this study is the classic inverted pendulum swing-up and balancing task. The continuous-time dynamics of the pendulum are governed by the second-order ordinary differential equation:

$$\ddot{\theta} = \frac{g}{l}\sin(\theta) - \frac{b}{ml^2}\dot{\theta} + \frac{1}{ml^2}u \tag{3.1}$$

where θ is the angle from the upward vertical position, $\dot{\theta}$ is the angular velocity, and $\ddot{\theta}$ is the angular acceleration. The physical parameters are set to match those in Wang and Fazlyab (2024): gravity $g = 9.81\,\mathrm{m/s^2}$, mass $m = 0.15\,\mathrm{kg}$, length $l = 0.5\,\mathrm{m}$, and friction coefficient $b = 0.1$ (dimensionless). The state of the system is the vector $x = [\theta, \dot{\theta}]^\top$.

The state space is constrained to $\theta \in [-\pi, \pi)$ radians and $\dot{\theta} \in [-8, 8]$ rad/s. The control input u, representing the applied torque, is constrained to $|u| \leq 1$ Nm. For all simulations, the system dynamics are integrated using a 4th-order Runge-Kutta (RK4) integrator (Butcher, 2003).

For the DRL-based agents (TD3 and LAS-TD3), the objective is to minimize a quadratic cost function. The instantaneous cost is defined as:

$$c(x, u) = \theta^2 + 0.1\dot{\theta}^2 + 0.001u^2 \tag{3.2}$$

This cost function penalizes deviations from the upright position ($\theta = 0$), angular velocity, and control effort.

3.2 Controller Architectures for Benchmarking

To provide a comprehensive evaluation, the LDP controller is compared against four other controllers, representing classical control, standard DRL, and state-of-the-art safe RL.

Linear-Quadratic Regulator (LQR) The classical control baseline is an infinite-horizon, continuous-time LQR controller. It is designed by linearizing the pendulum dynamics (Eq. 3.1) around the equilibrium at the origin ($x = 0$). The controller computes the optimal feedback gain K for the state-feedback control law $u = -Kx$, which minimizes the undiscounted quadratic cost function $\int_0^\infty (x^\top Q x + u^\top R u)dt$. The optimal gain matrix K is found by solving the continuous-time algebraic Riccati equation (CARE). The weighting matrices are set to the identity matrices, $Q = I$, $R = I$, matching the implementation on Wang and Fazlyab (2024) and avoiding separate hyperparameter searches for both the standalone LQR and the LQR component within our dual-policy controller.

Twin Delayed Deep Deterministic Policy Gradient (TD3) The standard DRL baseline is the TD3 algorithm (Fujimoto et al., 2018), representing a state-of-the-art model-free controller with no built-in safety guarantees. In our implementation, both the actor and the critic networks are constructed with two hidden layers of 256 nodes each. The networks are trained using separate Adam optimizers (Kingma and Ba, 2015), both configured with a learning rate of 1×10^{-4}.

Lyapunov Actor-Critic (LAC) We use a Lyapunov Actor-Critic (LAC) controller, a direct replication of the physics-informed architecture proposed by Wang and Fazlyab (2024). Following the hyperparameters specified in their work, the actor network has two hidden layers with 5 neurons each, while the critic network ($W^\phi(x)$) has two hidden layers with 10 neurons each. A single Adam optimizer is used with a learning rate of 2×10^{-3}. In the Zubov function $W = \tanh(\alpha V(x))$, we set $\alpha = 0.2$ to position the argument $\alpha V(x)$ in the transition region of the hyperbolic tangent, where slope is maximal. The critic network, $W^\phi(x)$, is trained to approximate a Zubov function by minimizing a loss function in Eq. (2.24), which minimizes the residual of Zubov's PDE (Eq. 2.14). The actor network $\pi^\theta(x)$, is concurrently trained to find actions that maximally decrease the value of the learned Zubov function, thereby driving the system towards stability. The full loss function structure described in Section 2.3.2 is used.

Locally Asymptotically Stable TD3 (LAS-TD3) Finally, we implement a Locally Asymptotically Stable TD3 (LAS-TD3) controller, replicating one of the main models in Zoboli and Dibangoye (n.d). Unlike the LAC controller, LAS-TD3 operates in a discrete-time, discounted setting. Its local law is a discrete-time LQR whose gain solves the discounted algebraic Riccati equation (DARE) as detailed in Section 4.2 of Zoboli and Dibangoye (n.d). The global policy is a standard TD3 agent, using the same network architecture and hyperparameters as our standalone TD3 baseline. We compute both the final action and the critic value

via the blended policy (Eq. 2.25) and blended Q-function (Eq. 2.26), respectively.

