Joint Differential Optimization and Verification for Certified Reinforcement Learning

Yixuan Wang*
Northwestern University
Evanston, IL, USA
wangyixu14@gmail.com

Simon Zhan*
UC Berkeley
Berkeley, CA, USA
simonzhan@berkeley.edu

Zhilu Wang
Northwestern University
Evanston, IL, USA

Chao Huang
University of Liverpool
Liverpool, UK

Zhaoran Wang
Northwestern University
Evanston, IL, USA

Zhuoran Yang
Yale University
New Haven, CT, USA

Qi Zhu
Northwestern University
Evanston, IL, USA

ABSTRACT

Model-based reinforcement learning has been widely studied for controller synthesis in cyber-physical systems (CPSs). In particular, for safety-critical CPSs, it is important to formally certify system properties (e.g., safety, stability) under the learned RL controller. However, as existing methods typically conduct formal verification *after* the controller has been learned, it is often difficult to obtain any certificate, even after many iterations between learning and verification. To address this challenge, we propose a framework that *jointly conducts reinforcement learning and formal verification* by formulating and solving a novel bilevel optimization problem, which is end-to-end differentiable by the gradients from the value function and certificates formulated by linear programs and semi-definite programs. In experiments, our framework is compared with a baseline model-based stochastic value gradient (SVG) method and its extension to solve constrained Markov Decision Processes (CMDPs) for safety. The results demonstrate the significant advantages of our framework in finding feasible controllers with certificates, i.e., Barrier functions and Lyapunov functions that formally ensure system safety and stability, available on Github.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Computer systems organization** → **Embedded and cyber-physical systems**.

KEYWORDS

RL, Safety, Barrier function, Stability, Lyapunov function.

1 INTRODUCTION

Applying machine learning techniques in cyber-physical systems (CPSs) has attracted much attention. In particular, reinforcement learning (RL) has shown great promise [46], such as in robotics [21] and smart buildings [50, 52], where RL trains control policy by maximizing the value function of the goal state [42]. However, there is still significant hesitation in applying RL to safety-critical applications [20, 56] of CPSs, such as in autonomous vehicles [26, 27, 49], because of the uncertain and potentially dangerous impact on system safety [47, 54, 55]. It is thus important to find RL-learned controllers that are certified, i.e., under which critical system properties such as safety and stability can be formally guaranteed. And a common approach to guarantee these properties is to find corresponding *certificates* for them, e.g., a barrier function for safety [34] and a Lyapunov function for stability [29].

In this work, we focus particularly on learning certified controllers with RL for CPSs that can be modeled as ordinary differential equations (ODEs) with unknown parameters, a common scenario in practice. Traditionally, this is typically done in a two-step *'open-loop'* process: 1) first, model-based reinforcement learning (MBRL) is conducted to learn the system model parameters and the controller simultaneously, and then 2) based on the identified dynamics, formal verification is performed to find certificates for various system properties by solving optimization problems. However, with such an open-loop paradigm, it is often difficult to find any feasible certificates even after many iterations of learning and verification steps, and the failed verification results are not leveraged sufficiently in the learning of a new controller. Thus, integrating controller synthesis and certificate generation in a more holistic manner has received increasing attention recently.

Pioneering works on control-certificate joint learning mainly focus on systems with known models, i.e., explicit models without any unknown parameters [5, 9, 24, 28, 36, 37, 44, 45]. Those methods typically collect samples from the system space, transform the certificate conditions into loss functions, and solve them via supervised learning methods. However, they cannot be directly used to address systems with unknown parameters, and the certificates obtained in those works are often tested/validated via sampling-based approaches without being formally verified.

Moreover, for safety properties, methods that are based on solving constrained Markov Decision Processes (CMDPs) are popular in the safe RL literature [40, 41, 43]. However, these methods typically try to achieve safety by restricting the expectation of the cumulative cost for the system's unsafe actions to be under a certain threshold, which can only be regarded as *soft* safety constraints as the system may still enter the unsafe region.

*Both authors contributed equally to this research.

ICCPS '23, May 9–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0036-1/23/05…$15.00
https://doi.org/10.1145/3576841.3585919

Contribution of our work: To address these challenges, we propose a **certified RL method with joint differentiable optimization and verification for systems with unknown model parameters**. As shown in Fig. 1, our approach seamlessly integrates RL optimization and formal verification by formulating and solving a *novel bilevel optimization problem*, which generates an optimal controller together with its certificates, e.g., barrier functions for safety and/or Lyapunov functions for stability. Different from CMDP-based methods, we address *hard* safety constraints where the system should never enter an unsafe region.

The upper-level problem in our bilevel optimization tries to learn the controller parameters θ and the unknown system model parameters α by MBRL; while the lower-level problem tries to verify system properties by searching for feasible certificates via SDP or LP with a slack variable c. Note that we propose LP relaxation in addition to SDP because while SDP may be more efficient for low-dimensional/low-degree systems, LP provides better scalability for higher-dimensional systems. When the lower-level problem fails to find any feasible certificate for given θ and α from the upper-level MBRL problem, it will return the gradient of the slack variable c over the control parameter θ, to guide the exploration of new θ. Our framework is *end-to-end fully differentiable* by the gradients from the value function in upper MBRL and from the certificates in lower SDP and LP and can be viewed as a *'closed-loop* learning-verification paradigm where the failed verification provides immediate gradient feedback to the controller synthesis. We conducted experiments on linear and non-linear systems with linear and non-linear controller synthesis, demonstrating significant advantages of our approach over model-based stochastic-value gradient (SVG) and its extension to solve CMDPs for safety. Our approach can find certificates for safety and stability in most cases based on identified dynamics and provide better results of those two properties in simulations.

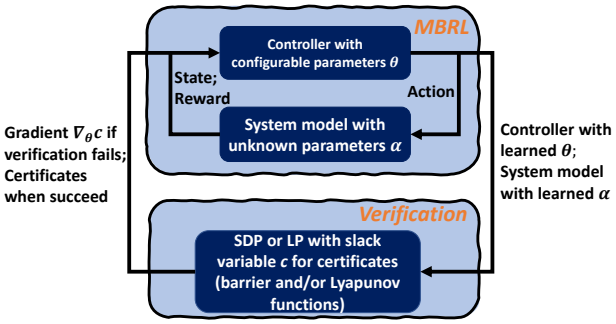

Figure 1: Overview of our joint differentiable optimization and verification framework for certified RL. Our approach integrates RL-based control optimization with formal verification in a closed-loop manner, by formulating and solving a bilevel optimization problem. The upper-level problem learns the controller parameters θ and the system model parameters α with MBRL, while the lower-level problem verifies system properties by searching for feasible certificates via SDP or LP with a slack variable c. The framework is end-to-end differentiable.

In the rest of the paper, Section 2 discusses related works. Section 3 formulates the problem to address. Section 4 presents our proposed approach, and Section 5 shows the experimental results. Section 6 concludes the paper.

2 RELATED WORKS

Certificate-based verification in our work is related to the literature on barrier function safety [34] and Lyapunov stability [29], which provide formal guarantees on the safe control of systems to avoid unsafe states and on the system stability around an equilibrium point, respectively. In classical control, finding barrier or Lyapunov functions is challenging [32] and often requires considerable expertise and manual effort [7] through optimization. Our approach, in contrast, automatically searches for certificates and provides gradient feedback from the failed searches to the learning process, to guide the exploration of control parameters for increasing the chance of finding feasible certificates.

Regarding safety in particular, our work aims at addressing *hard* safety constraints, by ensuring that the system never enters an unsafe region with formal guarantees [10, 15, 16], both during and after training. In contrast, in the popular CMDP-based methods [40, 41, 43], the agent aims to maximize the expected cumulative reward while restricting the expectation of cumulative cost for their unsafe interactions with the environment under a certain threshold. Since the agent can still take unsafe actions with some cost, the safety constraints in CMDPs can be regarded as addressing *soft* safety constraints without formal guarantees. There are also works that ensure safety by addressing stability [3] in RL, but we consider these two properties as different in this work, where safety is defined based on the reachability of the system state.

In terms of optimization techniques, our work leverages SVG [14] in MBRL and is a first-order, end-to-end differentiable approach with the computation of the analytic gradient of the RL value function. Our work also conducts convex optimization for the certification. As such an optimization problem may not be feasible to solve, our approach tries to 'repair' it via a slack variable, which is differentiable to control parameters. This is related to but different from the approach in [2], which tries to repair the infeasible problems by modifying program parameters. Finally, as a differentiable framework, our approach is related to safe PDP [17], which, different from ours, requires an explicit dynamical model with no unknown parameters and an initial safe policy.

Our work is also related to the joint learning of controller and verification by leveraging neural networks to represent certificates [5, 9, 18, 24, 28, 36–38, 44, 48]. These approaches first translate the certificate conditions into loss functions, sample and label data points from the system state, and then learn the certificate in a supervised learning manner. However, they require a known system dynamics model or safe demonstration data, and cannot be directly applied to systems models with unknown parameters or without safe data, which is the case our approach addresses via the RL process. Moreover, the neural network-based certificates generated by these approaches are often tested/validated through sampling-based methods and are not formally verified, while our approach provides formal and deterministic guarantees once the certificate is successfully obtained.

