SafeAdapt: Provably Safe Policy Updates in Deep Reinforcement Learning

Maksim Anisimov
Imperial College London
London, United Kingdom
m.anisimov23@imperial.ac.uk

Francesco Belardinelli
Imperial College London
London, United Kingdom
francesco.belardinelli@imperial.ac.uk

Matthew Wicker
Imperial College London
London, United Kingdom
m.wicker@imperial.ac.uk

ABSTRACT

Safety guarantees are a prerequisite to the deployment of reinforcement learning (RL) agents in safety-critical tasks. Often, deployment environments exhibit non-stationary dynamics or are subject to changing performance goals, requiring updates to the learned policy. This leads to a fundamental challenge: how to update an RL policy while preserving its safety properties on previously encountered tasks? The majority of current approaches either do not provide formal guarantees or verify policy safety only *a posteriori*. We propose a novel *a priori* approach to safe policy updates in continual RL by introducing the *Rashomon set*: a region in policy parameter space certified to meet safety constraints within the demonstration data distribution. We then show that one can provide formal, provable guarantees for arbitrary RL algorithms used to update a policy by projecting their updates onto the Rashomon set. Empirically, we validate this approach across grid-world navigation environments (Frozen Lake and Poisoned Apple) where we guarantee an *a priori* provably deterministic safety on the source task during downstream adaptation. In contrast, we observe that regularisation-based baselines experience catastrophic forgetting of safety constraints while our approach enables strong adaptation with provable guarantees that safety is preserved.

KEYWORDS

Reinforcement Learning, Continual Learning, AI Safety, Verification

1 INTRODUCTION

Reinforcement learning (RL) has achieved remarkable success in sequential decision-making, from game playing [26] to robotic control [22] and autonomous driving [19]. As RL agents move closer to real-world deployment in safety-critical domains – including autonomous vehicles, medical treatment planning, and industrial process control – it becomes essential to guarantee that learned policies satisfy safety constraints [7, 36]. At the same time, deployed agents must often operate in non-stationary settings: objectives evolve, dynamics drift, and new tasks appear. This continual adaptation introduces a central tension: policy updates aimed at improving performance often substantially degrade safety properties on previously encountered tasks.

Existing approaches to this problem fall short in important ways. Regularisation-based continual learning methods such as Elastic Weight Consolidation (EWC) [21] discourage parameter drift but do not provide formal guarantees that safety is preserved after adaptation. Shielding and action-space filtering methods [3, 12] can guarantee safety by overriding unsafe actions online, even after unrestricted policy adaptation. However, they do not certify that the adapted policy itself remains safe, they rely on a runtime intervention mechanism, and they may mask catastrophic forgetting of safe behaviour rather than prevent it. Safety-aware transfer and safe policy-update methods [11, 18] typically provide probabilistic guarantees via policy-ratio constraints or constrained optimisation, but they are unable to provide *a priori* guarantees of safety and instead require users to perform expensive evaluations to verify that updated policy satisfies safety. More broadly, safe RL methods that ensure safety during training on a single task [1, 10] do not directly address what happens when the resulting policy is later adapted to a new task. We provide the first, to our knowledge, *parameter-space certificate for source-task safety* during downstream policy adaptation in RL. In finite or discretised settings with an evaluable unsafety labelling function and greedy deployment, this certificate becomes *deterministic* and does not require full knowledge of transition dynamics.

Contributions. We propose SafeAdapt—an *a priori* approach to provably safe policy updates in continual RL. Our method leverages the Local Invariant Domain (LID) framework [13] to construct a *certified* region in policy parameter space – a *Rashomon set* – within which all policies satisfy a specified safety property on the source task. Concretely, given an unsafety labelling function that marks unsafe state–action pairs, we:

- **Formulate the notion of a *Rashomon Set* in policy parameter space** and show that computing a set of safe parameters can be posed as an optimisation problem using a sound, differentiable *safety surrogate*.
- **Compute certified Rashomon sets in parameter space** using Interval Bound Propagation (IBP) and our novel differentiable safety surrogate, ensuring that every policy inside the set satisfies the source-task safety specification under greedy deployment.
- **Perform downstream adaptation with constraints** by combining PPO updates with projected gradient descent, guaranteeing that parameter updates remain inside the certified safe region throughout training.

Empirically, we evaluate the method across grid-world navigation environments (Poisoned Apple and Frozen Lake). Our Rashomon-constrained updates preserve source-task safety after adaptation while achieving competitive downstream performance (see Section 5). Notably, we show that continual learning methods without formal guarantees (e.g. EWC) can lead to catastrophic forgetting of safe behaviour in the source task while our methods are able to at

Proc. of the Adaptive and Learning Agents Workshop (ALA 2026), Aydeniz, Delgrange, Mohammedalamen, Yang (eds.), May 25 – 26, 2026, Paphos, Cyprus, https://alaworkshop2026.github.io/. 2026.

once formally rule out this catastrophic forgetting while enabling strong adaptation to new tasks.

Paper organisation. This paper is organised as follows. Section 2 reviews related work on safe reinforcement learning, continual learning, and neural network verification. Section 3 provides background on MDPs with unsafe states, policy optimisation, and interval bound propagation. Section 4 presents the proposed method—SafeAdapt. Section 5 describes the experimental setup and results[1]. Section 6 discusses implications, limitations, and future work.

2 RELATED WORK

Our method lies at the intersection of safe reinforcement learning, continual learning, and neural network verification. We focus on *formally* certifying preservation of source-task safety during downstream adaptation in *parameter space* in the presence of an *unsafety labelling function*. We also note that when safe actions deterministically preserve safety under greedy deployment, this certification becomes deterministic. Table 1 summarises the key distinctions between our algorithm and related methods.

Table 1: Comparison of safety-aware adaptation methods. Safety structure: the type of prior safety/environment structure assumed by the method. Formal?: whether the method can provide a formal safety guarantee. Space: where the safety constraint is enforced. CL: whether the method addresses continual learning.

Method	Safety structure	Formal?	Space	CL
EWC [21]	None beyond source-task data	No	Parameters	Yes
Shielding [3]	Abstract environment model	Yes	Action	No
SPoRt [11]	Base policy + scenario-based safety data	Yes	Policy ratio	No
Zhang et al. [50]	Explicit transition dynamics model	Yes	Action/state	No
SaGui [46]	Safety guidance / exploration constraints	No	Action	No
Held et al. [18]	Learned damage model	Yes	Action	No
Berkenkamp et al. [6]	Learned dynamics model	Yes	State	No
P&C [31]	Transfer mechanism	No	Distillation	Yes
SafeAdapt (Ours)	**Unsafety labelling function $U(s, a)$**	**Yes**	**Parameters**	**Yes**

Safe Reinforcement Learning. Safe RL is commonly formalised via Constrained MDPs [4]—see García and Fernández [15], Gu et al. [17] for surveys. *Constrained policy optimisation* methods such as CPO [1], CRPO [45], and PCPO [47] guarantee near-constraint satisfaction during training but only for *individual* tasks; they do not address safety preservation when the policy is adapted to a new task. *Lyapunov-based* approaches provide safe updates at every iteration: Berkenkamp et al. [6] require a learned dynamics model (Gaussian processes), while Chow et al. [10] formulate Lyapunov constraints within a model-free primal-dual framework and operate in value space rather than parameter space. *Shielding* methods [3, 12, 14] correct unsafe actions at runtime via safety filters, typically requiring an abstract environment model for shield synthesis. In contrast, our method certifies safety in parameter space, thereby enforcing implicit safety of a neural policy which does not require any runtime intervention.

[1]Code is available at https://github.com/maxanisimov/provably-safe-policy-updates.

Continual Learning. Continual learning addresses catastrophic forgetting [27, 37] via the stability–plasticity trade-off. Regularisation methods such as EWC [21], SI [48], and MAS [2] penalise changes to important parameters but provide no formal guarantees on what knowledge the neural network retains. EWC is the most natural baseline for our approach: both methods operate in parameter space and aim to limit how far parameters move from a reference policy. However, EWC's soft quadratic penalty only *encourages* proximity to prior parameters – a sufficiently strong reward signal can override the penalty and cause arbitrary safety degradation. In contrast, our method enforces a *hard* constraint: parameters are projected onto the certified safe orthotope after every gradient step, providing a formal guarantee that no update can violate safety in a source task. Architecture and replay-based methods [23, 24, 29] address forgetting through structural or data-level mechanisms and are orthogonal to our parameter-space certification; in principle, they could be combined with our approach. In RL setting, continual methods such as Progress & Compress [31] and policy distillation [28] mitigate forgetting via knowledge transfer but provide no formal safety guarantees. Existing continual RL methods [20] do not provide formal guarantees on source-task safety preservation, and our paper addresses this gap.

Safety-Aware Policy Updates. A growing literature considers safety preservation during policy updates. SPoRt [11] bounds the violation probability of a task-specific policy using the scenario approach [9] and constrains the policy ratio $\pi_{\text{task}}(a|s)/\pi_{\text{base}}(a|s) \leq \alpha$ via a per-timestep convex projection. Its guarantees are probabilistic and the bound grows with episode length T, becoming vacuous for long horizons. Importantly, SPoRt addresses safe adaptation of a base policy to a single task – it does not consider catastrophic forgetting of safety requirements on prior tasks. Held et al. [18] formalise safe transfer via an expected damage bound, providing probabilistic guarantees that require a learned damage model. SaGui [46] provides empirical safety improvements without formal certificates. Zhang et al. [50] introduce a transfer-learning framework for safe RL that trains in a non-dangerous environment and then transfers the policy to the dangerous target system with theoretical stability and safety guarantees. Finally, Bou Ammar et al. [8] propose a lifelong policy-gradient method that learns multiple tasks online while enforcing safety constraints and achieving sublinear regret, but without any formal certificates.

Neural Network Verification. We leverage neural network verification to certify safety over parameter regions. Interval Bound Propagation (IBP) [16, 25] computes output bounds via a single forward pass but produces increasingly loose bounds with network depth, limiting the size of the certified region. Tighter methods such as CROWN [49] and α/β-CROWN [44] can be substituted at higher computational cost. While these methods focus on certifying input regions, work in Bayesian Neural Networks (BNN) introduced the notion of certifying parameter regions [40] which has also been completed for BNN policies in RL [41, 42]. Certified parameter regions were further optimised and studied by computing bounds over their derivatives [39] to proving dataset robustness [33] and privacy [34, 43]. Most closely related is the Local Invariant Domain (LID) framework of Elmecker-Plakolm et al. [13], which computes

maximal certified regions in parameter space via primal–dual optimisation over abstract domains. We adopt the LID framework and extend it to safe continual RL, where the specification is defined by an unsafety labelling function over state–action pairs and multiple safe actions per state are handled via a multi-label certificate. Prior verification in RL [5] has been used as a post-hoc check; our work uses it to define a safe region *within which* continual learning is allowed to proceed.

