Safer Policies via Affine Representations using Koopman Dynamics

Tanmay Ambadkar[1], Darshan Chudiwal[1], Greg Anderson[2], and Abhinav Verma[1]

[1] Pennsylvania State University, University Park, PA 16802, USA
{tsa5252, dsc5636, verma}@psu.edu
[2] Reed College, Portland, OR 97202, USA
grega@reed.edu

Abstract. Reinforcement learning for safety-critical tasks requires policies that are both high-performing and verifiably safe, particularly throughout the learning process. While model-predictive shielding offers a path to formal safety, existing methods are often computationally intractable for the high-dimensional, nonlinear systems where deep RL excels. We introduce SPARKD, a scalable shielding framework that overcomes this limitation. SPARKD leverages Koopman Operator theory to learn a globally linear representation of the environment's dynamics in a lifted feature space, enabling efficient formal safety analysis. We further introduce a slack-augmented shielding optimization that guarantees feasibility while enforcing strict safety on executed actions, making the framework robust to model uncertainty and long-horizon conservatism. Our experiments demonstrate that SPARKD significantly reduces safety violations compared to existing safe RL methods while maintaining strong task performance in complex control environments.

Keywords: Safe Exploration · Weakest Preconditions · Model Predictive Shielding

1 Introduction

Deep reinforcement learning (RL) has proven to be a powerful tool for controlling systems operating in complex environments and is increasingly being deployed in safety-critical applications like autonomous vehicles or industrial robotics [17]. In these domains it is critical that the controllers developed using RL techniques satisfy safety constraints. Moreover, because RL often uses real-world data for training it is often insufficient for just the final policy to be safe. Instead, we want to prioritize safe behavior even for *intermediate* policies learned during the training process. In recent years, a body of work has emerged to address this *safe exploration* problem [10, 21, 22].

One common approach to this problem is *model-predictive shielding* [3, 4, 6, 7, 15, 29]. In this general framework, a monitor is used to examine the actions proposed by a policy to determine whether they can lead to unsafe behavior. If they can, a *shield* is invoked which generates actions which are conservative but

safe. Shields and monitors may be constructed in many different ways, but can be broadly categorized as either *neural* or *symbolic*. Neural approaches (e.g., [9, 12]) use deep learning techniques to learn a function which predicts whether or not actions may lead to unsafe behavior. On the other hand, symbolic approaches (e.g., [3, 4, 8, 29]) either assume access to or construct a symbolic environment model which can be used to formally analyze the possible outcomes of potential actions.

The neural approaches are flexible because they only require the data to train a safety predictor. On the other hand, they require large amounts of data and are unable to effectively enforce safety until enough data is gathered. This leads to a relatively large number of safety violations, especially in the early stages of training. By contrast, symbolic approaches require much less training data because they work with more restricted classes of models. The tradeoff is that these models cannot handle complex behavior as precisely as the neural models. In addition, the formal analysis techniques used by symbolic approaches scale poorly with the dimension of the state space. This has historically limited their application to systems with only a handful of state variables, often four or fewer [3, 4, 29].

Our technique, SPARKD, builds on existing symbolic model predictive shielding techniques [3, 4, 6, 7, 15, 29] but enhances them with a learned lifted state space representation. The key idea is to estimate transition dynamics by learning a set of basis functions that extend the original feature space by capturing nonlinear transition behavior. This enables learning a globally linear model in a higher-dimensional space, allowing precise safety analysis within the model-predictive shielding framework. We leverage Koopman Operator Theory [11, 26] to learn both the lifted basis and the associated linear transition dynamics, which are then used by the safety monitor.

In addition, we introduce a slack-augmented quadratic programming formulation within the shielding optimization. By allowing controlled relaxation of future safety constraints while enforcing strict safety on the executed action, this formulation guarantees feasibility of the shielded optimization problem even under conservative model error bounds or long-horizon uncertainty. This design makes the shielding mechanism adaptive and robust in practice, avoiding unnecessary fallback behavior while preserving formal safety of the applied action.

SPARKD uses the simple structure of the learned lifted model for both policy optimization and safety analysis. As a result, it scales to higher-dimensional environments where prior symbolic approaches to safe exploration fail to construct effective shields. At the same time, it retains key advantages of symbolic methods, most notably sample efficiency. Compared to purely neural approaches to safe exploration, SPARKD requires significantly less data to learn an effective safety model, resulting in fewer safety violations during training.

Briefly, the contributions of this paper can be summarized as follows:

- We present SPARKD, a neurosymbolic framework for safe exploration via model predictive shielding that predicts whether current actions may lead to future safety violations,

- We leverage Koopman Operator theory to learn a globally linear representation of nonlinear transition dynamics in a lifted embedding space,
- We introduce a slack-augmented shielding optimization that guarantees feasibility while enforcing strict safety on executed actions, making the framework adaptive and robust to model uncertainty,
- Our empirical evaluation demonstrates that SPARKD learns performant policies with significantly fewer safety violations than existing neural or symbolic techniques in high-dimensional environments with complex dynamics.

2 Related Work

Work in safe reinforcement learning can be broadly categorized by what kinds of safety guarantees can be provided and when those guarantees apply. Safety is commonly defined using either a *cost-based* formulation, where each action may incur some cost and the total cost must be below some bound, or a *state-based* formulation where some regions of the state space are considered unsafe and must not be entered by the system. Our work fits into the state-based formulation. Within this framework, some work aims to formally verify that policies are safe in the worst case while other work provides high-probability guarantees based on statistical analysis. Further, some work focuses on guaranteeing the safety of the final output policy while other techniques provide guarantees on the entire learning process. In this work, we provide statistical guarantees on the entire learning process.

Several existing techniques aim to provide deterministic safety guarantees with respect to a worst-case environment model [2, 4, 5, 13, 14, 35]. In contrast to these approaches, SPARKD does not require a predefined environment model, making it easier to apply to a broad class of problems.

Because SPARKD does not assume a worst-case environment model, it cannot formally guarantee safety under all conditions. As a result, it is more closely related to a wide variety of approaches to safe RL which provide *statistical* guarantees [1, 23–25, 32, 34]. These approaches build an environment model and use a variety of different methods to generate policies which are likely to be safe with respect to that environment model. In contrast to these approaches, SPARKD generates a policy which *must* be safe with respect to the environment model, leading to fewer safety violations in practice.