For the dual-policy controllers (LAS-TD3 and LDP), performance is dependent on the blending hyperparameter β. For the primary comparative analysis presented in our results, we use $\beta = 0.9$ for both controllers, as further analysis (presented in Section 4.2) showed this value yielded the best overall performance.

3.3 The Lyapunov Dual-Policy (LDP) Controller

The LDP controller is our primary contribution. It integrates the dual-policy structure of Zoboli and Dibangoye (n.d) with the physics-informed learning objectives of Wang and Fazlyab (2024) in a continuous-time, undiscounted framework.

The LDP controller consists of a local and a global component. The local controller π_{loc} is a continuous-time, undiscounted LQR controller, identical to the LQR baseline. It provides a provably stable policy and a quadratic Lyapunov function, $V_{\text{loc}}(x) = x^\top P x$, where P is the solution to the CARE. The global controller (π^θ_{glob}) is a Lyapunov Actor-Critic (LAC) with identical network architectures and hyperparameters to the LAC baseline. The final blended policy π^θ follows Eq. 2.25.

A key innovation of our LDP controller is the adaptation of the blended Q-function from the LAS-TD3 framework into a blended Zubov function, $\hat{W}^\phi(x)$. This reframes the critic's role from estimating future discounted rewards to learning a verifiable stability certificate. This is an important distinction and improvement. The blended Zubov function is defined as:

$$\begin{aligned}\hat{W}^\phi(x) = W_{\text{loc}}(x) \\ + h_2(x)(W^\phi_{\text{glob}}(x) - W_{\text{loc}}(x))\end{aligned} \tag{3.3}$$

Here, $W_{\text{loc}}(x)$ is derived from the local LQR's Lyapunov function $V_{\text{loc}}(x) = x^\top P x$: specifically, $W_{\text{loc}}(x) = \tanh(\alpha\, V_{\text{loc}}(x))$, using $\alpha = 0.2$ identically to the global critic. $W^\phi(x)_{\text{glob}}$ is the output of the global critic network. This structure injects the known, provably correct local stability information directly into the learning target for the overall critic, providing a strong, physics-informed prior near the origin.

A small, yet important issue arises during training. The physics-informed loss from Wang and Fazlyab (2024) includes a component L_z (Eq. 2.24) that penalizes a non-zero value of the critic at the origin, i.e., $W^\phi(0)^2$. In our blended framework, as $x \to 0$, the blending function $h_2(x) \to 0$, causing the blended critic $\hat{W}^\phi(x)$ to be dominated by the fixed local critic $W_{\text{loc}}(x)$, which is already zero at the origin. This effectively blocks gradients from flowing back to the learnable global critic $W^\phi_{\text{glob}}(x)$ for this loss component. To solve this, for the L_z loss component only, we use the raw global critic output, $W^\phi_{\text{glob}}(x)$. All other loss components, as well as the policy rollouts during training, use the blended critic $\hat{W}^\phi(x)$ and blended policy $\pi^\theta(x)$. This ensures the global critic learns to be zero at the origin while the rest of the training benefits from the blended structure.

3.4 Training the Learning-based Lyapunov Controllers

The LAC and LDP controllers are trained using an identical training routine that combines learning with formal verification.

1. The controller is trained for 1000 steps. In each step, a batch of K initial states is sampled. The value of K is a compromise between performance and efficiency, and in this case we use $K = 8$. these samples come from R_1, a region defined by the certified DoA of the local LQR. Trajectories are simulated using the current (blended, for LDP) policy, and the physics-informed loss (Eq. 2.24) is minimized.

2. After 1000 steps, we find the largest certified sub-level set of the learned Lyapunov function, denoted by the value c^*. This is achieved by using a binary search to propose new candidate c^*, then formally verifying it using the dReal SMT solver. The solver is tasked with checking if the Lyapunov conditions hold for all states x, such that $W^\phi(x) \leq c^*$, using Eq. (2.17). In our experiments, mimicking Chang et al. (2019), we exclude a small ball of radius $\epsilon = 0.25$. If the solver returns `unsat` for the query that asserts a violation, the certificate is formally proven correct for the region $\mathcal{A}_{c^*}$

3. If a valid c^* greater than a certain numerical threshold is found (here chosen arbitrarily as $c^* > 0.3$), we estimate the area of the certified region $\mathcal{A}_{c^*} = \{x \mid W^\phi(x) \leq c^*\}$ using Monte Carlo sampling. The threshold $c^* > 0.3$ ensures the certificate is meaningful and sufficiently large, thereby excluding cases where verification typically struggles or fails, particularly in regions very close to the origin. Training stops if this area fails to increase by more than 1% for three consecutive certification loops. This prevents endless training for marginal gains.

4. When the SMT solver rejects a candidate c^*, it returns a counter-example state x_{ce} that violates the conditions. This state is added to a CEGAR buffer. In the next training phase, the batch is constructed from $K - n$

new samples and the n counter-examples from the buffer. This forces the network to correct the "artifacts" or local non-smooth regions in the learned Lyapunov landscape that cause verification to fail.

