FOCUSING ON THE RISKIEST: GAUSSIAN MIXTURE MODELS FOR SAFE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning under safety constraints remains a fundamental challenge. While primal–dual formulations provide a principled framework for enforcing such constraints, their effectiveness depends critically on accurate modeling of cost distributions. Existing approaches often impose Gaussian assumptions and approximate risk either by the mean or by CVaR, yet these formulations inherently fail to capture complex, multimodal, or heavy-tailed risks. To overcome these limitations, we propose **GMM-SSAC** (Gaussian Mixture Model-Based Supremum CVaR-Guided Safe Soft Actor-Critic), whose core is the Supremum Conditional Value-at-Risk (SCVaR) criterion: a coherent and robust safety measure that explicitly targets the worst-case tail across all components of a Gaussian mixture. To support accurate SCVaR estimation online, we introduce an incremental EM-based update that refines the GMM parameters by blending instantaneous safety samples with Bellman-transformed estimates—ensuring unbiased, convergent parameter estimates for reliable SCVaR computation. Empirical evaluations on standard safety benchmarks demonstrate that GMM-SSAC substantially improves risk sensitivity and safety while maintaining competitive task performance, validating SCVaR as a principled and effective cost estimator for safe reinforcement learning.

1 INTRODUCTION

Safe Reinforcement Learning (Safe RL) aims to enable autonomous agents to learn effective policies while satisfying safety constraints. With RL increasingly applied in safety-critical domains such as healthcare (Yu et al., 2021), robotics (Tang et al., 2024), finance (Hambly et al., 2023), and autonomous driving (Kiran et al., 2021), ensuring safe and reliable operation has become crucial. Standard RL methods, which focus on maximizing cumulative rewards, often overlook risks and safety violations during training, leading to unsafe behaviors and potentially catastrophic failures. This highlights the need for principled frameworks that explicitly integrate safety into the learning process.

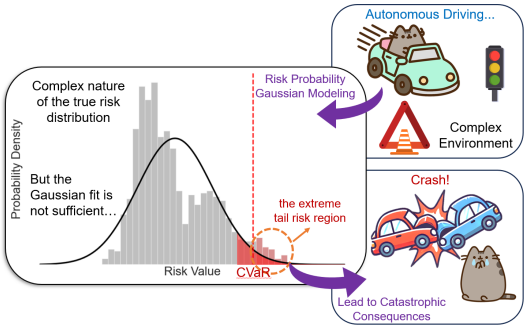


Figure 1: Demonstration of how Gaussian-based CVaR's underestimation of tail risks can lead to dangerous driving behaviors, as the model fails to properly account for extreme events that deviate from normal distribution assumptions.

A principled framework for Safe RL is the Constrained Markov Decision Process (CMDP) (Altman, 2021), where the agent maximizes rewards subject to cost constraints. Primal–dual methods are commonly used to solve CMDPs (Achiam et al., 2017; Tessler et al., 2018; Chow et al., 2018), with a safety critic estimating the distribution of cumulative costs, where "cost" denotes penalties from safety violations or undesirable actions. To better capture risk, prior studies have introduced measures such as Conditional Value-at-Risk (CVaR) (Tamar et al., 2015; Chow et al.,

2018; Coache et al., 2023), which focuses on tail losses, and upper confidence bounds (UCBs) (Wu et al., 2024), which provide conservative cost estimates.

However, most existing approaches approximate long-term costs with a single Gaussian, which is often too simplistic for safety-critical settings. As shown in Fig. 1, the empirical cost distribution can be complex and multi-modal, while its Gaussian fit fails to capture tail behavior, leading CVaR to underestimate extreme risks and induce hazardous policies. This motivates the need for more expressive distributional models. Gaussian Mixture Models (GMMs) provide a natural choice: they have universal approximation capability for continuous distributions (Chacko & Viceira, 2003; Jalali et al., 2019) and can represent distinct safety-critical modes through different components, offering both flexibility and interpretability.

GMMs have also been applied in RL for Q-function approximation (Agostini & Celaya, 2010; Vu & Slavakis, 2024), improving function accuracy, sample efficiency, and robustness in non-stationary environments. However, these works do not address the critical issue of risk modeling in Safe RL. To fill this gap, we introduce the **Supremum Conditional Value-at-Risk (SCVaR)**, defined as the maximum CVaR across GMM components. SCVaR is a coherent risk measure (Artzner et al., 1999), providing a conservative estimate that explicitly accounts for the worst-case risks captured by each component. To estimate GMM parameters robustly, we design an incremental EM refinement that blends Bellman-updated samples with new observations (Moon, 1996), ensuring unbiased online updates. Integrating this safety critic into the Soft Actor-Critic (SAC) framework (Haarnoja et al., 2018), we obtain the proposed **GMM-SSAC** algorithm.

Our contributions are threefold: (1) introducing SCVaR as a coherent risk measure that captures worst-case tail risks across mixture components; (2) developing an incremental EM-based update for accurate online GMM estimation; and (3) designing GMM-SSAC, which achieves improved safety with competitive performance on standard benchmarks.

2 Preliminaries

2.1 Constrained Markov Decision Processes

A CMDP extends the standard MDP by incorporating safety constraints into the optimization framework. Formally, a CMDP is defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, c, \gamma, D)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P(s'|s,a)$ is the transition probability from state s to state s' given action a, $r(s,a)$ is the reward function, and $c(s,a)$ is the cost function associated with safety risks. The discount factor $\gamma \in [0,1)$ balances immediate and future returns. The scalar D represents the threshold on the expected cumulative cost, defining the safety constraint.

The goal of a CMDP is to find a policy π that maximizes the expected cumulative reward while ensuring that the cumulative cost remains below the threshold D. This constrained optimization problem is formulated as:

$$\max_{\pi} \quad \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right], \quad \text{s.t.} \quad \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right] \le D, \tag{1}$$

where ρ_π is the state-action distribution induced by the policy π. The inequality ensures that the policy satisfies the specified safety requirement.

2.2 Primal–Dual Method for CMDPs

To solve the constrained optimization in Eq. 1, a standard approach is to adopt the primal–dual method, which introduces a non-negative Lagrange multiplier λ associated with the safety constraint. The resulting Lagrangian is given by:

$$\mathcal{L}(\pi,\lambda) = \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] - \lambda\left(\mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right] - D\right). \tag{2}$$

The optimization then proceeds by solving the following saddle-point problem:

$$\max_{\pi}\ \min_{\lambda\ge 0}\ \mathcal{L}(\pi,\lambda). \tag{3}$$

Intuitively, the policy π (primal variable) is updated to maximize the Lagrangian objective, while the multiplier λ (dual variable) is adjusted to penalize constraint violations. In practice, this leads to an iterative update scheme where policy optimization and dual variable adjustment are alternated, ensuring that the learned policy balances task performance and safety satisfaction.

2.3 Conditional Value at Risk

Conditional Value at Risk (CVaR) is a coherent risk measure that captures the expected loss in the tail of a distribution. For a random variable X, the Value at Risk (VaR) at confidence level $\alpha \in (0, 1]$ is defined as

$$\text{VaR}_\alpha(X) = \inf\{x \in \mathbb{R} \mid F_X(x) \geq 1 - \alpha\}, \tag{4}$$

where F_X denotes the cumulative distribution function of X.

The CVaR at level α is formally defined, for distributions that are absolutely continuous, as

$$\text{CVaR}_\alpha(X) = \mathbb{E}[X \mid X \geq \text{VaR}_\alpha(X)]. \tag{5}$$

For general distributions, equivalent formulations can be obtained through integral representations of the quantile function (see, e.g., Rockafellar & Uryasev (2000)), but in this work we focus on the Gaussian case, where absolute continuity holds.

For Gaussian random variables $X \sim \mathcal{N}(\mu, \sigma^2)$, the CVaR has the closed-form expression

$$\text{CVaR}_\alpha(X) = \mu + \sigma \cdot \frac{\phi(\Phi^{-1}(1-\alpha))}{1-\alpha}, \tag{6}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the PDF and CDF of the standard normal distribution, respectively. Here, α controls the degree of risk aversion: smaller values emphasize extreme tail risks, whereas larger values consider broader outcomes.

Several works (Yang et al., 2021; Wu et al., 2024) have extended the primal–dual formulation of CMDPs by replacing the expectation of cumulative costs in the Lagrangian (Eq. 2) with a CVaR-based risk term. Specifically, the modified Lagrangian becomes

$$\mathcal{L}_{\text{CVaR}}(\pi, \lambda) = \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right] - \lambda\left(\text{CVaR}_\alpha\left(\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)\right) - D\right). \tag{7}$$

This CVaR-based primal–dual formulation provides a more conservative alternative to expectation-based constraints by explicitly regularizing the tail of the cost distribution.

3 Methodology

3.1 Supremum Conditional Value at Risk

Fig. 1 illustrates the limitations of single Gaussian approximations in capturing complex safety cost distributions. To overcome this challenge, we propose a more expressive distributional framework using GMMs, defined as:

$$\mathcal{G}^\pi(s, a) \approx \sum_{k=1}^{K} \omega_k \mathcal{N}(\mu_k, \sigma_k^2), \tag{8}$$

where $\mathcal{G}^\pi(s, a)$, represents the probabilistic distribution of cumulative safety costs $\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)$ under the policy π, ω_k are the mixing coefficients satisfying $\sum_{k=1}^{K}\omega_k = 1$, and $\mathcal{N}(\mu_k, \sigma_k^2)$ represents the k-th Gaussian component with mean μ_k and variance σ_k^2.

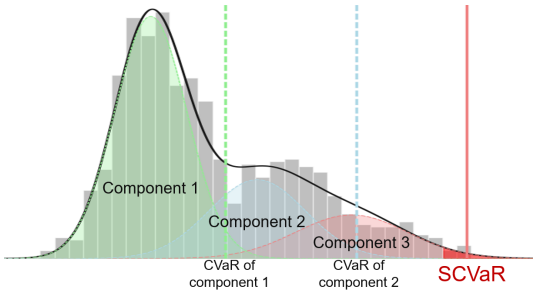


Figure 2: SCVaR illustration showing how it identifies the worst-case risk (red line) beyond individual CVaR components (green and blue) in a GMM.