SPARKD can be seen as an instance of model-predictive shielding [7]. This is a board class of approaches to safe exploration which either assume or learn an environment model and then construct a *shield* which can intervene if a policy attempts to take an unsafe action. A variety of model-predictive shielding algorithms have been proposed with different settings and different mechanisms for constructing environment models and shields [3, 4, 6, 15]. The distinguishing feature of SPARKD among these works is the use of a high-dimensional latent state-space embedding to provide a *linear* environment model. This allows SPARKD to work in higher-dimensional settings than prior work.

While we focus on state-based safety definitions, techniques which use cost-based definitions [1,16,27,28,31,33] can still be applied to this problem. Compared

to these approaches, SPARKD exhibits fewer safety violations due to a more stringent safety specification which is more amenable to formal analysis.

In summary, the current landscape of safe RL presents a challenging trade-off. On one hand, formal and symbolic methods provide strong, verifiable safety guarantees but are often computationally intractable. This struggle to scale limits their application to low-dimensional state and action spaces due to the exponential complexity of verification. On the other hand, statistical and cost-based optimization techniques are more scalable but typically learn safety through trial-and-error, often permitting numerous safety violations during the initial stages of training. Consequently, a critical open challenge is the development of a framework that can provide strong, shield-based safety assurances while remaining scalable enough to operate in the higher-dimensional and complex environments encountered in practical application

3 Preliminaries

3.1 Safe Exploration

The problem we aim to solve, *safe exploration*, may be described in terms of a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, p_0,)$ where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, $P'(x' \mid x, u)$ with $x, x' \in \mathcal{S}$ and $u \in \mathcal{A}$ is a probabilistic transition function, p_0 is an initial distribution over states. We consider a safety specifications which defines a distinguished set of unsafe states $\mathcal{S}_U$ [20]. A *policy* is a function π mapping states to distributions over actions. A policy and the environment interact to generate trajectories $s_0, a_0, s_1, a_1, \ldots, s_n \quad \text{where} \quad s_0 \sim p_0,\ a_i \sim \pi(s_i),\ s_{i+1} \sim P(s_i, a_i)$.

A policy induces a probability distributions $\mathcal{S}_t$ and $\mathcal{A}_t$ on the state and action at each time t. Given a discount factor $\gamma < 1$, the long-term return of a policy π is $R(\pi) = \mathbb{E}_{s_i, a_i \sim \pi}\Big[\sum_i \gamma^i r(s_i, a_i)\Big]$.

The goal of reinforcement learning is to find a policy π^* such that $\pi^* = \arg\max_\pi\ R(\pi)$. Most deep reinforcement learning algorithms find such a policy by generating a sequence of policies $\pi_0, \pi_1, \ldots, \pi_N$ such that $\pi_N \to \pi^*$. We refer to this sequence of policies as a *learning process*. Given a bound δ, the goal of *safe exploration* is to discover a learning process $\pi_0, \ldots, \pi_N$ such that $\pi_N = \arg\max_\pi R(\pi)$ and $\quad \forall\, 1 \le i \le N,\ \Pr_{x \sim \pi_i}(x \in \mathcal{S}_U) \le \delta$.

That is, the final policy in the sequence should have the highest return among safe policies and every policy in the sequence (except for π_0) should have a bounded probability δ of unsafe behavior. Following prior work [3, 29] this definition does not require π_0 to be safe because we assume that nothing is known about the environment *a priori*. This condition can be strengthened to include π_0 if we are given access to an initial (approximate) model of the environment dynamics, an initial dataset of trajectories, or an initial safe policy.

3.2 Model Predictive Shielding

Our approach to safe exploration is grounded in *model-predictive shielding*, a popular framework for safe reinforcement learning. Model-predictive shielding takes a neural policy π and delegates the responsibility for enforcing safety to a *shield* which typically consists of a *monitor* and a *backup policy*. At each time step, an action a^* is sampled from π. The monitor uses an model of the environment dynamics (either provided or learned) to determine whether the action a^* may lead to unsafe behavior. If it may, then the shield provides an alternative action which is guaranteed to be safe.

3.3 Safety Specifications

We define a set of states as safe or unsafe using unions of polyhedral conditions on the features defined by the state space. Specifically, we position our work among existing model-predictive approaches [3, 29], which define safe states as a union of convex polyhedra. That is, there exist matrices G_i and vectors h_i such that a state s is safe if for any i, $G_i s \leq h_i$. Such unions of convex polyhedra are sufficient to approximate any compact safe set with arbitrary precision.

4 Safer Policies via Affine Representations using Koopman Dynamics

Our approach, Safer Policies via Affine Representations using Koopman Dynamics (SPARKD), combines symbolic shielding with a lifted latent state space to provide strong safety even in complex, high-dimensional spaces. We borrow from Deep Koopman Operator with Control [26] to learn globally linear environment dynamics. We leverage the success of Koopman theory to retain constraints while introducing additional features to capture transitions. Our framework utilizes weakest precondition calculus on a learned environment model to find safe actions, a formal technique also employed by other symbolic shielding methods like SPICE. By applying weakest precondition calculus to the learned lifted space and projected safety constraint, we are able to work with much more complex environments than prior work in symbolic shielding.

SPARKD is outlined in Algorithm 1. It works by iterating through three steps: lifted space and model learning, policy optimization, and data gathering. The lifted space and associated environment model are used to construct a shield to ensure safety. The use of lifted dynamics allows SPARKD to work with higher dimensional environments than existing techniques. This is primarily because existing symbolic techniques use a learned family of linear models that become progressively complex and computationally intractable as the state and action dimensions grow. By contrast, the Koopman Operator theory is designed to support a globally linear environment model with richer features, allowing SPARKD to much more accurately enforce system safety.

Algorithm 1 SPARKD

Require: Environment $\mathcal{M}$, Safety Constraints ϕ
  Initialize policy π
  Collect initial dataset $D = (s_i, a_i, r_i, s_{i+1})$ of transitions
  for epoch in 1 to N **do**
    M = LEARNENVMODEL(D)
    Create shield π_s using π, M, ϕ
    Optimise π_s with shielding
    Collect new data (s_j, a_j, r_j, s_{j+1}) with shielded π
    $D = D \cup (s_j, a_j, r_j, s_{j+1})$
  end for

4.1 Learning Globally Linear Dynamics with Koopman Operators

A central challenge in controlling nonlinear dynamical systems is predicting their long-term behavior. Nonlinear state transition functions can quickly compound approximation errors, making long-horizon planning difficult. The Koopman operator framework offers an alternative perspective by shifting attention from the nonlinear evolution of states to the linear evolution of *observables*, which are functions that measure properties of the system state.