5. The entire loop (Step 1–4) is repeated. Training terminates via early stop once the controller has learnt a meaningful certificate (defined as $c^* > 0.3$) and the estimated DoA area fails to increase by more than 1% for three consecutive loops.

3.5 Evaluation Metrics

Once all the controllers are fully trained, they undergo a final evaluation to assess and compare their performance. Each controller is tested over 50 independent runs, each run consisting of 500 episodes, and each episode is simulated for a maximum of 3000 steps.

The following metrics are recorded:

- The mean cumulative cost per episode across all runs, using the cost function in Eq. (3.2). This measures the controller's ability to optimize the task objective.
- The mean percentage of the 500 episodes in a run that successfully stabilize the pendulum at the origin. An episode is considered successful if the state enters and remains within a small tolerance ($||x|| < 0.0005$) of the origin for 10 consecutive timesteps. This strict criterion is chosen to evaluate practical asymptotic stability, evaluating minor oscillations in the controller, rather than merely guaranteeing that trajectories are bounded within a stable region (boundedness).
- The mean number of steps required to achieve convergence under our definition. This is calculated for converging episodes only, and measures the controller's efficiency.
- For qualitative comparison, we plot the angle θ versus timesteps.

We assess normality with the Shapiro-Wilk test (Shapiro and Wilk, 1965), variance homogeneity with Levene's test (Levene, 1960), and all pairwise p-values were adjusted using the Holm-Bonferroni procedure (Holm, 1979).

4 Results

This section presents the empirical results of our comparative study. We begin with the statistical methodology used to analyse the data, then report quantitative performance metrics for each controller, and finally offer a qualitative discussion of the learned certificates and convergence dynamics.

To determine statistical significance, every metric collected from the 50 runs per controller followed the pipeline described in Section 3.5. Specifically, the Shapiro-Wilk test confirmed that mean reward and steps-to-converge were approximately normal ($p_{\text{reward}} = 0.21$, $p_{\text{steps}} = 0.34$), while Levene's test detected unequal variances for both metrics ($p < .001$ in each case). Consequently, we used Welch's two-sample t-test for all pairwise comparisons. Resulting p-values were adjusted with the Holm-Bonferroni procedure to control the family-wise error rate at $\alpha = 0.05$.

4.1 Quantitative Performance

The aggregated performance of the five primary controllers, using the best performing hyperparameter $\beta = 0.9$ for the dual-policy methods, is summarized in Table 4.1. The results demonstrate a clear hierarchy of performance and reliability.

In terms of mean cost, the LDP controller ($M = 738.15$, $SD = 36.30$) and the LAC controller ($M = 757.93$, $SD = 25.85$) were not statistically different from each other ($p_{\text{holm}} = .593$), but both were significantly higher than all other controllers (all $p_{\text{holm}} < .001$). The next tier consists of the classical LQR controller ($M = 809.53$, $SD = 28.85$), which achieved a significantly higher reward than both the LAS-TD3 ($M = 871.67$, $SD = 35.51$), $p_{\text{holm}} < .001$, and the standard TD3. The LAS-TD3, in turn, significantly outperformed the standard TD3 ($M = 997.86$, $SD = 42.73$), $p_{\text{holm}} < .001$, which was the worst-performing controller.

The LQR, LAS-TD3, and LDP controllers achieved a 100% convergence rate, whereas the TD3 and the pure LAC controllers failed to converge in any episode across all runs (under our definition of convergence), yielding a 0% convergence rate. Among the stable controllers, the LDP was the fastest, converging in significantly fewer steps ($M = 2114.1$) than both the LAS-TD3 ($M = 2238.5$, $p_{\text{holm}} < .001$) and the LQR ($M = 2331.12$, $p_{\text{holm}} < .001$). The LAS-TD3 was also significantly faster than the LQR ($p_{\text{holm}} < .001$).

The learning-based Lyapunov methods, LAC and LDP, successfully learned and verified large DoAs, certified by $c^* = 0.96$ and $c^* = 0.95$ respectively. These values, corresponding to the 1-level set of a bounded Zubov function, represent significantly larger certified regions than the quadratic certificate of the LQR ($c^* = 1.15$ for an unbounded Lyapunov function), as visualized in Figure 4.4.

Controller	Mean Cost (↓) (± SD)	Convergence Rate (%) (↑)	Mean Steps to Converge (↓) (± SD)
LQR	809.53 ± 28.85	**100%**	2331.12 ± 12.6
TD3	997.86 ± 42.73	0%	N/A
LAS-TD3 ($\beta = 0.9$)	871.67 ± 35.51	**100%**	2238.5 ± 11.1
LAC	**757.93 ± 25.85**	0%	N/A
LDP ($\beta = 0.9$)	**738.15 ± 36.30**	**100%**	**2114.1 ± 12.1**

Table 4.1: Aggregated Results over 50 Runs. Arrows indicate the desired direction for optimization (e.g., ↓ for lower is better). Bold values indicate the best performing controller(s) for each metric, based on statistical significance.