To quantify extreme tail risks in GMMs, we introduce the concept of SCVaR as shown in Fig. 2. SCVaR extends the traditional CVaR framework by focusing on the worst-case tail risk among all components in a multimodal distribution. We define SCVaR as follows.

Definition 1 (Supremum Conditional Value at Risk)**.** *SCVaR is the maximum CVaR across all components of the GMM, capturing the worst-case tail risk:*

$$SCVaR_\alpha = \sup_{k\in\{1,\dots,K\}} CVaR_\alpha^{(k)}, \tag{9}$$

where $CVaR_\alpha^{(k)}$ is the CVaR at level α of the k-th Gaussian component, and K is the total number of components.

Theorem 1 (Proof in Appendix A.1)**.** *SCVaR provides a conservative upper bound on the overall mixture CVaR under a GMM, i.e.,*

$$SCVaR_\alpha \geq CVaR_\alpha^{GMM},$$

where $CVaR_\alpha^{GMM}$ is the CVaR of the full GMM distribution at level α.

Proposition 1 (Proof in Appendix A.2)**.** *SCVaR is a coherent risk measure; it satisfies monotonicity, translation invariance, positive homogeneity, and subadditivity.*

To illustrate the distinction between SCVaR and CVaR, we provide an investment portfolio example involving bonds and stocks in Appendix A.3.

3.2 Incremental Updating Safety Critic with Bellman Operator

Our next challenge is to estimate the parameters $\{(\mu_k, \sigma_k, \omega_k)\}_{k=1}^K$ in $\mathcal{G}^\pi(s,a)$. Unlike traditional distribution parameter estimation problems, in the CMDP environment, since the safety cost is the discounted sum of the instantaneous safety measures over time, we cannot directly obtain samples of the cost. The only data we can collect are the instantaneous safety measures. To overcome these challenges, we employ an incremental updating approach that blends instantaneous safety measures from the Bellman operator with historical distribution estimates.

1) Sampling Operation $\mathcal{X}$.

We first sample from the existing parameter estimates, which results in a sampling operation:

$$\mathcal{X}(\mathcal{G}^\pi(s,a), N) = \bigcup_{k=1}^{K} \mathcal{X}_k(\mathcal{N}(\mu_k, \sigma_k^2), N_k), \tag{10}$$

where $\mathcal{X}_k$ denotes N_k samples drawn independently from the k-th Gaussian component. The allocation of N_k is calculated as:

$$N_k = \lfloor \omega_k N \rfloor, \quad \forall k \in \{1, \dots, K\}. \tag{11}$$

If $\sum_{k=1}^K N_k \neq N$ due to rounding, the remaining samples are assigned to components with the highest ω_k. We denote the final sample set as $\Psi(s,a) = \mathcal{X}(\mathcal{G}^\pi(s,a), N)$.

2) Bellman Sampling with Operator $\mathcal{B}$. Next, we use the Bellman Equation to generate a new estimate based on the instantaneous safety measures. For a given state-action pair (s,a), the Bellman operator $\mathcal{B}$ updates the safety critic by combining the real-time observed safety cost $c(s,a)$ and the expected discounted future safety costs. Samples x_i' are drawn from the target safety critic distribution $\mathcal{G}^\pi_{\text{target}}(s',a')$ using the sampling operator $\mathcal{X}$. The transformed sample set is then defined as:

$$\Psi_{\mathcal{B}}(s,a) = \{c(s,a) + \gamma x_i' \mid x_i' \in \mathcal{X}(\mathcal{G}^\pi_{\text{target}}(s',a'), M)\}, \tag{12}$$

where M represents the total number of samples.

3) Incremental Refinement with Operator $\mathcal{R}$.

The incremental refinement operator $\mathcal{R}$ blends the historical sample set $\Psi(s,a)$ with the Bellman-transformed sample set $\Psi_{\mathcal{B}}(s,a)$ by applying a weight parameter β. Specifically, it updates the target sample set by sampling from each set according to the weights $1-\beta$ and β, respectively. This results in a new target sample set: $\Psi_{\text{update}}(s,a)$. The parameter β allows for flexible control over the mixture, with larger values of β giving more weight to the Bellman-transformed samples, while smaller values emphasize the current estimation.

4) EM Estimation (Projection $\mathcal{P}$). The EM algorithm (Moon, 1996) is used to estimate the GMM parameters from samples in $\mathcal{R}$, alternating between the following steps:

E-step: Compute the responsibility of each Gaussian component k for each sample $x_m \in \Psi_{\text{update}}(s,a)$. Specifically:

$$\gamma_{mk} = \frac{\omega_k \varphi(x_m \mid \mu_k, (\sigma_k)^2)}{\sum_{j=1}^{K} \omega_j \varphi(x_m \mid \mu_j, (\sigma_j)^2)}, \tag{13}$$

where $\varphi(x_m \mid \mu_k, (\sigma_k)^2)$ is the Gaussian density function for the k-th component, defined as:

$$\varphi(x_m \mid \mu_k, (\sigma_k)^2) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x_m - \mu_k)^2}{2\sigma_k^2}\right). \tag{14}$$

M-step: Update the GMM parameters based on the computed responsibilities:

$$\mu_k^{\text{update}} = \frac{\sum_{m=1}^{M} \gamma_{mk} x_m}{\sum_{m=1}^{M} \gamma_{mk}}, \quad (\sigma_k^{\text{update}})^2 = \frac{\sum_{m=1}^{M} \gamma_{mk} (x_m - \mu_k^{\text{update}})^2}{\sum_{m=1}^{M} \gamma_{mk}}, \quad \omega_k^{\text{update}} = \frac{\sum_{m=1}^{M} \gamma_{mk}}{M}. \tag{15}$$

Finally, normalize the mixing coefficients ω_k^{update} to ensure they sum to 1:

$$\omega_k^{\text{update}} \leftarrow \frac{\omega_k^{\text{update}}}{\sum_{j=1}^{K} \omega_j^{\text{update}}}, \quad \forall k \in \{1, \ldots, K\}. \tag{16}$$

5) Neural Network Update. The neural network $\mathcal{F}$ for safety critic is updated by minimizing the MSE loss between predicted GMM parameters $(\mu_k, \sigma_k, \omega_k)$ and update parameters $(\mu_k^{\text{update}}, \sigma_k^{\text{update}}, \omega_k^{\text{update}})$.

By executing these steps in sequence, we obtain the Safety Critic with Mixture Gaussian Representation (SC-MGR) algorithm (detailed in Alg. 2), and we establish its convergence.

3.2.1 Convergence Analysis

We analyze the convergence of SC-MGR via the composite Bellman–EM operator $\mathcal{P}\mathcal{T}^\pi$, where $\mathcal{T}^\pi$ is the distributional Bellman operator and $\mathcal{P}$ the EM-based projection onto the GMM space.

Lemma 1. *For any $\nu_1, \nu_2 \in \mathcal{P}_1(\mathbb{R})$, the set of probability measures on $\mathbb{R}$ with finite first moment, we have*

$$W_1(\mathcal{T}^\pi \nu_1, \mathcal{T}^\pi \nu_2) \le \gamma W_1(\nu_1, \nu_2), \quad W_1(\mathcal{P}\nu_1, \mathcal{P}\nu_2) \le W_1(\nu_1, \nu_2),$$

so that

$$W_1(\mathcal{P}\mathcal{T}^\pi \nu_1, \mathcal{P}\mathcal{T}^\pi \nu_2) \le \gamma W_1(\nu_1, \nu_2), \quad \gamma \in (0,1).$$

Thus, $\mathcal{P}\mathcal{T}^\pi$ is a γ-contraction on $\mathcal{P}_1(\mathbb{R})$.

Theorem 2. *In a finite MDP with $|S|, |A| < \infty$ and $\gamma \in (0,1)$, let the safety critic be modeled as a GMM and updated by $\mathcal{P}\mathcal{T}^\pi$. Then:*

*1. (*Bellman contraction*) $\mathcal{T}^\pi$ is a γ-contraction in W_1 (Bellemare et al., 2017).*

*2. (*EM projection non-expansiveness*, proof in Appendix B) $\mathcal{P}$ is non-expansive in W_1, since EM matches first moments: $\mathbb{E}[\gamma_{mk}] = \omega_k \Rightarrow \mathbb{E}[\mu_k^{new}] = \mu_k$.*

*3. (*Preservation of contraction*) Combining (1) and (2), the composite operator $\mathcal{P}\mathcal{T}^\pi$ is a γ-contraction.*

Hence, SC-MGR converges to a unique fixed point $\mathcal{V}^* \in \mathcal{P}_1(\mathbb{R})$, defining the stable distributional safety critic.

The key is that EM yields unbiased GMM parameter estimates, allowing the contraction mapping property of $\mathcal{T}^\pi$ (Tsitsiklis & Van Roy, 1996) to ensure convergence.