SafeAdapt positioning. Our proposed method SafeAdapt assumes access to an unsafety labelling function $U : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$. This is motivated by the observation that engineers can often specify forbidden state-action pairs even when the transition dynamics is unknown. Using the LID framework [13], we construct a certified safe parameter region and constrain all downstream updates to remain within it via projected gradient descent. This yields, to our knowledge, the first method that can provide formal source-task safety guarantees during downstream adaptation.

3 PRELIMINARIES

We introduce the background required for our method: Markov Decision Processes with unsafe states and policy optimisation via reinforcement learning. We refer to Sutton and Barto [35] for broader background on RL.

MDPs with Unsafe States. We model sequential decision-making as a Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu_0)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $P(s'|s, a)$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, discount factor $\gamma \in [0, 1)$, and initial state distribution μ_0. Since not all actions may be executable in every state, we also define the set of state-specific actions as $\mathcal{A}(s)$. To capture safety, the state space is partitioned into safe and *unsafe regions*, which is common in safe RL literature [17]. Intuitively, the set of *unsafe states* $\mathcal{S}_u \subseteq \mathcal{S}$ is the set of states in which the agent is experiencing a safety violation. For example, those can be hazards such as the ice holes in the Frozen Lake environment, which is used in our experiments.

Definition 3.1 (Unsafety labelling function)**.** The *unsafety labelling function* $U : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$ assigns $U(s, a) = 1$ iff $P(s' \in \mathcal{S}_u|s, a) > 0$, that is, if taking action a in state s can lead to an unsafe state in the next step, and $U(s, a) = 0$ otherwise.

Note that we use a definition of U which is *conservative* since it does not allow *any* chance that the next state will be unsafe. Using U, we define the *set of safety-critical states* and *safe action set*:

Definition 3.2 (Safety-critical states)**.** The set of *safety-critical states* is $\mathcal{S}_{\text{sc}} = \{s \in \mathcal{S} : \exists\, a \in \mathcal{A}(s) \text{ s.t. } U(s, a) = 1\}$, i.e., states at which at least one action is unsafe.

Definition 3.3 (Safe action set)**.** The *safe action set* for a state s is a set of actions which guarantee that the next state will be safe, i.e.:

$$\mathcal{A}^{\text{safe}}(s) = \{a \in \mathcal{A}(s) : U(s, a) = 0\}. \tag{1}$$

Finally, we introduce safe policies as those that return safe actions in every safety-critical state. A policy π_θ, parametrised by θ, is a function that maps states into action distributions $\pi_\theta(a|s)$.

Definition 3.4 (Safe policy)**.** A policy π is *safe* if its deterministic (i.e. greedy) deployment leads to selecting safe actions in every state where unsafe actions exist, i.e.:

$$\arg\max_a \pi(a \mid s) \in \mathcal{A}^{\text{safe}}(s)\ \forall s \in \mathcal{S}_{\text{sc}}. \tag{2}$$

Reinforcement Learning. Provided an RL problem modelled as an MDP, the standard RL objective is to find a policy maximising the expected discounted return defined as:

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t\, r(s_t, a_t)\right], \tag{3}$$

where the expectation is over trajectories $\tau = (s_0, a_0, s_1, a_1, \ldots)$ with $s_0 \sim \mu_0$, $a_t \sim \pi_\theta(\cdot \mid s_t)$, and $s_{t+1} \sim P(\cdot \mid s_t, a_t)$.

Problem statement. Given a source-task policy $\pi_{\theta_{\text{source}}}$ trained on an MDP $\mathcal{M}_{\text{source}}$, an unsafety labelling function $U_{\text{source}} : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$ for $\mathcal{M}_{\text{source}}$, and a downstream MDP $\mathcal{M}_{\text{down}}$ defined over the same state–action space but different transition dynamics, we seek an adapted policy $\pi_{\theta_{\text{down}}}$ that maximises the downstream return $J_{\text{down}}(\theta_{\text{down}})$ while provably remaining safe on the source task, i.e.,

$$\forall\, s \in \mathcal{S}_{\text{sc}} : \quad \arg\max_{a\in\mathcal{A}(s)} \pi_{\theta_{\text{down}}}(a \mid s) \ \in\ \mathcal{A}^{\text{safe}}(s). \tag{4}$$

We require this guarantee to hold *a priori* — as a certified property of the policy parameters – without full access to the source-task transition dynamics and without runtime action filtering.

Definition 3.5 (Rashomon set)**.** We define a *Rashomon set* [32] Θ_i^{safe} for a task i as a parameter space in which any policy $\pi_{\theta'}$ parameterised with $\theta' \in \Theta_i^{\text{safe}}$ is safe.

4 METHODOLOGY

We present our method SafeAdapt for safe continual reinforcement learning. The key idea is to construct a certified region in parameter space – a *Rashomon set* [13] – around a safe source-task policy, and to constrain all downstream policy updates to remain within this region via projected gradient descent. We first present the algorithm (§4.1), describe each phase (§4.2–§4.4), and then state the assumptions and theoretical guarantees (§4.5).

4.1 Algorithm Overview

Given a source-task policy π_{source} and an unsafety labelling function U_{source}, we first construct a certified safe region $\Theta_{\text{source}}^{\text{safe}}$ (a Rashomon set) in parameter space within which all policies provably satisfy the safety specification in the source task. Downstream adaptation then proceeds as a constrained optimisation: any gradient-based method may be used, provided that after each update the parameters are projected back onto $\Theta_{\text{source}}^{\text{safe}}$ via element-wise clipping.

Algorithm 1 gives the complete procedure and Figure 1 illustrates its structure. The method proceeds in three phases: (1) construct a safe demonstration dataset, (2) compute the maximal locally invariant domain (LID), and (3) adapt to the downstream task with projected gradient descent.

4.2 Phase 1: Safe Behaviour Demonstration

In supervised learning, the LID specification is evaluated on a fixed dataset independent of model parameters. In RL, the state distribution shifts with the policy, creating a circular dependency. We

Algorithm 1: SafeAdapt: Safe Continual RL via Rashomon Set Computation

Input:
(1) Source-task policy π_{source} with parameters θ_{source}
(2) Unsafety labeller $U_{\text{source}} : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$
(3) Global safety specification Φ^{safe} and its surrogate $\widetilde{\Phi}^{\text{safe}}$
(4) Global safety surrogate threshold δ
Output: Policy $\pi_{\theta_{\text{down}}}$ adapted to a downstream task that is at least as safe as π_{source} in the source task

/* **Phase 1: Safe behaviour demonstration (§4.2)** */
$S_{\text{sc}} \leftarrow \{ s \in S : \exists a \in \mathcal{A}(s) \text{ s.t. } U_{\text{source}}(s, a) = 1 \}$
$\mathcal{A}^{\text{safe}}(s) \leftarrow \{ a \in \mathcal{A}(s) : U_{\text{source}}(s, a) = 0 \}$
$D^{\text{safe}}_{\text{source}} \leftarrow \{ (s, \mathcal{A}^{\text{safe}}(s)) : s \in \mathcal{S}_{\text{sc}} \}$
/* **Phase 2: Certified safe region (§4.3)** */
if $\widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}(\pi_{\text{source}}; D^{\text{safe}}_{\text{source}}) < \delta$ **then**
  $\Theta^{\text{safe}}_{\text{source}} \leftarrow \emptyset$
  return $\perp$ ▷ method cannot compute a provably safe parameter region
$\alpha^\star \leftarrow \text{MaxLID}(\theta_{\text{source}}, \widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}, \delta)$ ▷ primal–dual + IBP
$\Theta^{\text{safe}}_{\text{source}} \leftarrow \{\theta' : \theta_{\text{source}} - \alpha^\star \le \theta' \le \theta_{\text{source}} + \alpha^\star\}$ ▷ Rashomon set
/* **Phase 3: Safe downstream adaptation (§4.4)** */
(a) Initialise the downstream policy as a source-task policy:
$\pi_{\theta_{\text{down}}} \leftarrow \pi_{\text{source}}$
(b) Solve via a projected gradient descent:

$$\theta_{\text{down}} = \arg\max_{\theta \in \Theta^{\text{safe}}_{\text{source}}} J_{\text{down}}(\theta), \tag{5}$$

return $\pi_{\theta_{\text{down}}}$

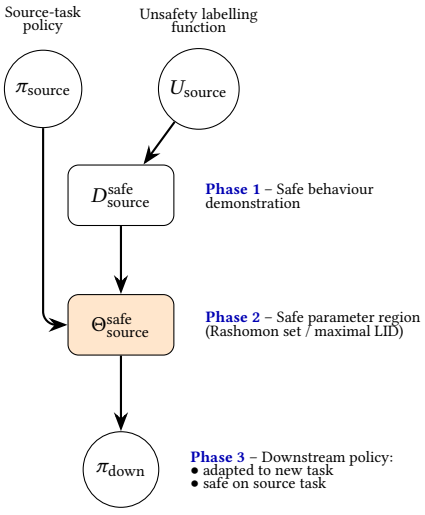

Figure 1: Overview of the proposed method. The unsafety labelling function U_{source} yields safe demonstrations $D^{\text{safe}}_{\text{source}}$ (Phase 1). A certified safe parameter region $\Theta^{\text{safe}}_{\text{source}}$ is computed around π_{source} (Phase 2), and the downstream policy π_{down} is adapted within this region (Phase 3).

resolve this by constructing a *distribution-independent* safety constraint: using U_{source}, we list all safe state–action pairs and form the

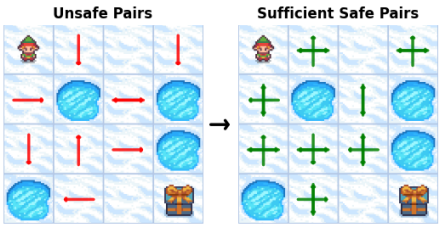

Figure 2: Safe demonstration dataset construction in Frozen Lake environment. We label state-action pairs that lead to ice holes as unsafe. Then, we derive a demonstration of safe state-action pairs, which exclude states without unsafe actions.

safe demonstration dataset:

$$D^{\text{safe}}_{\text{source}} = \left\{ (s, \mathcal{A}^{\text{safe}}(s)) : s \in \mathcal{S}_{\text{sc}} \right\}. \tag{6}$$

States where all actions are safe are excluded from the safety demonstration dataset due to redundancy. By enforcing safety at every state in $D^{\text{safe}}_{\text{source}}$, we guarantee safety along any trajectory that an updated policy can have in the source task.

4.3 Phase 2: Certified Safe Region

We compute the maximal $\Theta^{\text{safe}}_{\text{source}}$ centred at θ_{source}: the largest orthotope in parameter space within which policies are safe on $D^{\text{safe}}_{\text{source}}$. Note that if π_{source} does not satisfy the safety specification, the Rashomon set is empty and the algorithm terminates.