4.2 Sensitivity to Blending Factor β

The performance of the LDP and LAS-TD3 controllers was evaluated across a range of values for the blending hyperparameter β, which controls the influence of the global policy at the boundary of the local controller's DoA. The results are shown in Table 4.2.

The LDP controller achieves a 100% convergence rate across the entire tested range of β values. In contrast, the LAS-TD3 controller's convergence rate drops down to 61% for low values of β, only becoming reliable when $\beta \geq 0.4$. A low value of β implies a sharper transition from the local to the global policy, placing more stress on the global policy to perform well immediately outside the local region. For both controllers, performance generally improved with higher values of β, with the optimal performance for both reward and convergence speed occurring at $\beta = 0.9$. A notable observation is that even at $\beta = 0.9$, where the local controller influences the blended policy the least, the LDP achieves a 100% convergence rate, whereas the LAC controller, which conceptually is $\beta = 1.0$, achieves 0% convergence rate under our definition of convergence. Therefore, any influence from the local controller is enough to significantly improve the policy.

4.3 Qualitative Results

The training process for the LQR, LAC, and LDP controllers produces a verifiable stability certificate. Figure 4.1, Figure 4.2, and Figure 4.3 visualize these certificates in the state space. The LQR's Lyapunov function is a quadratic Lyapunov function, while the LAC and LDP controllers learn complex Zubov functions.

The LQR's quadratic certificate $V(x)$ only satisfies Lyapunov's decrease very close to the origin, resulting in a small ellipsoidal region. In contrast, both the LAC's $W^{\phi}(x)$ and the LDP's blended $\hat{W}^{\phi}(x)$ extend nearly to the boundary of the state space, as their estimated c^* values are very close to 1.0. Visually, the LAC and LDP surfaces look almost identical at larger radii since they share the same global actor-critic; the main difference is slightly smoother, more rounded contours near the origin in the LDP plot, as the local component dominates the Zubov function in that region. This is difficult to discern in the 3D plot, but more apparent in the contour lines within the same figure.

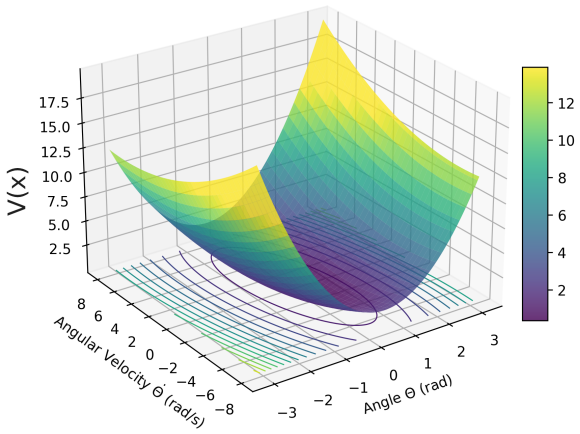

Figure 4.1: 3D surface of the quadratic Lyapunov function $V(x) = x^{\top}Px$ for the LQR controller over the pendulum's state space. Only in a small neighborhood around the origin does V satisfy the decrease condition, yielding a tight, ellipsoidal certified region.

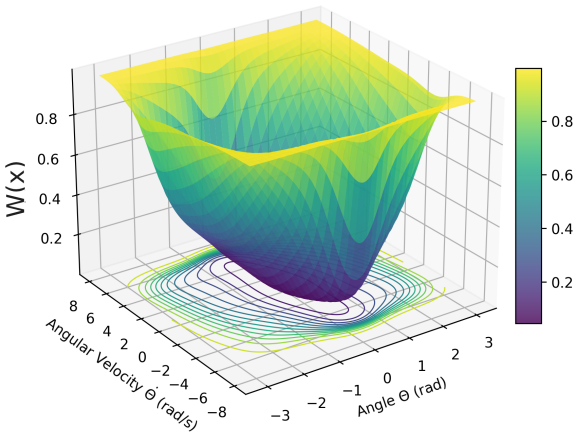

Figure 4.2: Learned Zubov function $W^{\phi}(x)$ from the LAC controller, plotted over $(\theta, \dot{\theta})$. The certificate extends smoothly from the origin to the top of the state space, reflecting a nearly global stability guarantee (with $c^* \approx 0.96$).