The Koopman Operator Consider a discrete-time dynamical system governed by

$$x_{k+1} = F(x_k),$$

where $x_k \in \mathbb{R}^n$ is the system state at time k and F is a possibly nonlinear transition function. The Koopman operator $\mathcal{K}$ is an infinite-dimensional linear operator that acts on observable functions $g : \mathbb{R}^n \to \mathbb{R}$. Its action is defined by composition with the system dynamics:

$$(\mathcal{K}g)(x) = g\big(F(x)\big).$$

Although the state evolution governed by F may be nonlinear, the Koopman operator advances observables forward in time in a linear manner. This linearity holds in an infinite-dimensional function space and is independent of the complexity of F.

Eigenfunctions and Global Linearization The key appeal of the Koopman framework lies in its spectral properties. If there exists a set of Koopman eigenfunctions $\{\psi_j\}$ satisfying

$$\mathcal{K}\psi_j = \lambda_j \psi_j,$$

for eigenvalues $\lambda_j \in \mathbb{C}$, then the system admits a global linear representation in the space spanned by these eigenfunctions. Any observable $g(x)$ can be expressed

as a linear combination of eigenfunctions, and its evolution over k time steps is given by

$$g(x_k) = (\mathcal{K}^k g)(x_0) = \sum_j c_j \lambda_j^k \psi_j(x_0).$$

This representation transforms the repeated application of a nonlinear transition function into a simple linear propagation governed by powers of eigenvalues, greatly simplifying long-horizon prediction.

Data-Driven Approximation of the Koopman Operator In practice, the Koopman operator and its eigenfunctions are rarely available in closed form. As a result, data-driven methods are used to approximate finite-dimensional representations of $\mathcal{K}$. Classical approaches such as Extended Dynamic Mode Decomposition (EDMD, [30]) project the operator onto a predefined dictionary of basis functions. More recent methods replace hand-crafted dictionaries with learned representations based on deep neural networks.

Following the framework of [26], we adopt an end-to-end learning approach that jointly learns a feature embedding and a finite-dimensional linear Koopman approximation tailored for control. The central idea is to construct a *lifted* state that augments the original system state with learned nonlinear features.

Lifted State Representation We define the lifted state z_k as

$$z_k = \begin{bmatrix} x_k \\ g_\theta(x_k) \end{bmatrix},$$

where g_θ is a neural network encoder parameterized by θ. By explicitly including the original state x_k as part of the lifted state, we ensure that the original physical variables remain directly accessible. This design choice is particularly important for control and safety, since constraints are often naturally specified in terms of the original state variables.

In contrast to the original formulation in [26], which introduces an auxiliary network to embed control inputs, we assume that the control input u_k enters the dynamics linearly. This assumption allows us to preserve the original control bounds and simplifies the subsequent optimization problems used for shielding.

Linear Dynamics in the Lifted Space In the lifted space, the system dynamics are approximated by a linear model:

$$z_{k+1} = Az_k + Bu_k,$$

where the matrices A and B are learned parameters. This linear model serves as a finite-dimensional approximation of the Koopman operator restricted to the learned feature space.

End-to-End Training Objective The encoder g_θ and the matrices A and B are trained jointly using trajectory data. The training objective is designed to encourage accurate multi-step prediction in the lifted space, thereby promoting global linearity over long horizons. Specifically, we minimize a discounted K-step prediction loss:

$$\mathcal{L} = \sum_{i=1}^{K} \gamma^{\,i-1} \left\| z_{k+i} - \hat{z}_{k+i} \right\|^2 ,$$

where $\hat{z}_{k+i}$ denotes the i-step prediction obtained by iteratively applying the learned linear dynamics, and $\gamma \in (0, 1]$ is a discount factor that places greater emphasis on near-term accuracy while still penalizing long-horizon errors.

This loss can be interpreted as combining two complementary objectives. First, it enforces accurate state prediction by penalizing deviations between the true next state x_{k+1} and the predicted state extracted from $\hat{z}_{k+1} = Az_k + Bu_k$. Second, it enforces a linearity constraint on the learned feature space by aligning the predicted lifted features with those obtained by directly encoding the true next state through $g_\theta(x_{k+1})$. Together, these effects regularize the encoder to discover features in which the dynamics are well approximated by a globally linear system.

Implications for Safe Shielding. This Koopman-based formulation is particularly well suited for the shielding framework developed in this work. Because the original state x_k is embedded directly within the lifted state z_k, safety constraints can be specified directly on the physical state variables without requiring any additional transformation into the latent space. Enforcing safety constraints on the relevant components of z_k therefore guarantees safety of the underlying system state.

At the same time, the globally linear structure of the lifted dynamics enables the use of weakest precondition reasoning and convex optimization for safety enforcement over long horizons. This combination preserves the expressiveness needed to model complex nonlinear dynamics while retaining the tractability required for real-time model predictive shielding.

4.2 Shielding via Weakest Preconditions

This section presents a unified and detailed exposition of the shielding framework based on weakest preconditions. We first derive polyhedral action constraints for convex polyhedral safety regions using a globally linear latent dynamics model. We then extend the construction to safety regions represented as unions of convex polyhedra. Finally, we describe how these constraints are used online to compute shielded actions via efficient optimization. The presentation closely follows the weakest precondition based shielding framework of [3], with additional details included for clarity and completeness.

Problem Setup and Symbolic Trajectories We consider a system operating in a latent space with states z_i and actions a_i. Using the Koopman operator, the learned environment model is globally linear and given by

$$z_{i+1} = f(z_i, a_i) + \Delta_i = Az_i + Ba_i + \Delta_i,$$

where Δ_i is a nondeterministic disturbance term representing a PAC-style bound on model error. We assume that there exists ε, δ such that with probability at least $1 - \delta$, each component of Δ_i has magnitude bounded by ε.