Rashomon set computation. We define the safe region as a centre-symmetric orthotope $\{\theta' : \theta_{\text{source}} - \alpha \le \theta' \le \theta_{\text{source}} + \alpha\}$ with half-widths $\alpha \in \mathbb{R}^p_{\ge 0}$. We maximise its volume subject to the lower bound of the safety surrogate inside the orthotope:

$$\max_{\alpha \ge 0} \sum_{i=1}^{p} \log \alpha_i \quad \text{s.t.} \quad \min_{\alpha \ge 0} \widetilde{\Phi}(\theta_{\text{source}}, \alpha) \ge \delta, \tag{7}$$

solved via the primal–dual algorithm [13].

Safety Specification: Hard vs Surrogate. We distinguish between a *hard* (exact) safety specification, which is exact but non-differentiable, and a *soft* (surrogate) specification, which is differentiable but only provides a one-sided guarantee.

Definition 4.1 (Hard safety specification)**.** A policy satisfies the hard safety specification at state s if $\phi^{\text{safe}}(s) = 1$, where:

$$\phi^{\text{safe}}(s) := \mathbb{I}\left\{ \arg\max_{a \in \mathcal{A}(s)} z_{\mathcal{A}}(s) \in \mathcal{A}^{\text{safe}}(s) \right\}, \tag{8}$$

assuming no ties at the maximum.

The hard specification is:

- **sound and complete**: it exactly captures safety,
- **non-differentiable**: due to the arg max and indicator.

We want to verify that a policy is safe in any safety-critical state:

$$\phi^{\text{safe}}(s) = 1 \quad \forall s \in \mathcal{S}_{\text{sc}}.$$

To enable optimisation, we introduce a smooth *safety surrogate*:

Definition 4.2 (Safety surrogate)**.** A safety surrogate is $\widetilde{\Phi}^{\text{safe}}_{\tau}(s)$:

$$\widetilde{\phi}^{\text{safe}}_{\tau}(s) := \sum_{a \in \mathcal{A}^{\text{safe}}(s)} \pi(a|s;\tau), \tag{9}$$

where

$$\pi(a|s;\tau) = \frac{e^{z_{\mathcal{A}}(s)/\tau}}{\sum_{a' \in \mathcal{A}(s)} e^{z_{a'}(s)/\tau}}, \quad \tau > 0.$$

Proposition 1 (Properties of the surrogate)**.** Assume no ties at the maximum. Then:

(1) **Differentiability:** $\widetilde{\phi}^{\text{safe}}_{\tau}(s)$ is smooth in $z_{\mathcal{A}}(s)$ for all $\tau > 0$.
(2) **Consistency:**

$$\lim_{\tau \to 0^+} \widetilde{\phi}^{\text{safe}}_{\tau}(s) = \phi^{\text{safe}}(s).$$

(3) **Soundness (one-sided guarantee):** for all $\tau > 0$,

$$\widetilde{\phi}^{\text{safe}}_{\tau}(s) > \frac{|\mathcal{A}^{\text{safe}}(s)|}{1 + |\mathcal{A}^{\text{safe}}(s)|} \quad \Rightarrow \quad \phi^{\text{safe}}(s) = 1.$$

(4) **Non-completeness (for finite τ):**

$$\widetilde{\phi}^{\text{safe}}_{\tau}(s) \leq \frac{|\mathcal{A}^{\text{safe}}(s)|}{1 + |\mathcal{A}^{\text{safe}}(s)|} \quad \not\Rightarrow \quad \phi^{\text{safe}}(s) = 0.$$

The surrogate provides a *sufficient but not necessary* condition for safety: high surrogate values certify safety, but low values are inconclusive. Therefore, we impose the following *sound* constraint:

$$\widetilde{\phi}^{\text{safe}}_{\tau}(s) > \frac{|\mathcal{A}^{\text{safe}}(s)|}{1 + |\mathcal{A}^{\text{safe}}(s)|}. \tag{10}$$

Global Safety Specification. We now lift the specification from states to policies.

Definition 4.3 (Critical State Safety Rate)**.** Define a critical state safety rate as a proportion of safety-critical states in which the policy is safe:

$$\Phi^{\text{safe}}_{\text{sc}}(\pi) = \frac{1}{|\mathcal{S}_{\text{sc}}|} \sum_{s \in \mathcal{S}_{\text{sc}}} \phi^{\text{safe}}(s).$$

Definition 4.4 (Trajectory Safety Rate)**.** Define a trajectory safety rate as an expected proportion of episodes in which the policy trajectory does not experience any unsafe state-action pairs:

$$\Phi^{\text{safe}}_{\text{traj.}}(\pi) = \mathbb{E}_{s \sim d^{\pi}} \left[\phi^{\text{safe}}(s) \right],$$

where $d^{\pi}(s)$ is the state visitation distribution induced by π.

Global Safety Surrogate. During optimisation, we replace the hard global specification with a smooth lower bound.

Definition 4.5 (Critical State Safety Surrogate)**.** The critical state safety surrogate is defined as the minimum per-state safety surrogate in safety-critical states:

$$\widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}(\pi) = \min_{s \in \mathcal{S}_{\text{sc}}} \widetilde{\phi}^{\text{safe}}_{\tau}(s).$$

Definition 4.6 (Sound global bound)**.** Let

$$M := \max_{s \in \mathcal{S}_{\text{sc}}} |\mathcal{A}^{\text{safe}}(s)|.$$

Then

$$\widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}(\pi) > \frac{M}{1+M} \quad \Rightarrow \quad \Phi^{\text{safe}}_{\text{sc}}(\pi) = 1.$$

Key implication. The surrogate enables gradient-based optimisation while preserving a *global safety certificate*: if the global surrogate constraint is satisfied with $\delta^{\star} = \frac{M}{1+M}$, the global hard safety specification constraint $\Phi_{sc}(\pi) = 1$ is satisfied as well.

4.4 Phase 3: Safe Downstream Adaptation

We adapt to the downstream task by solving $\max_{\theta \in \Theta^{\text{safe}}_{\text{source}}} J_{\text{down}}(\theta)$ via a gradient-based policy optimisation method (e.g. PPO) with an additional projection step. After each gradient step $\hat{\theta} = \theta + \eta\, g$, we project back via element-wise clipping:

$$\theta_{\text{down}} \leftarrow \text{clip}\big(\hat{\theta},\ \ \theta_{\text{source}} - \alpha^{*},\ \ \theta_{\text{source}} + \alpha^{*}\big). \tag{11}$$

4.5 Assumptions and Theoretical Guarantees

We now state the assumptions under which our safety guarantees hold.

Assumption 1 (Finite discrete state–action space)**.** The state space $\mathcal{S}$ and action space $\mathcal{A}$ are discrete and finite. This enables using unsafety labelling function U_{source} to generate the sufficient safety demonstration dataset $D^{\text{safe}}_{\text{source}}$ for the source task.

Assumption 2 (Existence of safe actions)**.** For every safety-critical state, there exists at least one safe action: $\mathcal{A}^{\text{safe}}(s) \neq \emptyset\ \forall s \in \mathcal{S}_{\text{sc}}$. Otherwise, the method cannot guarantee safe behaviour in any state.

Assumption 3 (Safe source policy)**.** The source-task policy $\pi_{\theta_{\text{source}}}$ satisfies the sound safety surrogate constraint at the required threshold:

$$\widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}(\pi) > \frac{M}{1+M}.$$

This enables building a Rashomon set, which is convex by design.

Assumption 4 (Greedy action selection)**.** At deployment, the agent selects actions greedily, i.e. $a^{\star} = \arg\max_{a' \in \mathcal{A}(s)} \pi_{\theta}(a'|s)$. This is a standard assumption about test-time policy deployment which also avoids formal analysis of stochastic sampling from $\pi_{\theta}(\cdot \mid s)$.

Assumption 1 restricts the formulation to discretised state-action spaces, ensuring that the safe demonstration dataset $D^{\text{safe}}_{\text{source}}$ can be constructed exhaustively. We note that this does not require manual enumeration: in practice, the unsafety labelling function is specified as a computable predicate grounded in domain knowledge (e.g., "if adjacent to a hole, moving toward it is unsafe"), and the dataset is constructed programmatically. Assumption 2 excludes states from which failure is unavoidable under any policy. Assumption 3 is a necessary precondition: no certified convex region can exist around a policy that is itself unsafe. Algorithm 1 checks this explicitly and returns $\perp$ if the condition is violated. Assumption 4 avoids dealing with stochastic action sampling when an unsafe action can be drawn even if the greedy action is safe.

Theorem 1 (Provably safe policy updates)**.** Let Assumptions 1–4 hold, and let

$$\Theta^{\text{safe}}_{\text{source}} = \{\theta' : \theta_{\text{source}} - \alpha^{\star} \leq \theta' \leq \theta_{\text{source}} + \alpha^{\star}\}$$

be the certified orthotope returned by Phase 2 of Algorithm 1. Let

$$M := \max_{s \in \mathcal{S}_{\text{sc}}} |\mathcal{A}^{\text{safe}}(s)|, \qquad \delta^{\star} := \frac{M}{1+M}.$$

Assume that the verification procedure used in Phase 2 provides a sound lower bound for the safety surrogate $\widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}$ such that

$$\min_{\theta' \in \Theta^{\text{safe}}_{\text{source}}} \widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}(\pi_{\theta'}) > \delta^{\star}.$$

Then, $\forall\ \theta' \in \Theta^{\text{safe}}_{\text{source}}$,

$$\Phi^{\text{safe}}_{\text{sc}}(\pi_{\theta'}) = 1.$$

In particular, if Phase 3 updates the policy by projected gradient descent onto $\Theta^{\text{safe}}_{\text{source}}$, then in every iteration θ_t satisfies

$$\Phi^{\text{safe}}_{\text{sc}}(\pi_{\theta_t}) = 1.$$

Proof. Take any $\theta' \in \Theta^{\text{safe}}_{\text{source}}$ and any $s \in \mathcal{S}_{\text{sc}}$. By Definition 4.5,

$$\widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}(\pi_{\theta'}) = \min_{s' \in \mathcal{S}_{\text{sc}}} \widetilde{\phi}^{\text{safe}}_{\tau}(s'; \pi_{\theta'}),$$

hence

$$\widetilde{\phi}^{\text{safe}}_{\tau}(s; \pi_{\theta'}) \geq \widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}(\pi_{\theta'}).$$

Because the verification lower bound is sound over $\Theta^{\text{safe}}_{\text{source}}$,

$$\widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}(\pi_{\theta'}) \geq \min_{\theta \in \Theta^{\text{safe}}_{\text{source}}} \widetilde{\Phi}^{\text{safe}}_{\tau,\text{sc}}(\pi_{\theta}) > \delta^{\star} = \frac{M}{1+M}.$$

Since $|\mathcal{A}^{\text{safe}}(s)| < M$ and the map $x \mapsto x/(1+x)$ is increasing on $[0, \infty)$,

$$\frac{M}{1+M} > \frac{|\mathcal{A}^{\text{safe}}(s)|}{1+|\mathcal{A}^{\text{safe}}(s)|}.$$

Therefore

$$\widetilde{\phi}^{\text{safe}}_{\tau}(s; \pi_{\theta'}) > \frac{|\mathcal{A}^{\text{safe}}(s)|}{1+|\mathcal{A}^{\text{safe}}(s)|}.$$