β	LDP Mean Steps (↓) (± SD)	LDP Conv. Rate (%) (↑)	LAS-TD3 Mean Steps (↓) (± SD)	LAS-TD3 Conv. Rate (%) (↑)
0.1	2319.6 (± 10.6)	**100%**	2333.2 (± 22.8)	61%
0.2	2301.7 (± 16.6)	**100%**	2349.8 (± 25.1)	70%
0.3	2284.3 (± 10.9)	**100%**	2439.1 (± 17.3)	70%
0.4	2261.5 (± 10.7)	**100%**	2506.8 (± 16.1)	**100%**
0.5	2346.5 (± 14.7)	**100%**	2426.4 (± 14.4)	**100%**
0.6	2215.8 (± 11.9)	**100%**	2353.5 (± 12.7)	**100%**
0.7	2422.4 (± 13.2)	**100%**	2429.0 (± 12.5)	**100%**
0.8	2172.5 (± 9.7)	**100%**	2314.3 (± 12.8)	**100%**
0.9	**2114.1 (± 12.1)**	**100%**	**2238.5 (± 11.1)**	**100%**

Table 4.2: Sensitivity to the blending factor β for dual-policy controllers over 50 runs. Arrows indicate the desired direction for optimization. Bold values indicate the best performing hyperparameter for each controller.

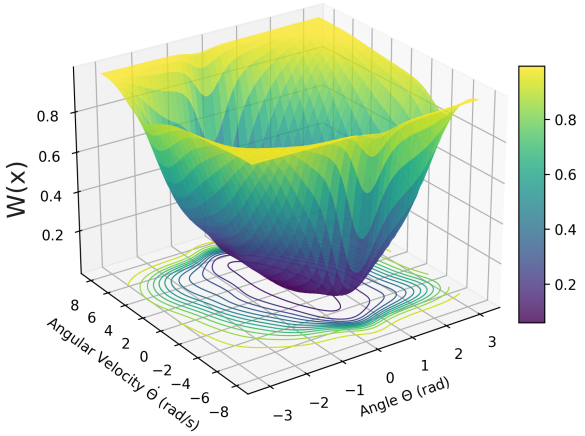

Figure 4.3: Blended Zubov-Lyapunov function $W^{\phi}(x)$ from the LDP controller. Visually almost identical to the pure LAC $W^{\phi}(x)$, as they share the same global controller, but transitions to the local quadratic form near the origin, yielding a slightly smoother contour plot.

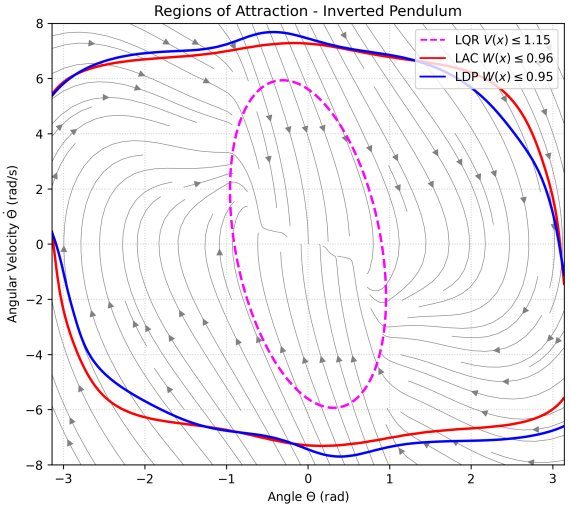

Figure 4.4: 2D projection of the verified domains of attraction in the pendulum state plane for LQR, LAC, and LDP. Both learning-based controllers are represented by very similar boundaries, far larger than the conservative LQR region.

Figure 4.4 provides a 2D projection of the final, formally verified DoAs for the LQR, LAC, and LDP controllers. Both the LAC and LDP controllers learn complex, non-ellipsoidal regions that are visibly larger than the region certified for the LQR, demonstrating their ability to capture the nonlinear stability boundaries of the system.

Finally, the evolution of the system state over time, aggregated over 500 new episodes and plotted in Figure 4.5 and Figure 4.6, provides a direct comparison of the controllers' convergence dynamics.

The angle evolution plot, Figure 4.5, highlights significant differences in performance. The standard TD3 controller fails to converge to the origin, instead stabilizing at a fixed, non-zero angle of $\theta \approx -0.21$ radians. The LAS-TD3 controller exhibits a large initial overshoot, passing the $\theta = 0$ setpoint before slowly converging back to it. In contrast, the LQR and pure LAC controllers show a much slower, more gradual convergence toward the origin without overshooting. The LDP controller demonstrates the most effective response, driving the angle to zero almost immediately and holding it there with no visible overshoot or subsequent oscillation.