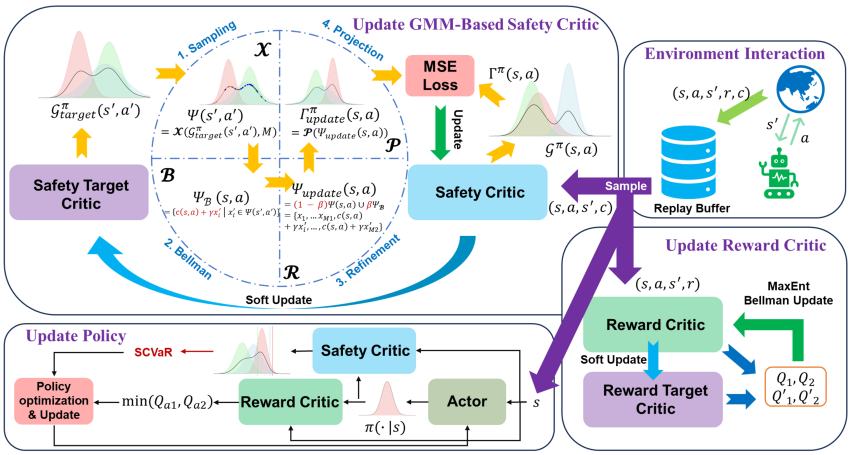


Figure 3: Framework overview of our proposed method. The architecture consists of three main components: (1) Update GMM-Based Safety Critic (top), which employs a cycle of sampling, Bellman updates, refinement, and EM projection operations to maintain the safety distribution; (2) Update Reward Critic (right), which follows the standard soft actor-critic architecture with twin Q-functions; and (3) Update Policy (bottom), which integrates both safety and reward signals through SCVaR-constrained policy optimization.

3.3 Policy Update with SCVaR

After obtaining the GMM parameters using the safety critic, we use SCVaR_α to guide safe exploration and improve policy optimization. At each iteration, the GMM parameters $\{(\mu_k, \sigma_k, \omega_k)\}_{k=1}^K$ are estimated, enabling the computation of the safety measure $\Lambda^\pi(s, a, \alpha)$ for a specified risk level α:

$$\Lambda^\pi(s, a, \alpha) \doteq \text{SCVaR}_\alpha\left(\{(\mu_k, \sigma_k, \omega_k)\}_{k=1}^K\right). \tag{17}$$

The policy is optimized under the constraint:

$$\Lambda^\pi(s, a, \alpha) \leq d, \quad \forall t, \tag{18}$$

where d is a discounted safety threshold derived from the episodic constraint D. The threshold is given by

$$d = \frac{D \cdot (1 - \gamma^T)}{(1 - \gamma) \cdot T}, \tag{19}$$

where γ is the discount factor and T is the maximum episode length. This formulation ensures consistency between the per-step discounted constraint and the original episodic bound D; the detailed derivation is provided in Appendix D.3.

To balance task performance and safety, we extend the SAC framework (Haarnoja et al., 2018) by introducing a safety-adjusted target distribution. Specifically, the policy π_θ is optimized by minimizing the KL divergence between the current policy and a target distribution that incorporates both reward maximization and safety regularization:

$$\min_\pi D_{\text{KL}}\left(\pi(\cdot \mid s_t) \middle\| \frac{\exp\left(\frac{1}{\lambda}\left(Q_r^\pi(s_t, \cdot) - \kappa\Lambda^\pi(s_t, \cdot, \alpha)\right)\right)}{Z^\pi(s_t)}\right), \tag{20}$$

where $Z^\pi(s_t)$ is the partition function ensuring normalization, $\lambda > 0$ is the temperature parameter controlling entropy, and $\kappa > 0$ is the safety weight trading off rewards against constraint violations.

From this formulation, we derive the actor loss function:

$$J_\pi(\theta) = \mathbb{E}_{(s_t, a_t) \sim \rho_{\pi_\theta}}\left[\lambda \log \pi_\theta(a_t \mid s_t) - X_{\alpha,\kappa}^{\pi_\theta}(s_t, a_t)\right], \tag{21}$$

where $X_{\alpha,\kappa}^{\pi_\theta}(s, a) = Q_r^\pi(s, a) - \kappa\Lambda^\pi(s, a, \alpha)$, with $Q_r^\pi(s, a)$ denoting the standard state-action value function.

To further guarantee adherence to safety constraints, the safety weight κ is not fixed but adaptively tuned. This is achieved by minimizing

$$J_s(\kappa) = \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\theta}}\left[\kappa\left(d - \Lambda^\pi(s_t, a_t, \alpha)\right)\right]. \tag{22}$$

3.4 Overall Framework

The overall framework of our GMM SCVaR-Guided SAC (GMM-SSAC) is illustrated in Fig. 3, which integrates reward maximization with explicit safety regulation via SCVaR. Concretely, the framework follows the SAC structure but augments it with a GMM-based safety critic, SCVaR-based risk evaluation, and adaptive adjustment of the safety weight κ. For clarity, we also present a compact pseudocode overview of the training loop in Alg. 1, while full details and derivations are deferred to Appendix C.

Algorithm 1 Overview of GMM-SSAC

1: Initialize policy, reward critics, safety critic (GMM), weights λ, κ, replay buffer $\mathcal{D}$.
2: **while** not converged **do**
3: Collect transitions and store in $\mathcal{D}$.
4: Update safety critic via Bellman–EM and GMM fitting.
5: Update reward critic via double-Q learning.
6: Compute SCVaR from GMM components.
7: Update policy with reward–risk trade-off.
8: Adapt safety weight κ to enforce constraint.
9: Soft update target networks.
10: **end while**
11: Output optimized policy π_θ.

4 Experiments

We conduct a comparative evaluation using CarGoal1, CarButton1, and CarCircle1 from the Safety-Gymnasium benchmark [1] (Ji et al., 2023) and Hopper and Ant from the velocity-constrained MuJoCo benchmark [2] (Todorov et al., 2012). A comprehensive description of the tested tasks and hyper-parameters is provided in Appendix D.

The baselines considered in our experiments are as follows: (1) SAC (Haarnoja et al., 2018), a method without safety constraints, allowing us to analyze the reward-cost trade-offs in each experimental environment; (2) SAC-Lag (Stooke et al., 2020), which employs a Lagrange multiplier update method that leverages the derivatives of the safety constraint; (3) WC-SAC (Yang et al., 2021), which uses a Gaussian distribution and CVaR estimation to model the safety constraint; and (4) CAL (Wu et al., 2024), which models the safety constraint using multiple Gaussian distributions and derives an UCB by aggregating multiple independent cost distribution estimates.

In addition to the baseline methods, we categorize the GMM-SSAC method into three variants based on different risk level values α for SCVaR: GMM-0.1 ($\alpha = 0.1$, the most conservative), GMM-0.5 ($\alpha = 0.5$, moderate risk aversion), and GMM-0.9 ($\alpha = 0.9$, nearly disregarding risk). The implementation detail and hyperparameter settings of all models are given in Appendix D.2.

We conducted several ablation studies to assess the robustness and interpretability of our approach. In the main text, we highlight two key experiments: (i) varying the number of Gaussian components and (ii) tuning the sample–set blending ratio β. Additional ablations, including comparisons with alternative risk measures, integration into baseline methods, and component-level interpretability analyses, are provided in the Appendix E.

4.1 Main Results

Fig. 4 presents the benchmark results averaged over 5 random seeds, demonstrating the effectiveness of the GMM-based approach compared to conventional Gaussian-based methods. The top/middle rows present the tested reward/cost using extra episodes after each training iteration. The bottom row presents the cost induced during training, where the training episodes are quartered in time order and respectively represented by the four boxes for each algorithm. In CarGoal1, CarButton1, CarCircle1, and Hopper environments, GMM-SSAC with $\alpha = 0.1$ and $\alpha = 0.5$ demonstrates reduced safety violations after 750k steps while achieving comparable or superior reward and constraint satisfaction levels compared to state-of-the-art off-policy methods. By tuning the α parameter, we can effectively

[1] https://github.com/PKU-Alignment/safety-gymnasium
[2] https://github.com/google-deepmind/mujoco

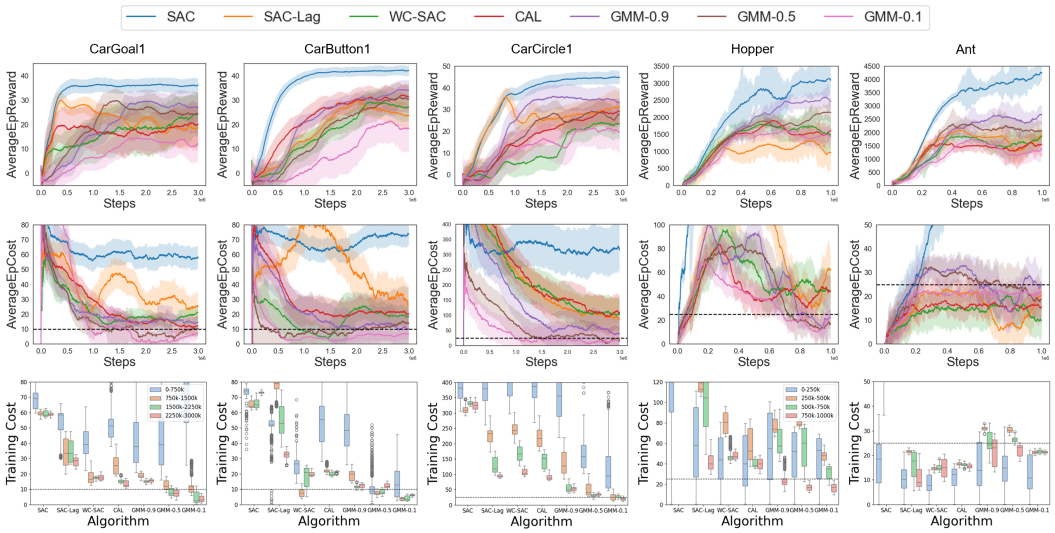


Figure 4: Comparison with off-policy baselines on five benchmark environments. Top/middle rows show average rewards and costs, bottom row shows training costs (divided into four phases). GMM-SSAC consistently reduces safety violations while maintaining competitive rewards, with α controlling the reward–cost trade-off. In Ant and Hopper, SCVaR adapts to the intrinsic reward–cost coupling, achieving balanced performance.