By Proposition 1(3), this implies

$$\phi^{\text{safe}}(s; \pi_{\theta'}) = 1.$$

Since this holds for every $s \in \mathcal{S}_{\text{sc}}$, Definition 4.3 yields

$$\Phi^{\text{safe}}_{\text{sc}}(\pi_{\theta'}) = 1.$$

Finally, Phase 3 projects each update back into $\Theta^{\text{safe}}_{\text{source}}$, so every iterate θ_t remains in $\Theta^{\text{safe}}_{\text{source}}$; applying the same argument to each θ_t proves the second claim. □

Corollary 1 (Per-state certified safety)**.** Under the assumptions of Theorem 1, for every $\theta' \in \Theta^{\text{safe}}_{\text{source}}$ and every $s \in \mathcal{S}_{\text{sc}}$,

$$\arg\max_{a \in \mathcal{A}(s)} \pi_{\theta'}(a \mid s) \in \mathcal{A}^{\text{safe}}(s).$$

Corollary 2 (Safety preservation throughout adaptation)**.** Let $(\theta_t)_{t \geq 0}$ be the sequence of policy parameters produced by Phase 3. Then $\theta_t \in \Theta^{\text{safe}}_{\text{source}}$ for all t, and therefore

$$\Phi^{\text{safe}}_{\text{sc}}(\pi_{\theta_t}) = 1 \qquad \forall t \geq 0.$$

Corollary 3 (Distribution-independent source-task safety)**.** Under the assumptions of Theorem 1, greedy execution of π_{θ_t} on the source task never selects an unsafe action at any iteration t. Consequently, for any initial state distribution μ_0, the source-task occupancy measure of π_{θ_t} satisfies

$$\mathbb{E}_{(s,a) \sim d^{\pi_{\theta_t}}}[U_{\text{source}}(s,a)] = 0.$$

Note that Theorem 1 guarantees preservation of source-task safety, but it does not guarantee that either source or downstream goal-reaching policy exists inside the certified region. Whether such a policy exists depends on the overlap between the certified safe set and high-performing source-task and downstream-task policies.

5 EXPERIMENTS

We consider continual adaptation from a source task (Task 1) to a downstream task (Task 2) in discrete-state, discrete-action environments with known unsafe state-action pairs. In all experiments, π_{source} is safe on the source-task safety dataset before adaptation. In our finite-state settings, this corresponds to full safety coverage of the source safety-critical states. We compare four policies:

- **Source**: train the policy on the source task and without updating parameters to the downstream task.
- **UnsafeAdapt**: unconstrained PPO fine-tuning on Task 2.
- **EWC**: PPO fine-tuning on Task 2 with EWC regularisation.
- **SafeAdapt (ours)**: PPO fine-tuning on Task 2 with per-update projection onto the Rashomon set, which is a parameter set certified to be safe in the source task.

Policy training. In each experiment run, policies are trained with PPO [30] using the same MLP actor and critic with softmax over action logits. This ensures that the expressiveness of the neural policy is the same across methods. We run 10 seeds and report mean ± standard deviation. Detailed settings are specified in Tables 6-9.

Metrics. For each environment and policy π, we calculate the following metrics for comparison:

- **Critical-state safety rate** $\Phi^{\text{safe}}_{\text{sc}}(\pi)$: fraction of safety-critical states in which the policy is safe.
- **Trajectory safety rate** $\Phi^{\text{safe}}_{\text{traj.}}(\pi)$: fraction of episodes in which the policy trajectory is safe.
- **Total reward**: Episodic return of the policy.
- **Success rate**: environment-defined task completion rate.

In each experiment run, our method SafeAdapt is the only method that provides a formal guarantee that $\Phi^{\text{safe}}_{\text{sc}}(\pi) = 1$ in a source task.

5.1 Environments

Frozen Lake. Frozen Lake is a grid-world Gymnasium environment [38] where the agent must reach the goal while avoiding holes in a frozen lake. State is represented with a one-hot encoded location and task indicator, and there are 4 actions available (go up, down, left, or right). Reward is +1 at goal and 0 otherwise. Task 1 and Task 2 differ by hole placement. Falling into a hole is unsafe, and an episode is terminated upon this event.

Poisoned Apple. Poisoned Apple is a custom grid world in which an agent's goal is to collect safe apples while avoiding poisoned ones. Collecting a safe apple yields +1, collecting a poisoned apple yields −1 and is unsafe. A state is represented as flattened grid with locations of safe and poisoned apples indicated therein. There are 4 available actions: up, down, left and right. Task 1 and Task 2 differ in safe/poisoned apple placement. A trajectory is safe if no poisoned apple is collected. Once all safe apples are collected, the environment terminates.

Figure 3 demonstrates the environments and their corresponding source and downstream tasks. Table 2 showcases that we have a setup which tests SafeAdapt across heterogeneous environments. Frozen Lake is a Task-Incremental Learning (TIL) setting, since the task ID is included in the state and the policy can condition

explicitly on task context. In contrast, Poisoned Apple is a Domain-Incremental Learning (DIL) setting, where adaptation must occur without explicit task identity, making it a harder regime than TIL. We use small discrete environments because they permit a straightforward construction of the safety-critical-state dataset and end-to-end validation of the deterministic certificate under the assumptions of Theorem 1.

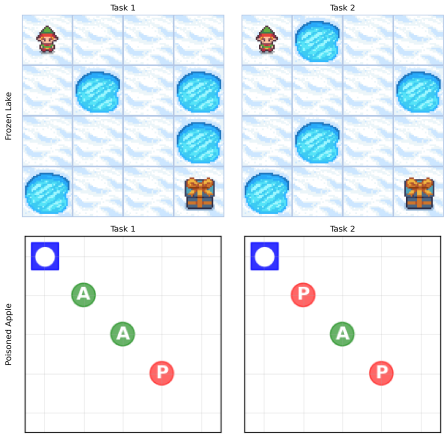

Figure 3: Examples of experiment environments: Frozen Lake `standard_4x4` (first row) and Poisoned Apple `simple_5x5` (second row).

Table 2: Frozen Lake vs Poisoned Apple experiment structure.

Setting	Frozen Lake	Poisoned Apple
Task ID in state representation	Yes	No
Termination when unsafe	Yes	No
Single destination	Yes	No

5.2 Results

Safety retention in the source task. Table 3 demonstrates how our method (SafeAdapt) retains agent's safety in the source task after the update to a new task. Out of all adaptation methods, only SafeAdapt has the average critical state safety rate $\Phi_{\text{sc}}^{\text{safe}}(\pi)$ of 1. That is, UnsafeAdapt and EWC methods *forget* how to be safe in some safety-critical states. The average trajectory safety rate $\Phi_{\text{traj.}}^{\text{safe}}(\pi)$ of SafeAdapt agent is 1, which follows from SafeAdapt's safety in all safety-critical states. UnsafeAdapt and EWC agents have unsafe policy trajectories in the source task in Frozen Lake environment, while their trajectories are safe in Poisoned Apple. Finally, the Total Reward column demonstrates degradation of agent's performance in the source task across all adaptation methods. However, due to the nature of Frozen Lake and Poisoned Apple, safety retention allows the SafeAdapt to maintain competitive performance in the source task compared to other adaptation methods.

Adaptation to downstream task. Table 4 shows that SafeAdapt retains substantial downstream plasticity despite the hard source-task safety constraint. In Frozen Lake, all three adaptation methods achieve optimal downstream performance, so the relevant distinction is not Task-2 total reward but whether that reward is obtained without sacrificing source-task safety. In this respect, SafeAdapt is strictly preferable, as it matches the best downstream performance while being the only method with a source-task safety certificate. In Poisoned Apple, SafeAdapt again matches the best-performing baseline, achieving the same average total reward as UnsafeAdapt (0.96) and outperforming EWC (0.86). Overall, these results show that constraining updates to the certified Rashomon set can allow for effective downstream adaptation and can preserve source-task safety without sacrificing downstream performance.

Scalability analysis. To study scalability, we evaluate how SafeAdapt behaves as the Frozen Lake layout size increases. Figure 4 shows a stability–plasticity trade-off across scales. The source policy has perfect Task-1 safety and total reward, but fails completely on Task 2. UnsafeAdapt and EWC exhibit strong downstream performance, yet incur substantial degradation on the source task. Namely, UnsafeAdapt exhibits a large drop in critical-state safety and zero trajectory safety rate, while EWC retains partial source-task safety and near-zero Task-1 total reward on the larger layouts. In contrast, SafeAdapt maintains perfect source-task safety across all sizes and remains competitive on the downstream task, with only a modest drop at 8×8. To keep the comparison controlled, we use the same policy architecture and training configuration across all Frozen Lake layouts. Larger layouts may benefit from increased model capacity and optimised training settings. Exploring scaling of the environment, network architecture, and training procedure is a potential direction for future work.

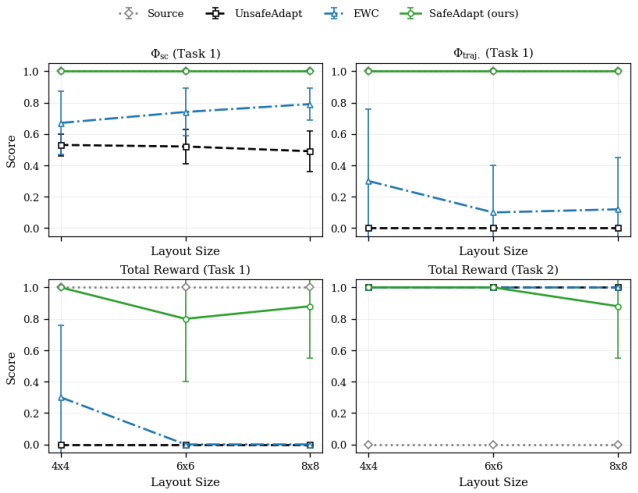

Figure 4: Scalability analysis across diagonal Frozen Lake layouts.

Appendix D.2 visualises the Rashomon set for Frozen Lake. We leave analysis of certified-region size and its dependence on neural network architecture to future work.

Table 3: Source Task safety and performance metrics in Frozen Lake and Poisoned Apple (mean ± standard deviation over 10 seeds). We report (i) critical state safety rate $\Phi^{\text{safe}}_{\text{sc}}(\pi)$, measuring the fraction of safety-critical states in which the policy selects only safe actions, (ii) trajectory-level safety $\Phi^{\text{safe}}_{\text{traj.}}(\pi)$, measuring the probability of executing a fully safe trajectory, and (iii) total reward. Standard continual learning baselines (UnsafeAdapt and EWC) may violate safety constraints, as reflected by degraded $\Phi^{\text{safe}}_{\text{sc}}$ in both Frozen Lake and Poisoned Apple and degraded $\Phi^{\text{safe}}_{\text{traj.}}$ in Frozen Lake. In contrast, SafeAdapt (ours) consistently retains perfect safety ($\Phi^{\text{safe}}_{\text{sc}} = \Phi^{\text{safe}}_{\text{traj.}} = 1$) across all environments while maintaining competitive total reward in the source task.