3 PROBLEM FORMULATION

We consider a continuous CPS whose dynamics can be expressed as an ordinary differential equation (ODE):

$$\dot{x} = f(x, u; \alpha), \tag{1}$$

where $x \in X \subset \mathbf{R}^n$ is a vector denoting the system state within the state space X and $u \in U \subset \mathbf{R}^m$ is the control input variable. $f : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}^n$ is a locally Lipschitz-continuous function ensuring that there exists a unique solution for the system ODE. Without loss of generality, f is a polynomial function, as common elementary functions, such as $\sin(x), \cos(x), \sqrt{x}, \frac{1}{x}, e^x, \ln(x)$ and their combinations, can be equivalently transformed to polynomials [25]. $\alpha \in [\underline{\alpha}, \bar{\alpha}] \subset \mathbf{R}^{|\alpha|} \in$ is a vector denoting the unknown system model parameters, which are within a lower bound $\underline{\alpha}$ and an upper bound $\bar{\alpha}$. The system has an initial state set $X_0 \subset X$, an unsafe state set $X_u \subset X$, and a goal state set $X_g \subset X$. Without loss of generality, X_g is assumed as the origin point in this paper. These sets are semi-algebraic, which can be expressed as: $X_0 = \{x | \xi_i(x) \geq 0, i = 1, \cdots, s\}, X_u = \{x | \zeta_i(x) \geq 0, i = s+1, \cdots, s+q\}, X = \{x | \psi_i(x) \geq 0, i = s+q+1, \cdots, s+q+r\}$. The partial *derivatives* f_x and f_u can be computed with the parameters α. We abbreviate partial differentiation or gradient using subscripts, e.g, $\frac{\partial f(x,u;\alpha)}{\partial x} \triangleq f_x$ with *gradient* or *derivative* in front.

Such a continuous system can be controlled by a feedback deterministic controller $\pi(x; \theta) = \theta \cdot w^\theta(x) : X \to U$, which is parameterized by vector $\theta \in \mathbb{R}^{|\theta|}$ and monomial basis $w^\theta(x)$. Note that $w^\theta(x)$ can contain non-linear terms. Given any time $\forall t \geq 0$, the controller π reads the system state $x(t)$ at t, and computes the control input as $u = \pi(x(t); \theta)$. Overall, the system evolves by following $\dot{x} = f(x, \pi(x; \theta))$ with π.

A *flow* function $\varphi(x(0), t) : X_0 \times \mathbb{R}_+ \to X$ maps any initial state $x(0)$ to the system state $\varphi(x(0), t)$ at time $t, \forall t \geq 0$. Mathematically, φ satisfies 1) $\varphi(x(0), 0) = x(0)$, and 2) φ is the solution of the $\dot{x} = f(x, u)$. Thus, the system safety and stability properties and their corresponding certificates are defined as follows.

Definition 3.1. ***(Infinite-time Safety Property)*** *Starting from any initial state $x(0) \in X_0$, the system defined in* (1) *is considered as meeting the safety property if and only if its flow never enters into the unsafe set X_u: $\forall t \geq 0,\ \varphi(x(0), t) \notin X_u$.*

This safety property can be formally guaranteed if the controller π can obtain a barrier function as:

Definition 3.2. ***(Exponential Condition based Barrier Function [22])*** *Given a controller $\pi(x; \theta)$, $B(x; \beta^B)$ is a safety barrier function parameterized by vector $\beta^B \in \mathbb{R}^{|\beta|}$ with $\lambda \in \mathbb{R}$ if:*

$$B(x; \beta^B) \leq 0, \forall x \in X_0;\ B(x; \beta^B) > 0, \forall x \in X_u,$$
$$\frac{\partial B}{\partial x} \cdot f(x, \pi(x); \alpha) - \lambda B(x; \beta^B) \leq 0, \forall x \in X.$$

Remark 3.3. ***(Shielding-based One-Step Safety)*** *Another way to ensure safety is to check during run-time a pre-defined shield for the system and stop the system when finding a hazard affront. Note that such a shielding mechanism is reactional. It tries to protect the system from danger by only looking one step forward, which may degrade the overall performance, as shown in the experiments. Moreover, the system could still be led toward the unsafe region after several steps when taking the current action and has to be stopped eventually. In contrast, a barrier function guarantees infinite-time safety.*

Definition 3.4. ***(Stability Property)*** *Starting from any initial state $x(0) \in X_0$, the system defined in* (1) *is stable around the goal set X_g if there exists a $\mathcal{KL}$ function τ [19] such that for any $x(0) \in X_0$, $\|\varphi(x(0), t)\|_{X_g} \leq \tau(\|x(0)\|_{X_g}, t)$, where $\|x\|_{X_g} = \inf_{x_g \in X_g} \|x - x_g\|$, with $\|\cdot\|$ denoting the Euclidean distance.*

This stability property can be formally guaranteed if there exists a Lyapunov function for π as:

Definition 3.5. ***(Lyapunov Function)*** *$V(x; \beta^V)(\beta^V \in \mathbb{R}^{|\beta|})$ is a Lyapunov function of controller $\pi(x; \theta)$ if:*

$$V(x; \beta^V) \geq 0, x \in X;\ \frac{\partial V}{\partial x} \cdot f(x, \pi(x; \theta); \alpha) \leq 0, x \in X.$$

Considering the safety and stability certificates, the problem in this paper can be defined as a certified control learning problem:

Problem 3.6. ***(Certified Control Learning)*** *Given a continuous system defined as in* (1)*, learn the unknown dynamical parameters α and a feedback controller $\pi(x; \theta)$ so that the system formally satisfies the safety property and/or the stability property with barrier function $B(x; \beta^B)$ and/or Lyapunov function $V(x; \beta^V)$ as certificates.*

4 OUR APPROACH

In this section, we present our *certified differentiable reinforcement learning framework* to solve the Problem 3.6 defined above. We first introduce a novel bilevel optimization formulation for Problem 3.6 in Section 4.1, by treating the learning of the controller and the system model parameters as an upper-level MBRL problem and formulating the verification as a lower-level SDP or LP problem. We connect these two sub-problems with a slack variable based on certification results. We then solve the bilevel optimization problem with the Algorithm 1 introduced in Section 4.2, which leverages the gradients of the slack variable and the value function in RL, and includes techniques for variable transformation, safety shielding, and parameter identification. Finally, we conduct theoretical analysis on the soundness, incompleteness, and optimality in Section 4.3.

4.1 Bilevel Optimization Problem Formulation

In this section, we introduce a novel and general bilevel optimization problem for the certified control learning framework and then its specific extension to SDP and LP.

In general, we can formulate a constrained optimization problem for the certified control learning defined in Problem 3.6 as:

$$\begin{aligned} &\max_{\theta, \alpha} \mathbb{E}_{f, x(0) \in X_0}[\mathcal{V}(x(0))], \\ &\text{s.t. } \mathcal{I}^i(x; \theta, \alpha, \beta) \geq 0;\ \mathcal{E}^j(x; \theta, \alpha, \beta) = 0. \end{aligned} \tag{2}$$

Here, $\mathcal{V}(x(0))$ is the value function on the initial state $x(0) \in X_0$ in RL. $\mathcal{I}^i(x; \theta, \alpha, \beta)$, $\mathcal{E}^j(x; \theta, \alpha, \beta)$ are the inequality and equality constraints encoded from barrier certificate and Lyapunov function via various relaxation techniques (such as SDP and LP that are later introduced), with $\theta \in \mathbb{R}^{|\theta|}$, $\alpha \in \mathbb{R}^{|\alpha|}$, $\beta \in \mathbb{R}^{|\beta|}$ as the vector of controller parameters, unknown system parameters, and parameters for certificates, respectively. As RL builds on the discrete-time MDPs, we need to discretize continuous dynamics f to compute $\mathbb{E}[\mathcal{V}(x(0))]$ by simulating different traces with the controller.

Specifically, in RL, $\mathcal{V}(x)$ satisfies the Bellman equation as: $\mathcal{V}(x) = r(x, \pi(x)) + \gamma\mathcal{V}'(x')$, where r is a reward function at the state-action pair $(x, \pi(x))$, encoding the desired learning goal for the controller, x' is the next system state, $\mathcal{V}'$ is the value function of x', and constant factor $\gamma < 1$.