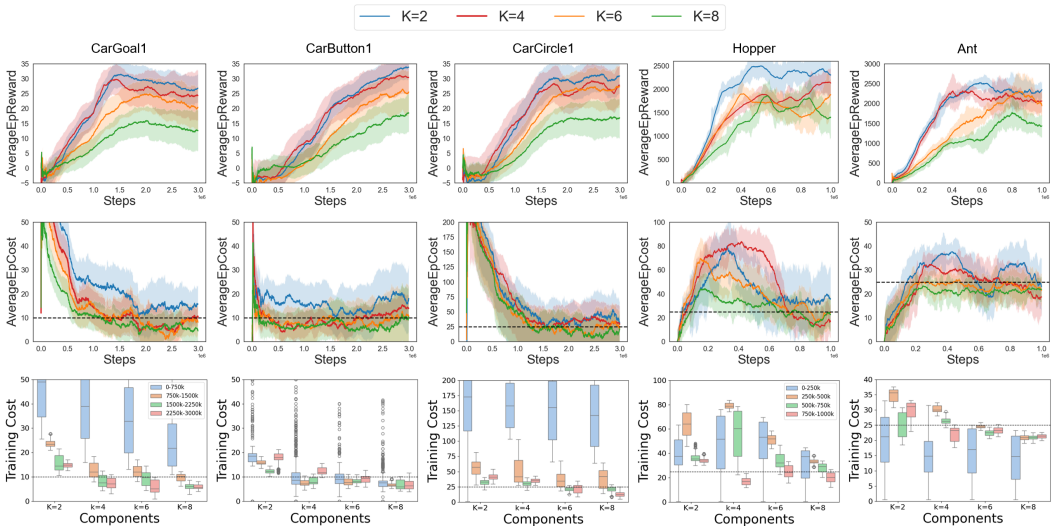


Figure 5: Ablation on the number of GMM components ($K = 2, 4, 6, 8$). Fewer components yield higher rewards but weaker cost control, while larger K improves safety by better modeling heavy-tailed distributions. Results show that $K = 6, 8$ achieve lower and more stable training costs across environments, especially in CarGoal1, CarButton1, and CarCircle1.

balance the reward-cost trade-off. With $\alpha = 0.9$, the model approaches the reward performance of unconstrained SAC while eventually reducing costs to acceptable levels, attributed to SCVaR's inherent conservative property in considering worst-case scenarios. In the Ant environment, while our model achieves higher rewards and maintains acceptable costs compared to baselines, it incurs slightly higher costs due to its adaptive balancing of efficiency and risk. Rewards and costs in Ant are intrinsically linked to velocity, where surpassing a predefined threshold incurs penalties, while higher speeds correspond to increased rewards. Our GMM-based method iteratively adjusts the mixture distribution, driving SCVaR closer to the threshold d while maintaining it just below the limit. Models like $\alpha = 0.1$ exhibit greater risk sensitivity, cautiously approaching D without exceeding it, unlike other baselines that fail to achieve this precise balance.

4.2 Ablation Study

4.2.1 Number of Gaussian Components

Fig. 5 compares the performance of models using 2, 4, 6, and 8 GMM components. The results demonstrate a clear trade-off between reward performance and safety constraints across environments. While models with fewer components ($K = 2, 4$) achieve higher rewards, configurations with more components ($K = 6, 8$) consistently maintain lower costs during training. This is particularly evident in CarGoal1, CarButton1, and CarCircle1, where $K = 8$ achieves the lowest safety violations. In the Hopper environment, $K = 6$ and $K = 8$ demonstrate more stable cost control compared to $K = 2$, which shows higher variance in constraint satisfaction. The bottom row statistics further confirm this pattern, showing that models with more components generally maintain lower training costs across different stages, especially in the later phases (750k-3000k steps). Given the complex nature of cumulative cost value distributions, incorporating additional components enables more accurate modeling of heavy-tailed values. This improved representation of tail distributions leads to more precise estimation of worst-case scenarios, thereby enhancing the model's conservative behavior and safety guarantees.

4.2.2 Sample-set Blending Ratio

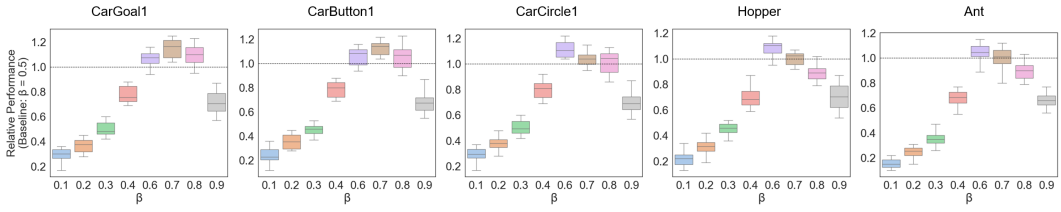


Figure 6: Ablation Study on the Impact of Blending Ratio β.

To examine the impact of the blending ratio β in the incremental updating process, we use $\beta = 0.5$ as the baseline. The relative performance is defined as:

$$P_R(x) = \frac{1}{2} \times \frac{R_{\beta=x}}{R_{\beta=0.5}} + \frac{1}{2} \times \frac{C_{\beta=0.5}}{C_{\beta=x}}, \tag{23}$$

where $R_{\beta=x}$ and $C_{\beta=x}$ are the reward and cost values under a specific β. This metric reflects the balance between rewards and costs for different β settings.

Fig. 6 shows the performance comparison of various blending ratios β during incremental updates, based on results averaged over ten runs with the same random seed in the same environment. For Safety-Gymnasium, the optimal β lies between 0.5 and 0.8, while for Mujoco, it is between 0.5 and 0.7. The optimal range highlights the importance of weighting newer learning targets more heavily in RL to avoid early stagnation due to the dynamic nature of the targets. We observe that setting $\beta = 0.9$ yields suboptimal performance. This can be attributed to the variance introduced in GMM parameters when using EM updates alongside Bellman equation updates.

5 Conclusion

We present GMM-SSAC, a Safe RL framework that combines GMMs with SCVaR-based risk assessment to improve safety in complex environments. Experiments show that GMM-SSAC achieves stronger safety guarantees while maintaining competitive reward performance. Key findings include: (1) GMM-based safety critics better capture complex risk distributions compared to single Gaussian approaches; (2) SCVaR effectively manages worst-case risks by considering the maximum CVaR across mixture components; and (3) the incremental updating mechanism enables stable learning in dynamic environments. These results suggest that more expressive risk modeling through mixture distributions, combined with conservative risk measures, provides a promising direction for developing robust safe RL systems. Future work could explore adaptive component selection strategies and extend the framework to handle more complex safety constraints.

Ethics Statement

This work adheres to the ICLR Code of Ethics.[3] Our research focuses on methodological advances in safe reinforcement learning and does not involve human subjects, personally identifiable information, or sensitive user data. The experiments are conducted entirely in simulated environments, ensuring that no harm is caused to humans, animals, or the environment.

The proposed methods aim to improve the safety and reliability of reinforcement learning algorithms, particularly in high-stakes applications. While our approach provides a more conservative mechanism for risk-sensitive decision making, it does not directly enable harmful applications. We emphasize that any future deployment in real-world safety-critical domains (e.g., healthcare, autonomous driving, finance) must carefully account for regulatory, ethical, and societal considerations beyond the scope of this work.

We disclose no conflicts of interest, external sponsorship, or ethical concerns related to fairness, discrimination, privacy, or security. All experiments were designed and conducted in compliance with established standards of research integrity.

Reproducibility Statement

We have made significant efforts to ensure the reproducibility of our work. All theoretical results are presented with complete proofs in the appendix, and detailed algorithmic descriptions, including pseudocode, are provided in Section 3 and Appendix C. Experimental setups, hyperparameters, and evaluation protocols are fully described in Section 4 and Appendix D. In addition, we will release our anonymized implementation in the supplementary materials to facilitate independent verification of the reported results.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pp. 22–31. PMLR, 2017.

Andrea Agostini and Eloy Celaya. Reinforcement learning with a gaussian mixture model. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2010.

Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999. doi: 10.1111/1467-9965.00068.

Nir Baram, Guy Tennenholtz, and Shie Mannor. Maximum entropy reinforcement learning with mixture policies. *arXiv preprint arXiv:2103.10176*, 2021.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *ICML*, pp. 449–458. PMLR, 2017.

George Chacko and Luis M Viceira. Spectral gmm estimation of continuous-time processes. *Journal of Econometrics*, 116(1-2):259–292, 2003.

Yunho Choi, Kyungjae Lee, and Songhwai Oh. Distributional deep reinforcement learning with a mixture of gaussians. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9791–9797. IEEE, 2019.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167): 1–51, 2018.

[3] https://iclr.cc/public/CodeOfEthics

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Alexandre Coache, Shaji Jaimungal, and Áureo Cartea. Conditionally elicitable dynamic risk measures for deep reinforcement learning. *SIAM Journal on Financial Mathematics*, 14(4):1249–1289, 2023.

Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. Natural policy gradient primal-dual method for constrained markov decision processes. *Advances in Neural Information Processing Systems*, 33:8378–8390, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement. In *Proceedings of the 35th International Conference on Machine Learning. July 10th-15th, Stockholm, Sweden*, volume 1870, 1861.

Tuomas Haarnoja, Aviral Zhou, Kaisa Hartikainen, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Ben Hambly, Renyuan Xu, and Huining Yang. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3):437–503, 2023.

Subin Huh and Insoon Yang. Safe reinforcement learning for probabilistic reachability and safety specifications: A lyapunov-based approach. *arXiv preprint arXiv:2002.10126*, 2020.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Shirin Jalali, Carl Nuzman, and Iraj Saniee. Efficient deep approximation of gmms. *Advances in Neural Information Processing Systems*, 32, 2019.

Ashkan B Jeddi, Nariman L Dehghani, and Abdollah Shafieezadeh. Lyapunov-based uncertainty-aware safe reinforcement learning. *arXiv preprint arXiv:2107.13944*, 2021.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. *Advances in Neural Information Processing Systems*, 36, 2023.