For the purpose of symbolic reasoning, we introduce a symbolic trajectory

$$\chi_0, \omega_0, \chi_1, \omega_1, \ldots, \chi_{H-1}, \omega_{H-1}, \chi_H,$$

where each χ_i represents a concrete latent state z_i and each ω_i represents a concrete action a_i. The symbolic dynamics are expressed as

$$\chi_{i+1} \in A\chi_i + B\omega_i + \Delta_i.$$

Convex Polyhedral Safety Constraints Let S_U denote the unsafe region. We assume first that S_U is a convex polyhedron. Its complement, the safe region, can be expressed as an intersection of half-spaces. The safety requirement over a horizon H is

$$\phi = \bigwedge_{i=1}^{H} \chi_i \notin S_U.$$

Equivalently, if S_U is represented as $Px \leq q$, the safety constraint becomes

$$\phi = \bigwedge_{i=1}^{H} P\chi_i \leq q.$$

We focus on a single conjunct $\phi_i = P\chi_i \leq q$. The weakest precondition calculus is applied by eliminating future symbolic states and expressing ϕ_i as a constraint over earlier states and actions.

Substituting the dynamics into ϕ_i yields

$$\begin{aligned}\phi_i' &= P(A\chi_{i-1} + B\omega_{i-1} + \Delta_{i-1}) \leq q \\ &= (PA)\chi_{i-1} + (PB)\omega_{i-1} + P\Delta_{i-1} \leq q.\end{aligned}$$

Repeating this backward substitution recursively eliminates $\chi_{i-1}, \chi_{i-2}, \ldots$, eventually yielding

$$\begin{aligned}(PA^i)\chi_0 + (PA^{i-1}B)\omega_0 + (PA^{i-2}B)\omega_1 + \cdots + (PB)\omega_{i-1} \\ +PA^{i-1}\Delta_0 + PA^{i-2}\Delta_1 + \cdots + P\Delta_{i-1} \leq q.\end{aligned}$$

Since χ_0 corresponds to the known current latent state z_0, it can be substituted and moved to the right-hand side:

$$\begin{aligned}(PA^{i-1}B)\omega_0 + (PA^{i-2}B)\omega_1 + \cdots + (PB)\omega_{i-1} \\ +PA^{i-1}\Delta_0 + \cdots + P\Delta_{i-1} \leq q - PA^i z_0.\end{aligned}$$

Worst-Case Noise Overapproximation Each disturbance term Δ_j is bounded component-wise by ε. Since the disturbance appears on the left-hand side of an inequality, we conservatively overapproximate its worst-case contribution. For each term $PA^{i-j-1}\Delta_j$, the maximum value is achieved by choosing the extreme values of Δ_j aligned with the signs of the coefficients. This yields the safe overapproximation

$$PA^{i-j-1}\Delta_j \;\leq\; |PA^{i-j-1}|\,\boldsymbol{\varepsilon},$$

where $|\cdot|$ denotes component-wise absolute value and $\boldsymbol{\varepsilon}$ is a vector whose components are all equal to ε.

Applying this bound to all disturbance terms results in the final action constraint

$$\begin{aligned}&(PA^{i-1}B)\omega_0 + (PA^{i-2}B)\omega_1 + \cdots + (PB)\omega_{i-1}\\ &\leq q - PA^i z_0 - \left(|PA^{i-1}| + |PA^{i-2}| + \cdots + |P|\right)\boldsymbol{\varepsilon}.\end{aligned}$$

Collecting constraints for all $i = 1, \ldots, H$ yields a polyhedral constraint of the form

$$F_0 a_0 + F_1 a_1 + \cdots + F_{H-1} a_{H-1} \leq u,$$

which constrains the sequence of actions over the horizon.

Extension to Unions of Convex Polyhedra The above derivation assumes that the safe region is defined by a single convex polyhedron. This approach extends naturally to safety constraints defined as a finite union of convex polyhedra. In this case, the safety specification becomes

$$\bigwedge_{i=1}^{H} \bigvee_{j=1}^{N} P_j \chi_i \leq q_j,$$

where each pair (P_j, q_j) defines a convex polyhedron.

To enable tractable weakest precondition reasoning, we strengthen the specification by swapping conjunction and disjunction:

$$\bigvee_{j=1}^{N} \bigwedge_{i=1}^{H} P_j \chi_i \leq q_j.$$

This modified formula requires the system to remain within a single polyhedron over the entire horizon, rather than allowing arbitrary switching at each step. Although stronger, this constraint remains safe and enables efficient computation.

For each polyhedron j, we independently apply the weakest precondition construction described above, resulting in N separate quadratic programs. An action is considered safe if it satisfies the constraints of at least one of these programs. Among all feasible solutions, we select the one that minimally deviates from the policy action.

Slack-Augmented Online Shielded Action Selection At runtime, the shield operates on the current latent state z_0 and a proposed action a_π generated by the neural policy. Using the precomputed matrices F_i and vector u, the shield computes a safe action by solving a slack-augmented quadratic program. The introduction of slack variables is a deliberate modification to the standard weakest precondition shielding formulation and is motivated by robustness and feasibility considerations.

Motivation for Slack Variables In the standard formulation, the quadratic program may become infeasible. This can occur even when a safe first action exists, for example due to conservative overapproximation of model error through large disturbance bounds or because safety cannot be guaranteed for all future steps in the horizon, even though an immediate safe action is available. In such cases, infeasibility would force the shield to trigger a backup policy unnecessarily.

To address this issue, we introduce slack variables that relax the safety constraints. This trades some conservatism for guaranteed feasibility while preserving safety of the executed action.

Slack-Augmented Optimization Problem We introduce a nonnegative slack variable $s_i \geq 0$ for each action constraint associated with a_i. Let $s = (s_0, s_1, \ldots, s_{H-1})$ denote the vector of slack variables. The slack-augmented optimization problem is given by

$$\begin{aligned}(a_0^*, a_1^*, \ldots, a_{H-1}^*, s_0^*, \ldots, s_{H-1}^*) = \arg\min \ \|a_0 - a_\pi\|_2^2 + \lambda \sum_{i=0}^{H-1} s_i \\ \text{s.t. } F_0 a_0 + F_1 a_1 + \cdots + F_{H-1} a_{H-1} \leq u + s, \\ s_i \geq 0 \quad \forall i \in \{0, \ldots, H-1\}.\end{aligned}$$

Here, λ is a large penalty coefficient. In our implementation, we set $\lambda = 10^6$, ensuring that slack usage is heavily discouraged unless strictly necessary.

The presence of slack variables guarantees that the quadratic program is always feasible. Any violation of the safety constraints is explicitly captured by a positive slack value, making constraint violations observable and quantifiable.