Such a constrained optimization problem in RL is often infeasible. To leverage the gradient from the verification results and form an *end-to-end differentiable* framework, the above problem can be modified to our **bilevel optimization problem** by introducing a slack variable $c \in \mathbb{R}^+$. Specifically, the upper problem tries to solve:

$$\max_{\theta,\alpha} \mathbb{E}[\mathcal{V}(x(0); \theta, \alpha)] - \lambda(c^*(\theta, \alpha))^2 - \|\alpha - \alpha_0\|^2,$$

where $c^*(\theta, \alpha)$ is the solution to a lower-level problem:

$$\min_{\beta} c,$$
$$\text{subject to } \begin{cases} \mathcal{I}^i(x; \theta, \alpha, \beta) + c \geq 0, \\ \mathcal{E}^i(x; \theta, \alpha, \beta) \leq c, \\ \mathcal{E}^i(x; \theta, \alpha, \beta) \geq -c, \\ c \geq 0, \end{cases} \tag{3}$$

where $\alpha_0 \in \mathbb{R}^{|\alpha|}$ is the unknown ground truth value of the unknown system model parameter vector and needs to be estimated in learning. $\lambda \geq 0$ is a penalty multiplier. Overall, the lower-level problem tries to search for a feasible solution for the certificates while reducing the slack variable c. The upper-level problem tries to maximize the value function in RL, reduce the penalty from the lower-level slack variable, and learn the uncertain parameters. Once the lower optima $c^* = 0$, we can obtain a feasible solution for the original problem (2) and therefore generate a certificate. In this way, by differentiating the lower-level problem, the *gradient* c^*_θ of c^* wrt θ can be combined with the gradient of MBRL in the upper problem, and the entire bilevel optimization problem is differentiable.

4.1.1 SDP Relaxation for Bilevel Formulation.

Definition 4.1. ***(Sum-of-Squares)*** *A polynomial $p(x)$ is a sum-of-squares (SOS) if there exist polynomials $f_1(x), f_2(x), \cdots, f_m(x)$ such that $p(x) = \sum_{i=1}^m f_i(x)^2$. It is easy to infer that $p(x) \geq 0$.*

For the positivity of SOS, the three conditions in a barrier function as defined in Definition 3.2 can be relaxed into three SOS constraints based on Putinar's Positivstellensatz theorem [30]:

$$-B(x) - \sum_{i=1}^{s} \sigma_i(x) \cdot \xi_i(x) \in \Sigma[x];\ B(x) - \sum_{i=s+1}^{s+q} \sigma_i(x) \cdot \zeta_i(x) \in \Sigma[x];$$
$$-\frac{\partial B}{\partial x} \cdot f(x, \pi(x)) + \lambda B(x) - \sum_{i=s+q+1}^{s+q+r} \sigma_i(x)\psi_i(x) \in \Sigma[x].$$

Here, $\sigma_i \in \Sigma[x] \geq 0, i = 1, \cdots, s+q+r$. $\Sigma[x]$ denotes a set that contains all SOSs over x. $\xi_i(x) \geq 0$, $\zeta_i(x) \geq 0$, $\psi_i(x) \geq 0$ are the semi-algebraic constraints on X_0, X_u, and X, respectively. Note that in the above formulation, barrier parameter β is the decision variable while controller parameter θ and dynamics parameterα are fixed. If such SOS constraints can be solved, i.e., a feasible barrier function $B(x)$ exists, the system is proved to be always safe under the controller π and identified α.

Similarly, a Lyapunov function can be formulated into two SOS constraints as the following, and if a solution is obtained, the system stability can be guaranteed:

$$V(x) - \sum_{i=1}^{r} \sigma_i(x) \cdot \psi_i(x) \in \Sigma[x],$$
$$-\frac{\partial V}{\partial x} \cdot f(x, \pi(x)) - \sum_{i=r+1}^{2r} \sigma_i(x) \cdot \psi_i(x) \in \Sigma[x].$$

Next, we are going to show how to transform a problem with SOS constraints into an SDP, which is used in our framework. Given a polynomial $h(x)$ in SOS with the degree bound $2D$, we have:

$$h(x; \theta, \alpha, \beta) \in \Sigma[x] \Longleftrightarrow h(x; \theta, \alpha, \beta) = w(x)^T Q(\theta, \alpha, \beta) w(x),$$
$$Q(\theta, \alpha, \beta) \geq 0.$$

Here, $w(x) = (1, x_1, \cdots, x_n, x_1x_2, \cdots, x_n^D)$ is a vector of monomials, and Q is a $d^Q \times d^Q$ positive semi-definite matrix, where $d^Q = \binom{D+n}{D}$, called *Gram matrix* of $h(x)$ [35]. The problem in (2) is then:

$$\max_{\theta,\alpha} \mathbb{E}[V(x(0))],$$
$$\text{s.t. } h^i(x; \theta, \alpha, \beta) = w(x)^T Q^i(\theta, \alpha, \beta) w(x);\ Q^i(\theta, \alpha, \beta) \geq 0.$$

To make $h(x) = w(x)^T Q w(x)$, we need to list all the equations for coefficients in each monomial. Given an upper bound of degree 2D of the polynomial $h(x)$, let $a = (a_1, a_2, \cdots, a_n)$, $b = (b_1, b_2, \cdots, b_n)$, $d = (d_1, d_2, \cdots, d_n)$, $(a_i, b_i, d_i \in \mathbb{N})$ be the n-dimensional vectors indicating the degree of $x = (x_1, x_2, \cdots, x_n)$. Let $h(x) = \sum_{||a||_1 \leq 2D} h_a x_a$, where $||\cdot||_1$ is the 1-norm operator and $h_a(\theta, \alpha, \beta)$ is the coefficient of $x_a = \prod_{i=1}^n x_i^{a_i}$. Let $Q = Q_{bd}$, where $\{Q_{bd}\}$ represents the entry corresponding to x_b and x_d in the base vector $w(x)$. By equating the coefficients for all the monomials, we have

$$h(x; \theta, \alpha, \beta) = w(x)^T Q(\theta, \alpha, \beta) w(x) \Longleftrightarrow$$
$$\forall ||a||_1 \leq 2D, h_a(\theta, \alpha, \beta) = \sum_{b+d=a} Q_{bd}$$

as the equality constraints. Along with $Q \geq 0$, the problem in (2) can now be written as an SDP problem:

$$\max_{\theta,\alpha} \mathbb{E}[\mathcal{V}(x(0))],$$
$$\text{s.t. } h_a^i(\theta, \alpha, \beta) = \textstyle\sum_{b+d=a} Q_{bd}^i, \forall ||a||_1 \leq 2D,\ Q^i(\theta, \alpha, \beta) \geq 0. \tag{4}$$

Therefore, the bilevel optimization problem in (3) can be written as the following problem with SDP formulation:

$$\max_{\theta,\alpha} \mathbb{E}[\mathcal{V}(x(0); \theta, \alpha)] - \lambda(c^*(\theta, \alpha))^2 - \|\alpha - \alpha_0\|^2,$$

where $c^*(\theta, \alpha)$ is the solution to a lower-level problem:

$$\min_{\beta} c,$$
$$\text{subject to } \begin{cases} h_a^i(\theta, \alpha, \beta) \leq \sum_{b+d=a} Q_{bd}^i + c, \forall ||a||_1 \leq 2D, \\ h_a^i(\theta, \alpha, \beta) \geq \sum_{b+d=a} Q_{bd}^i - c, \forall ||a||_1 \leq 2D, \\ Q^i(\theta, \alpha, \beta) \geq 0, \\ c \geq 0, \end{cases} \tag{5}$$

where for barrier function, $i = (1, 2, 3)$, and for Lyapunov function, $i = (1, 2)$.

4.1.2 LP Relaxation for Bilevel Formulation. In addition to SDP, to improve scalability for higher-dimensional/higher-degree systems, we introduce the Handelman Representation [39] to encode the bilevel optimization formulation into an LP problem.

THEOREM 4.2. ***(Handelman)*** *$p(x)$ is defined over a compact and semi-algebraic set $K = \{x|f_j(x) \geq 0\}(j = 1, \cdots, m)$. If p can be represented as a positive linear combination of the inequalities f_j:*

$$p(x) = \sum_{k=1}^{n} \lambda_k \prod_{j=1}^{m} f_j(x)^{n_{k,j}}, \ \ for \ \lambda_k > 0 \ \ and \ \ n_{k,j} \in \mathbb{N}.$$

Then, $p(x)$ is non-negative on K.

Theorem 4.2 is proved in [13]. Based on it, finding the general Handelman Representation of a function $p(x)$ within a semi-algebraic set K defined by inequalities $\bigwedge_{i=1}^{m} f_j \geq 0$ is as follows.

(1) Fix the demand degree D. Generate all possible positive power product polynomials upto degree D in the form $p_a = \prod_{j=1}^{m} f_j^{a_j}$, where each $a_j \in \mathbb{N}$ and $\sum_{j=1}^{m} a_j = \|a\|_1 \leq D$, and thus $p_a \geq 0$ and we have $PP = \{p_a \mid \forall \|a\|_1 \leq D\}$.
(2) Express $p(x) = \sum_{p_a \in PP} c_s p_a$ and equate the known polynomial p with the set of power products from the last step to get the linear equality constraint involving c_s.
(3) If c_s exists, then $p(x) \geq 0$ is proved.