Dohyeong Kim, Jaeseok Heo, and Songhwai Oh. Safetac: Safe tsallis actor-critic reinforcement learning for safer exploration. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4070–4075. IEEE, 2022.

B. R. Kiran, I. Sobh, V. Talpaert, et al. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.

Zhanhong Liu, Zhiwei Cen, Vladimir Isenbaev, et al. Constrained variational policy optimization for safe reinforcement learning. In *International Conference on Machine Learning*, pp. 13644–13668. PMLR, 2022.

T. K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6): 47–60, 1996.

Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.

Santiago Paternain, Miguel Calvo-Fullana, Luiz FO Chamon, and Alejandro Ribeiro. Safe policies for reinforcement learning via primal-dual methods. *IEEE Transactions on Automatic Control*, 68(3):1321–1336, 2022.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pp. 9133–9143. PMLR, 2020.

Aviv Tamar, Yoni Glassner, and Shie Mannor. Policy gradients beyond expectations: Conditional value-at-risk. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Chao Tang, Ben Abbatematteo, Jian Hu, et al. Deep reinforcement learning for robotics: A survey of real-world successes. *Annual Review of Control, Robotics, and Autonomous Systems*, 8, 2024.

Chen Tessler, David J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

Minh Vu and Kimon Slavakis. Gaussian-mixture-model q-functions for reinforcement learning by riemannian optimization. *arXiv preprint arXiv:2409.04374*, 2024.

Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, et al. Diffusion actor-critic with entropy regulator. *Advances in Neural Information Processing Systems*, 37:54183–54204, 2024.

Yue Wang and Shaofeng Zou. Online robust reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems*, 34:7193–7206, 2021.

Zhi Wu, Baoxu Tang, Qiang Lin, et al. Off-policy primal-dual safe reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Qiang Yang, Tiago D. Simão, Stefan H. Tindemans, et al. Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10639–10646, 2021.

Chao Yu, Jie Liu, Saman Nemati, et al. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

TABLE OF CONTENTS

A SCVaR Advantage Analysis

A.1 Theoretical Proof: SCVaR Exceeds CVaR in GMMs

Proof. Assume the loss variable L follows a GMM with K components:

$$f_L(x) = \sum_{i=1}^{K} w_i \cdot \phi_i(x), \tag{24}$$

where w_i are the mixture weights ($\sum w_i = 1$), and $\phi_i(x) = \mathcal{N}(x \mid \mu_i, \sigma_i^2)$ is the PDF of the i-th Gaussian component.

The VaR at confidence level α is defined as the $(1-\alpha)$-quantile:

$$\text{VaR}_\alpha(L) = \inf\{q \mid \Pr[L \geq q] \leq 1-\alpha\} = \text{ICDF}_L(1-\alpha),$$

which for a GMM satisfies:

$$\sum_{i=1}^{K} w_i \cdot \Phi\left(\frac{\text{VaR}_\alpha - \mu_i}{\sigma_i}\right) = 1-\alpha,$$

where $\Phi(\cdot)$ is the standard normal CDF.

The Conditional Value-at-Risk (CVaR) is defined as the expected loss conditional on exceeding VaR:

$$\text{CVaR}_\alpha(L) = \mathbb{E}[L \mid L \geq \text{VaR}_\alpha] = \frac{1}{1-\alpha}\int_{\text{VaR}_\alpha}^{\infty} x \cdot f_L(x)\,dx.$$

Expanding over the mixture:

$$\text{CVaR}_\alpha^{\text{GMM}} = \frac{1}{1-\alpha}\sum_{i=1}^{K} w_i \int_{\text{VaR}_\alpha}^{\infty} x \cdot \phi_i(x)\,dx.$$

For each Gaussian component, we apply the change of variables:

$$z_i = \frac{x-\mu_i}{\sigma_i}, \quad t = \mu_i + \sigma_i z_i, \quad dt = \sigma_i\,dz_i.$$

This transforms the integral:

$$\int_{\text{VaR}_\alpha}^{\infty} t \cdot \phi_i(t)\,dt = \int_{z_i}^{\infty} (\mu_i + \sigma_i z) \cdot \phi(z)\,dz, \quad z_i = \frac{\text{VaR}_\alpha - \mu_i}{\sigma_i}.$$

Splitting the terms, we get:

$$\mu_i \int_{z_i}^{\infty} \phi(z)\,dz + \sigma_i \int_{z_i}^{\infty} z \cdot \phi(z)\,dz.$$

These evaluate as:

$$\int_{z_i}^{\infty} \phi(z)\,dz = 1 - \Phi(z_i), \quad \int_{z_i}^{\infty} z \cdot \phi(z)\,dz = \phi(z_i),$$

yielding:

$$\int_{\text{VaR}_\alpha}^{\infty} t \cdot \phi_i(t)\,dt = \mu_i(1-\Phi(z_i)) + \sigma_i\phi(z_i).$$

Therefore, the overall CVaR is:

$$\text{CVaR}_\alpha^{\text{GMM}} = \frac{1}{1-\alpha}\sum_{i=1}^{K} w_i\left[\mu_i(1-\Phi(z_i)) + \sigma_i\phi(z_i)\right].$$

To express this as a convex combination, define the posterior tail weights:

$$\text{PostWeight}_i = \frac{w_i(1-\Phi(z_i))}{1-\alpha},$$

which satisfy $\sum_{i=1}^{K} \text{PostWeight}_i = 1$.

Thus, we can write:

$$\text{CVaR}_\alpha^{\text{GMM}} = \sum_{i=1}^{K} \text{PostWeight}_i \cdot \text{CVaR}_\alpha^{(i)}.$$

We define the SCVaR as:

$$\text{SCVaR}_\alpha = \max_i \text{CVaR}_\alpha^{(i)},$$

which captures the worst-case per-component tail risk (based only on its own distribution).

Since $\text{CVaR}_\alpha^{\text{GMM}}$ is a convex combination of per-component contributions, it holds that:

$$\sum p_i a_i \le \max_i a_i \quad \text{for} \quad p_i \ge 0, \sum p_i = 1,$$

implying:

$$\text{CVaR}_\alpha^{\text{GMM}} \le \text{SCVaR}_\alpha.$$

Therefore, SCVaR always provides a conservative upper bound on the mixture CVaR, explicitly focusing on the worst-case per-component tail risk.

This completes the proof.

□

A.2 Detailed Proof of SCVaR Coherence

Proof. Let the (cost) distribution be modeled by a Gaussian mixture. For

$$X = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mu_k, \sigma_k^2), \qquad \sum_k \omega_k = 1,\ \omega_k \ge 0,$$

define

$$\text{SCVaR}_\alpha(X) := \sup_{k=1,\dots,K} \text{CVaR}_\alpha^{(k)}(X), \qquad \text{CVaR}_\alpha\big(\mathcal{N}(\mu,\sigma^2)\big) = \mu + c_\alpha \sigma, \quad c_\alpha := \frac{\phi\big(\Phi^{-1}(1-\alpha)\big)}{1-\alpha}.$$

Note that $\text{CVaR}_\alpha^{(k)}(X)$ depends only on the component parameters (μ_k, σ_k), not on the mixture weight.

Monotonicity. If $X \le Y$ almost surely, then for all k, $\text{CVaR}_\alpha^{(k)}(X) \le \text{CVaR}_\alpha^{(k)}(Y)$ (monotonicity of CVaR). Taking the supremum over k yields

$$\text{SCVaR}_\alpha(X) = \sup_k \text{CVaR}_\alpha^{(k)}(X) \le \sup_k \text{CVaR}_\alpha^{(k)}(Y) = \text{SCVaR}_\alpha(Y).$$

Translation invariance. For any constant $a \in \mathbb{R}$,

$$\text{CVaR}_\alpha^{(k)}(X+a) = \text{CVaR}_\alpha^{(k)}(X) + a \quad \Rightarrow \quad \text{SCVaR}_\alpha(X+a) = \text{SCVaR}_\alpha(X) + a.$$

Positive homogeneity. For any $\lambda \ge 0$,

$$\text{CVaR}_\alpha^{(k)}(\lambda X) = \lambda\, \text{CVaR}_\alpha^{(k)}(X) \quad \Rightarrow \quad \text{SCVaR}_\alpha(\lambda X) = \lambda\, \text{SCVaR}_\alpha(X).$$

Subadditivity. Let

$$X = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mu_k, \sigma_k^2), \qquad Y = \sum_{l=1}^{L} \lambda_l \mathcal{N}(\nu_l, \tau_l^2), \qquad \sum_k \omega_k = \sum_l \lambda_l = 1,\ \omega_k, \lambda_l \geq 0,$$

and set $Z := X + Y$. Each pair (k, l) induces a Gaussian component of Z with

$$\mu_{k,l} = \mu_k + \nu_l, \qquad \sigma_{k,l} = \sqrt{\sigma_k^2 + \tau_l^2}, \qquad \text{weight } \omega_k \lambda_l.$$

Using the closed form of Gaussian CVaR,

$$\mathrm{CVaR}_\alpha^{(k,l)}(Z) = (\mu_k + \nu_l) + c_\alpha \sqrt{\sigma_k^2 + \tau_l^2} \ \leq\ (\mu_k + c_\alpha \sigma_k) + (\nu_l + c_\alpha \tau_l) = \mathrm{CVaR}_\alpha^{(k)}(X) + \mathrm{CVaR}_\alpha^{(l)}(Y),$$

where $\sqrt{a^2 + b^2} \leq a + b$ for $a, b \geq 0$. Taking the supremum and using $\sup_{(k,l)}(a_k + b_l) \leq \sup_k a_k + \sup_l b_l$,

$$\begin{aligned}\mathrm{SCVaR}_\alpha(Z) &= \sup_{(k,l)} \mathrm{CVaR}_\alpha^{(k,l)}(Z) \ \leq\ \sup_{(k,l)} \left(\mathrm{CVaR}_\alpha^{(k)}(X) + \mathrm{CVaR}_\alpha^{(l)}(Y)\right) \\ &\leq\ \sup_k \mathrm{CVaR}_\alpha^{(k)}(X) + \sup_l \mathrm{CVaR}_\alpha^{(l)}(Y) = \mathrm{SCVaR}_\alpha(X) + \mathrm{SCVaR}_\alpha(Y).\end{aligned}$$