Safety Interpretation The resulting action sequence may violate safety constraints for future time steps if one or more slack variables s_i with $i > 0$ are strictly positive. However, the shielding objective does not require the entire action sequence to be safe. Instead, we only require that the *first* action be certifiably safe.

Accordingly, the shield enforces the condition

$$s_0^* = 0.$$

If the optimal solution satisfies $s_0^* = 0$, then the first action a_0^* respects all safety constraints without relaxation, guaranteeing that executing a_0^* is safe. The values

of s_i for $i > 0$ indicate potential future violations but do not invalidate the safety of the immediate action.

This design ensures that the shield always returns an action while avoiding unnecessary fallback behavior caused by conservative infeasibility. It also explicitly exposes uncertainty about future safety, rather than masking it through infeasibility.

Algorithm 2 Slack-Augmented Shielded Action Selection

Require: current state z_0, policy action a_π, horizon H
Assemble constraints $F_0 a_0 + \cdots + F_{H-1} a_{H-1} \leq u$
Introduce slack variables $s_0, \ldots, s_{H-1}$
Solve QP minimizing $\|a_0 - a_\pi\|_2^2 + \lambda \sum_i s_i$
Require $s_0 = 0$
$a_0 \leftarrow a_0^*$
return a_0

Algorithm By augmenting the weakest precondition based shield with slack variables, we ensure feasibility of the optimization problem at every time step while still guaranteeing safety of the executed action. This modification reduces unnecessary conservatism arising from worst-case error bounds and improves robustness of the shielding mechanism in practice.

5 Analysis

SPARKD learns an unknown transition function by estimating globally linear dynamics in a lifted space, allowing a single pair of transition matrices to define the system dynamics over the entire state space. In contrast, existing symbolic shielding and verification techniques typically rely on families of locally linear models learned directly in the original state space. These approaches often struggle in high-dimensional settings, where local linearizations fail to capture complex nonlinear behavior accurately.

By lifting the system into a higher-dimensional space and learning globally linear dynamics, SPARKD avoids the need to manage multiple local models. Imposing safety constraints directly on the original state variables, while bounding the learned basis functions, yields a framework that is both expressive and tractable. This design enables SPARKD to reduce safety violations in high-dimensional systems with nonlinear dynamics, even when long-horizon predictions are required.

This approach represents a departure from traditional formal verification paradigms. Verification methods commonly reduce dimensionality or construct coarse abstractions to preserve tractability. In contrast, SPARKD intentionally expands the state representation by learning nonlinear features that admit a single global linear model. This shift trades the complexity of reasoning over

many local abstractions for the simplicity of analyzing one higher-dimensional linear system. As a result, SPARKD can perform efficient reachability analysis in the lifted space and provide finite-horizon safety guarantees using weakest precondition reasoning.

A key advantage of SPARKD is its decoupling of safety from the learning process. The lifted state space is learned solely from environment dynamics and does not encode any specific safety constraint. Consequently, safety specifications can be imposed after training by writing polyhedral constraints over the original state variables. This enables SPARKD to handle dynamic or time-varying safety constraints without retraining. Moreover, any existing policy can be made safe by adding the shield, and offline datasets can be used to train the dynamics model, ensuring safety from the first real-world interaction.

5.1 Effect of Slack-Augmented Shielding

The introduction of slack variables into the shielding optimization further improves robustness. In the standard weakest precondition formulation, the quadratic program may be infeasible due to conservative overapproximation of model error or uncertainty about long-horizon safety, even when a safe immediate action exists. In such cases, infeasibility would unnecessarily trigger fallback behavior.

By augmenting the optimization problem with nonnegative slack variables and assigning them a large penalty, the shielding QP is guaranteed to be feasible at every time step. This modification trades strict long-horizon guarantees for feasibility, while preserving safety of the executed action. Importantly, SPARKD only requires the slack associated with the first action to be zero. When this condition holds, the executed action satisfies the safety constraints without relaxation, guaranteeing immediate safety. Positive slack values for future actions explicitly signal uncertainty or potential violations beyond the first step, rather than masking them through infeasibility.

This design choice aligns with the practical goal of shielding, which is to ensure that the action actually taken by the agent is safe. It also mitigates the impact of overly conservative error bounds, allowing SPARKD to avoid unnecessary intervention when the learned model is sufficiently accurate in the short term.

5.2 Probabilistic Safety Guarantees

The probability of unsafe behavior decreases as the accuracy of the learned environment model improves. This relationship is captured by the following theorem, adapted to the slack-augmented formulation.

Theorem 1 (Soundness with Slack-Augmented Shielding)**.** *Assume the Koopman-based dynamics model satisfies a PAC-style bound: there exist δ, ε such that*

$$\|Az_k + Bu_k - z_{k+1}\| < \varepsilon$$

with probability at least $1 - \delta$ at each step. If the solution to the slack-augmented QP satisfies $s_0 = 0$, then the executed action a_0^* maintains safety in the true

environment for the next step with probability at least $1 - \delta$*. Moreover, if all slack variables satisfy* $s_i = 0$ *for* $i = 0, \ldots, H - 1$*, then the resulting action sequence maintains safety over the entire horizon* H *with probability at least* $(1 - \delta)^H$*.*

The proof follows the weakest precondition argument presented in Theorem 1 of [3], with the additional observation that slack variables explicitly encode constraint relaxation. Enforcing $s_0 = 0$ ensures that no relaxation is applied to the first action, and thus the standard soundness argument applies to the executed action. The probability bound compounds multiplicatively over the horizon when no slack is used.

5.3 Justification of the PAC Assumption

The PAC-style assumption on the Koopman operator is supported by the following convergence result.