According to Theorem 4.2, Definition 3.2, and Krivine-Vasilescu-Handelman's Positivstellensatz [23], barrier function can be relaxed into following three LP constraints:

$$B(x) = \sum_{deg(p_\xi^\delta) \leq D} \lambda_\delta p_\xi^\delta; \quad -B(x) > \sum_{deg(p_\zeta^\omega) \leq D} \lambda_\omega p_\zeta^\omega;$$
$$\frac{\partial B}{\partial x} \cdot f(x, \pi(x;\theta);\alpha) - \lambda B(x) > \sum_{deg(p_\psi^\tau) \leq D} \lambda_\tau p_\psi^\tau$$

Here, $\lambda_\delta, \lambda_\tau, \lambda_\omega \geq 0$, p_ξ^δ, p_ψ^τ, and p_ζ^ω are all power products of polynomials on the initial space, state space, and unsafe space as $\xi_i(x) \geq 0, \zeta_i(x) \geq 0$, and $\psi_i(x) \geq 0$. Thus, according to Theorem 4.2, the positivity of each barrier certificate property can be ensured from the above formulation. Similarly, the Lyapunov function can be encoded into the following equations for stability properties.

$$V(x;\beta^V) > \sum_{deg(p_\psi^\gamma) \leq D} \lambda_\epsilon p_\psi^\gamma;$$
$$-\frac{\partial V}{\partial x} \cdot f(x, \pi(x;\theta);\alpha) > \sum_{deg(p_\psi^\gamma) \leq D} \lambda_\nu p_\psi^\gamma; \ \lambda_\epsilon, \lambda_\nu \geq 0.$$

We can conduct the same constraint generation for LP as SDP by equating the coefficients of all the possible monomials. Let $\lambda_i p_i^a$ denote the coefficient of monomial $x_a = \prod_{i=1}^{n} x_i^{a_i}, a = (a_1, \cdots, a_n)$, and thus $h_a^i(\theta, \alpha, \beta) = \lambda_i p_i^a$. Then the bilevel optimization problem in (3) can be written as the following problem with LP formulation:

$$\max_{\theta,\alpha} \mathbb{E}[\mathcal{V}(x(0);\theta,\alpha)] - \lambda(c^*(\theta,\alpha))^2 - \|\alpha - \alpha_0\|^2,$$

where $c^*(\theta,\alpha)$ is the solution to a lower-level problem:

$$\min_{\beta} c,$$
$$\text{subject to} \begin{cases} h_a^i(\theta,\alpha,\beta) \leq \lambda_i p_i^a + c, \forall ||a||_1 \leq D, \\ h_a^i(\theta,\alpha,\beta) \geq \lambda_i p_i^a - c, \forall ||a||_1 \leq D, \\ \forall \lambda_i \geq 0, \\ c \geq 0. \end{cases} \tag{6}$$

REMARK 4.3. *Note that both SOS and Handelman relaxations are incomplete, meaning that it is possible that a polynomial $p(x)$ is positive but cannot be expressed by SOS or Handelman representations.*

4.2 Bilevel Optimization Algorithm

We develop the following Algorithm 1 to solve the bilevel optimization problem by SDP and LP. The inputs to Algorithm 1 include the system model (with unknown parameters), the step length, a shielding set for ensuring the system safety during learning (more details below), and the form of polynomials for the certificates and the controller. The outputs include the learned controller and its certificates (barrier and/or Lyapunov function). There are four major modules in Algorithm 1, including variable transformation for the system model, shielding-based safe learning, parameter identification, and gradient computation for RL and certificates, as below.

Algorithm 1 End-to-end MBRL with Certification

1: **Input:** Nominal dynamics $\dot{x} = f(x,u;\alpha)$ and its discretized form f^d, with unknown system model parameters $\alpha \in [\underline{\alpha}, \bar{\alpha}]$, step length l, shielding set $\mathcal{S}$, barrier function form $B(x) = \beta^B \cdot w^B(x)$, Lyapunov function form $V(x) = \beta^V \cdot w^V(x)$, and controller form $\pi = \theta \cdot w^\theta(x)$ (β^B, β^V, θ unknown).
2: $\theta = 0$, $\beta^B = 0$, $\beta^V = 0$.
3: Conduct variable transformation if necessary.
4: **repeat**
5: Sample trajectory with θ; stop early if state $x \in \mathcal{S}$.
6: $\mathcal{V}'_{x'} = 0, \mathcal{V}'_{\theta'} = 0$.
7: **for** $t = T$ **down to** 0 with trajectory **do**
8: $\alpha \leftarrow \alpha + \gamma \frac{\Delta x}{\delta t}$; compute *gradients* $\mathcal{V}_x, \mathcal{V}_\theta$ as in Eq. (7).
9: Solve lower-level SDP or LP; compute c^*, β^*; compute *gradient* c^*_θ as in Eq. (8).
10: $\theta \leftarrow \theta + l(\mathcal{V}_\theta - 2\lambda c^* \cdot c^*_\theta)$, increase λ.
11: **until** $c^* = 0$
12: **Output:** $\pi(x;\theta), B(x;\beta^B), V(x;\beta^V)$.

Variable Transformation: If the system model contains non-polynomial univariate basic elementary functions such as $\sin(x)$, $\cos(x)$, $\exp(x)$, $\log(x)$, $1/x$, $\sqrt{x}$ or their combinations, we can **equivalently** transform them into polynomial terms with additional variables [25]. For example if $\dot{x} = \sin(x)$, we can let $m = \sin(x), n = \cos(x)$, and we then have $\dot{x} = m, \dot{m} = \cos(x)\dot{x} = nm, \dot{n} = -\sin(x)\dot{x} = -m^2$ as a polynomial system.

Shielding-based Safe Learning for Training: We compute a shielding set $\mathcal{S}$ to ensure system safety *during learning*, by stopping the current learning process if the system is within $\mathcal{S}$ (line 5 in Algorithm 1). Specifically, we can construct $\mathcal{S}$ offline, based on the definition that the system may enter the unsafe state set X_u in the

next step when it is in $\mathcal{S}$, i.e., $\mathcal{S} = \{x | \min_\alpha \min_{x_u \in X_u} \|x' - x_u\| = 0\}$ s.t. $\underline{\alpha} \le \alpha \le \bar{\alpha}$. Here, x is the current state and x' is the predicted next state based on some $\alpha \in [\underline{\alpha}, \bar{\alpha}]$ for the discretized system model f^d (by applying zeroth-order hold to the continuous system model f). During the learning process, when the system is within the shielding set $\mathcal{S}$, the current learning process will stop and start over again. Note that we compute the set with the entire interval of α and it only provides *one-step safety* as explained in Remark 3.3. Also, note that the shielding set is for ensuring safety *during* learning. Regardless of whether we use it, the controller we obtained *after* learning if it exists with the generated barrier function based on the identified dynamics, is always guaranteed to be *infinite-time safe* as defined in Definition 3.1.

Parameter Identification for System Model: To learn the unknown parameters of the system model during RL, we can compute the state difference between any two adjacent control time steps and then compute the approximated gradient for parameters α. For instance, for a one-dimensional system $\dot{x} = \alpha x$, we can perform $\alpha \leftarrow \alpha + \gamma \cdot [f^d(x(\delta t), \pi(x(\delta t)); \alpha) - x(\delta t)]/\delta$, where γ is the learning rate and f^d is the discretized system model from the continuous system model f, as long as the sampling period δ is small enough according to the Nyquist–Shannon sampling theorem. In the experiments, we observe that α always converges to its ground truth with the learning (i.e., $\|\alpha - \alpha_0\|^2 \to 0$), albeit we cannot guarantee the convergence. Note that the safety or stability guarantee is established on the identified system parameters, and we have the following remark.

Remark 4.4. ***(Parameter Identification Error and Certification)*** *The certification is built on the identified system parameters. However, due to the errors from the discretization and gradient approximation, the final identified parameters may be close to the ground truth value but not the same (the ground truth value is in fact assumed as unknown in this paper). However, if the identification error can be quantified, it can then be viewed as a bounded disturbance to the system. In which case our approach can be easily extended to such disturbed systems as barrier functions can be built on uncertain parameters [34] and parametric Lyapunov function can also be synthesized for LTI systems [11].*

Computing the Value Function Gradient in MBRL: As mentioned above, by applying zeroth-order hold to the continuous system model f, we can obtain the discrete-time model $x(t+1) = f^d(x(t), \pi(x(t)))$. Note that f^d contains the unknown system parameters α, which are updated at run-time. By differentiating the Bellman equation $\mathcal{V}(x) = r(x, \pi(x)) + \gamma \mathcal{V}'(f^d(x, \pi(x)))$ [14], we can obtain the value function *gradients* $\mathcal{V}_x, \mathcal{V}_\theta$ with respect to the state x and the controller parameters θ:

$$\begin{aligned} \mathcal{V}_x &= r_x + r_u \pi_x + \gamma \mathcal{V}'_{x'}(f^d_x + f^d_u \pi_x), \\ \mathcal{V}_\theta &= r_u \pi_\theta + \gamma \mathcal{V}'_{x'} f^d_u \pi_\theta + \gamma \mathcal{V}'_\theta, \end{aligned} \tag{7}$$

where every subscript is a partial *derivative*. $\mathbb{E}[\mathcal{V}(x(0))]$ will be increased by updating θ with the direction as *gradient* $\mathcal{V}_\theta(x(0))$. For the implementation, we can collect a trajectory $\{x(0), u(0), r(0), \cdots, x(T), u(T), r(T)\}$ of the discrete-time system by the controller, let $\mathcal{V}'_{x^T} = 0, \mathcal{V}'_\theta = 0$ and roll back to the initial state $x(0)$, and obtain the *gradient* $\mathcal{V}_\theta(x(0))$ based on equation (7).

