State-Wise Constrained Policy Shaping: Runtime Behavior Steering for Safe Reinforcement Learning [*]

Thomas Howell[1], Phi Bui[1], Robert McPherson[1], Vasanth Sarathy[1]

[1]Tufts University
thomas.howell@tufts.edu, phi@cyvl.ai, rob.mcpherson97@gmail.com, vasanth.sarathy@tufts.edu

Abstract

While reinforcement learning can learn effective policies for maximizing reward, it remains difficult to encode complex behavioral preferences through reward engineering alone, especially for safety-critical applications. We present *State-wise Constrained Policy Shaping* (SCPS), a general-purpose algorithm for steering agent behavior at runtime that guarantees state-wise safety constraint satisfaction when feasible and encourages compliance with behavioral norms. SCPS minimizes the expected norm violation cost within a trust region around the original policy, balancing task performance with norm compliance at runtime. Behavioral norms are specified post-training as soft constraints, enabling the agent to adapt to evolving requirements without relearning its base policy. We evaluate SCPS in the HighwayEnv autonomous driving environment using a Deep Q-Network, where it reduces the collision rate by 97% and the norm violation cost rate by 89% in-distribution relative to the base policy. SCPS also generalizes robustly under zero-shot evaluation, achieving significant improvements in safety and norm compliance.

Code — https://github.com/thowell332/state-wise-constrained-policy-shaping

1 Introduction

In reinforcement learning (RL), an agent learns by trial and error to maximize reward through interactions with its environment. In practice, reward functions are often misspecified or incomplete, giving rise to the *value alignment* problem (Gabriel 2020). As a result, agents may learn to exploit loopholes in the reward or behave undesirably in scenarios not foreseen during training, which is especially acute in safety-critical domains (Amodei et al. 2016; Skalse et al. 2022). Addressing value alignment requires methods that go beyond optimizing task performance to explicitly promote compliance with behavioral norms and safety constraints.

A growing body of work in safe RL aims to enforce hard constraints with provable safety guarantees at runtime (Gu et al. 2024; Brunke et al. 2022). Among this category of approaches, we focus on non-learned methods for constraint set certification, which offer two key advantages: (i) they can provide explicit and interpretable justifications for safety-critical decisions, and (ii) they offer greater flexibility in adapting to evolving norms and constraints.

This paper presents *State-wise Constrained Policy Shaping* (SCPS), a principled, learning-free method for runtime policy augmentation that enforces state-wise safety constraints and steers agent behavior to comply with behavioral norms. SCPS operates entirely post-training to support dynamic requirements with a closed-form policy adjustment. At each timestep, SCPS guarantees safety constraint satisfaction when feasible and minimizes the expected norm violation cost within a trust region around the original policy, balancing task performance with the post-training objective of norm compliance. We demonstrate our approach using a Deep Q-Network (DQN) in the HighwayEnv autonomous driving environment (Leurent 2018). SCPS is deployed as a supervisor module that modifies the DQN agent's action distribution at each step. We report significant improvements in safety and norm compliance both in-distribution and under zero-shot evaluation in a more complex scenario that subsumes the training environment.

2 Related Work

Much of the prior work on enforcing norms and constraints in RL can be grouped into three methodological categories: approaches that modify the training procedure (Achiam et al. 2017; Yang et al. 2020; Zhao et al. 2024), approaches grounded in control theory (Dalal et al. 2018; Ames et al. 2019; Hobbs et al. 2023), and formal methods-based approaches that derive runtime enforcement layers from symbolic specifications (Alshiekh et al. 2018; Shalev-Shwartz, Shammah, and Shashua 2016; Neufeld et al. 2021). These methods often require retraining, rely on system dynamics, or filter actions without regard for the original objective.

Fine-tuning and residual learning methods adapt pretrained policies to new objectives through continued optimization. A notable class of fine-tuning approaches introduces regularized control objectives that constrain updates to remain close to the base policy while improving alignment with new goals (Jaques et al. 2017; Ziegler et al. 2019). Residual approaches instead keep the base policy fixed and train a residual function to augment the base behavior for improved performance or policy customization. (Johannink et al. 2019; Li et al. 2023). In contrast, SCPS explicitly separates hard safety constraints from soft normative costs and requires no additional training.

[*]Presented at the AAAI 2026 AI Governance Workshop.

3 Preliminaries

The agent's environment is modeled as a constrained Markov decision process (CMDP), with state space $\mathcal{S}$ and action space $\mathcal{A}$, where the set of allowable policies is restricted by a set of hard constraints. These constraints are specified as cost functions $\mathcal{C} = \{c_1, \ldots, c_m\}$, where each $c_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$ defines the immediate cost associated with constraint i. We focus on the state-wise CMDP, as formalized by Zhao et al. (2023), where constraints are enforced at every step rather than cumulatively or in expectation. The set of feasible stationary policies is therefore:

$$\Pi_{\mathcal{C}} = \{\pi \in \Pi \mid \forall (s, a) \sim \tau,\ \forall i,\ c_i(s, a) \leq d_i\} \tag{1}$$

where Π is the set of all stationary policies and d_i is the threshold for constraint i. The objective is to maximize the performance measure $\mathcal{J}$ subject to these constraints.