Environment	Policy	Provably Safe?	$\Phi^{\text{safe}}_{\text{sc}}(\pi)$	$\Phi^{\text{safe}}_{\text{traj.}}(\pi)$	Total Reward
Frozen Lake (`standard_4x4`)	Source	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	UnsafeAdapt	×	0.88 ± 0.00	0.10 ± 0.30	0.00 ± 0.00
	EWC	×	0.94 ± 0.06	0.60 ± 0.49	0.30 ± 0.46
	SafeAdapt (ours)	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	0.90 ± 0.30
Poisoned Apple (`simple_5x5`)	Source	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.96 ± 0.00**
	UnsafeAdapt	×	0.93 ± 0.11	**1.00 ± 0.00**	0.91 ± 0.00
	EWC	×	0.93 ± 0.11	**1.00 ± 0.00**	1.02 ± 0.32
	SafeAdapt (ours)	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	0.91 ± 0.00

Table 4: Downstream Task performance metrics in Frozen Lake and Poisoned Apple (mean ± standard deviation over 10 seeds).

Environment	Policy	Total Reward	Success Rate
Frozen Lake (`standard_4x4`)	Source	0.00 ± 0.00	0.00 ± 0.00
	UnsafeAdapt	**1.00 ± 0.00**	**1.00 ± 0.00**
	EWC	**1.00 ± 0.00**	**1.00 ± 0.00**
	SafeAdapt (ours)	**1.00 ± 0.00**	**1.00 ± 0.00**
Poisoned Apple (`simple_5x5`)	Source	−0.04 ± 0.00	**1.00 ± 0.00**
	UnsafeAdapt	**0.96 ± 0.00**	**1.00 ± 0.00**
	EWC	0.86 ± 0.30	**1.00 ± 0.00**
	SafeAdapt (ours)	**0.96 ± 0.00**	**1.00 ± 0.00**

6 CONCLUSION

We introduced SafeAdapt, an approach to provably safe policy updates in continual deep reinforcement learning. The key idea is to compute a certified Rashomon set in policy parameter space around a safe source-task policy and to constrain downstream adaptation to remain within this set via projected gradient descent. Our method yields an *a priori* guarantee that source-task safety is preserved throughout downstream adaptation, rather than only being checked *a posteriori* or enforced by a runtime intervention mechanism.

Empirically, we showed across grid-world environments Frozen Lake and Poisoned Apple that SafeAdapt is the only adaptation method in our study that preserves perfect source-task safety after updates to a downstream task, while exhibiting competitive downstream performance. In contrast, unconstrained adaptation (UnsafeAdapt) and EWC can both exhibit catastrophic forgetting of safe behaviour in a source task while adapting to a new task. The scalability study further suggests that constrained adaptation within the safe region may exhibit trade-off between safety retention and downstream plasticity as environment size increases.

The current formulation is limited to finite discrete state–action spaces as it relies on exhaustive coverage of safety-critical states to obtain safety guarantees. In addition, the certified parameter region may become conservative when using IBP, especially for larger networks or more complex environments. Extending the framework to infinite and continuous state-action spaces and understanding how certified regions behave in larger environments and across longer task sequences are important directions for future work. Another interesting idea is to derive the unsafety labelling function from a synthesised shield, which could extend our framework to more expressive multi-step safety specifications. Finally, probabilistic verification is a complementary direction for more complex settings where deterministic parameter-space certification becomes too conservative or intractable.

Overall, our results show that parameter-space certification is a viable route to preventing safety forgetting in continual RL and provide, to our knowledge, the first parameter-space certificate for preserving source-task safety during downstream adaptation.

ACKNOWLEDGMENTS

This work was supported by the UKRI Centre for Doctoral Training in Safe and Trusted AI [EP/S0233356/1] and the EPSRC grant number EP/X015823/1, "An abstraction-based Technique for Safe Reinforcement Learning".

REFERENCES

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. https://proceedings.mlr.press/v70/achiam17a.html

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory Aware Synapses: Learning What (Not) to Forget. In *European Conference on Computer Vision*. 139–154.

[3] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Mykel J. Kochenderfer, Scott Niekum, and Ufuk Topcu. 2018. Safe Reinforcement Learning via Shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. https://ojs.aaai.org/index.php/AAAI/article/view/11573

[4] Eitan Altman. 1999. *Constrained Markov Decision Processes*. CRC Press.

[5] Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. 2018. Verifiable Reinforcement Learning via Policy Extraction. In *Advances in Neural Information Processing Systems*, Vol. 31.

[6] Felix Berkenkamp, Matteo P. Turchetta, Angela P. Schoellig, and Andreas Krause. 2017. Safe Model-based Reinforcement Learning with Stability Guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper/2017/hash/3a1dd3879d8c38fefcd1b582f44f0f42-Abstract.html

[7] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

[8] Haitham Bou Ammar, Rasul Tutunov, and Eric Eaton. 2015. Safe Policy Search for Lifelong Reinforcement Learning with Sublinear Regret. In *International Conference on Machine Learning*. PMLR, 2361–2369.

[9] Marco C. Campi and Simone Garatti. 2008. The Exact Feasibility of Randomized Solutions of Uncertain Convex Programs. *SIAM Journal on Optimization* 19, 3 (2008), 1211–1230.

[10] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. 2018. Lyapunov-based Safe Policy Optimization for Continuous Control. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. https://ojs.aaai.org/index.php/AAAI/article/view/11682

[11] Jacques Cloete, Niklas Vertovec, and Kostas Margellos. 2025. SPoRt – Safe Policy Ratio: Certified Training and Deployment of Task Policies in Model-Free RL. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[12] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. 2018. Safe Exploration in Continuous Action Spaces. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. https://proceedings.mlr.press/v80/dalal18a.html

[13] Leo Elmecker-Plakolm, Pierre Fasterling, Philip Sosnin, Calvin Tsay, and Matthew Wicker. 2025. Provably Safe Model Updates. *arXiv preprint arXiv:2512.01899* (2025). Submitted to IEEE SaTML 2026.

[14] Ingy ElSayed-Aly, Suda Bharadwaj, Christopher Amato, Rüdiger Ehlers, Ufuk Topcu, and Lu Feng. 2021. Safe Multi-Agent Reinforcement Learning via Shielding. In *Proceedings of the 20th International Conference on Autonomous Agents and Multi-Agent Systems*.

[15] Javier García and Fernando Fernández. 2015. A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research* 16 (2015), 1437–1480.

[16] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2018. On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models. *arXiv preprint arXiv:1810.12715* (2018).

[17] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. 2024. A Review of Safe Reinforcement Learning: Methods, Theories and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[18] David Held, Zoe McCarthy, Michael Zhang, Fred Shentu, and Pieter Abbeel. 2017. Probabilistically Safe Policy Transfer. In *IEEE International Conference on Robotics and Automation (ICRA)*. 5798–5805.

[19] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. 2019. Learning to Drive in a Day. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*. IEEE, 8248–8254. https://doi.org/10.1109/ICRA.2019.8793742

[20] Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. 2022. Towards Continual Reinforcement Learning: A Review and Perspectives. *Journal of Artificial Intelligence Research* 75 (2022), 1401–1476.

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwińska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* 114, 13 (2017), 3521–3526.

[22] Jens Kober, J. Bagnell, and Jan Peters. 2013. Reinforcement Learning in Robotics: A Survey. *The International Journal of Robotics Research* 32 (09 2013), 1238–1274. https://doi.org/10.1177/0278364913495721

[23] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems*, Vol. 30.

[24] Arun Mallya and Svetlana Lazebnik. 2018. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7765–7773.

[25] Matthew Mirman, Timon Gehr, and Martin Vechev. 2018. Differentiable Abstract Interpretation for Provably Robust Neural Networks. In *International Conference on Machine Learning*. PMLR, 3578–3586.

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *CoRR* abs/1312.5602 (2013). arXiv:1312.5602 http://arxiv.org/abs/1312.5602

[27] Mark B. Ring. 1994. Continual Learning in Reinforcement Environments. *PhD thesis, University of Texas at Austin* (1994).

[28] Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. 2015. Policy Distillation. *arXiv preprint arXiv:1511.06295* (2015). https://arxiv.org/abs/1511.06295

[29] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671* (2016).

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. In *arXiv preprint arXiv:1707.06347*. https://arxiv.org/abs/1707.06347

[31] Jonathan Schwarz, Wojciech Marian Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwińska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & Compress: A scalable framework for continual learning. *arXiv preprint arXiv:1805.06370* (2018).

[32] Lesia Semenova and Cynthia Rudin. 2019. A Study in Rashomon Curves and Volumes: A New Perspective on Generalization and Model Simplicity in Machine Learning. *arXiv preprint arXiv:1908.01755* (2019). arXiv:1908.01755 [cs.LG] https://arxiv.org/abs/1908.01755

[33] Philip Sosnin, Mark Niklas Müller, Maximilian Baader, Calvin Tsay, and Matthew Robert Wicker. 2025. Certified Robustness to Data Poisoning in Gradient-Based Training. *Transactions on Machine Learning Research* (2025). https://openreview.net/forum?id=9WHifn9ZVX

[34] Philip Sosnin, Matthew Wicker, Josh Collyer, and Calvin Tsay. 2025. Abstract Gradient Training: A Unified Certification Framework for Data Poisoning, Unlearning, and Differential Privacy. *arXiv preprint arXiv:2511.09400* (2025).

[35] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press. http://incompleteideas.net/book/the-book-2nd.html

[36] Lukasz Szpruch, Agni Orfanoudaki, Carsten Maple, Matthew Wicker, Yoshua Bengio, Kwok-Yan Lam, and Marcin Detyniecki. 2025. Insuring AI: Incentivising Safe and Secure Deployment of AI Workflows. *Available at SSRN 5505759* (2025).

[37] Sebastian Thrun. 1998. *Lifelong Learning Algorithms*. Springer. 181–209 pages.

[38] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. 2025. Gymnasium: A Standard Interface for Reinforcement Learning Environments. arXiv:2407.17032 [cs.LG] https://arxiv.org/abs/2407.17032

[39] Matthew Wicker, Juyeon Heo, Luca Costabello, and Adrian Weller. 2023. Robust Explanation Constraints for Neural Networks. In *The Eleventh International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=_hHYaKu0jcj

[40] Matthew Wicker, Luca Laurenti, Andrea Patane, and Marta Kwiatkowska. 2020. Probabilistic safety for bayesian neural networks. In *UAI*. PMLR, 1198–1207.

[41] Matthew Wicker, Luca Laurenti, Andrea Patane, Nicola Paoletti, Alessandro Abate, and Marta Kwiatkowska. 2021. Certification of iterative predictions in bayesian neural networks. In *UAI*. PMLR, 1713–1723.

[42] Matthew Wicker, Luca Laurenti, Andrea Patane, Nicola Paoletti, Alessandro Abate, and Marta Kwiatkowska. 2024. Probabilistic reach-avoid for Bayesian neural networks. *Artificial Intelligence* 334 (2024), 104132.