Computing the Certification Gradient: To solve the bilevel problem in an end-to-end manner, the slack variable c^* should be differentiable to the controller parameters θ as it connects the two sub-problems. The lower-level SDP or LP belongs to the disciplined parameterized programming problem where the optimization variables are c, β and the parameters are θ. And the lower-level problem defined in (3) can be viewed as a function mapping of θ to the optimal solution (c^*, β^*), e.g., $\mathcal{F} : \theta \to (c^*, \beta^*)$. According to [1], function $\mathcal{F}$ can be expressed as the composition $\mathcal{R} \circ s \circ C$, where C represents the canonical mapping of θ to a cone problem (A, e), which is then solved by a cone solver s and returns $(\bar{c}^*, \bar{\beta}^*)$. Finally, the retriever $\mathcal{R}$ translates the cone solution $(\bar{c}^*, \bar{\beta}^*)$ to the original solution (c^*, β^*). Thus, according to the chain rule, we have

$$c^*_\theta = \mathcal{R}_{(\bar{c}^*, \bar{\beta}^*)} \cdot s_{(A,e)} \cdot C_\theta \tag{8}$$

as a part of the *gradient* in the upper-level objective. Overall, the controller is updated as $\theta \leftarrow \theta + l(\mathcal{V}_\theta - 2\lambda c^* \cdot c^*_\theta)$. As the termination condition in Algorithm 1, $c^* = 0$ indicates that the original constrained problem has a feasible solution β^* given the current θ, meaning that there exists a certificate for the learned controller.

4.3 Theoretical Analysis on Optimality

Proposition 4.5. ***(Soundness)*** *Our approach is sound as the final learned controller is formally guaranteed to hold the barrier certificate for safety and the Lyapunov function for stability, only if the identified parameters are the truth values in the system model.*

Take the SDP relaxation as an example, the soundness is easy to check as the slack variables c of the learned controller is reduced to 0 and thus the solution of the bilevel optimization problem (5) is a solution to problem (4). And it is similar to the LP relaxation.

Remark 4.6. ***(Incompleteness)*** *Our approach is incomplete as we cannot guarantee our approach will always be able to search a controller with a barrier certificate and Lyapunov function. This is due to 1) the incompleteness of the SDP and LP relaxation approaches we utilized, 2) the limited controller parameter space we optimize on, and 3) the gradients of RL and slack variable affect each other.*

However, with some mild assumptions, we can provide the following completeness analysis for our framework. Here we take the SDP relaxation of the bilevel problem as an example, and the analysis can also be applied to the LP-based bilevel problem.

Proposition 4.7. ***(Stationary Point)*** *Suppose that there exists a step length l satisfying the Wolfe conditions [51] for the value gradient $\mathcal{V}_{\theta_k}$ and the verification gradient $c^*_{\theta_k}$ at the k-th update. Then Algorithm 1 will reach a stationary point for problem* (5).

$\mathbb{E}[\mathcal{V}(x(0); \theta, \alpha)] - \lambda(c^*(\theta, \alpha))^2$ can be viewed as an unconstrained optimization problem over θ. For this problem, since we can compute its closed-form *gradient* as $\mathcal{V}_\theta$ and c^*_θ, we can choose the step length l by the Wolfe conditions, which lead problem (5) to a stationary point when given a λ. The proof of Proposition 4.7 can be adapted from the general analysis in [31] and is shown below.

Reaching a stationary point does not necessarily mean that $c^* = 0$ in problem (5) and thus does not necessarily lead to a solution of (4). However, with stronger assumptions, we can guarantee $c^* = 0$ (and thus a solution of (4)) as follows in Theorem 4.8. The proof of this theorem is also adapted from [31] and shown below.

Theorem 4.8. ***(Global Solution)*** *Suppose that θ_k is the exact global maximizer of the objective function in problem* (5) *at the k-th update iteration, and that $\lambda_k \to \infty$. Then every limit point θ^* of the sequence $\{\theta_k\}$ is a global solution of* (4).

Proof for Proposition 4.7: Let $g(\theta_k) = -\mathbb{E}[\mathcal{V}(x(0);\theta_k,\alpha)] + \lambda_k c^2(\theta_k)$, which is the negative value of the objective in the bilevel problem (5) and needs to be minimized. Let $p_k = -g_{\theta_k}$ denote the line search direction as the gradient. According to the second Wolfe condition with two constant numbers $0 < d_1 < d_2 < 1$, we have

$$(g_{\theta_{k+1}} - g_{\theta_k})^T p_k \geq (d_2 - 1) g_{\theta_k}^T p_k.$$

Assume that the *gradient* g_θ of $g(\theta)$ is Lipschitz continuous, which implies that there exists a constant value L such that

$$\frac{g_{\theta_{k+1}} - g_{\theta_k}}{\theta_{k+1} - \theta_k} = \frac{g_{\theta_{k+1}} - g_{\theta_k}}{l_k p_k} \leq L;\ (g_{\theta_{k+1}} - g_{\theta_k})^T p_k \leq l_k L \|p_k\|^2,$$

where l_k is the step length. Combine the two inequalities, we have

$$l_k \geq \frac{d_2 - 1}{L} \frac{g_{\theta_k}^T p_k}{\|p_k\|^2}.$$

According to the first Wolfe condition, we can obtain

$$g(\theta_{k+1}) \leq g(\theta_k) - d_1 \frac{1 - d_2}{L} \frac{(g_{\theta_k}^T p_k)^2}{\|p_k\|^2},$$
$$g(\theta_{k+1}) \leq g(\theta_k) - d \left\|g_{\theta_k}\right\|^2,$$

where $d = \frac{d_1(1-d_2)}{L}$. We can then extend the inequality to the initial value as

$$g(\theta_{k+1}) \leq g(\theta_0) - d \sum_{i=0}^{k} \left\|g_{\theta_i}\right\|^2.$$

Since we are considering the episodic RL and c^* is bounded from the lower-level SDP, function g is bounded, and thus there is a positive number N such that

$$\sum_{k=0}^{\infty} \left\|g_{\theta_k}\right\|^2 < N \Longrightarrow \lim_{k\to\infty} \left\|g_{\theta_k}\right\| = 0,$$

which indicates that the solving the problem (5) with Algorithm 1 eventually reaches a stationary point. □

Proof for Theorem 4.8: Problem (4) can be viewed as a constrained problem with constraint $c(\theta) = 0$. Suppose that $\bar{\theta}$ is a global solution of problem (4) with $c(\bar{\theta}) = 0$ (meaning that there exists a feasible certificate), and name the objective function of problem (5) $\max_{\theta,\alpha} \mathbb{E}[\mathcal{V}(x(0);\theta,\alpha)] - \lambda(c^*(\theta,\alpha))^2 - \|\alpha - \alpha_0\|^2$ as $g(\theta)$, we then have

$$g(\bar{\theta}) \geq g(\theta)\ \forall \theta, c(\theta) = 0.$$

Since θ_k maximizes $g(\theta, \lambda_k)$ for each iteration k, we then have $g(\theta_k, \lambda_k) \geq g(\bar{\theta}, \lambda_k)$, resulting in the following inequality:

$$\begin{aligned}\mathbb{E}[\mathcal{V}(x(0);\theta_k,\alpha)] - \lambda_k c^2(\theta_k) &\geq \mathbb{E}[\mathcal{V}(x(0);\bar{\theta},\alpha)] - \lambda_k c^2(\bar{\theta}) \\ &= \mathbb{E}[\mathcal{V}(x(0);\bar{\theta},\alpha)],\end{aligned}$$

and thus,

$$c^2(\theta_k) \leq \frac{\mathbb{E}[\mathcal{V}(x(0);\theta_k,\alpha)] - \mathbb{E}[\mathcal{V}(x(0);\bar{\theta},\alpha)]}{\lambda_k}.$$

Suppose that θ^* is a limit point of the sequence $\{\theta_k\}$, so that there exists an infinite sub-sequences $\mathcal{K}$ such that $\lim_{k\in\mathcal{K}} \theta_k = \theta^*$. When $k \to \infty$, we then have

$$c^2(\theta^*) = \lim_{k\in\mathcal{K}} c^2(\theta_k) \leq \lim_{k\in\mathcal{K}} \frac{\mathbb{E}[\mathcal{V}(x(0);\theta_k,\alpha)] - \mathbb{E}[\mathcal{V}(x(0);\bar{\theta},\alpha)]}{\lambda_k}.$$

For $\mathbb{E}[\mathcal{V}(x(0);\theta_k,\alpha)]$ and $\mathbb{E}[\mathcal{V}(x(0);\bar{\theta},\alpha)]$, since they follow the same distribution on $x(0) \in X_0$, and we deal with the episodic setting in RL, their difference is bounded. Because we have $\lambda_k \to \infty$ when $k \to \infty$, so $c(\theta^*) = 0$, meaning that θ^* is the feasible solution of the problem (4).