Beyond safety constraint satisfaction, many applications require agents to exhibit behavior aligned with domain-specific norms or ethical guidelines. We formalize these considerations as a norm base, defined as a set of cost functions $\mathcal{N} = \{n_1, \ldots, n_k\}$, where each $n_j : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$ defines the immediate cost associated with norm j. Each norm n_j is assigned a non-negative weight $w_j \geq 0$ reflecting its relative importance. We define the overall norm violation cost with respect to a norm base $\mathcal{N}$ as the weighted sum of costs:

$$\phi_{\mathcal{N}}(s, a) = \sum_{j=1}^{k} w_j \cdot n_j(s, a). \tag{2}$$

For a stationary policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$, which maps states to probability distributions over actions, we denote the action distribution at state s under policy π as $\pi_s := \pi(\cdot \mid s)$. We define the state-wise norm-centric performance measure as the total norm violation cost in expectation over actions:

$$\mathcal{J}_{\mathcal{N}}(\pi_s) = -\mathbb{E}_{a \sim \pi_s}\left[\phi_{\mathcal{N}}(s, a)\right] \tag{3}$$

where the negative sign ensures that maximizing $\mathcal{J}_{\mathcal{N}}$ corresponds to minimizing expected norm violations at each state.

4 State-wise Constrained Policy Shaping

The goal of SCPS is to augment an agent's action distribution at runtime, ensuring state-wise safety constraint satisfaction while improving compliance with domain-specific norms. Given a policy π_s at each step, we seek to derive a shaped policy π'_s guided by three criteria:

1. **Constraint Satisfaction:** $\pi'_s \in \Pi_{\mathcal{C}}$, where $\Pi_{\mathcal{C}}$ is the set of feasible policies under the safety constraint set.
2. **Norm Compliance:** π'_s steers behavior toward higher state-wise norm-centric performance $\mathcal{J}_{\mathcal{N}}$.
3. **Objective Retention:** π'_s retains competency in maximizing the original objective $\mathcal{J}$.

To retain the original objective, the shaped policy must preserve information from the original policy about reward-maximizing behaviors. To limit the amount of information discarded from the original policy, we formalize SCPS as the solution to a constrained optimization problem:

$$\begin{aligned} \max_{\pi'_s \in \Pi_{\mathcal{C}}} \quad & \mathcal{J}_{\mathcal{N}}(\pi'_s) \\ \text{subject to} \quad & \mathcal{D}\left(\pi'_s \| \pi_s\right) \leq \delta_s \end{aligned} \tag{4}$$

where $\mathcal{D}$ is a generic distance measure between probability distributions, and the state-wise threshold δ_s controls the trade-off between pursuing the original objective and the post-training norm-centric objective. In our approach, we instantiate $\mathcal{D}$ as the Kullback–Leibler (KL) divergence, which is formalized in Section 4.2. Bounding the divergence between the shaped and unshaped policies constrains the extent to which π' can deviate from behaviors which were learned to maximize the original objective, preserving the task-relevant competencies of π.

4.1 Permissible Action Filtering

SCPS enforces safety constraint satisfaction by filtering the action set at each timestep, only retaining the actions that satisfy all of the hard constraints. We denote the state-wise permissible action space under the constraint set $\mathcal{C}$ as:

$$\mathcal{A}_{\mathcal{C}}(s) := \{a \in A \mid \forall i,\ c_i(s, a) \leq d_i\}\,. \tag{5}$$

The filtered policy $\bar{\pi}_s$ is constructed by renormalizing the original policy π_s over the permissible action set, operating under Assumption 1. For cases where $\mathcal{A}_{\mathcal{C}}(s) = \emptyset$, the default behavior of SCPS is to retain the actions which minimize constraint violations; however, in some domains, it may be more appropriate to fall back on a known safe policy or terminate the agent.

Assumption 1. *At every state s, the base policy π_s assigns nonzero probability to at least one action in $\mathcal{A}_{\mathcal{C}}(s)$.*

Remark 1. *SCPS guarantees that $\bar{\pi}_s \in \Pi_C$ by construction of the permissible action filter under Assumption 1, given $\mathcal{A}_{\mathcal{C}}(s) \neq \emptyset$.*

4.2 Trust Region Policy Shaping

It follows from Remark 1 that the permissibility constraint $\bar{\pi} \in \Pi_{\mathcal{C}}$ is satisfied for all filtered policies $\bar{\pi}$ if a feasible solution exists. We reduce the remaining problem to maximizing norm-centric performance measure within a trust region around the filtered policy, operationalizing the distance constraint with the KL divergence, denoted as $\mathcal{D}_{\mathrm{KL}}$:

$$\begin{aligned} \max_{\pi'_s \in \Pi_{\mathcal{C}}} \quad & \mathcal{J}_{\mathcal{N}}\left(\pi'_s\right) \\ \text{subject to} \quad & \mathcal{D}_{\mathrm{KL}}\left(\pi'_s \parallel \bar{\pi}_s\right) \leq \bar{\delta}. \end{aligned} \tag{6}$$

where $\bar{\delta}$ is a state-invariant threshold which is related to the earlier δ_s by the log-probability mass assigned by π_s to impermissible actions. The direction of the KL constraint is chosen to penalize the shaped policy for overweighting actions deemed unlikely by the base policy, while allowing it to suppress norm-violating actions without prohibitive penalty. This reduced problem admits the closed-form solution:

$$\pi'_s(a) \propto \bar{\pi}_s(a) \cdot \exp\left(-\frac{1}{\beta}\hat{\phi}_{\mathcal{N}}(s, a)\right). \tag{7}$$

where $\hat{\phi}_{\mathcal{N}}(s, a)$ is the normalized cost, scaled for numerical stability, and the parameter $\beta > 0$ is numerically computed to satisfy the KL constraint. The remainder of this section establishes the global optimality and uniqueness of this solution, with degenerate cases addressed in the next section.