Theorem 2 (Convergence of Koopman Operator Approximation)**.** *Let* $\mathcal{K}$ *be the Koopman operator associated with a discrete-time dynamical system on a space* $\mathcal{M}$*. Let* $\{\psi_i\}_{i=1}^{\infty}$ *be a complete basis for a function space on* $\mathcal{M}$*. Let* K_N *be the finite-dimensional approximation of* $\mathcal{K}$ *obtained by projecting* $\mathcal{K}$ *onto the span of the first* N *basis functions. Then, for any observable function* g *in the function space,*

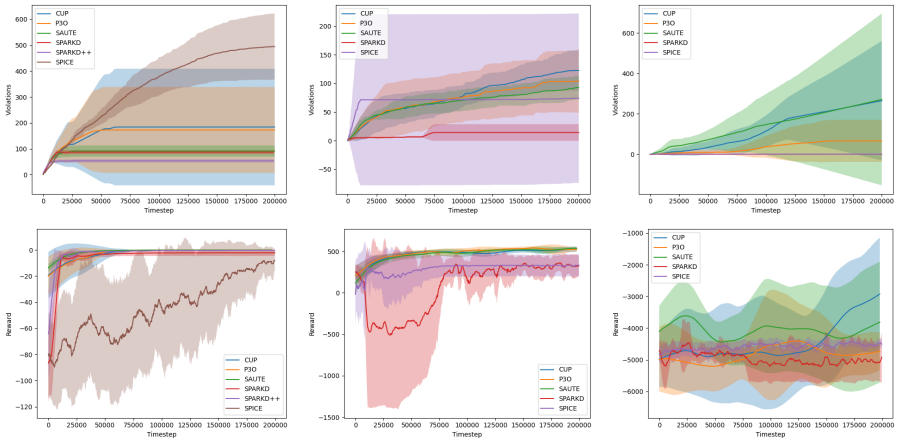
$$\lim_{N \to \infty} \|\mathcal{K}g - K_N g\| = 0.$$

This theorem provides the formal grounding for our approach. The Koopman operator yields an exact linear representation of nonlinear dynamics in an infinite-dimensional space, but practical implementations require finite-dimensional approximations. By selecting a sufficiently rich set of basis functions and increasing the dimensionality of the lifted space, the approximation error of the Koopman operator can be made arbitrarily small.

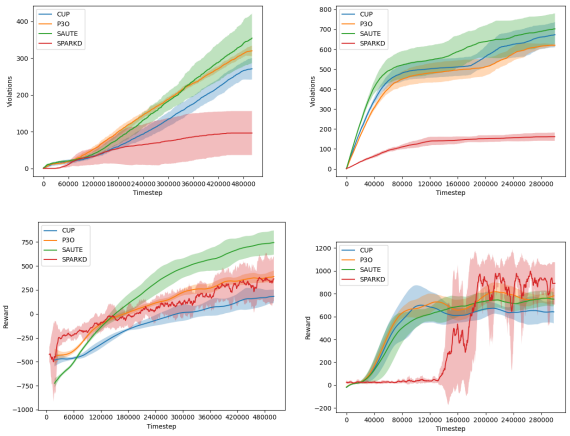
In SPARKD, the learned neural features serve as data-driven basis functions. The theorem supports the claim that a high-dimensional lifted representation is not merely heuristic but is theoretically capable of converging to the true nonlinear dynamics. As the approximation error decreases, the disturbance bound ε tightens, reducing the need for slack usage and increasing the probability that the shield enforces safety over longer horizons without relaxation. Empirically, we demonstrate that the learned model achieves sufficient accuracy within a modest computational budget, enabling SPARKD to maintain superior safety performance compared to prior work.

6 Experimental Evaluation

To evaluate the performance of SPARKD, we conducted experiments comparing it with SPICE [3], PPOSaute [27], P3O [33] and CUP [31]. For Saute and P3O, we use the implementation available in the OmniSafe-RL library [19]. The neural policy in SPARKD is trained with PPO.



(a) Pendulum (b) ACC (c) Car Racing



(d) Cheetah (e) Hopper

Fig. 1: Cumulative safety violations (top) and reward curves (bottom).

We evaluated the algorithms on five environments: acc, pendulum, car-racing, SafeHopper, and SafeCheetah. The first three experiments have been borrowed from SPICE. The last two environments have been borrowed from safety-gymnasium [18], with the safety specifications from v0 versions of the environment. We train the acc, car-racing, and pendulum benchmarks for 200K timesteps, SafeHopper and SafeCheetah for 500K timesteps, counting the cumulative safety violations over each respective training period. For all algorithms, a penalty reward of -100 was given upon entering an unsafe state to ensure all methods received a strong and unambiguous signal for safety violations. In addition, the episode was terminated upon violation. We found this to be a

Method	SafeHopper	SafeCheetah	acc	car-racing	pendulum
SauteRL	702 ± 77	354 ± 67	92 ± 20	271 ± 424	91 ± 21
CUP	673 ± 63	271 ± 29	122 ± 35	265 ± 295	184 ± 225
P3O	620 ± 6	319 ± 14	104 ± 55	66 ± 103	172 ± 165
SPICE	Failed	Failed	74 ± 147	$\mathbf{0 \pm 0}$	494 ± 127
Sparkd	$\mathbf{161 \pm 20}$	$\mathbf{96 \pm 59}$	$\mathbf{14 \pm 14}$	0.5 ± 0.71	$\mathbf{87 \pm 7}$

Table 1: Cumulative safety violations during training.

Horizon	acc		pendulum		car racing		cheetah		hopper	
	r^2	MSE	r^2	MSE	r^2	MSE	r^2	MSE	r^2	MSE
1	0.98	0.0004	0.98	0.0016	0.99	0.00006	0.89	2.57	0.98	0.029
5	0.92	0.0025	0.89	0.0095	0.96	0.0009	0.84	3.27	0.81	0.36
10	0.88	0.0071	0.80	0.0132	0.91	0.022	0.81	4.37	0.73	0.53
20	0.82	0.0228	0.70	0.0355	0.81	0.047	0.81	4.34	0.72	0.55
50	0.68	0.0190	0.37	0.0551	0.78	0.0286	0.81	4.35	0.51	0.126

Table 2: Model accuracy metrics for all environments after training for 10000 steps.

necessary modification for the Lagrangian-based baselines, which otherwise failed to learn a safe policy on these challenging benchmarks. Additionally, for P3O, PPOSaute and CUP the cost is 1 whenever a violation occurs, otherwise 0. The boolean cost for failure has been borrowed from safety-gym.

6.1 Environment Descriptions

We evaluate our approach on five distinct environments, ranging from classic control tasks to more complex locomotion challenges, to demonstrate its efficacy across varying dimensionalities and dynamics.

ACC The Autonomous Cruise Control (ACC) environment involves controlling an ego vehicle to safely follow a lead car. The state space is 2-dimensional, representing the relative distance and velocity, while the action space is 1-dimensional (acceleration of the ego vehicle). A state is designated as unsafe if the relative position becomes non-negative, which signifies a collision.

Pendulum In this classic control benchmark, the goal is to swing up and stabilize an inverted pendulum. The environment has a 3-dimensional state space (encoding the pendulum's angle and angular velocity) and a 1-dimensional action space (torque). A state is considered unsafe if the pendulum's angle $|\theta|$ exceeds 0.4 radians.