Moreover, follow the inequality of θ_k with $k \to \infty$, we have

$$\begin{aligned}\lim_{k\in\mathcal{K}} \mathbb{E}[\mathcal{V}(x(0);\theta_k,\alpha)] - \lambda_k c^2(\theta_k) &\geq \mathbb{E}[\mathcal{V}(x(0);\bar{\theta},\alpha)], \\ \mathbb{E}[\mathcal{V}(x(0);\theta^*,\alpha)] - \lambda_k c^2(\theta^*) &\geq \mathbb{E}[\mathcal{V}(x(0);\bar{\theta},\alpha)], \\ \mathbb{E}[\mathcal{V}(x(0);\theta^*,\alpha)] &\geq \mathbb{E}[\mathcal{V}(x(0);\bar{\theta},\alpha)].\end{aligned}$$

Since θ^* is a feasible solution with $c(\theta^*) = 0$, whose objective is not smaller than that of the global solution $\bar{\theta}$, we can conclude that θ^* is a global solution as well, as claimed in Theorem 4.8. □

5 EXPERIMENTAL RESULTS

Experimental Settings: As the focus of our work is to learn certified controllers that formally guarantee system safety and stability, we will compare our approach with an SVG(∞) [14] based method over a variety of benchmarks. For a fair comparison, the SVG is equipped with parameter identification and shielding *during both training and testing*. For our approach, shielding is used *only during training* but not needed at testing, as safety is already guaranteed by the barrier function. For the safety property, the baseline SVG is further extended to solve a CMDP with a safety constraint on the system state's Euclidean distance to the unsafe set being greater than a threshold. This CMDP is solved by the standard penalty method for its augmented Lagrangian problem. For the stability property, we encode it in the reward function as the negative L2 norm to the origin point (target). We apply formal verification (i.e., search for barrier or Lyapunov function) at *each iteration* of the SVG to check whether safety or stability can hold.

In our comparison, we carefully select benchmarks with 2-6 states from [4, 6, 15, 34]. It is worth noting that generating certificates for a dynamical system with *a given controller* is already an NP-hard problem [33] in theory and is difficult to solve in practice. Current state-of-the-art works of learning-based controller synthesis with certificates therefore mainly focus on low-dimensional systems with fewer than six-dimensional states [5, 8, 12, 24, 28, 37, 53]. Thus, we believe that the chosen benchmarks can well reflect the advantages and limitations of our approach. We test the examples on an Intel-i7 machine with 16GB memory.

Comparison on Safety and Stability: Table 1 summarizes the certification results of our approach with SDP and LP and the SVG-based method on four examples, as explained below.

PJ (Safety). We consider a modified example from [34], whose dynamics is expressed as $\dot{x_1} = \alpha_1 x_2, \dot{x_2} = \alpha_2 x_1^3 + u$ where state $x_1, x_2 \in [-100, 100]$, and $\alpha_1, \alpha_2 \in [-1.5, 1.5]$. The initial and unsafe sets are $X_0 = \{(x_1-1.5)^2+x_2^2 \leq 0.25\}, X_u = \{(x_1+0.8)^2+(x_2+1)^2 \leq 0.25\}$. We focus on finding a linear controller and barrier certificate

Table 1: Certification results by our approach with SDP and LP, and the SVG-based method for benchmarks. B^d denotes the successfully obtained barrier function with polynomial degree d for safety guarantees, and L^d denotes the successfully obtained Lyapunov function with polynomial degree d for stability guarantee. '×' means there does not exist a certificate as the controller is unsafe. '–' means the failure to find a certificate with degrees up to 6. Our approach (both SDP-based and LP-based) succeeds in finding a certified controller in all cases while SVG cannot in most cases.

Examples	PJ	Pendulum	LK	Att. Control
Ours (LP)	B^2	L^2	L^2, B^6	L^2
Ours (SDP)	B^2	L^2	L^2, B^4	L^2
SVG (CMDP)	×	–	L^2, –	–

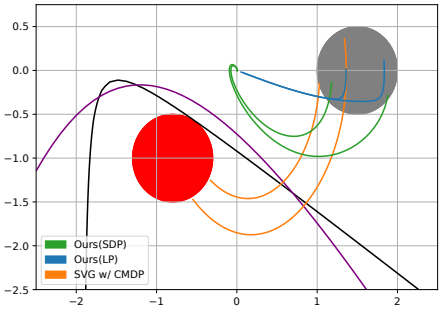

Figure 2: Trajectories under the learned controllers from our approach with SDP and LP, and from SVG (with shielding at testing) for the PJ example. Initial space X_0 is in grey, and unsafe region X_u is in red. The barrier function 0-level plot by ours with the SDP formulation is in black and with the LP formulation is in purple.

for system safety in this example. Fig. 2 shows the simulated system trajectories by the learned controllers from our approach and from the SVG method with CMDP. It also shows the 0-level contour plot of the barrier functions from our approach. We can see that the controller learned by the SVG is unsafe (entering the unsafe region in red and has to be stopped by shielding) and thus has no safety certificate. Our approach is safe during and after learning with shielding and the learned barrier certificate.

Inverted Pendulum (Stability). We consider the inverted pendulum example from the gym environment [4] by a non-linear controller with $\sin\varphi$ and $\cos\varphi$ terms. The pendulum can be expressed $\ddot{\varphi} = -\frac{g}{l}sin(\varphi) - \frac{d}{ml^2}\dot{\varphi} + \frac{u}{ml^2}$ where φ is the angle deviation. System state $(\varphi, \dot{\varphi}) \in \{\varphi^2 + \dot{\varphi}^2 \leq 2\}$. $m = 1, l = 1, d = 0.1$. Unknown parameter $g \in [9, 10.5]$. $X_0 = \{\varphi^2 + \dot{\varphi}^2 \leq 1\}$. For the sin function, we conduct variable transformation with $p = \sin(\varphi)$ and $q = \cos(\varphi)$, so that the dynamics can be transformed into a 4D polynomial system. Fig. 3 shows the trajectories from our approach (with SDP and LP) and from SVG, along with the Lyapunov function generated by our approach with SDP. The SVG fails to generate a Lyapunov function with a polynomial degree up to 6 during the entire learning, as shown in Table 1 while our approach succeeds with quadratic certificates by both SDP and LP.

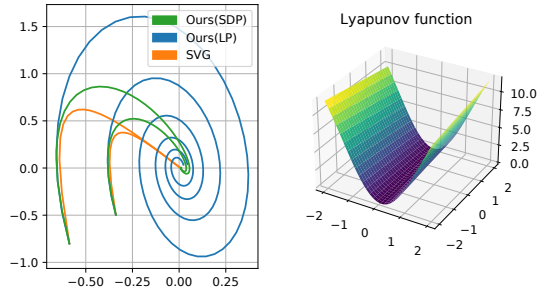

Figure 3: Trajectories on $(\varphi, \dot{\varphi})$ under the learned controllers from our approach by SDP and LP, and from baseline SVG for the Pendulum example. The right subplot shows the affiliated Lyapunov function obtained by SDP for the learned controller in our approach.

Lane Keeping (Safety and Stability). We consider a lane-keeping example [6], where we try to derive a linear controller with barrier and Lyapunov certificate functions for both safety and stability. The system can be expressed as $\dot{x} = Ax + Bu$, where

$$A = \begin{bmatrix} 0 & 1 & v_x & 0 \\ 0 & \frac{-(C_{\alpha f}+C_{\alpha r})}{mv_x} & 0 & \frac{bC_{\alpha r}-aC_{\alpha f}}{mv_x} - v_x + \alpha_1 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{bC_{\alpha r}-aC_{\alpha f}}{I_z v_x} & 0 & \frac{-(a^2C_{\alpha f}+b^2C_{\alpha r})}{I_z v_x} + \alpha_2 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{C_{\alpha f}}{m} \\ 0 \\ \frac{aC_{\alpha f}}{I_z} \end{bmatrix}.$$

Here, $x = (y, v_y, \psi_e, r)^T$ is the system state with lateral displacement error y, lateral velocity v_y, yaw angle error ψ_e and yaw rate r. Control input u represents the steering angle at the front tire. v_x is the longitudinal vehicle velocity. $\alpha_1 \in [-15, 5], \alpha_2 \in [-10, -1]$ are the unknown parameters and other symbols are all known constants. $X_0 = \{\|x - x_0\|_2 \leq 0.2\}$, $X_u = \{\|x - x_u\|_2 \leq 1\}$, and $X = \{\|x\|_2 \leq 3\}$, where $x_0 = (0.4, 2, 0.5, 0)^T$ and $x_u = (2, 2, 0, 1)^T$. Fig. 4 shows the simulated system trajectories under the controllers from our approach with SDP and LP, and from the SVG with CMDP. It also shows the barrier function value and the Lyapunov function value generated by our approach with LP, along with the trajectories over time. The SVG with CMDP can generate a quadratic Lyapunov function for stability but fails to find a barrier function for safety, as also shown in Table 1. LP succeeds with polynomial degree 6 for barrier function and SDP succeeds with degree 4. Therefore, SDP takes a shorter time for each iteration in this example.