[43] Matthew Robert Wicker, Philip Sosnin, Igor Shilov, Adrianna Janik, Mark Niklas Mueller, Yves-Alexandre De Montjoye, Adrian Weller, and Calvin Tsay. 2025. Certification for Differentially Private Prediction in Gradient-Based Training. In *International Conference on Machine Learning*. PMLR, 66726–66745.

[44] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. 2021. Fast and Complete: Enabling Complete Neural Network Verification with Rapid and Massively Parallel Incomplete Verifiers. *International Conference on Learning Representations* (2021).

[45] Tengyu Xu, Yingbin Liang, and Guanghui Lan. 2021. CRPO: A New Approach for Safe Reinforcement Learning with Convergence Guarantee. In *International Conference on Machine Learning*. PMLR.

[46] Qisong Yang, Thiago D. Simão, Nils Jansen, Simon H. Tindemans, and Matthijs T. J. Spaan. 2023. Reinforcement Learning by Guided Safe Exploration. In *ECAI*

2023 – 26th European Conference on Artificial Intelligence (Frontiers in Artificial Intelligence and Applications, Vol. 372). IOS Press, 2858–2865. https://doi.org/10.3233/FAIA230598

[47] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J. Ramadge. 2020. Projection-Based Constrained Policy Optimization. In *International Conference on Learning Representations*.

[48] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual Learning Through Synaptic Intelligence. *arXiv preprint arXiv:1703.04200* (2017).

[49] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. 2018. Efficient Neural Network Robustness Certification with General Activation Functions. *Advances in Neural Information Processing Systems* 31 (2018).

[50] Quanqi Zhang, Chengwei Wu, Haoyu Tian, Yabin Gao, Weiran Yao, and Ligang Wu. 2024. Safety Reinforcement Learning Control via Transfer Learning. *Automatica* 166 (2024), 111714. https://doi.org/10.1016/j.automatica.2024.111714

A METHODOLOGY DETAILS

Here we provide some additional details on our method SafeAdapt.

A.1 Safety surrogate sound constraint

Consider a multi-label classification setting with K actions, and without loss of generality, assume the number of actions is the same for each state, i.e. $K = |\mathcal{A}| = |\mathcal{A}(s)|\ \forall s \in \mathcal{S}$. Also, without loss of generality, assume in each state M actions are safe, i.e. $|\mathcal{A}^{\text{safe}}(s)| = M\ \forall s \in \mathcal{S}$. Let $\pi(a \mid s)$ denote the policy's probability of selecting action a in state s.

The *hard safety specification* is defined as:

$$\phi^{\text{safe}}(s) = \mathbb{I}\left\{\arg\max_a \pi(a \mid s) \in \mathcal{A}^{\text{safe}}(s)\right\},$$

which equals 1 if and only if the greedy action under the policy is safe. arg max and indicator function make ϕ^{safe} non-differentiable. To enable gradient-based optimisation, we introduce a safety surrogate, which is defined as:

$$\tilde{\phi}^{\text{safe}}(s) = \sum_{a \in \mathcal{A}^{\text{safe}}(s)} \pi(a \mid s),$$

which represents the total softmax probability mass assigned to safe actions.

A.2 Proof for Proposition 1(3)

Proof of Proposition 1(3). From the inequality

$$\sum_{a \in \mathcal{A}^{\text{safe}}(s)} \pi(a|s) > \frac{|\mathcal{A}^{\text{safe}}(s)|}{1 + |\mathcal{A}^{\text{safe}}(s)|},$$

we get:

$$\frac{\sum_{a \in \mathcal{A}^{\text{safe}}(s)} \pi(a|s)}{|\mathcal{A}^{\text{safe}}(s)|} > \frac{1}{1 + |\mathcal{A}^{\text{safe}}(s)|}.$$

Since the average safe action probability, exceeds $\frac{1}{1+|\mathcal{A}^{\text{safe}}(s)|}$, the maximum safe action probability must exceed $\frac{1}{1+|\mathcal{A}^{\text{safe}}(s)|}$ as well:

$$\max_{a \in \mathcal{A}^{\text{safe}}(s)} \pi(a|s) > \frac{1}{1 + |\mathcal{A}^{\text{safe}}(s)|} \tag{12}$$

Since the sum of safe action probabilities exceeds $\frac{|\mathcal{A}^{\text{safe}}(s)|}{1+|\mathcal{A}^{\text{safe}}(s)|}$, the sum of unsafe action probabilities cannot exceed $\frac{1}{1+|\mathcal{A}^{\text{safe}}(s)|}$:

$$\sum_{a \notin \mathcal{A}^{\text{safe}}(s)} \pi(a|s) < \frac{1}{1 + |\mathcal{A}^{\text{safe}}(s)|}$$

The upper bound for the sum of non-negative values is also the upper bound for the maximum term in the sum:

$$\max_{a \notin \mathcal{A}^{\text{safe}}(s)} \pi(a|s) < \frac{1}{1 + |\mathcal{A}^{\text{safe}}(s)|} \tag{13}$$

From Eqs. 12 and 13, we get:

$$\max_{a \in \mathcal{A}^{\text{safe}}(s)} \pi(a|s) > \max_{a \notin \mathcal{A}^{\text{safe}}(s)} \pi(a|s)$$

□

As the number of safe actions per state goes to infinity ($|\mathcal{A}^{\text{safe}}(s)| \to \infty$), the surrogate sound constraint becomes very conservative since the threshold approaches 1.

A.3 Relationship Between $\tilde{\phi}^{\text{safe}}$ and ϕ^{safe}

We proved that if the safety surrogate exceeds the threshold $\delta^\star = \frac{|\mathcal{A}^{\text{safe}}(s)|}{1+|\mathcal{A}^{\text{safe}}(s)|}$, the policy is guaranteed to be safe. Note that this condition is sufficient but not necessary.

Proof that $\delta^\star$ implies a non-necessary constraint. Let us provide an example. Consider a state in which one action is safe and 4 actions are unsafe: $|\mathcal{A}(s)| = 5$, $|\mathcal{A}^{\text{safe}}(s)| = 1$. Then $\delta^\star = \frac{1}{2}$. However, a policy with the following action probability distribution is safe:

$$\pi(a|s) = \begin{cases} 0.25, & a \in \mathcal{A}^{\text{safe}}(s), \\ 0.1875 & \text{otherwise.} \end{cases} \tag{14}$$

Even though $\max_{a\in\mathcal{A}^{\text{safe}}(s)} \pi(a|s) < \delta^\star$ $(0.25 < 0.5)$, it holds that $\max_{a\in\mathcal{A}^{\text{safe}}(s)} \pi(a|s) > \max_{a\notin\mathcal{A}^{\text{safe}}(s)} \pi(a|s)$ $(0.25 > 0.1875)$. This completes the proof that the constraint $\max_{a\in\mathcal{A}^{\text{safe}}(s)} \pi(a|s) < \delta^\star$ is not necessary for safety guarantee. □

Sufficient condition for unsafety. In order to prove that a policy is *unsafe*, it is sufficient to show that the following safety surrogate inequality holds:

$$\tilde{\phi}^{\text{safe}}(s) < \frac{1}{1+K-M},$$

where $K = |\mathcal{A}(s)|$ and $M = |\mathcal{A}^{\text{safe}}(s)|$.

Proof. Using the definition of the safety surrogate, we write:

$$\sum_{a\in\mathcal{A}^{\text{safe}}(s)} \pi(a|s) < \frac{1}{1+K-M}$$

$$\sum_{a\notin\mathcal{A}^{\text{safe}}(s)} \pi(a|s) = 1 - \sum_{a\in\mathcal{A}^{\text{safe}}(s)} \pi(a|s) > 1 - \frac{1}{1+K-M}$$

$$\sum_{a\notin\mathcal{A}^{\text{safe}}(s)} \pi(a|s) > \frac{K-M}{1+K-M}$$

From this it follows that:

- $\max_{a\notin\mathcal{A}^{\text{safe}}(s)} \pi(a|s) > \frac{1}{1+K-M}$
- $\max_{a\in\mathcal{A}^{\text{safe}}(s)} \pi(a|s) < \frac{1}{1+K-M}$

Therefore:

$$\max_{a\notin\mathcal{A}^{\text{safe}}(s)} \pi(a|s) > \max_{a\in\mathcal{A}^{\text{safe}}(s)} \pi(a|s)$$

□

However, this sufficient unsafety condition is not necessary:

Proof. To prove that, we will provide a counterexample for $K = 3$ and $M = 1$. The threshold is then $\frac{1}{2}$.

$$\pi(a|s) = \begin{cases} 0.4, & a \in \mathcal{A}^{\text{safe}}(s), \\ 0.3 & \text{otherwise.} \end{cases} \tag{15}$$

Even though $\tilde{\phi}^{\text{safe}}(s) < \frac{1}{1+K-M}$ $(0.4 < 0.5)$, the maximum probability of a safe action is higher than the maximum probability of an unsafe action $(0.4 > 0.3)$. □

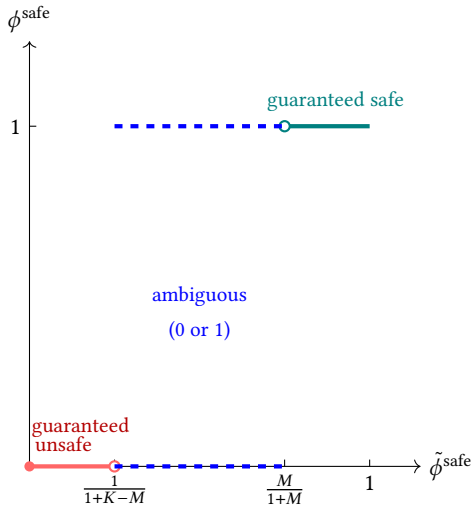

Figure 5: Relationship between the safety surrogate and the hard safety specification.

The discussion above showcases that the relationship between the hard safety and safety surrogate is helpful, but there is a region of "uncertainty". Namely, when $\tilde{\phi}^{\text{safe}}(s) \in \left[\frac{1}{1+K-M}; \frac{M}{1+M}\right]$, we cannot infer the value of $\phi^{\text{safe}}(s)$ from $\tilde{\phi}^{\text{safe}}(s)$ alone. Figure 5 illustrates the relationship between the hard safety specification and its surrogate.

The Surrogate is Not Monotonically Related to the Hard Specification. An appealing property would be that increasing $\tilde{\phi}^{\text{safe}}$ never decreases ϕ^{safe}, i.e., informally, $\partial\phi^{\text{safe}}/\partial\tilde{\phi}^{\text{safe}} \geq 0$. If this held, $\tilde{\phi}^{\text{safe}}$ would be a suitable optimisation target *without enforcing a lower bound on it.* However, the following counterexample demonstrates that this *monotonicity property does not hold for* $\tilde{\phi}^{\text{safe}} \in \left[\frac{1}{1+K-M}; \frac{M}{1+M}\right]$.