Car-Racing This environment tasks the agent with navigating a car in a 2D plane while avoiding a defined obstacle. The state space is 4-dimensional (x-y position, heading, and velocity), and the action space is 2-dimensional (acceleration and steering angle). A state is unsafe if the car enters the rectangular obstacle region defined by $1 \leq x, y \leq 2$.

SafeHopper To test our method on more complex dynamics, we use the SafeHopper environment. This task involves controlling a two-legged robot, presenting a higher-dimensional challenge with an 11-dimensional state space and a 3-dimensional action space. Safety is defined by constraints on the robot's velocity to ensure stable hopping.

SafeCheetah The SafeCheetah environment is our most complex benchmark. The agent must control a planar cheetah-like robot to run forward. This environment features a 17-dimensional state space and a 6-dimensional action space. Similar to SafeHopper, the safety specifications impose constraints on the robot's velocity to prevent unstable or dangerous movements.

6.2 Safety and Performance

The results show that SPARKD is able to maintain safety competitively during training in challenging environments. As detailed in Table **??**, SPARKD achieves the lowest number of safety violations across all five benchmarks, outperforming cost-based optimization methods (SAUTERL, CUP, P3O) and the symbolic-based SPICE. The difference is particularly stark in the more complex, higher-dimensional **SafeHopper** and **SafeCheetah** environments. Here, SPICE's underlying model fails entirely, while SPARKD reduces violations by over 70% and 74% respectively, compared to the next best algorithm. This demonstrates the effectiveness of the learned Koopman dynamics in scaling symbolic shielding to environments where prior methods were not applicable.

Performance and the Safety-Reward Tradeoff A critical aspect of safe RL is balancing safety with task performance. The reward curves in Figure 1 reveal how SPARKD navigates this tradeoff. In environments like **ACC** and **Car-Racing**, SPARKD prioritizes safety, achieving nearly zero violations. This conservative approach, driven by the shield's foresight, results in a lower average reward compared to less safe methods. The tendency of an algorithm to prioritize safety over performance is a desirable feature for safety-critical applications, where a safety violation could lead to catasrophic results. It should be noted that CUP and SAUTE obtain higher rewards in Car Racing due to early termination following safety violations. The Car Racing reward function assigns a negative reward proportional to the remaining distance to the finish at each step. As a result, terminating an episode early leads to fewer time steps accumulating this negative reward, which can artificially inflate the total return despite poorer safety performance. In case of ACC, all cost based techniques heavily prioritize reward exploitation instead of maintaining safety, thus they never learn safe behavior

(indicated by the fact that the violation curves do not plateau), compared to SPARKD which learns to maintain safety early on.

However, SPARKD is also able to find performant policies in environments where the safety constrain is better aligned with the performance objective. In **SafeCheetah**, **SafeHopper**, and **Pendulum**, it achieves both superior safety and competitive, often higher, task rewards than the other baselines. This shows that when the dynamics allow, the shield permits the policy to explore high-reward regions without compromising on safety. It can be similarly noted that that none of the baselines learn safe behavior in both environments, and prioritize reward exploitation. Thus, their violation curves keep increasing while SPARKD learns safe behavior in both cases (evidenced by the flattening curve).

To summarize, SPARKD can obey desired safety specifications during training by sacrificing some reward performance, but other techniques fail to scale well with increasing dimensions.

Impact of Model Accuracy on Shielding The performance of the shield is directly tied to the accuracy of the learned dynamics model. Table 2 shows that our Koopman-based model provides excellent short-term predictions (e.g., horizon of 1 to 5 steps) but, as expected, its accuracy degrades over longer horizons. This finding informs our choice of a planning horizon of $H = 5$ for all experiments. This represents a principled tradeoff between providing the shield with sufficient foresight to avoid future hazards and ensuring its decisions are based on a reliable model of the future. The high variance noted in the reward plots for SPICE and SPARKD is a methodological artifact, as our reward is recorded at the end of each episode, unlike the more smoothed reporting in the omnisafe library used by the other baselines.

7 Conclusion

We present SPARKD, a scalable model-predictive shielding technique that uses learned environment dynamics in a globally linear lifted space to learn performant policies while maintaining safety during training and deployment. SPARKD is a sample-efficient technique that enables safe policy learning, where actions are monitored with a robust shield to reduce safety violations. Experiments on a suite of seven environments show the efficacy of SPARKD, where it can maintain safety during training to ensure the policy never takes unsafe actions. SPARKD can be used with any existing policy with just a few real-world environment samples to learn the model and project the safety constraints to create the shield, thus making it policy-agnostic and particularly well suited for real-world deployment.

References

1. Achiam, J., Held, D., Tamar, A., Abbeel, P.: Constrained policy optimization. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. p. 22–31. ICML'17, JMLR.org (2017)
2. Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U.: Safe reinforcement learning via shielding. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 2669–2678. AAAI Press (2018), https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17211
3. Anderson, G., Chaudhuri, S., Dillig, I.: Guiding safe exploration with weakest preconditions. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net (2023), https://iclr.cc/virtual/2023/poster/12258
4. Anderson, G., Verma, A., Dillig, I., Chaudhuri, S.: Neurosymbolic reinforcement learning with formally verified exploration. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 6172–6183. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/448d5eda79895153938a8431919f4c9f-Paper.pdf
5. Bacci, E., Giacobbe, M., Parker, D.: Verifying reinforcement learning up to infinity. In: Zhou, Z.H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 2154–2160. International Joint Conferences on Artificial Intelligence Organization (8 2021). https://doi.org/10.24963/ijcai.2021/297, https://doi.org/10.24963/ijcai.2021/297, main Track
6. Banerjee, A., Rahmani, K., Biswas, J., Dillig, I.: Dynamic model predictive shielding for provably safe reinforcement learning. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), https://openreview.net/forum?id=x2zY4hZcmg
7. Bastani, O.: Safe reinforcement learning with nonlinear dynamics via model predictive shielding. In: 2021 American Control Conference (ACC). pp. 3488–3494 (2021). https://doi.org/10.23919/ACC50511.2021.9483182
8. Berkenkamp, F., Turchetta, M., Schoellig, A., Krause, A.: Safe model-based reinforcement learning with stability guarantees. Advances in neural information processing systems **30** (2017)
9. Bharadhwaj, H., Kumar, A., Rhinehart, N., Levine, S., Shkurti, F., Garg, A.: Conservative safety critics for exploration. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=iaO86DUuKi
10. Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J., Schoellig, A.P.: Safe learning in robotics: From learning-based control to safe reinforcement learning. ArXiv **abs/2108.06266** (2021), https://arxiv.org/pdf/2108.06266.pdf
11. Brunton, S.L., Budišić, M., Kaiser, E., Kutz, J.N.: Modern koopman theory for dynamical systems. arXiv preprint arXiv:2102.12086 (2021)
12. Dalal, G., Dvijotham, K., Vecerík, M., Hester, T., Paduraru, C., Tassa, Y.: Safe exploration in continuous action spaces. CoRR **abs/1801.08757** (2018), http://arxiv.org/abs/1801.08757