The angular velocity plot, Figure 4.6, further clarifies these behaviors. Both the TD3 and LAS-TD3 controllers induce a massive initial overshoot in angular velocity. The other controllers also exhibit an initial velocity overshoot, but the LDP controller's is minimal and barely perceptible. After this initial overshoot, the LDP controller's performance is again notable: once the angular velocity reaches zero, it remains there with no visible oscillations. The plot for the TD3 controller confirms that it successfully drives the angular velocity to zero, which corresponds to it settling into the fixed, non-zero angular position observed in the previous figure.

5 Conclusion

This thesis addresses the gap between the high performance of DRL and the formal stability guarantees required for its safe deployment in critical applications. We introduced the Lyapunov Dual-

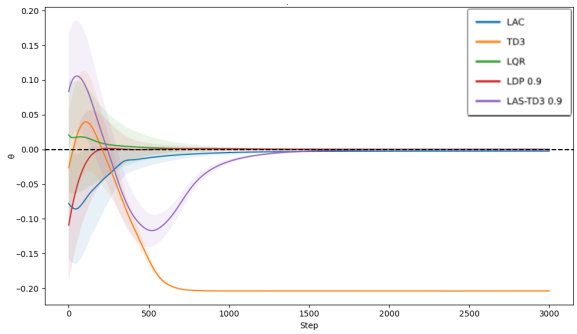

Figure 4.5: The evolution of the pendulum angle θ ($\pm$ SD) versus timesteps for all controllers. The LDP converges rapidly without overshoot. The LAS-TD3 overshoots significantly, while LQR and LAC converge slowly. The TD3 stabilizes at a non-zero angle.

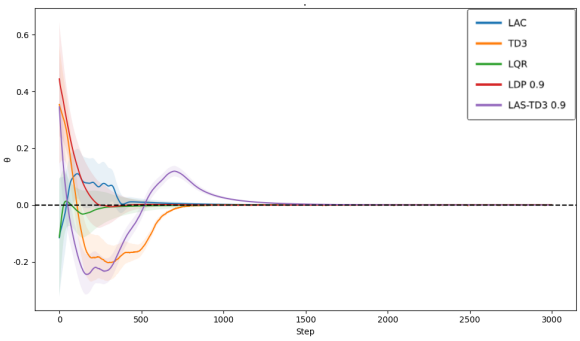

Figure 4.6: The evolution of the pendulum angular velocity $\dot{\theta}$ ($\pm$ SD) versus timesteps. Both TD3 and LAS-TD3 show a large initial overshoot. The LDP controller exhibits a minimal, barely noticeable overshoot and maintains zero velocity with no visible oscillations upon convergence.

Policy (LDP) controller, a novel approach combining a dual-policy structure for guaranteed local asymptotic stability and for injecting information during learning, a physics-informed learning objective for maximizing the formally certifiable Domain of Attraction (DoA), and a Counter-Example Guided Abstraction Refinement (CEGAR) loop for stable training.

Our empirical evaluation on the inverted pendulum demonstrated that the LDP controller outperforms baseline methods under all evaluated metrics. Specifically, the LDP achieves the best cost and convergence speed, maintains a 100% convergence rate, and yields a large formally verified DoA. In comparison, the TD3 baseline achieved a higher cost, lower convergence speed, and failed our convergence criteria. The agent failed by finding a suboptimal local minimum of the cost function (Eq. (3.2)), stabilizing at a fixed, non-zero angle while minimizing angular velocity. This behavior highlights a classic failure mode for model-free agents that lack physical insight into the true objective. The LAC controller, while obtaining a large certified DoA, lacked practical convergence. Finally, the LQR and LAS-TD3 controllers, despite guaranteeing stability, notably obtained a higher cost and slower convergence.

A key advantage of the LDP framework, inherited from the dual-policy structure proposed by Zoboli and Dibangoye (n.d), is that it provides formal local stability guarantees from the start of training. Because the local controller is a precomputed LQR and the blending function ensures dominance of the local policy near the equilibrium, the blended system's equilibrium point remains provably locally asymptotically stable at all times. This ensures that any trajectory sufficiently close to the origin is captured by the local controller's stabilizing influence, guaranteeing convergence.

These stability guarantees mark a significant improvement over the pure LAC approach by Wang and Fazlyab (2024), whose stability is only certified post-training. The LDP framework thus ensures continuous local stability throughout the training process, in addition to providing terminal stability guarantees upon completion. Furthermore, the LDP controller addresses the limitations of its constituent approaches, yielding not only improved stability but also superior overall performance.

Although the physics-informed approach proposed by Wang and Fazlyab (2024) can learn large DoAs, our results show that this approach alone fails to produce practically stable policies. Specifically, the pure LAC policy demonstrates bounded trajectories within the stable region but fails to completely eliminate minor oscillations around the equilibrium, thereby failing our strict convergence criteria. In contrast, the LDP controller capitalizes on the physics-informed objective to achieve a similarly large certified region, but uses the dual-policy structure to achieve reliable convergence.