Therefore, SCVaR satisfies monotonicity, translation invariance, positive homogeneity, and subadditivity, and is thus a coherent risk measure. □

A.3 Investment Portfolio Example Involving Bonds and Stocks

Consider a portfolio modeled by a two-component Gaussian Mixture Model (GMM), where each component represents a distinct investment class:

$$f(x) = \omega_1 \mathcal{N}(x; \mu_1, \sigma_1^2) + \omega_2 \mathcal{N}(x; \mu_2, \sigma_2^2).$$

where the first component represents stocks with parameters:

$$\mu_1 = 10, \quad \sigma_1 = 20, \quad \omega_1 = 0.6,$$

and the second component represents bonds with parameters:

$$\mu_2 = 5, \quad \sigma_2 = 5, \quad \omega_2 = 0.4.$$

The mixture weights indicate a 60% probability to stocks and 40% to bonds. For a confidence level $\alpha = 0.95$, we employ Monte-Carlo sampling to compute both CVaR and SCVaR for this portfolio. The results are:

$$\mathrm{CVaR}_{0.95} = 29.1713, \quad \mathrm{SCVaR}_{0.95} = \mathrm{CVaR}_{0.95}^{\text{Stock}} = 30.3136.$$

In this example, SCVaR accounts for the inherent possibility that **an investor may concentrate investments in stocks**. By recognizing that the maximum risk exposure stems from the stock component's inherent volatility, SCVaR provides a more conservative risk estimate compared to the standard CVaR.

Given that the parameters $\{\mu_1, \mu_2, \sigma_1, \sigma_2, \omega_1, \omega_2\}$ are outputs of a neural network, the safety measure as a function of ω_k within the interval $(0, 1]$ satisfies the following two properties:

Theorem 3. *For a given risk level α, the safety measure $\Lambda^\pi(s, a, \alpha) = SCVaR_\alpha$ is **invariant** to $\omega_k \in (0, 1]$ for each $k \in \{1, 2, \cdots, K\}$. Specifically,*

$$\frac{\partial \Lambda^\pi(s, a, \alpha)}{\partial \omega_k} = 0, \quad \textit{for} \quad k \in \{1, 2, \cdots, K\}, \quad \omega_k \in (0, 1].$$

Theorem 4. *For a given risk level α, the safety measure $\Lambda^\pi(s, a, \alpha) = SCVaR_\alpha$ is **invariant** to μ_k and σ_k for all $k \in \{1, 2, \ldots, K\}$ except for the k_{max}-th components associated with the largest CVaR. Specifically,*

$$\frac{\partial \Lambda^\pi(s, a, \alpha)}{\partial \mu_k} = 0 \quad \textit{and} \quad \frac{\partial \Lambda^\pi(s, a, \alpha)}{\partial \sigma_k} = 0, \quad \forall k \in \{1, 2, \ldots, K\} \setminus \{k_{max}\},$$

where k_{max} is the index of the component associated with the largest CVaR, i.e.,

$$k_{max} = \arg \max_{k \in \{1, 2, \ldots, K\}} CVaR_\alpha^{(k)}.$$

Through the above two theorems, we know that under the guidance of SCVaR, the RL agent tends to completely exclude components that may lead to higher risks. In this case, if a stock is considered a high-risk investment, the RL agent will completely avoid this option, or SCVaR will mitigate the risk associated with the stock choice (for example, by selecting another stock with relatively lower return variance). If an investor completely abandons stocks and chooses a less risky investment compared to bonds, SCVaR would encourage the investor to entirely forgo the bond option as well.

B Incremental Updating for Safety Critic with Bellman Operator and Its Convergence

Algorithm 2 Incremental Updating for Safety Critic with Bellman Operator

1: **Input:** Current GMM parameters $\Gamma^\pi(s,a) = \{(\mu_k, \sigma_k, \omega_k)\}_{k=1}^K$, Immediate cost $c(s,a)$, Discount factor γ, Total samples M, Blending coefficient β, Neural network $\mathcal{F}$ for predicting $\Gamma^\pi(s,a)$.
2: **Output:** Updated network parameters for $\mathcal{F}$.
3: **Initialize:** Generate samples $\Psi(s,a) = \mathcal{X}(\mathcal{G}^\pi(s,a), M)$ using the current GMM.
4: Generate samples $\Psi(s',a') = \mathcal{X}(\mathcal{G}^\pi_{\text{target}}(s',a'), M)$ using the target safety critic GMM.
Step 1: Bellman Sampling
5: Transform samples using the Bellman operator:

$$\Psi_{\mathcal{B}}(s,a) = \{\hat{x}_i \mid \hat{x}_i = c(s,a) + \gamma x_i',\ x_i' \in \mathcal{X}(\mathcal{G}^\pi_{\text{target}}(s',a'), M)\}.$$

Step 2: Incremental Refinement
6: Blend current and Bellman-transformed samples:

$$\Psi_{\text{update}}(s,a) = \mathcal{R}(\Psi(s,a), \Psi_{\mathcal{B}}(s,a), \beta) = \{x_i, ..., x_{M_1}, c(s,a)+\gamma x_1', ..., c(s,a)+\gamma x_{M_2}' | M_1 : M_2 = (1-\beta) : \beta\}$$

Step 3: Projection Operation
7: Perform EM to update GMM parameters:
8: **E-step:** :

$$\gamma_{mk} = \frac{\omega_k \varphi(x_m \mid \mu_k, (\sigma_k)^2)}{\sum_{j=1}^K \omega_j \varphi(x_m \mid \mu_j, (\sigma_j)^2)}.$$

9: **M-step:** :

$$\mu_k^{\text{update}} = \frac{\sum_{m=1}^M \gamma_{mk} x_m}{\sum_{m=1}^M \gamma_{mk}}, \quad (\sigma_k^{\text{update}})^2 = \frac{\sum_{m=1}^M \gamma_{mk}(x_m - \mu_k^{\text{update}})^2}{\sum_{m=1}^M \gamma_{mk}}, \quad \omega_k^{\text{update}} = \frac{\sum_{m=1}^M \gamma_{mk}}{M}.$$

10: Normalize mixing coefficients:

$$\omega_k^{\text{update}} \leftarrow \frac{\omega_k^{\text{update}}}{\sum_{j=1}^K \omega_j^{\text{update}}}, \quad \forall k \in \{1, \ldots, K\}.$$

Step 4: Neural Network Update
11: Compute the Mean Squared Error (MSE) loss between predicted GMM parameters $(\mu_k, \sigma_k, \omega_k)$ and update parameters $(\mu_k^{\text{update}}, \sigma_k^{\text{update}}, \omega_k^{\text{update}})$:

$$\mathcal{L} = \sum_{k=1}^K \left[(\mu_k - \mu_k^{\text{update}})^2 + (\sigma_k - \sigma_k^{\text{update}})^2 + (\omega_k - \omega_k^{\text{update}})^2 \right].$$

12: Perform gradient descent on the network parameters of $\mathcal{F}$ to minimize $\mathcal{L}$.
13: **Return:** Updated GMM parameters $\Gamma^\pi_{\text{update}}(s,a)$ and updated network $\mathcal{F}$.

Proof. **Setup.** Fix a policy π. Let $\nu(\cdot\,|s,a) \in \mathcal{P}_1(\mathbb{R})$ denote the (cost) value distribution at (s,a). Project $\nu(\cdot\,|s,a)$ onto a Gaussian mixture:

$$\mathcal{V}(s,a) = \sum_{k=1}^{K} \omega_k \mathcal{N}(x \mid \mu_k, \sigma_k), \qquad \omega_k \geq 0,\ \sum_{k=1}^{K} \omega_k = 1.$$

The distributional Bellman operator (cost form) is

$$(\mathcal{T}^\pi \nu)(\cdot\,|s,a) = \mathsf{Law}(c(s,a) + \gamma\, X')\,, \quad X' \sim \nu(\cdot\,|s',a'),\ (s',a') \sim P(\cdot|s,a) \times \pi(\cdot|s').$$

One SC-MGR update is the Bellman–EM composition

$$\mathcal{B}^\pi \nu \ \equiv\ \mathcal{P}\big(\mathcal{T}^\pi \nu\big),$$

where $\mathcal{P}$ denotes the EM projection onto the GMM family.