Theorem 1. *The shaped policy π'_s as defined in Equation 7 is globally optimal for some $\beta > 0$ if the resulting policy satisfies $\mathcal{D}_{\mathrm{KL}}(\pi'_s \parallel \bar{\pi}_s) = \bar{\delta}$ with $\bar{\delta} > 0$.*

Proof. Let $f(\pi'_s) = \mathcal{J}_{\mathcal{N}}(\pi'_s)$ be the maximization objective function $g(\pi'_s) = \mathcal{D}_{\mathrm{KL}}(\pi'_s \parallel \bar{\pi}_s) - \bar{\delta} \leq 0$ be the inequality constraint. Solving the Karush–Kuhn–Tucker (KKT) *stationary* condition yields the closed-form solution provided in Equation 7. This solution is only defined for $\beta > 0$, ensuring that the *dual feasibility* condition $\beta \geq 0$ is satisfied. The *primal feasibility* condition $g(\pi'_s) \leq 0$ is satisfied by choosing β such that the KL constraint is active. The *complementary slackness* condition holds by construction: $g(\pi'_s) = 0$. These conditions are necessary and sufficient for global optimality, since $f(\pi'_s)$ is differentiable and concave, $g(\pi'_s)$ is differentiable and convex, and Slater's condition holds because the constraint set admits the reference policy $\pi'_s = \bar{\pi}_s$, which strictly satisfies $g(\bar{\pi}_s) = -\bar{\delta} < 0$ for $\bar{\delta} > 0$. □

Theorem 2. *Any globally optimal solution for a given state s, satisfying the conditions specified by Theorem 1, is unique if* $\mathrm{Var}_{\bar{\pi}_s}(\phi_{\mathcal{N}}(s,a)) > 0$.

Proof Sketch. Let $D(\beta) := \mathcal{D}_{\mathrm{KL}}(\pi'_s \parallel \bar{\pi}_s)$ where π'_s is defined by Equation 7. Differentiating $D(\beta)$:

$$\frac{d}{d\beta} D(\beta) = -\frac{1}{\beta^3} \mathrm{Var}_{\pi'_s}(\phi_{\mathcal{N}}(s,a)) .$$

Therefore, $\mathrm{Var}_{\pi'_s}(\phi_{\mathcal{N}}(s,a)) > 0$ is a sufficient condition to guarantee that $D(\beta)$ is strictly decreasing for all $\beta > 0$. Since π'_s matches the support of $\bar{\pi}_s$, $\mathrm{Var}_{\bar{\pi}_s}(\phi_{\mathcal{N}}(s,a)) > 0$ is also a sufficient condition. Hence, any solution $\beta > 0$ for which $D(\beta) = \bar{\delta}$ must be unique. □

Method Variants The primary formulation of the SCPS algorithm is referred to as the *adaptive-β* method, where β is computed using a root-finding algorithm, like bisection or Brent's method, such that $\mathcal{D}_{\mathrm{KL}}(\pi'_s \parallel \bar{\pi}_s) = \bar{\delta}$.

An alternative formulation uses a fixed parameter β_{fixed} instead of solving the state-wise KL divergence constraint. This method, referred to as the *fixed-β* method, does not guarantee that the KL constraint is satisfied, but is included for comparison with the adaptive-β method. The fixed-β method is executed with slight modifications to Algorithm 1, where the KL constraint is ignored at step 9 and β_{fixed} is used directly at step 13 without using a solver.

4.3 Cost-optimal Projection

In this section, we derive the optimal solution for cases not covered by the previous section. Specifically, we address the case where $\mathcal{D}_{\mathrm{KL}}(\pi'_s \parallel \bar{\pi}_s) < \bar{\delta}$ for all $\beta > 0$, and the case where $\mathrm{Var}_{\bar{\pi}_s}(\phi_{\mathcal{N}}(s,a)) = 0$. We begin by defining the cost-optimal permissible action set:

$$\mathcal{A}_{\mathcal{N}}(s) := \left\{ a \in \mathcal{A}_{\mathcal{C}}(s) \;\middle|\; \phi_{\mathcal{N}}(s,a) = \min_{a' \in \mathcal{A}_{\mathcal{C}}(s)} \phi_{\mathcal{N}}(s,a') \right\} \tag{8}$$

where $\mathcal{A}_{\mathcal{C}}(s)$ is the permissible action set in state s under the constraint set $\mathcal{C}$. The cost-optimal projection π_s^{proj} is then constructed by renormalizing the filtered policy $\bar{\pi}_s$ over the cost-optimal permissible action set. The remainder of this section justifies the use of the cost-optimal projection in the adaptive-β method to handle the aforementioned degenerate cases with provable guarantees under Assumption 2. These cases correspond to steps 4-6 and 9-11 in Algorithm 1.