Attitude Control(Stability). The attitude control example [15] is the most complex one we tested. It has 6D state and 3D control input, which can be expressed as

$$\dot{\omega}_1 = \alpha_1(u_0 + \omega_2\omega_3);\ \dot{\omega}_2 = \alpha_2(u_1 - 3\omega_1\omega_3);\ \dot{\omega}_3 = u_2 + 2\omega_1\omega_2,$$

$$\dot{\psi}_1 = 0.5\left(\omega_2(\psi_1\psi_2 - \psi_3) + \omega_3(\psi_1\psi_3 + \psi_2) + \omega_1(\psi_1^2 + 1)\right),$$

$$\dot{\psi}_2 = 0.5\left(\omega_1(\psi_1\psi_2 + \psi_3) + \omega_3(\psi_2\psi_3 - \psi_1) + \omega_2(\psi_1^2 + 1)\right),$$

$$\dot{\psi}_3 = 0.5\left(\omega_1(\psi_1\psi_3 - \psi_2) + \omega_2(\psi_2\psi_3 + \psi_1) + \omega_3(\psi_3^2 + 1)\right),$$

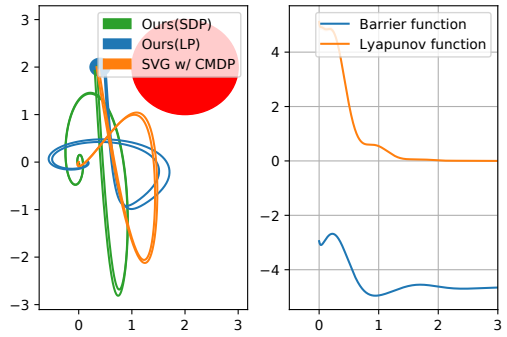

Figure 4: *Left:* Trajectories on (y, v_y) under the learned controllers from ours and from SVG with CMDP for the Lane Keeping example. *Right:* Barrier and Lyapunov function values generated by our approach with LP, with trajectories.

where the state $x = (\omega, \psi)$ consists of the angular velocity vector in a body-fixed frame $\omega \in \mathbb{R}^3$ and the Rodrigues parameter vector $\psi \in \mathbb{R}^3$. $u \in \mathbb{R}^3$ is the control torque. State space $X = \{x \mid ||x||_2 \leq 2\}$, unknown dynamical parameters $\alpha_1 \in [-1, 2], \alpha_2 \in [-0.5, 1.5]$.

Our approach can successfully find a cubic polynomial controller with a quadratic Lyapunov function for stability with both SDP and LP, while SVG cannot generate a Lyapunov function for the entire learning process. Fig. 5 shows the simulated trajectories by learned controllers from different approaches.

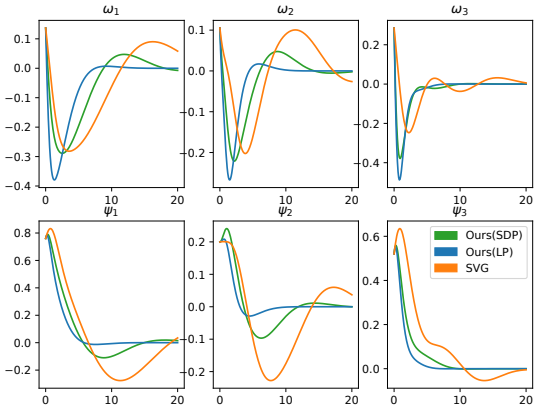

Figure 5: Trajectory on each dimension under the learned controllers from SVG and our approach for Attitude Control.

Timing Complexity of SDP and LP Relaxation: We first test the timing efficiency of SDP and LP in each iteration for the examples introduced before, with the polynomial degrees for certificates set as in Table 1, and the timing results are summarized in Table 2. LP does not show a big advantage, but note that most certificate functions are quadratic in Table 1.

Thus, to further test the scalability of SDP and LP for higher-dimensional systems, we raise the degree of certificates from 2 up to 6 in testing, with runtime shown in Fig. 6. We also show that the total number of variables of LP is fewer than SDP's in the Attitude Control example in Table 3, especially for higher-dimensional systems. These demonstrate LP's advantage in scalability, and we plan to explore it further in future work for larger examples.

Table 2: Averaged running time of each iteration by SDP and LP in our approach, with the degree shown in Table 1. LP is typically a bit more efficient than SDP, except for generating barrier certificate in the Lane Keeping example, where SDP succeeds in polynomial degree 4 while LP needs degree 6.

	PJ	Pendulum	Lane Keep	Att. Control
SDP(s)	0.95(B)	1.03(L)	2.55(B), 0.45(L)	7.32(L)
LP(s)	0.58(B)	0.81(L)	14.8(B), 0.4(L)	6.52(L)

Table 3: Number of variables in the Attitude Control example under different degrees of certificate functions.

Degree	2	3	4	5	6
SDP	475	4042	4168	26078	26502
LP	302	664	1380	2675	4849

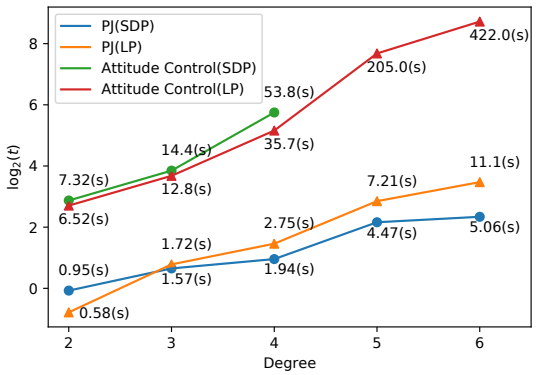

Figure 6: Averaged running time of each iteration by LP and SDP in PJ and Attitude Control examples under different degrees of certificate functions, shown in $\log_2$ magnitude with values on the plot. SDP reports timeout with degrees 5 and 6 in Attitude Control, while LP can succeed.

6 CONCLUSION

In this paper, we present a joint differentiable optimization and verification framework for certified reinforcement learning, by formulating and solving a novel bilevel optimization problem in an end-to-end differentiable manner, leveraging the gradients from both the certificates and the value function. Experimental results demonstrate the effectiveness of our approach in finding controllers with certificates for guaranteeing system safety and stability.

ACKNOWLEDGMENTS

This work is supported in part by NSF grants 1834701, 1724341, 2038853, and Office of Naval Research grant N00014-19-1-2496.

REFERENCES

[1] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. 2019. Differentiable Convex Optimization Layers. *Advances in Neural Information Processing Systems* 32 (2019), 9562–9574.

[2] Shane Barratt, Guillermo Angeris, and Stephen Boyd. 2021. Automatic repair of convex optimization problems. *Optimization and Engineering* 22, 1 (2021).

[3] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. 2017. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems* 30 (2017).

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).

[5] Ya-Chien Chang, Nima Roohi, and Sicun Gao. 2019. Neural lyapunov control. *Advances in neural information processing systems* 32 (2019).

[6] Bo-Chiuan Chen, Bi-Cheng Luan, and Kangwon Lee. 2014. Design of lane keeping system using adaptive model predictive control. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 922–926.

[7] Shaoru Chen, Mahyar Fazlyab, Manfred Morari, George J Pappas, and Victor M Preciado. 2020. Learning lyapunov functions for piecewise affine systems with neural network controllers. *arXiv preprint arXiv:2008.06546* (2020).

[8] Charles Dawson, Sicun Gao, and Chuchu Fan. 2022. Safe Control with Learned Certificates: A Survey of Neural Lyapunov, Barrier, and Contraction methods. *arXiv preprint arXiv:2202.11762* (2022).

[9] Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. 2022. Safe nonlinear control using robust neural lyapunov-barrier functions. In *Conference on Robot Learning*. PMLR, 1724–1735.

[10] Jiameng Fan, Chao Huang, Xin Chen, Wenchao Li, and Qi Zhu. 2020. ReachNN*: A Tool for Reachability Analysis of Neural-Network Controlled Systems. In *Automated Technology for Verification and Analysis*, Dang Van Hung and Oleg Sokolsky (Eds.). Springer International Publishing, Cham, 537–542.

[11] Minyue Fu and Soura Dasgupta. 2000. Parametric Lyapunov functions for uncertain systems: The multiplier approach. In *Advances in linear matrix inequality methods in control*. SIAM, 95–108.