Bellman contraction under W_1. By the Kantorovich–Rubinstein duality,

$$W_1(\mu,\nu) = \sup_{\|f\|_{\mathrm{Lip}} \leq 1} \Big|\, \mathbb{E}_\mu[f] - \mathbb{E}_\nu[f] \,\Big|.$$

Two basic invariances of W_1 (for any constant b and any $a \geq 0$) are

$$W_1(\mathsf{Law}(X+b), \mathsf{Law}(Y+b)) = W_1(\mathsf{Law}(X), \mathsf{Law}(Y)),$$
$$W_1(\mathsf{Law}(aX), \mathsf{Law}(aY)) = a\, W_1(\mathsf{Law}(X), \mathsf{Law}(Y)).$$

Fix (s,a) and two distributions ν_1, ν_2. Conditioning on (s',a') and applying the two invariances gives

$$\begin{aligned} W_1\big((\mathcal{T}^\pi \nu_1)(\cdot|s,a), (\mathcal{T}^\pi \nu_2)(\cdot|s,a)\big) &= W_1\big(\mathsf{Law}(c+\gamma X_1'), \mathsf{Law}(c+\gamma X_2')\big) \\ &= \gamma\, W_1\big(\mathsf{Law}(X_1'), \mathsf{Law}(X_2')\big) \\ &\leq \gamma\, \mathbb{E}_{(s',a')}\Big[W_1\big(\nu_1(\cdot|s',a'), \nu_2(\cdot|s',a')\big)\Big] \\ &\leq \gamma \sup_{(s',a')} W_1\big(\nu_1(\cdot|s',a'), \nu_2(\cdot|s',a')\big). \end{aligned}$$

Hence, uniformly over (s,a),

$$W_1\big(\mathcal{T}^\pi \nu_1, \mathcal{T}^\pi \nu_2\big) \ \leq\ \gamma\, W_1(\nu_1,\nu_2), \qquad \gamma \in (0,1).$$

EM projection: responsibilities and unbiased M-step. Given samples $\{x_m\}_{m=1}^M \sim p(x)$ (here $p = \mathcal{T}^\pi \nu$), the E-step responsibility for component k is

$$\gamma_{mk} \ =\ \frac{\omega_k\, \varphi(x_m \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^K \omega_j\, \varphi(x_m \mid \mu_j, \sigma_j^2)}, \qquad \varphi(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\tfrac{(x-\mu)^2}{2\sigma^2}\Big).$$

In the population limit ($M \to \infty$), the M-step updates are

$$\mu_k^{\text{new}} = \frac{\mathbb{E}_p[\gamma_k(X)\, X]}{\mathbb{E}_p[\gamma_k(X)]}, \qquad \omega_k^{\text{new}} = \mathbb{E}_p[\gamma_k(X)].$$

If p equals the current GMM (or is at an EM fixed point), i.e., $p(x) = \sum_j \omega_j \varphi(x \mid \mu_j, \sigma_j^2)$, then

$$\mathbb{E}_p[\gamma_k(X)] = \int \frac{\omega_k \varphi(x \mid \mu_k, \sigma_k^2)}{\sum_j \omega_j \varphi(x \mid \mu_j, \sigma_j^2)}\, p(x)\, dx = \int \omega_k \varphi(x \mid \mu_k, \sigma_k^2)\, dx = \omega_k,$$

and

$$\mathbb{E}_p[\gamma_k(X)\, X] = \int \omega_k \varphi(x \mid \mu_k, \sigma_k^2)\, x\, dx = \omega_k\, \mu_k.$$

Therefore,

$$\mathbb{E}[\mu_k^{\text{new}}] = \frac{\omega_k\, \mu_k}{\omega_k} = \mu_k, \qquad \mathbb{E}[\omega_k^{\text{new}}] = \omega_k,$$

i.e., the EM projection preserves component means and weights in expectation (population unbiasedness), hence does not distort first-moment structure.

Bellman–EM composition and convergence. Let $\nu_{t+1} = \mathcal{P}\mathcal{T}^\pi \nu_t$. Using the Bellman contraction above and the non-expansiveness of $\mathcal{P}$ in W_1 (i.e., $W_1(\mathcal{P}\mu, \mathcal{P}\nu) \le W_1(\mu, \nu)$),

$$\begin{aligned} W_1(\nu_{t+1}, \nu'_{t+1}) &= W_1\big(\mathcal{P}\mathcal{T}^\pi \nu_t,\ \mathcal{P}\mathcal{T}^\pi \nu'_t\big) \ \le\ W_1\big(\mathcal{T}^\pi \nu_t,\ \mathcal{T}^\pi \nu'_t\big) \\ &\le\ \gamma\, W_1(\nu_t, \nu'_t). \end{aligned}$$

Thus $\mathcal{P}\mathcal{T}^\pi$ is a γ-contraction under W_1. By the Banach fixed-point theorem, there exists a unique $\mathcal{V}^*$ such that

$$\mathcal{P}\mathcal{T}^\pi \mathcal{V}^* = \mathcal{V}^*, \qquad W_1(\nu_t, \mathcal{V}^*) \ \le\ \gamma^t\, W_1(\nu_0, \mathcal{V}^*).$$

Together with the population unbiasedness above, the Bellman–EM updates of SC-MGR converge stably to the unique distributional fixed point $\mathcal{V}^*$. □

C Detail Derivation of Policy Update & Complete Algorithm: GMM-SSAC

C.1 Detail Derivation of Policy Update

Leveraging the GMM-based distributional safety critic, we propose a novel safety metric, SCVaR_α, to guide safe exploration and improve policy optimization. At each iteration, the GMM parameters, $\Gamma^\pi(s,a) = \{(\mu_k, \sigma_k, \omega_k)\}_{k=1}^K$, are estimated, enabling the computation of the safety measure $\Lambda^\pi(s,a,\alpha)$ for a specified risk level α:

$$\begin{aligned} \Lambda^\pi(s,a,\alpha) &\doteq \text{SCVaR}_\alpha \\ &= \sup_{k\in\{1,\dots,K\}} \text{CVaR}_\alpha^{(k)} \\ &= \sup_{k\in\{1,\dots,K\}} \left(\mu_k + \sigma_k \frac{\phi\left(\Phi^{-1}(\alpha)\right)}{1-\alpha}\right). \end{aligned}$$

The policy is optimized under the constraint:

$$\Lambda^\pi(s,a,\alpha) \le d, \quad \forall t,$$

where d is a predefined safety threshold.

To achieve a balance between performance and safety, inspired by the SAC framework (Haarnoja et al., 2018), we optimize the policy π_θ by minimizing the Kullback-Leibler (KL) divergence between the current policy and a safety-adjusted target distribution:

$$\min_\pi D_{\text{KL}}\left(\pi(\cdot \mid s_t) \middle\| \frac{\exp\left(\frac{1}{\lambda}\left(Q_r^\pi(s_t,\cdot) - \kappa\Lambda^\pi(s_t,\cdot,\alpha)\right)\right)}{Z^\pi(s_t)}\right),$$

where $Z^\pi(s_t)$ is the partition function ensuring normalization, $\lambda > 0$ represents the temperature parameter controlling entropy, and $\kappa > 0$ is the safety weight regulating the trade-off between maximizing rewards and adhering to safety constraints.

The KL divergence can be equivalently expressed as:

$$\begin{aligned} &D_{\text{KL}}\left(\pi_\theta(\cdot \mid s_t) \middle\| \exp\left(\frac{1}{\lambda} X_{\alpha,\kappa}^{\pi_\theta}(s_t,\cdot) - \log Z^{\pi_\theta}(s_t)\right)\right) \\ &\quad = \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\theta}}\left[-\log \frac{\pi_\theta(a_t \mid s_t)}{\exp\left(\frac{1}{\lambda} X_{\alpha,\kappa}^{\pi_\theta}(s_t,a_t) - \log Z^{\pi_\theta}(s_t)\right)}\right] \\ &\quad = \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\theta}}\left[\log \pi_\theta(a_t \mid s_t) - \frac{1}{\lambda} X_{\alpha,\kappa}^{\pi_\theta}(s_t,a_t) + \log Z^{\pi_\theta}(s_t)\right], \end{aligned}$$

where

$$X^{\pi_\theta}_{\alpha,\kappa}(s,a) = Q^{\pi}_{r}(s,a) - \kappa\Lambda^{\pi}(s,a,\alpha),$$

and $Q^{\pi}_{r}(s,a)$ denotes the state-action value function.

As the partition function $Z^{\pi_\theta}(s_t)$ does not influence the gradient of θ, it can be excluded from the optimization. This results in the actor loss function:

$$J_\pi(\theta) = \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\theta}}\left[\lambda \log \pi_\theta(a_t \mid s_t) - X^{\pi_\theta}_{\alpha,\kappa}(s_t,a_t)\right].$$

To ensure the policy adheres to safety constraints, the safety weight κ is adjusted dynamically by minimizing the following loss function:

$$J_s(\kappa) = \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\theta}}\left[\kappa\left(d - \Lambda^{\pi}(s_t,a_t,\alpha)\right)\right],$$

where d represents a predefined safety threshold.

This approach enables an adaptive trade-off between performance and safety by dynamically updating κ. The reward critic $Q^{\pi_\theta}_{r}$ and the entropy weight λ are updated following the SAC method. Details on the loss functions $J_e(\lambda)$ for entropy adaptation and $J_r(\psi)$ for the reward critic can be found in Haarnoja et al. (2018).

C.2 Complete Algorithm: GMM-SSAC

See Alg. 3.

D Experiment Details

D.1 Task Description

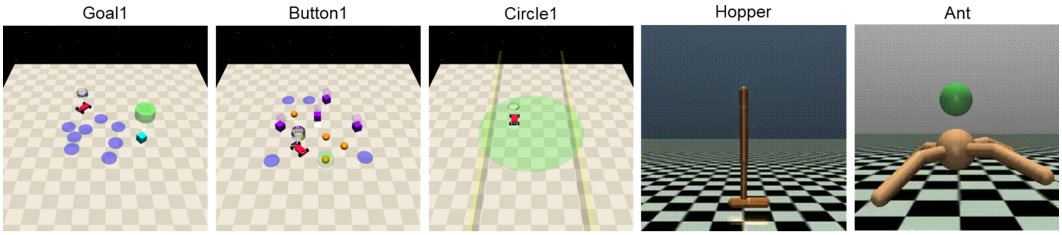


Figure 7: Illustration of five different tasks: Goal1, Button1, Circle1, Hopper, and Ant.

Goal. The agent's objective is to reach the goal buttons while avoiding static obstacles. Once the agent presses the correct button, a new goal button is randomly selected. The agent earns positive rewards for moving toward the goal and a bonus for successfully reaching it. Penalties are applied as costs for violating safety constraints, such as colliding with static obstacles or pressing the wrong button. The observation space includes the agent's ego states and sensory information about the obstacles and the goal, represented by pseudo LiDAR points. We use a Car robot in this environment and set the difficulty to level 1, naming it *CarGoal1*.