Algorithm 1: Adaptive-β SCPS

```
Require: $\pi, \mathcal{C}, \mathcal{N}, \bar{\delta} > 0, s \in \mathcal{S}$
 1: Filter the permissible action set $\mathcal{A}_{\mathcal{C}}(s)$ (Eq. 5)
 2: Renormalize $\pi_s$ over $\mathcal{A}_{\mathcal{C}}(s)$ to obtain $\bar{\pi}_s$
 3: Compute the cost vector $\phi_{\mathcal{N}}(\cdot|s)$ over $\mathcal{A}_{\mathcal{C}}(s)$ (Eq. 2)
 4: if $\phi_{\mathcal{N}}(s,a)$ is uniform over $\mathcal{A}_{\mathcal{C}}(s)$ then
 5:     return $\bar{\pi}_s$
 6: end if
 7: Identify the minimum cost action set $\mathcal{A}_{\mathcal{N}}(s)$ (Eq. 8)
 8: Renormalize $\bar{\pi}_s$ over $\mathcal{A}_{\mathcal{N}}(s)$ to obtain $\pi_s^{\mathrm{proj}}$
 9: if $\mathcal{D}_{\mathrm{KL}}(\pi_s^{\mathrm{proj}} \parallel \bar{\pi}_s) \leq \bar{\delta}$ then
10:     return $\pi_s^{\mathrm{proj}}$
11: end if
12: Normalize $\phi_{\mathcal{N}}(\cdot|s)$ s.t. $\sum_{a \in \mathcal{A}_{\mathcal{C}}(s)} \phi_{\mathcal{N}}(s,a) = 1$
13: Estimate $\beta > 0$ via root-finding to satisfy the active KL
    divergence constraint (Eq. 6)
14: Shape $\bar{\pi}_s$ using the estimated $\beta$ and renormalize over
    $\mathcal{A}_{\mathcal{C}}(s)$ to obtain $\pi'_s$ (Eq. 7)
15: return $\pi'_s$
```

Assumption 2. *At every state s, the filtered policy $\bar{\pi}_s$ assigns nonzero probability to at least one action in $\mathcal{A}_{\mathcal{N}}(s)$.*

Theorem 3. *If $\mathcal{D}_{\mathrm{KL}}(\pi_s^{\mathrm{proj}} \parallel \bar{\pi}_s) \leq \bar{\delta}$, then π_s^{proj} is the unique KL-minimizing cost-optimal solution.*

Proof Sketch. The policy which minimizes the KL divergence to $\bar{\pi}_s$ over the cost-minimizing action set $\mathcal{A}_{\mathcal{N}}(s)$ must be proportional to $\bar{\pi}_s$, since any deviation from this weighting would increase the KL divergence. Therefore, if $\mathcal{D}_{\mathrm{KL}}(\pi_s^{\mathrm{proj}} \parallel \bar{\pi}_s) \leq \bar{\delta}$, then π_s^{proj} is feasible and by construction the unique cost-optimal solution that minimizes the KL divergence to $\bar{\pi}_s$. □

Corollary 1. *If no $\beta > 0$ exists such that $\mathcal{D}_{\mathrm{KL}}(\pi'_s \parallel \bar{\pi}_s) = \bar{\delta}$, then π_s^{proj} is the unique KL-minimizing cost-optimal solution by Theorem 3 under Assumption 2.*

Proof Sketch. As $\beta \to 0^+$, the shaped policy π'_s converges to π_s^{proj}, and as $\beta \to \infty$, it converges to $\bar{\pi}_s$. Let $D(\beta) := \mathcal{D}_{\mathrm{KL}}(\pi'_s \parallel \bar{\pi}_s)$. In the limits of β:

$$\lim_{\beta \to 0^+} D(\beta) = \mathcal{D}_{\mathrm{KL}}(\pi_s^{\mathrm{proj}} \parallel \bar{\pi}_s), \quad \lim_{\beta \to \infty} D(\beta) = 0.$$

Thus, if no β yields $D(\beta) = \bar{\delta}$, since $D(\beta)$ is continuous for $\beta > 0$, Intermediate Value Theorem requires that $\mathcal{D}_{\mathrm{KL}}(\pi_s^{\mathrm{proj}} \parallel \bar{\pi}_s) \leq \bar{\delta}$. □

Remark 2. *If $\mathrm{Var}_{\bar{\pi}_s}(\phi_{\mathcal{N}}(s,a)) = 0$, then $\phi_{\mathcal{N}}(s,a)$ is uniform over $\mathcal{A}_{\mathcal{N}}(s) = \mathcal{A}_{\mathcal{C}}(s)$, and $\pi_s^{\mathrm{proj}} = \bar{\pi}_s$ trivially satisfies the KL divergence constraint with $\mathcal{D}_{\mathrm{KL}}(\pi_s^{\mathrm{proj}} \parallel \bar{\pi}_s) = 0$. By Theorem 3, π_s^{proj} is therefore the unique KL-minimizing cost-optimal solution if $\mathrm{Var}_{\bar{\pi}_s}(\phi_{\mathcal{N}}(s,a)) = 0$.*

5 Experimental Setup

We evaluate our approach using the HighwayEnv simulator for autonomous driving (Leurent 2018) using a Stable-Baselines3 DQN model (Raffin et al. 2021) for the ego vehicle (EV) with the hyperparameters and reward structure described in Appendix A. All non-ego vehicles are controlled by the Intelligent Driver Model (Treiber, Hennecke, and Helbing 2000). The base model is trained in a four-lane highway environment with 20 vehicles, shown in Figure 1. At test time, we evaluate in-distribution performance in the same environment used for training, as well as zero-shot performance in a more complex environment with six lanes and 50 vehicles. All results are reported as the mean and standard error of the mean (SEM) aggregated over 5 independent experiments with 100 episodes each.