Proposition 2 (ϕ^{safe} is not globally monotonic in $\tilde{\phi}^{\text{safe}}$)**.** Consider $K = 3$ actions with $\mathcal{A}^{\text{safe}}(s) = \{a_1, a_2\}$ (two safe actions, one unsafe action a_3). For this case, the sound bound for the safety surrogate is $\delta^\star = \frac{2}{3}$. To provide a non-monotonicity example, we should consider action distributions such that $\tilde{\phi}^{\text{safe}} \in \left[\frac{1}{1+K-M}; \frac{M}{1+M}\right]$.

Proof. We will show that increasing the surrogate value can both decrease and increase the value of hard safety specification.

Increasing surrogate can decrease the hard safety.
Policy A: $\pi_A = (0.45,\ 0.15,\ 0.40)$.

- $\tilde{\phi}^{\text{safe}} = 0.45 + 0.15 = 0.60$. Note: $\tilde{\phi}^{\text{safe}} < \frac{2}{3}$.
- $\arg\max_a \pi_A(a \mid s) = a_1$ (safe), so $\phi^{\text{safe}} = 1$.

Policy B: $\pi_B = (0.38,\ 0.23,\ 0.39)$.

- $\tilde{\phi}^{\text{safe}} = 0.38 + 0.23 = 0.61 > 0.60$. Note: $\tilde{\phi}^{\text{safe}} < \frac{2}{3}$.
- $\arg\max_a \pi_B(a \mid s) = a_3$ (unsafe), so $\phi^{\text{safe}} = 0$.

Therefore, the surrogate increased $(0.60 \to 0.61)$ while the hard specification decreased $(1 \to 0)$.

Increasing surrogate can increase the hard safety.
Policy B: $\pi_B = (0.38,\ 0.23,\ 0.39)$.

- $\tilde{\phi}^{\text{safe}} = 0.38 + 0.23 = 0.61$. Note: $\tilde{\phi}^{\text{safe}} < \frac{2}{3}$.
- $\arg\max_a \pi_B(a \mid s) = a_3$ (unsafe), so $\phi^{\text{safe}} = 0$.

Policy C: $\pi_B = (0.39,\ 0.23,\ 0.38)$.

- $\tilde{\phi}^{\text{safe}} = 0.39 + 0.23 = 0.62 > 0.61$. Note: $\tilde{\phi}^{\text{safe}} < \frac{2}{3}$.
- $\arg\max_a \pi_C(a \mid s) = a_1$ (safe), so $\phi^{\text{safe}} = 1$.

Therefore, the surrogate increased ($0.61 \rightarrow 0.62$) and the hard specification increased ($0 \rightarrow 1$). □

Remark 1 (Mechanism of non-monotonicity)**.** The non-monotonic relationship arises because the surrogate aggregates probability mass over *all* safe actions, but the hard specification depends on which *single* action has the highest probability. Redistributing mass among safe actions (e.g. from a_1 to a_2) can increase the surrogate while simultaneously eroding the lead of the top safe action, allowing an unsafe action to become the argmax.

Remark 2 (Implications for optimisation)**.** This shows that gradient-based optimisation of $\tilde{\phi}^{\text{safe}}$ can, in principle, move the policy *away* from safety when there is no lower bound imposed on $\tilde{\phi}^{\text{safe}}$. This is exactly what we do in the optimisation – by requiring $\tilde{\phi}^{\text{safe}}(s) > \frac{M}{1+M}\ \forall s \in \mathcal{S}_{\text{sc}}$, we ensure that $\phi^{\text{safe}}(s) = 1\ \forall s \in \mathcal{S}_{\text{sc}}$.

A.4 Proofs of corollaries

Corollary 1 restated (per-state certified safety): Under the assumptions of Theorem 1, for every $\theta' \in \Theta^{\text{safe}}_{\text{source}}$ and every $s \in \mathcal{S}_{\text{sc}}$,

$$\arg\max_{a \in \mathcal{A}(s)} \pi_{\theta'}(a \mid s) \in \mathcal{A}^{\text{safe}}(s).$$

Proof. By Theorem 1, for every $\theta' \in \Theta^{\text{safe}}_{\text{source}}$,

$$\Phi^{\text{safe}}_{\text{sc}}(\pi_{\theta'}) = 1.$$

By Definition 4.3,

$$\Phi^{\text{safe}}_{\text{sc}}(\pi_{\theta'}) = \frac{1}{|\mathcal{S}_{\text{sc}}|} \sum_{s \in \mathcal{S}_{\text{sc}}} \phi^{\text{safe}}(s; \pi_{\theta'}).$$

Since each term $\phi^{\text{safe}}(s; \pi_{\theta'})$ is binary, the average can equal 1 only if

$$\phi^{\text{safe}}(s; \pi_{\theta'}) = 1 \qquad \forall s \in \mathcal{S}_{\text{sc}}.$$

By Definition 4.1 of the hard safety specification, $\phi^{\text{safe}}(s; \pi_{\theta'}) = 1$ is equivalent to

$$\arg\max_{a \in \mathcal{A}(s)} \pi_{\theta'}(a \mid s) \in \mathcal{A}^{\text{safe}}(s).$$

This proves the claim. □

Corollary 2 (safety preservation throughout adaptation) restated: Let $\{\theta_t\}_{t \geq 0}$ be the sequence of policy parameters produced by Phase 3. Then $\theta_t \in \Theta^{\text{safe}}_{\text{source}}$ for all t, and therefore

$$\Phi^{\text{safe}}_{\text{sc}}(\pi_{\theta_t}) = 1 \qquad \forall t \geq 0.$$

Proof. Phase 3 performs projected gradient descent with projection onto the certified orthotope $\Theta^{\text{safe}}_{\text{source}}$. Therefore, after each gradient step, the updated parameter vector is projected back into $\Theta^{\text{safe}}_{\text{source}}$, so

$$\theta_t \in \Theta^{\text{safe}}_{\text{source}} \qquad \forall t \geq 0.$$

Applying Theorem 1 to each iterate θ_t yields

$$\Phi^{\text{safe}}_{\text{sc}}(\pi_{\theta_t}) = 1 \qquad \forall t \geq 0.$$

□

Corollary 3 (distribution-independent source-task safety) restated: Under the assumptions of Theorem 1, greedy execution of π_{θ_t} on the source task never selects an unsafe action at any iteration t. Consequently, for any initial state distribution μ_0, the source-task occupancy measure of π_{θ_t} satisfies

$$\mathbb{E}_{(s,a) \sim d^{\pi_{\theta_t}}} [U_{\text{source}}(s, a)] = 0.$$

Proof. Fix any iteration $t \geq 0$. By Corollary 1, for every safety-critical state $s \in \mathcal{S}_{\text{sc}}$,

$$\arg\max_{a \in \mathcal{A}(s)} \pi_{\theta_t}(a \mid s) \in \mathcal{A}^{\text{safe}}(s).$$

By Definition 3.3, every action $a \in \mathcal{A}^{\text{safe}}(s)$ satisfies

$$U_{\text{source}}(s, a) = 0.$$

Now consider any state $s \notin \mathcal{S}_{\text{sc}}$. By Definition 3.2, such a state has no unsafe actions, so

$$U_{\text{source}}(s, a) = 0 \qquad \forall a \in \mathcal{A}(s).$$

Therefore, under greedy execution, every action selected by π_{θ_t} satisfies $U_{\text{source}}(s, a) = 0$, regardless of which source-task states are actually visited.

Hence $U_{\text{source}}(s, a) = 0$ for all state-action pairs (s, a) in the support of the source-task occupancy measure $d^{\pi_{\theta_t}}$. Since $U_{\text{source}}(s, a) \in \{0, 1\}$, it follows that

$$\mathbb{E}_{(s,a) \sim d^{\pi_{\theta_t}}} [U_{\text{source}}(s, a)] = 0.$$

□

B ENVIRONMENT CONFIGURATIONS

Table 5 shows how environments are configured and Figure 6 demonstrates initial-state frames for every environment configuration.

Table 5: Frozen Lake vs. Poisoned Apple experiment setup.

Setting	Frozen Lake	Poisoned Apple
Environment	`FrozenLake-v1` with custom wrappers	`PoisonedAppleEnv` (custom Gymnasium environment)
State representation	One-hot-encoded position over grid cells and appended task ID	Flat grid vector with the following entries: `0=empty`, `1=agent`, `2=safe apple`, `3=poisoned apple`
Action space	`Discrete(4)`: Left, Down, Right, Up	`Discrete(4)`: Left, Down, Right, Up
Unsafe event	Entering a hole tile (`H`)	Stepping onto a poisoned apple (`info["safe"]=False`, `cost=1`)
Termination on unsafe event?	Yes	No
What changes from Task 1 to Task 2?	Positions of ice holes	Safe/poisoned apple positions

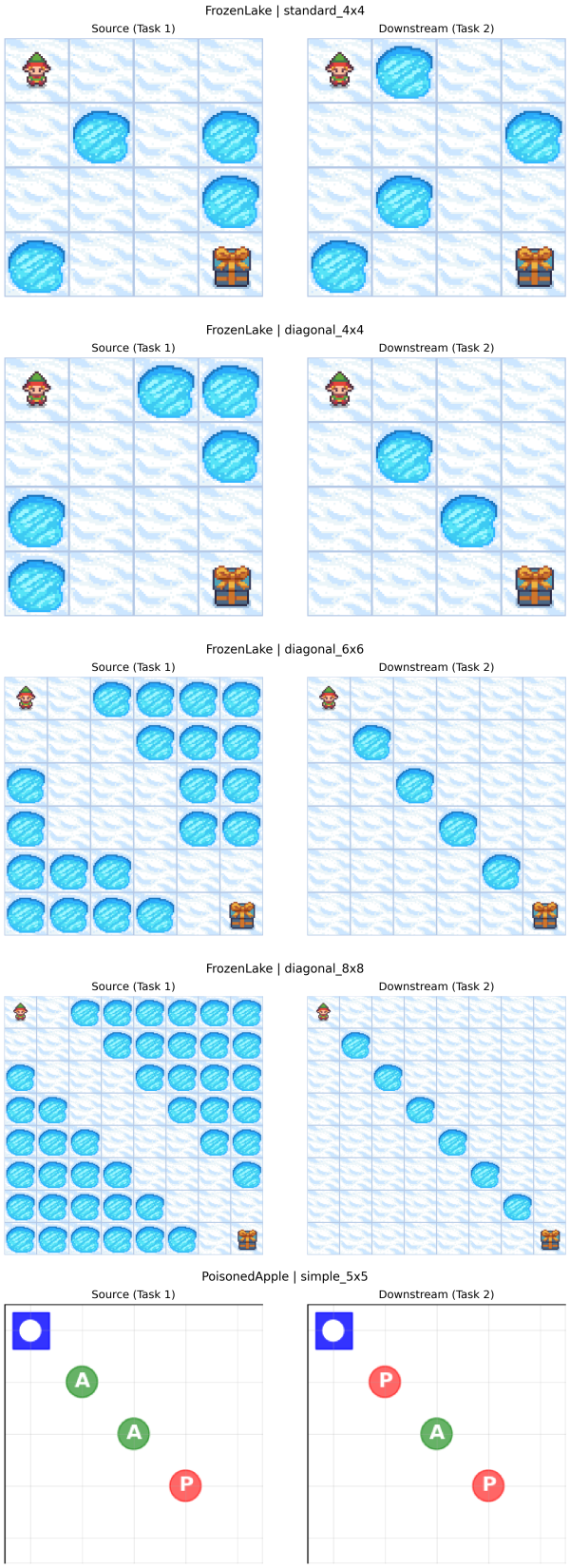

Figure 6: Environment frames at initial states.