13. Fulton, N., Platzer, A.: Verifiably safe off-model reinforcement learning. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 413–430. Springer (2019)
14. Gillula, J.H., Tomlin, C.J.: Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor. In: IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA. pp. 2723–2730 (2012). https://doi.org/10.1109/ICRA.2012.6225136, https://doi.org/10.1109/ICRA.2012.6225136
15. Goodall, A.W., Belardinelli, F.: Approximate model-based shielding for safe reinforcement learning. In: Gal, K., Nowé, A., Nalepa, G.J., Fairstein, R., Radulescu, R. (eds.) ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023). Frontiers in Artificial Intelligence and Applications, vol. 372, pp. 883–890. IOS Press (2023). https://doi.org/10.3233/FAIA230357, https://doi.org/10.3233/FAIA230357
16. Gu, S., Sel, B., Ding, Y., Wang, L., Lin, Q., Jin, M., Knoll, A.: Balance reward and safety optimization for safe reinforcement learning: A perspective of gradient manipulation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 21099–21106 (2024)
17. Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., Yang, Y., Knoll, A.: A review of safe reinforcement learning: Methods, theory and applications. ArXiv **abs/2205.10330** (2022), https://api.semanticscholar.org/CorpusId:248965265
18. Ji, J., Zhang, B., Zhou, J., Pan, X., Huang, W., Sun, R., Geng, Y., Zhong, Y., Dai, J., Yang, Y.: Safety gymnasium: A unified safe reinforcement learning benchmark. In: Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2023), https://openreview.net/forum?id=WZmlxIuIGR
19. Ji, J., Zhou, J., Zhang, B., Dai, J., Pan, X., Sun, R., Huang, W., Geng, Y., Liu, M., Yang, Y.: Omnisafe: An infrastructure for accelerating safe reinforcement learning research. Journal of Machine Learning Research **25**(285), 1–6 (2024), http://jmlr.org/papers/v25/23-0681.html
20. Jothimurugan, K., Alur, R., Bastani, O.: A composable specification language for reinforcement learning tasks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper_files/paper/2019/file/f5aa4bd09c07d8b2f65bad6c7cd3358f-Paper.pdf
21. Jovanovi'c, M.R., Ding, D., Wei, X., Yang, Z., Wang, Z.: Provably efficient safe exploration via primal-dual policy optimization. In: International Conference on Artificial Intelligence and Statistics (2020), https://api.semanticscholar.org/CorpusId:211677570
22. Ladosz, P., Weng, L., Kim, M., Oh, H.: Exploration in deep reinforcement learning: A survey. ArXiv **abs/2205.00824** (2022), https://api.semanticscholar.org/CorpusId:247900285
23. Liu, Z., Zhou, H., Chen, B., Zhong, S., Hebert, M., Zhao, D.: Constrained model-based reinforcement learning with robust cross-entropy method (2020)
24. Ma, Y.J., Shen, A., Bastani, O., Jayaraman, D.: Conservative and adaptive penalty for model-based safe reinforcement learning (2021). https://doi.org/10.48550/ARXIV.2112.07701, https://arxiv.org/abs/2112.07701

25. Satija, H., Amortila, P., Pineau, J.: Constrained markov decision processes via backward value functions. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20, JMLR.org (2020)
26. Shi, H., Meng, M.Q.H.: Deep koopman operator with control for nonlinear systems (2022), https://arxiv.org/abs/2202.08004
27. Sootla, A., Cowen-Rivers, A.I., Jafferjee, T., Wang, Z., Mguni, D.H., Wang, J., Ammar, H.: Sauté rl: Almost surely safe reinforcement learning using state augmentation. In: International Conference on Machine Learning. pp. 20423–20443. PMLR (2022)
28. Sootla, A., Cowen-Rivers, A.I., Wang, J., Ammar, H.B.: Effects of safety state augmentation on safe exploration. arXiv preprint arXiv:2206.02675 (2022)
29. Wang, Y., Zhu, H.: Safe Exploration in Reinforcement Learning by Reachability Analysis over Learned Models, pp. 232–255 (07 2024). https://doi.org/10.1007/978-3-031-65633-0_11
30. Williams, M.O., Kevrekidis, I.G., Rowley, C.W.: A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. Journal of Nonlinear Science **25**, 1307–1346 (2015)
31. Yang, L., Ji, J., Dai, J., Zhang, L., Zhou, B., Li, P., Yang, Y., Pan, G.: Constrained update projection approach to safe policy optimization. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 9111–9124. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/3ba7560b4c3e66d760fbdd472cf4a5a9-Paper-Conference.pdf
32. Yang, T.Y., Rosca, J., Narasimhan, K., Ramadge, P.J.: Projection-based constrained policy optimization. arXiv preprint arXiv:2010.03152 (2020)
33. Zhang, L., Shen, L., Yang, L., Chen, S., Wang, X., Yuan, B., Tao, D.: Penalized proximal policy optimization for safe reinforcement learning. In: Raedt, L.D. (ed.) Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22. pp. 3744–3750. International Joint Conferences on Artificial Intelligence Organization (7 2022). https://doi.org/10.24963/ijcai.2022/520, https://doi.org/10.24963/ijcai.2022/520, main Track
34. Zhang, Y., Vuong, Q., Ross, K.: First order constrained optimization in policy space. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 15338–15349. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/af5d5ef24881f3c3049a7b9bfe74d58b-Paper.pdf
35. Zhu, H., Xiong, Z., Magill, S., Jagannathan, S.: An inductive synthesis framework for verifiable reinforcement learning. In: ACM Conference on Programming Language Design and Implementation (SIGPLAN) (2019)