5.1 State Space and Action Space

The state space of the agent is defined by kinematic observations of the EV and surrounding vehicles, describing position, velocity, and orientation. We use a discrete high-level action space for controlling the EV, comprised of *Slower*, *Idle*, *Faster*, *Lane Change Right*, and *Lane Change Left*.

5.2 Behavior Profiles and Constraints

To demonstrate runtime behavior steering, we construct *cautious* and *efficient* behavior profiles, which reflect conservative and permissive driving preferences, respectively. Both profiles instantiate the same set of hard constraints, and variants of the same norm base. For both behavior profiles, the constraint set $\mathcal{C}$ prohibits actions which violate a one second threshold for the time-to-collision (TTC), which is defined as the amount of time it would take for a vehicle to collide with another object on their current trajectories. The norm base $\mathcal{N}$ consists of six boolean-valued cost functions of equal weight, described in Appendix B, which encourage the agent to comply with a maximum speed, minimum following distance, minimum TTC, and lane keeping behavior. For lane change actions, TTC and following distance are evaluated in the target lane between the EV and its leading vehicle, and between the following vehicle and the EV. The configured values for each profile are given in Table 1.

Norm	Cautious	Efficient
Speed Limit	25 m/s	30 m/s
Following Distance	$3L$	$2L$
Minimum TTC	3 seconds	2 seconds
Lane Preference	Right	Left

Table 1: Profile thresholds. L represents one car length.

5.3 Baselines

In addition to adaptive-β and fixed-β SCPS, we evaluate the unsupervised base policy, the cost-optimal projection, and a naive policy augment method that reweights the filtered policy inversely proportionally to the normalized norm violation cost, using the state-wise scaling factor: $\frac{1}{1+\hat{\phi}_{\mathcal{N}}(s,a)}$.

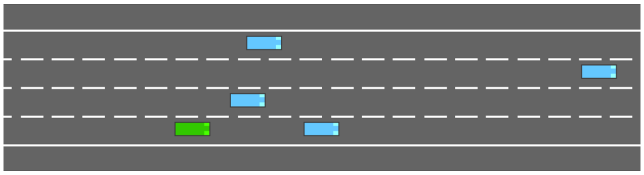

Figure 1: Screenshot from the HighwayEnv environment.

The cost-optimal projection represents the behavior of adaptive-β SCPS with an inactive KL divergence constraint, while the naive augment method isolates the effect of simple cost-based policy shaping within the permissible action set without controlling the divergence from the base policy.

6 Results & Discussion

Our results demonstrate safe and effective behavior steering across behavior profiles in-distribution, along with improved safety and norm compliance in the zero-shot environment. Ablation studies to isolate the effect of the permissible action filter are provided in Appendix C.

6.1 In-Distribution Safety and Norm Compliance

Performance metrics for safety, norm compliance, and efficiency across methods and behavior profiles are summarized in Table 2. To represent the adaptive-β and fixed-β methods, we report results for $\bar{\delta} = 0.1$ and $\beta = 0.1$, respectively. Both SCPS variants achieve significant reductions in collisions and norm violations compared to base policy, reducing the collision rate by 97% and the norm violation cost rate by at least 89% under the cautious profile. The naive method shows marked improvement over the other baselines, but consistently underperforms SCPS. The cost-optimal projection achieves a lower cost rate than the adaptive-β method, but, notably, a slightly higher collision rate under the cautious profile. This tradeoff highlights the role of the KL budget as an optimization parameter: a looser budget is guaranteed to reduce norm violations, but may also limit the supervisor's ability to preserve safety-critical behaviors from the base policy (Appendix D). In these experiments, the permissible action filter only marginally improved safety relative to the base policy, suggesting that the agent already learned to avoid constraint-violating actions when possible.

Across behavior profiles, SCPS exhibits behavior consistent with the specified norms. Table 2 shows that, while both profiles significantly improve safety relative to the base policy, the cautious profile yields a lower collision rate whereas the efficient profile maintains higher vehicle speeds. To better understand how these high-level differences emerge from local decision-making, Figure 2 shows action selection rates for adaptive-β SCPS and the base policy in low-TTC states. Under both profiles, SCPS is significantly more likely to choose *Slower* than the unsupervised model, contributing to improved safety. Between profiles, SCPS favors *Faster* and *Idle* when conditioned on the efficient profile, reflecting a more aggressive driving style that prioritizes speed. Under the cautious profile, SCPS displays a lower risk tolerance, consistently choosing *Slower* or *Lane Change* actions.