C TRAINING, ADAPTATION, AND CERTIFICATION SETTINGS

We report the full experimental configuration used in both environments (Frozen Lake and Poisoned Apple), organized by training stage. Table 6 summarises source-task policy training (PPO and safety finetuning), Table 7 reports Rashomon-set computation settings, Table 8 details downstream PPO adaptation settings for UnsafeAdapt and SafeAdapt, and Table 9 lists EWC-PPO-specific hyperparameters. Together, these tables make explicit which choices are shared across environments and which are environment-specific.

Setting	Frozen Lake	Poisoned Apple
Source PPO max timesteps	500,000	20,000
PPO eval episodes	1	1
Actor/Critic architecture	MLP 64-64	MLP 256-256
Rollout steps	256	256
Update epochs	8	6
Mini-batch size	64	64
γ / GAE λ	0.99 / 0.95	0.99 / 0.95
Clip / value coef	0.2 / 0.5	0.2 / 0.5
Entropy coef	0.01	0.01
Learning rate / max grad norm	3×10^{-4} / 0.5	3×10^{-4} / 0.5
Source PPO early stopping	Enabled; deterministic reward ≥ 1.0 (1 eval ep)	Disabled (`early_stop=False`)
Safety finetuning	Enabled (default)	Enabled (default)
Safety finetuning objective	Allowed-action log-prob on combined safety+trajectory states (`overlap_mode=policy`)	Multi-label BC (`BCEWithLogitsLoss`) on safety-critical states
Target safety rate	1.0	1.0
Safety finetuning optimiser	Adam, lr $= 10^{-2}$, max epochs $= 3000$	Adam, lr $= 2 \times 10^{-3}$, epochs $= 2000$, batch $= 64$
Extra source acceptance check	Deterministic Task-1 reward must be 1.0	Task-1 overall success ≥ 0.95 + global safety check

Table 6: Source-task policy training and safety finetuning settings.

Setting	Frozen Lake	Poisoned Apple
Rashomon dataset	Task-1 safety-critical states	Task-1 safety dataset
Label type / aggregation	Multi-label safe-action masks; `aggregation=min`	Multi-label safe-action masks; `aggregation=min`
Rashomon iterations (`n_iters`)	5,000	20,000
Surrogate threshold	$\max_s \lvert\mathcal{A}^{\text{safe}}(s)\rvert / (1 + \max_s \lvert\mathcal{A}^{\text{safe}}(s)\rvert)$	Same formula
Inverse-temperature search	Smallest $T \in [10, 1000]$ satisfying surrogate mass constraint	Smallest $T \in [10, 1000]$ satisfying surrogate mass constraint
`smin_acc_limit`	Surrogate threshold	Surrogate threshold
`min_acc_increment`	0.0	0.0
`checkpoint`	100	100
Hard certificate threshold	1.0	`min_safety_accuracy` (default 1.0 here)
Selected bound	Last certificate index meeting hard threshold	Last certificate index meeting hard threshold
Used downstream as	Actor parameter bounds (`param_l`, `param_u`) for SafeAdapt PPO	Actor parameter bounds (`param_l`, `param_u`) for SafeAdapt PPO

Table 7: Rashomon set computation settings.

Setting	Frozen Lake	Poisoned Apple
Downstream max timesteps	50,000	20,000
Entropy coef	0.1	0.01
Learning rate	3×10^{-4} (PPO default)	3×10^{-4} (PPO default)
Rollout / epochs / minibatch	2048 / 10 / 64	2048 / 10 / 64
γ / GAE λ / clip / vf / grad norm	0.99 / 0.95 / 0.2 / 0.5 / 0.5	0.99 / 0.95 / 0.2 / 0.5 / 0.5
Eval episodes	1	1
Early stopping enabled	Yes	Yes
Early-stop reward threshold	1.0	0.96
Early-stop min steps	0	0
Early-stop check cadence	Every 20,480 steps	Every 20,480 steps

Table 8: Downstream adaptation PPO settings for UnsafeAdapt and SafeAdapt.

Setting	Frozen Lake	Poisoned Apple
EWC λ	5000	5000
Fisher data source	Source training states	Source training states
Fisher sample-size cap	$\min(1000, N_{\text{source states}})$	$\min(1000, N_{\text{source states}})$
Compute critic Fisher?	No	No
Apply EWC to critic during training?	No (default)	No (explicit)
EWC adaptation timesteps	50,000	20,000
Entropy coef in EWC-PPO	0.1	0.01
PPO backbone in EWC	lr = 3×10^{-4}, rollout 2048, epochs 10, minibatch 64	lr = 3×10^{-4}, rollout 2048, epochs 10, minibatch 64
Early-stop reward threshold	1.0	0.96
Early-stop eval episodes (internal)	10	10

Table 9: EWC-PPO settings.

D DETAILED EXPERIMENT RESULTS

D.1 Scalability analysis

Tables 10 and 11 report results of experiments with the diagonal Frozen Lake configurations to highlight how retention and adaptation scale with layout size.

D.2 Rashomon set visualisations

Figures 7 and 8 visualise the safe logit intervals and worst-case action probabilities for Frozen Lake (`standard_4x4`).

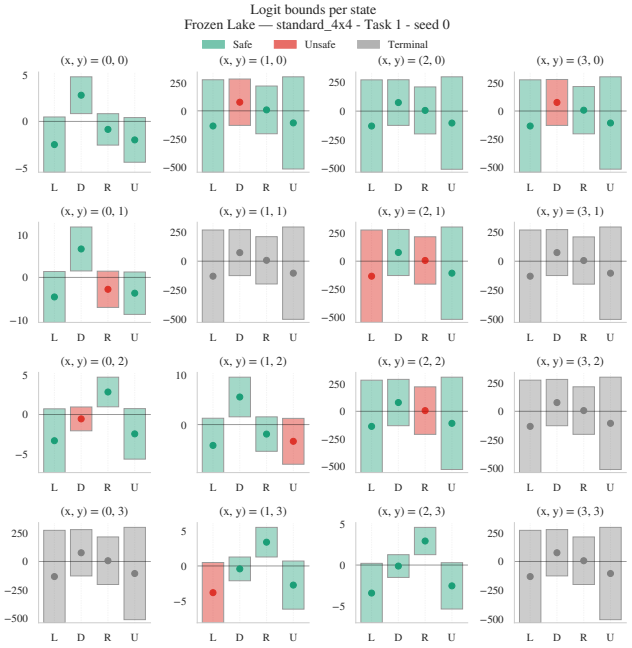

Figure 7: Logit bounds illustrate the following guarantee of the Rashomon set: in any state, there is a safe action whose lower bound logit is greater than upper bound of any unsafe action's logit. States are represented using their x and y coordinates in the grid world.

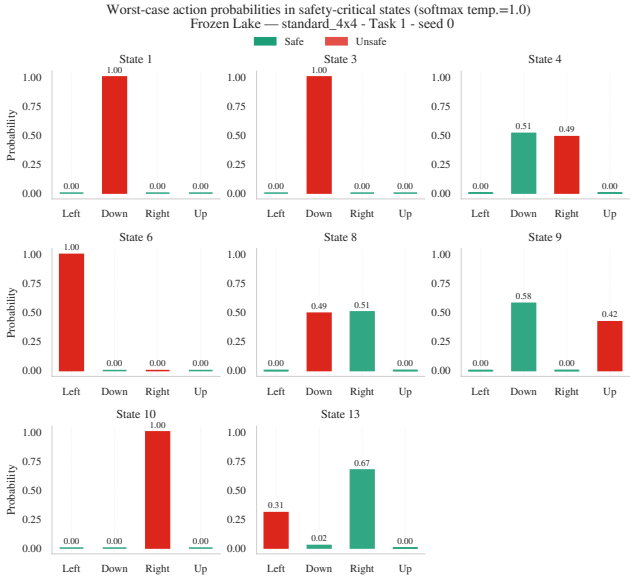

Figure 8: Neural policy probabilities for the worst-case logit vector in the source task of Frozen Lake (`standard 4x4`).

Table 10: Scalability analysis: Frozen Lake Task 1 results across configurations (mean ± std over 10 seeds).

Environment	Policy	Provably Safe?	$\Phi_{\text{sc}}(\pi)$	$\Phi_{\text{traj.}}(\pi)$	Total Reward
diagonal_4x4	Source	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	UnsafeAdapt	×	0.53 ± 0.07	0.00 ± 0.00	0.00 ± 0.00
	EWC	×	0.67 ± 0.20	0.30 ± 0.46	0.30 ± 0.46
	SafeAdapt (ours)	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
diagonal_6x6	Source	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	UnsafeAdapt	×	0.52 ± 0.11	0.00 ± 0.00	0.00 ± 0.00
	EWC	×	0.74 ± 0.15	0.10 ± 0.30	0.00 ± 0.00
	SafeAdapt (ours)	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	0.80 ± 0.40
diagonal_8x8	Source	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	UnsafeAdapt	×	0.49 ± 0.13	0.00 ± 0.00	0.00 ± 0.00
	EWC	×	0.79 ± 0.10	0.12 ± 0.33	0.00 ± 0.00
	SafeAdapt (ours)	✓	**1.00 ± 0.00**	**1.00 ± 0.00**	0.88 ± 0.33

Table 11: Scalability analysis: Frozen Lake Task 2 results across configurations (mean ± std over 10 seeds).

Environment	Policy	Total Reward	Success Rate
diagonal_4x4	Source	0.00 ± 0.00	0.00 ± 0.00
	UnsafeAdapt	**1.00 ± 0.00**	**1.00 ± 0.00**
	EWC	**1.00 ± 0.00**	**1.00 ± 0.00**
	SafeAdapt (ours)	**1.00 ± 0.00**	**1.00 ± 0.00**
diagonal_6x6	Source	0.00 ± 0.00	0.00 ± 0.00
	UnsafeAdapt	**1.00 ± 0.00**	**1.00 ± 0.00**
	EWC	**1.00 ± 0.00**	**1.00 ± 0.00**
	SafeAdapt (ours)	**1.00 ± 0.00**	**1.00 ± 0.00**
diagonal_8x8	Source	0.00 ± 0.00	0.00 ± 0.00
	UnsafeAdapt	**1.00 ± 0.00**	**1.00 ± 0.00**
	EWC	**1.00 ± 0.00**	**1.00 ± 0.00**
	SafeAdapt (ours)	0.88 ± 0.33	0.88 ± 0.33