Furthermore, the LDP significantly outperforms the LAS-TD3 approached proposed by Zoboli and Dibangoye (n.d); Zoboli et al. (2021). The LAS-TD3's tendency to overshoot the setpoint, as seen in our results, is a characteristic behavior of underdamped second-order systems (Oliveira and Vrančić, 2012) and might indicate a lack of precision in its learned global policy. We attribute the LDP's superior performance to two main design choices that address this. First, the continuous-time, undiscounted formulation more accurately represents the physical system's dynamics, avoiding the approximation errors inherent to the discrete-time, discounted model used by LAS-TD3. Second, the introduction of our blended Zubov critic provides the learning agent with a direct and explicit objective of maximizing the verifiable stable region. This represents a more powerful control

objective than the blended Q-function's indirect objective of accumulating future rewards. Consequently, the LDP achieves a solution that combines stability guarantees with better practical performance, clearly surpassing the capabilities of either its physics-informed or dual-policy predecessors alone.

5.1 Limitations

Despite the promising results, this work has several limitations that frame the scope of our claims and motivate future research.

First, the LDP controller's formal guarantees critically depend on the fidelity of the analytical model used during training and verification. Any discrepancies between the modeled and real-world system dynamics, such as unmodeled friction, parameter drift, or actuator delay, can compromise these guarantees and, at worst, lead to instability. To bridge this simulation-to-reality gap, future implementations must incorporate measures such as conservative uncertainty margins or online system identification. Until then, stability assurances should be viewed as conditional upon accurate modeling rather than true guarantees of real-world performance.

Second, while the DoA is explicitly maximized, it remains a finite, local region of the state space. The controller's behavior outside this verified region is not guaranteed. A sufficiently large real-world disturbance could push the system beyond the certificate's boundary, potentially leading to instability. The method provides no guarantee of recovering from all possible initial states.

Third, the training and verification process is computationally intensive, presenting a fundamental barrier to scalability. This is highlighted by the underlying physics-informed loss, which relies on solving Zubov's PDE. While feasible in low dimensions, evaluating the PDE's residual becomes computationally intractable in high-dimensional state spaces. Additionally, the larger or higher-dimensional the state-space, the more expensive will the SMT solver check be. This "curse of dimensionality" currently restricts the practical application of this specific physics-informed approach to systems with relatively few degrees of freedom (Bellman, 1984).

5.2 Future Work

This research opens several promising avenues for future investigation, each addressing a limitation of the current work.

A primary direction for future work is to test the scalability and generality of the LDP framework. This involves applying the controller to higher-dimensional and more complex benchmark problems, such as robotic arms, or simplified autonomous vehicle lane-keeping tasks. Such experiments would help determine the practical limits of the current approach and highlight the engineering and algorithmic challenges that arise in more complex state and action spaces, particularly concerning the computational cost of SMT-based verification.

A second research direction is the development of a fully data-driven LDP controller. As suggested by Zoboli and Dibangoye (n.d), a key step would be to replace the model-based local LQR with a component that learns the local system dynamics and forms a local stabilizing controller online from data. This could involve techniques like Koopman operator theory for linearizing dynamics from trajectories or using Gaussian Processes to learn a model with quantifiable uncertainty. A truly model-free LDP would drastically expand its applicability to systems where no first-principles model is known.

Finally, a dedicated study should be conducted to quantify the contribution of the CEGAR loop to learning stability. Such an investigation could compare training dynamics, final DoA size, and overall controller performance of the LDP framework with and without the CEGAR loop, thereby providing rigorous evidence for its stabilizing effect. Likewise, evaluating the blended Zubov critic against alternative formulations would help isolate its specific contribution. These analyses would yield deeper insights into why the LDP framework is successful and guide the design of future safe RL architectures.

6 Broader Impact Statements

The Lyapunov Dual-Policy (LDP) controller presents a significant improvement in safety and reliability of deploying deep reinforcement learning (DRL) in safety-critical settings by providing formally verifiable stability guarantees both during training and inference, as well as a clearly defined Domain of Attraction (DoA). Moreover, it consistently outperforms classical and learning-based baselines in control performance and convergence speed. However, the method requires an accurate analytical model of the system dynamics, presenting practical challenges when translating results to real-world environments with modeling uncertainties or environmental shifts. Additionally, while training is computationally intensive and its complexity rapidly scales with system dimensionality, inference is notably resource-efficient, owing to the small network sizes relative to standard DRL methods. Thus, for stable environments without frequent need for re-certification, the LDP controller provides an effective, sustainable, and resource-

efficient solution for embedded deployment.

References

Richard Bellman. *Dynamic programming*. Princeton University Press, 1984. ISBN 978-0-691-07951-6. OCLC: 476091680.

Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022. ISSN 2573-5144. doi:10.1146/annurev-control-042920-020211. URL https://www.annualreviews.org/doi/10.1146/annurev-control-042920-020211.

Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2 edition, May 2022. ISBN 978-1-009-08951-7 978-1-009-09848-9. doi:10.1017/9781009089517. URL https://www.cambridge.org/highereducation/product/9781009089517/book.

J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 1 edition, 2003. ISBN 978-0-471-96758-3 978-0-470-86827-0. doi:10.1002/0470868279. URL https://onlinelibrary.wiley.com/doi/book/10.1002/0470868279.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural Lyapunov control. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2647c1dba23bc0e0f9cdf75339e120d2-Paper.pdf.

Emilia Fridman. Tutorial on Lyapunov-based methods for time-delay systems. *European Journal of Control*, 20(6):271–283, November 2014. ISSN 09473580. doi:10.1016/j.ejcon.2014.10.001. URL https://linkinghub.elsevier.com/retrieve/pii/S0947358014000764.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Sicun Gao, Soonho Kong, and Edmund M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In Maria Paola Bonacina, editor, *Automated Deduction – CADE-24*, volume 7898, pages 208–214. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-38573-5 978-3-642-38574-2. doi:10.1007/978-3-642-38574-2_14. URL http://link.springer.com/10.1007/978-3-642-38574-2_14. Series Title: Lecture Notes in Computer Science.

Dongkun Han, Ahmed El-Guindy, and Matthias Althoff. Estimating the domain of attraction based on the invariance principle. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 5569–5576. IEEE, 2016. ISBN 978-1-5090-1837-6. doi:10.1109/CDC.2016.7799125. URL https://ieeexplore.ieee.org/document/7799125/.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. ISSN 0303-6898. URL https://www.jstor.org/stable/4615733. Publisher: [Board of the Foundation of the Scandinavian Journal of Statistics, Wiley].

Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023. ISSN 1476-4687. doi:10.1038/s41586-023-06419-4. URL https://doi.org/10.1038/s41586-023-06419-4.

H.K. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002. ISBN 978-0-13-067389-3. URL https://books.google.nl/books?id=t_d1QgAACAAJ.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, pages 1–13. OpenReview.net, 2015.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Howard Levene. Robust tests for equality of variances. In *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, pages 278–292. Stanford University Press, 1960.

P. B. Moura Oliveira and Damir Vrančić. Underdamped second-order systems overshoot control. *IFAC Proceedings Volumes*, 45(3):518–523,

2012. ISSN 1474-6670. doi:10.3182/20120328-3-it-3014.00088. URL https://linkinghub.elsevier.com/retrieve/pii/S1474667016310783. Publisher: Elsevier BV.

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3):591, 1965. ISSN 00063444. doi:10.2307/2333709. URL https://www.jstor.org/stable/2333709?origin=crossref.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018. ISSN 0036-8075, 1095-9203. doi:10.1126/science.aar6404. URL https://www.science.org/doi/10.1126/science.aar6404.

A. Vannelli and M. Vidyasagar. Maximal lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica*, 21(1): 69–80, 1985. ISSN 00051098. doi:10.1016/0005-1098(85)90099-8. URL https://linkinghub.elsevier.com/retrieve/pii/0005109885900998.

M. Vidyasagar. *Nonlinear Systems Analysis*. Society for Industrial and Applied Mathematics, second edition, 2002. ISBN 978-0-89871-526-2 978-0-89871-918-5. doi:10.1137/1.9780898719185. URL https://epubs.siam.org/doi/book/10.1137/1.9780898719185.

Jiarui Wang and Mahyar Fazlyab. Actor–critic physics-informed neural Lyapunov control. *IEEE Control Systems Letters*, 8:1751–1756, 2024. ISSN 2475-1456. doi:10.1109/lcsys.2024.3416235. URL https://dx.doi.org/10.1109/LCSYS.2024.3416235.

Rory Young and Nicolas Pugeault. Enhancing robustness in deep reinforcement learning: A Lyapunov exponent approach. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 86102–86123. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9c4bbdad95f6ffed1a15c06b491e0a3e-Paper-Conference.pdf.

Samuele Zoboli and Jilles S. Dibangoye. Asymptotically stable deep actor-critic algorithms for quadratic cost setpoint optimal control problems, n.d. Manuscript.

Samuele Zoboli, Vincent Andrieu, Daniele Astolfi, Giacomo Casadei, Jilles S. Dibangoye, and Madiha Nadri. Reinforcement learning policies with local LQR guarantees for nonlinear discrete-time systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2258–2263. IEEE, 2021. ISBN 978-1-6654-3659-5. doi:10.1109/CDC45484.2021.9683721. URL https://ieeexplore.ieee.org/document/9683721/.

Zubov. *Methods of A. M. Lyapunov and Their Application*. Prentice-Hall, 1964. Originally published in Russian in 1961.