Method	Cautious Profile			Efficient Profile		
	Collision Rate (hr^{-1})	Cost Rate $(10^2 \cdot \mathrm{hr}^{-1})$	Speed $(\mathrm{m/s})$	Collision Rate (hr^{-1})	Cost Rate $(10^2 \cdot \mathrm{hr}^{-1})$	Speed $(\mathrm{m/s})$
Unsupervised	33.07 ± 2.62	62.90 ± 0.21	29.71 ± 0.01	33.07 ± 2.62	17.43 ± 0.24	29.71 ± 0.01
Filter-Only	31.54 ± 2.57	62.37 ± 0.29	29.67 ± 0.02	31.54 ± 2.57	17.00 ± 0.34	29.67 ± 0.02
Naive Augment	10.20 ± 1.53	23.01 ± 0.76	26.32 ± 0.06	32.80 ± 2.60	7.65 ± 0.22	29.24 ± 0.05
Adaptive $(\bar{\delta} = 0.1)$	$\mathbf{0.96 \pm 0.48}$	6.51 ± 0.47	24.35 ± 0.02	$\mathbf{11.98 \pm 1.66}$	3.37 ± 0.37	27.81 ± 0.07
Fixed $(\beta = 0.1)$	$\mathbf{0.96 \pm 0.48}$	4.44 ± 0.45	24.02 ± 0.04	$\mathbf{11.19 \pm 1.61}$	2.94 ± 0.28	27.70 ± 0.09
Projection	$\mathbf{1.21 \pm 0.54}$	4.44 ± 0.45	24.03 ± 0.05	$\mathbf{11.19 \pm 1.61}$	2.97 ± 0.27	27.69 ± 0.09

Table 2: In-distribution performance metrics across methods and behavior profiles, reported as the mean and SEM across experiments. Note that the unsupervised and filter-only methods are unaffected by the profile, but incur different costs under each norm base. Bolded values are within one SEM of the minimum collision rate for each profile.

6.2 Zero-Shot Generalization

We evaluate SCPS for zero-shot generalization to a more complex environment by deploying the pre-trained model in a highway with six lanes and 50 vehicles. Table 3 reports performance across methods under the cautious profile. Both SCPS variants significantly improve safety and norm compliance relative to the base policy, achieving a 99% reduction in collision rate and at least an 82% reduction in norm violation cost rate. This improvement reflects the influence of the cautious norm base, which explicitly promotes safe driving practices. Compared to the in-distribution case, the permissible action filter has a greater impact in the complex environment, suppressing unsafe behaviors that emerge more frequently under the distribution shift. The naive policy augment method significantly improves safety and norm compliance over the other baseline methods, but again underperforms SCPS on both metrics.

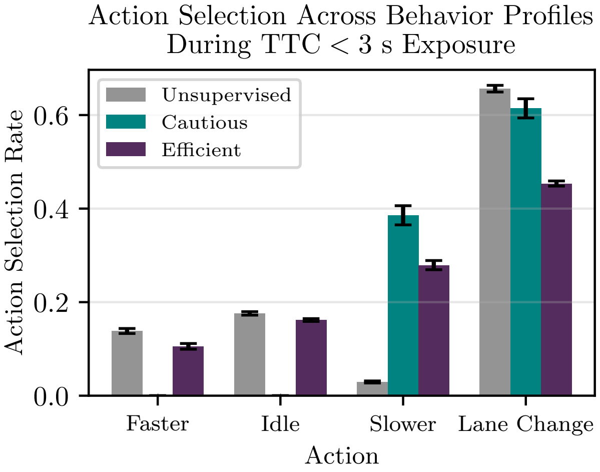

Figure 2: In-distribution action selection rates across behavior profiles for TTC < 3 s. The cautious and efficient profiles are represented by adaptive-β SCPS with $\bar{\delta} = 0.10$. Error bars represent one standard error.

Method	Collision Rate (hr^{-1})	Cost Rate $(10^2 \cdot \mathrm{hr}^{-1})$
Unsupervised	93.95 ± 3.69	63.60 ± 0.60
Filter-Only	56.64 ± 3.21	61.86 ± 0.70
Naive Augment	14.49 ± 1.81	28.73 ± 1.05
Adaptive $(\bar{\delta} = 0.1)$	$\mathbf{1.20 \pm 0.54}$	11.46 ± 0.77
Fixed $(\beta = 0.1)$	$\mathbf{0.96 \pm 0.48}$	8.76 ± 0.82
Projection	$\mathbf{0.96 \pm 0.48}$	8.62 ± 0.77

Table 3: Zero-shot performance under the cautious profile, reported as the mean and SEM across experiments. Bolded values are within one SEM of the minimum collision rate.

7 Limitations & Future Work

While SCPS provides formal guarantees for runtime constraint satisfaction and norm compliance, it relies on a competent base policy and assumes a feasible action exists at each step. Extending SCPS to handle infeasible states, continuous action spaces, and partial observability would broaden its applicability. Future work should also explore integrating SCPS with learned or symbolic representations of norms to enable adaptive, preference-aligned supervision.

8 Conclusion

In this work, we introduce SCPS, a runtime behavior steering method that provides formal guarantees on safety constraint satisfaction and expected norm violations within a trust region. Empirical results in an autonomous driving domain show that SCPS significantly improves safety and norm compliance both in-distribution and under zero-shot generalization, all without additional training.