[12] Meichen Guo, Claudio De Persis, and Pietro Tesi. 2020. Learning control for polynomial systems using sum of squares relaxations. In *CDC*. IEEE, 2436–2441.

[13] David Handelman. 1988. Representing polynomials by positive linear functions on compact convex polyhedra. *Pacific J. Math.* 132, 1 (1988), 35–62.

[14] Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. 2015. Learning Continuous Control Policies by Stochastic Value Gradients. *Advances in Neural Information Processing Systems* 28 (2015).

[15] Chao Huang, Jiameng Fan, Xin Chen, Wenchao Li, and Qi Zhu. 2022. Polar: A polynomial arithmetic framework for verifying neural-network controlled systems. In *ATVA*. Springer, 414–430.

[16] Chao Huang, Jiameng Fan, Wenchao Li, Xin Chen, and Qi Zhu. 2019. ReachNN: Reachability analysis of neural-network controlled systems. *ACM Transactions on Embedded Computing Systems (TECS)* 18, 5s (2019), 1–22.

[17] Wanxin Jin, Shaoshuai Mou, and George J Pappas. 2021. Safe pontryagin differentiable programming. *NeurIPS* 34 (2021), 16034–16050.

[18] Wanxin Jin, Zhaoran Wang, Zhuoran Yang, and Shaoshuai Mou. 2020. Neural certificates for safe control policies. *arXiv preprint arXiv:2006.08465* (2020).

[19] Hassan K Khalil. 2002. Nonlinear systems third edition. (2002).

[20] John C Knight. 2002. Safety critical systems: challenges and directions. In *Proceedings of the 24th international conference on software engineering*. 547–550.

[21] William Koch, Renato Mancuso, Richard West, and Azer Bestavros. 2019. Reinforcement learning for UAV attitude control. *TCPS* 3, 2 (2019), 1–21.

[22] Hui Kong, Fei He, Xiaoyu Song, William NN Hung, and Ming Gu. 2013. Exponential-condition-based barrier certificate generation for safety verification of hybrid systems. In *CAV*. Springer, 242–257.

[23] Jean B Lasserre. 2005. Polynomial programming: LP-relaxations also converge. *SIAM Journal on Optimization* 15, 2 (2005), 383–393.

[24] Lars Lindemann, Haimin Hu, Alexander Robey, Hanwen Zhang, Dimos Dimarogonas, Stephen Tu, and Nikolai Matni. 2021. Learning Hybrid Control Barrier Functions from Data. In *Conference on Robot Learning*. PMLR, 1351–1370.

[25] Jiang Liu, Naijun Zhan, Hengjun Zhao, and Liang Zou. 2015. Abstraction of elementary hybrid systems by variable transformation. In *International Symposium on Formal Methods*. Springer, 360–377.

[26] Xiangguo Liu, Chao Huang, Yixuan Wang, Bowen Zheng, and Qi Zhu. 2022. Physics-aware safety-assured design of hierarchical neural network based planner. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 137–146.

[27] Xiangguo Liu, Ruochen Jiao, Bowen Zheng, Dave Liang, and Qi Zhu. 2023. Safety-driven Interactive Planning for Neural Network-based Lane Changing. In *Proceedings of the 28th Asia and South Pacific Design Automation Conference*. 39–45.

[28] Yuping Luo and Tengyu Ma. 2021. Learning Barrier Certificates: Towards Safe Reinforcement Learning with Zero Training-time Violations. *NeurIPS* 34 (2021).

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

[30] Jiawang Nie and Markus Schweighofer. 2007. On the complexity of Putinar's Positivstellensatz. *Journal of Complexity* 23, 1 (2007), 135–150.

[31] Jorge Nocedal and Stephen J Wright. 1999. *Numerical optimization*. Springer.

[32] Antonis Papachristodoulou and Stephen Prajna. 2002. On the construction of Lyapunov functions using the sum of squares decomposition. In *CDC*, Vol. 3. IEEE, 3482–3487.

[33] Stephen Prajna. 2006. Barrier certificates for nonlinear model validation. *Automatica* 42, 1 (2006), 117–126.

[34] Stephen Prajna and Ali Jadbabaie. 2004. Safety verification of hybrid systems using barrier certificates. In *HSCC*. Springer, 477–492.

[35] Stephen Prajna, Antonis Papachristodoulou, and Pablo A Parrilo. 2002. Introducing SOSTOOLS: A general purpose sum of squares programming solver. In *Proceedings of the 41st CDC, 2002.*, Vol. 1. IEEE, 741–746.

[36] Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. 2021. Learning Safe Multi-Agent Control with Decentralized Neural Barrier Certificates. *arXiv preprint arXiv:2101.05436* (2021).

[37] Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. 2020. Learning control barrier functions from expert demonstrations. In *CDC*. IEEE, 3717–3724.

[38] Alexander Robey, Lars Lindemann, Stephen Tu, and Nikolai Matni. 2021. Learning robust hybrid control barrier functions for uncertain systems. *IFAC-PapersOnLine* 54, 5 (2021), 1–6.

[39] Sriram Sankaranarayanan, Xin Chen, et al. 2013. Lyapunov Function Synthesis using Handelman Representations. *IFAC Proceedings Volumes* 46, 23 (2013).

[40] Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. 2020. Learning to be safe: Deep rl with a safety critic. *arXiv preprint arXiv:2010.14603* (2020).

[41] Adam Stooke, Joshua Achiam, and Pieter Abbeel. 2020. Responsive safety in reinforcement learning by pid lagrangian methods. In *ICML*. PMLR, 9133–9143.

[42] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[43] Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. 2021. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters* 6, 3 (2021), 4915–4922.

[44] Hiroyasu Tsukamoto and Soon-Jo Chung. 2020. Neural contraction metrics for robust estimation and control: A convex optimization approach. *IEEE Control Systems Letters* 5, 1 (2020), 211–216.

[45] Yixuan Wang, Chao Huang, Zhaoran Wang, Zhilu Wang, and Qi Zhu. 2022. Design-while-verify: correct-by-construction control learning with verification in the loop. In *Proceedings of the 59th Design Automation Conference*. 925–930.

[46] Yixuan Wang, Chao Huang, Zhilu Wang, Shichao Xu, Zhaoran Wang, and Qi Zhu. 2021. Cocktail: Learn a better neural network controller from multiple experts via adaptive mixing and robust distillation. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 397–402.

[47] Yixuan Wang, Chao Huang, and Qi Zhu. 2020. Energy-efficient control adaptation with safety guarantees for learning-enabled cyber-physical systems. In *2020 International Conference On Computer Aided Design (ICCAD)*. IEEE, 1–9.

[48] Yixuan Wang, Simon Sinong Zhan, Ruochen Jiao, Zhilu Wang, Wanxin Jin, Zhuoran Yang, Zhaoran Wang, Chao Huang, and Qi Zhu. 2022. Enforcing Hard Constraints with Soft Barriers: Safe Reinforcement Learning in Unknown Stochastic Environments. *arXiv preprint arXiv:2209.15090* (2022).

[49] Zhilu Wang, Chao Huang, and Qi Zhu. 2022. Efficient Global Robustness Certification of Neural Networks via Interleaving Twin-Network Encoding. In *DATE'22: Proceedings of the Conference on Design, Automation and Test in Europe*.

[50] T. Wei, Yanzhi Wang, and Q. Zhu. 2017. Deep reinforcement learning for building HVAC control. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6. https://doi.org/10.1145/3061639.3062224

[51] Philip Wolfe. 1969. Convergence conditions for ascent methods. *SIAM review* 11, 2 (1969), 226–235.

[52] Shichao Xu, Yixuan Wang, Yanzhi Wang, Zheng O'Neill, and Qi Zhu. 2020. One for many: Transfer learning for building hvac control. In *Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 230–239.

[53] Hengjun Zhao, Xia Zeng, Taolue Chen, and Zhiming Liu. 2020. Synthesizing barrier certificates using neural networks. In *HSCC*. 1–11.

[54] Qi Zhu, Chao Huang, Ruochen Jiao, Shuyue Lan, Hengyi Liang, Xiangguo Liu, Yixuan Wang, Zhilu Wang, and Shichao Xu. 2021. Safety-assured design and adaptation of learning-enabled autonomous systems. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*. 753–760.

[55] Qi Zhu, Wenchao Li, Hyoseung Kim, Yecheng Xiang, Kacper Wardega, Zhilu Wang, Yixuan Wang, Hengyi Liang, Chao Huang, Jiameng Fan, and Hyunjong Choi. 2020. Know the Unknowns: Addressing Disturbances and Uncertainties in Autonomous Systems. In *Proceedings of the 39th International Conference on Computer-Aided Design* (Virtual Event, USA) *(ICCAD '20)*.

[56] Qi Zhu and Alberto Sangiovanni-Vincentelli. 2018. Codesign Methodologies and Tools for Cyber–Physical Systems. *Proc. IEEE* 106, 9 (Sep. 2018), 1484–1500. https://doi.org/10.1109/JPROC.2018.2864271