Unlike methods that only filter or override unsafe actions, SCPS reshapes the full policy distribution at runtime. Through its trust region formulation, SCPS preserves competencies from the learned policy, balancing the original objective with norm compliance. SCPS supports post-training specifications for norms and constraints, enabling runtime adaptation to behavior preferences. As safety and value alignment become increasingly critical for RL systems in high-stakes domains, SCPS offers a flexible and principled framework for runtime control of agent behavior.

A Training Details

We used the Stable-Baselines3 default hyperparameter values for the underlying DQN model, shown in Table 4.

Parameter	**Value**
net_arch	[256, 256]
learning_rate	0.0005
buffer_size	15000
learning_starts	200
batch_size	32
gamma	0.8
train_freq	1
gradient_steps	1
target_update_interval	50
total_timesteps	100000

Table 4: Hyperparameters for the DQN model.

During training, the agent received a scalar reward at each timestep: $r = \sum_i w_i \cdot R_i$, where R_i is the value of each reward component and w_i is its corresponding weight. We used the following reward components from HighwayEnv:

- *Collision penalty* ($w = 1.0$): Evaluates to -1 if the ego vehicle crashes, 0 otherwise.
- *High-speed reward* ($w = 0.4$): Encourages maintaining a velocity v in the target range: $R_{\text{speed}} = \text{clip}\left(\frac{v-20}{10}, 0, 1\right)$.

B Safety Constraints and Norms

We define norms and constraints as binary cost functions $f : \mathcal{S} \times \mathcal{A} \to [0, 1]$. *LaneChange* variants evaluate whether the lane change is feasible and penalize the action if it would result in a violation for the EV with respect to its leading vehicle or for the following vehicle with respect to the EV.

B.1 Constraints

1. *CollisionConstraint*: Prohibits actions that produce or fail to mitigate a violation of the TTC threshold to the leading vehicle, set to one second for our experiments.
2. *LaneChangeCollisionConstraint*: Prohibits actions that cause a lane change and a *CollisionConstraint* violation for the EV or following vehicle in the target lane.

B.2 Norms

1. *SpeedNorm* ($w = 1.0$): Discourages actions that produce or fail to mitigate a violation of the speed threshold.
2. *TailgatingNorm* ($w = 1.0$): Discourages actions that produce or fail to mitigate a violation of the configured threshold for following distance to the leading vehicle.
3. *BrakingNorm* ($w = 1.0$): Discourages actions that produce or fail to mitigate a violation of the configured threshold for TTC to the leading vehicle.
4. *LaneChangeTailgatingNorm* ($w = 1.0$): Discourages actions that cause a lane change and a *TailgatingNorm* violation for the EV or following vehicle in the target lane.
5. *LaneChangeBrakingNorm* ($w = 1.0$): Discourages actions that cause a lane change and a *BrakingNorm* violation for the EV or following vehicle in the target lane.
6. *LaneKeepingNorm* ($w = 1.0$): Discourages actions that cause a lane change outside of the configured preference.

C Ablation Studies

To isolate the effect of the action filter, we tested all policy augment methods without hard constraints. Table 5 shows results under the cautious profile for the in-distribution and complex zero-shot scenarios. In-distribution, removing the action filter significantly increased collisions for the naive method, but had negligible impact on SCPS. Combined with Table 2, which shows that the base policy naturally avoids constraint-violating actions in-distribution when feasible, this suggests that SCPS preserved the relevant competencies of the base policy whereas the naive method disrupted them. In the zero-shot environment, where Table 3 shows that the base policy frequently selects constraint-violating actions even when safer alternatives exist, Table 5 confirms that the permissible action filter plays a critical role in SCPS's ability to steer behavior safely, particularly for the adaptive-β method which enforces the trust region constraint.

Notably, the fixed-β and projection methods are less reliant on the action filter for safe behavior under the distribution shift. This suggests that enforcing the trust region constraint can hinder safe adaptation when the reference policy fails to prioritize safe actions. SCPS addresses this limitation by shaping around the filtered policy, constructing the trust region around a distribution that systematically suppresses impermissible actions while preserving a meaningful notion of proximity to the original policy.

Method	**Collision Rate** (%)	**Cost Rate** (%)
In-Distribution		
Naive Augment	$+15.00$	$+1.84$
Adaptive $\left(\bar{\delta} = 0.1\right)$	0.00	$+0.04$
Fixed ($\beta = 0.1$)	0.00	-0.01
Projection	0.00	0.00
Complex Zero-Shot		
Naive Augment	$+29.84$	-1.36
Adaptive $\left(\bar{\delta} = 0.1\right)$	$+67.03$	-0.41
Fixed ($\beta = 0.1$)	0.00	$+2.62$
Projection	0.00	0.00

Table 5: Effect of removing the permissible action filter under the cautious profile, reported as percent change.

D KL-Budget Sensitivity Analysis

Figure 3 shows trends in collision and cost rates as the parameter $\bar{\delta}$ varies for adaptive-β SCPS. As $\bar{\delta}$ increases, the shaped policy deviates further from the base policy to reduce norm violations until convergence to the cost-optimal projection. Importantly, the collision rate reaches a minimum before this convergence, illustrating how appropriate tuning of $\bar{\delta}$ balances safety with improved norm compliance.

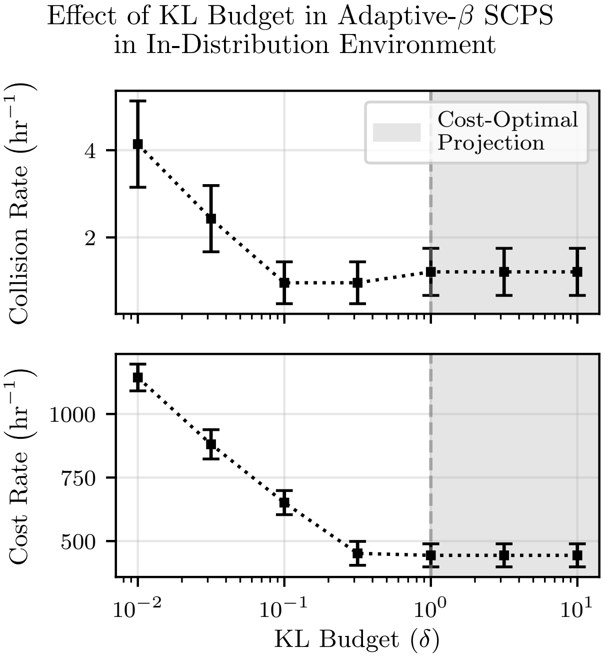

Figure 3: Collision and cost rates as functions of the KL budget $\bar{\delta}$ for the cautious profile in-distribution. The shaded region indicates convergence to the cost-optimal projection.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 22–31. PMLR.

Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; and Topcu, U. 2018. Safe Reinforcement Learning via Shielding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Ames, A. D.; Coogan, S.; Egerstedt, M.; Notomista, G.; Sreenath, K.; and Tabuada, P. 2019. Control Barrier Functions: Theory and Applications. In *2019 18th European Control Conference (ECC)*, 3420–3431.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete Problems in AI Safety. arXiv:1606.06565.

Brunke, L.; Greeff, M.; Hall, A. W.; Yuan, Z.; Zhou, S.; Panerati, J.; and Schoellig, A. P. 2022. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(Volume 5, 2022): 411–444.

Dalal, G.; Dvijotham, K.; Vecerík, M.; Hester, T.; Paduraru, C.; and Tassa, Y. 2018. Safe Exploration in Continuous Action Spaces. arXiv:1801.08757.

Gabriel, I. 2020. Artificial Intelligence, Values, and Alignment. *Minds and Machines*, 30(3): 411–437.

Gu, S.; Yang, L.; Du, Y.; Chen, G.; Walter, F.; Wang, J.; and Knoll, A. 2024. A Review of Safe Reinforcement Learning: Methods, Theories, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 11216–11235.

Hobbs, K. L.; Mote, M. L.; Abate, M. C.; Coogan, S. D.; and Feron, E. M. 2023. Runtime Assurance for Safety-Critical Systems: An Introduction to Safety Filtering Approaches for Complex Control Systems. *IEEE Control Systems Magazine*, 43(2): 28–65.

Jaques, N.; Gu, S.; Bahdanau, D.; Hernández-Lobato, J. M.; Turner, R. E.; and Eck, D. 2017. Sequence Tutor: Conservative Fine-Tuning of Sequence Generation Models with KL-control. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1645–1654. PMLR.

Johannink, T.; Bahl, S.; Nair, A.; Luo, J.; Kumar, A.; Loskyll, M.; Ojea, J. A.; Solowjow, E.; and Levine, S. 2019. Residual Reinforcement Learning for Robot Control. In *2019 International Conference on Robotics and Automation (ICRA)*, 6023–6029.

Leurent, E. 2018. An Environment for Autonomous Driving Decision-Making. https://github.com/eleurent/highway-env.

Li, C.; Tang, C.; Nishimura, H.; Mercat, J.; Tomizuka, M.; and Zhan, W. 2023. Residual Q-learning: offline and online policy customization without value. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Curran Associates Inc.

Neufeld, E.; Bartocci, E.; Ciabattoni, A.; and Governatori, G. 2021. A Normative Supervisor for Reinforcement Learning Agents. In *Automated Deduction – CADE 28*, 565–576. Springer International Publishing.

Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; and Dormann, N. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268): 1–8.

Shalev-Shwartz, S.; Shammah, S.; and Shashua, A. 2016. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. arXiv:1610.03295.

Skalse, J.; Howe, N.; Krasheninnikov, D.; and Krueger, D. 2022. Defining and Characterizing Reward Gaming. In *Advances in Neural Information Processing Systems*, volume 35, 9460–9471. Curran Associates, Inc.

Treiber, M.; Hennecke, A.; and Helbing, D. 2000. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*, 62: 1805–1824.

Yang, T.-Y.; Rosca, J.; Narasimhan, K.; and Ramadge, P. J. 2020. Projection-Based Constrained Policy Optimization. In *International Conference on Learning Representations*.

Zhao, W.; Chen, R.; Sun, Y.; Li, F.; Wei, T.; and Liu, C. 2024. State-Wise Constrained Policy Optimization. *Transactions on Machine Learning Research*.

Zhao, W.; He, T.; Chen, R.; Wei, T.; and Liu, C. 2023. State-wise Safe Reinforcement Learning: A Survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 6814–6822. International Joint Conferences on Artificial Intelligence Organization.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2019. Fine-Tuning Language Models from Human Preferences. arXiv:1909.08593.