Button. This task is a more challenging version of Goal, featuring dynamic obstacles in addition to static ones. The dynamic obstacles move continuously along circular paths, requiring the agent to navigate to the goal while avoiding both static and dynamic obstacles. Compared to Circle and Goal tasks, Button demands greater inference capabilities as the agent must deduce the states of surrounding obstacles from raw sensory data. We use a Car robot and set the difficulty to level 1, naming it *CarButton1*.

Circle. In this task, the agent controls a robot to move clockwise along a circular path. Rewards increase as the agent's velocity rises and it stays closer to the circle's boundary. The safety zone is defined by two parallel plane boundaries intersecting the circle, and the agent incurs a penalty of 1 for leaving this zone. The observation space includes the car's ego states and sensory information

Algorithm 3 GMM-SSAC: Gaussian Mixture Model-Based Supremum CVaR-Guided Safe Soft Actor-Critic

Input: Initial policy π_θ, safety critic parameters ϕ, reward critic parameters ψ_1, ψ_2, entropy weight λ, safety weight κ, risk level α, safety threshold d, learning rates $\eta_\theta, \eta_\phi, \eta_\kappa$, target smoothing factor τ, and replay buffer $\mathcal{D}$.
Initialize target networks for safety and reward critics with parameters ϕ', ψ_1', ψ_2'.
while not converged **do**
  Sample action $a_t \sim \pi_\theta(\cdot \mid s_t)$, observe next state s_{t+1}, reward r_t, and cost c_t.
  Store transition $(s_t, a_t, r_t, c_t, s_{t+1})$ in replay buffer $\mathcal{D}$.
  for each gradient step **do**
    Sample a mini-batch of transitions $(s, a, r, c, s') \sim \mathcal{D}$.
    Update Safety Critic:
    Perform incremental Bellman update for safety critic:

$$\begin{aligned}\Psi_{\text{update}}(s,a) &= \mathcal{R}(\Psi(s,a), \Psi_{\mathcal{B}}(s,a), \beta) \\ &= \{x_1, \ldots, x_{M_1},\, c(s,a) + \gamma x_1', \ldots, c(s,a) + \gamma x_{M_2}' \mid M_1 : M_2 = (1-\beta) : \beta\}\end{aligned}$$

    Fit GMM parameters $\Gamma^{\pi}_{\text{update}}(s,a) = \{(\mu_k^{\text{update}}, \sigma_k^{\text{update}}, \omega_k^{\text{update}})\}_{k=1}^{K}$ to $\Psi_{\text{update}}(s,a)$.
    Update safety critic network parameters ϕ by minimizing the MSE loss:

$$\mathcal{L}_{\text{safety}} = \sum_{k=1}^{K} \Big[(\mu_k - \mu_k^{\text{update}})^2 + (\sigma_k - \sigma_k^{\text{update}})^2 + (\omega_k - \omega_k^{\text{update}})^2 \Big].$$

    Update Reward Critic:
    Minimize Bellman residuals for $Q_r^{\pi_\theta}$ using double-Q learning.
    Compute SCVaR:

$$\Lambda^{\pi}(s,a,\alpha) = \sup_{k \in \{1,\ldots,K\}} \left(\mu_k + \sigma_k \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha} \right).$$

    Update Policy:
    Minimize actor loss:

$$J_\pi(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\lambda \log \pi_\theta(a \mid s) - \big(Q_r^{\pi}(s,a) - \kappa \Lambda^{\pi}(s,a,\alpha) \big) \right].$$

    Update Safety Weight:
    Minimize safety loss:

$$J_s(\kappa) = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\kappa \big(d - \Lambda^{\pi}(s,a,\alpha) \big) \right].$$

    Perform gradient steps for $\theta, \kappa, \phi, \psi_1, \psi_2$.
    Target Network Updates:
    Update target networks:

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi', \quad \psi_1' \leftarrow \tau\psi_1 + (1-\tau)\psi_1', \quad \psi_2' \leftarrow \tau\psi_2 + (1-\tau)\psi_2'.$$

  end for
end while
Output: Optimized policy π_θ.

about the boundary. We use a Car robot in this environment and set the difficulty to level 1, naming it *CarCircle1*.

HopperVelocity. This task requires the agent to control a hopper robot to move as quickly as possible while adhering to velocity constraints. Rewards are given for achieving high speeds, while penalties of 1 are applied if the velocity exceeds a predefined threshold, set to 50% of the hopper's maximum velocity determined after Proximal Policy Optimization (PPO) training for 10^7 steps. This task emphasizes balancing speed optimization with safety constraints, naming it *Hopper*.

AntVelocity. Similar to HopperVelocity, this task involves controlling a quadruped ant robot under the same velocity constraints and reward structure. The velocity threshold is set to 50% of the ant's maximum velocity obtained after PPO training for 10^7 steps. Due to the ant's higher degrees of freedom, this task presents additional challenges in balancing speed, stability, and adherence to safety constraints, naming it *Ant*.

D.2 Implementation details & Hyper-parameter settings

Compute Resources. All experiments were run locally on a machine with an NVIDIA RTX 3090 GPU (24GB), 32GB RAM, and an Intel Core i7-12700KF CPU. The system ran Ubuntu 20.04 with Python 3.9, PyTorch 2.0, and CUDA 11.8. Each training run took 8–12 hours depending on the environment and GMM complexity, with total compute estimated at 4,000 GPU-hours. Early-stage experiments and failed runs are not included in this estimate.

Baselines. We use the official implementations from the respective codebases:

- WC-SAC(Yang et al., 2021): `https://github.com/AlgTUDelft/WCSAC`
- CAL(Wu et al., 2024): `https://github.com/ZifanWu/CAL`
- SAC & SAC-Lag: `https://github.com/PKU-Alignment/omnisafe` (a comprehensive framework for Safe RL algorithms (Ji et al., 2023))

We adopt the default hyper-parameter settings from the original implementations. Additionally, for WC-SAC, the risk hyperparameter α is set to 0.1. The safety threshold d is configured as follows: 10 for CarGoal1 and CarButton1, and 25 for CarCircle1, Hopper, and Ant. These settings are consistent with the on-policy method CVPO (Liu et al., 2022) and the OmniSafe framework (Ji et al., 2023).

GMM-SSAC. The detailed settings for GMM-SSAC are summarized in Table 1.

Table 1: Hyper-parameter settings for GMM-SSAC.

Parameter	**Setting**
Policy network sizes	[256, 256]
Q network sizes	[256, 256]
Network activation	ReLU
Discount factor γ	0.99
Reward Critics learning rate	1×10^{-3}
Cost Critics learning rate	1×10^{-3}
Actor learning rate	3×10^{-4}
NN optimizer	Adam
Number of GMM components(K)	4
Blending ratio (β)	0.6
Number of samples(M)	500

D.3 Derivation of the Discounted Threshold

In Safe RL, the cost threshold D ensures safety constraints during training. When using discounted costs, the total cost must account for the discount factor γ.

The discounted threshold d adjusts the cost limit D to reflect discounting. The total cost in an episode, discounted by γ^t, is:

$$C_{\text{Total}} = \sum_{t=1}^{T} \gamma^t \cdot C_t.$$

Assuming constant cost per time step $C_t = \frac{D}{T}$, we have:

$$C_{\text{Total}} = \frac{D}{T} \cdot \sum_{t=1}^{T} \gamma^t = \frac{D}{T} \cdot \frac{1-\gamma^T}{1-\gamma}.$$

Thus, the discounted threshold d is:

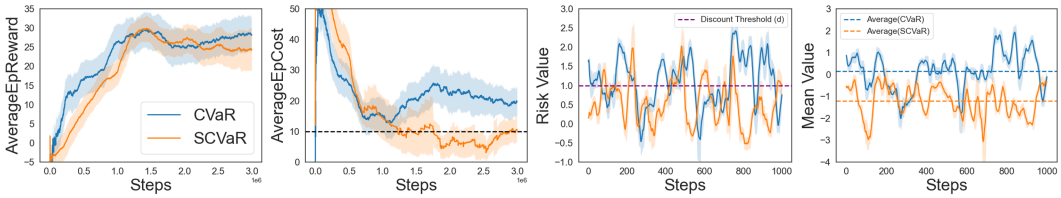
$$d = \frac{D \cdot (1-\gamma^T)}{(1-\gamma) \cdot T}.$$

Here, D is the cost limit, γ is the discount factor, and T is the maximum episode length (e.g., $T = 1000$).

The discounted threshold d ensures the total cost stays within the original limit D, even with discounted future costs, and is essential for enforcing safety constraints in RL tasks.

E Ablation Study

E.1 SCVaR vs. CVaR



(a) Reward comparison. (b) Cost comparison. (c) SCVaR/CVaR values. (d) Mean values.

Figure 8: Comparison of SCVaR and CVaR in terms of reward and cost performance, and GMM distribution.

We compare GMM-SSAC models trained in the CarGoal1 environment using SCVaR and Monte Carlo-based CVaR, as shown in Fig. 8.

To establish a fair baseline, CVaR is estimated empirically via Monte Carlo sampling. Given a learned GMM distribution

$$f_L(x) = \sum_{k=1}^{K} w_k \cdot \mathcal{N}(x \mid \mu_k, \sigma_k^2),$$

we draw $N = 5{,}000$ i.i.d. samples $\{x_j\}_{j=1}^{N}$ at each state-action pair. The empirical α-level VaR is the $(1-\alpha)$-quantile of the sampled values:

$$\widehat{\text{VaR}}_\alpha = \text{Quantile}_{1-\alpha}(\{x_j\}_{j=1}^{N}),$$

and the corresponding CVaR is the average of samples beyond this threshold:

$$\widehat{\text{CVaR}}_\alpha = \frac{1}{|\mathcal{I}|} \sum_{j \in \mathcal{I}} x_j, \quad \mathcal{I} = \{j \mid x_j \geq \widehat{\text{VaR}}_\alpha\}.$$

This approach avoids reliance on closed-form solutions and provides a flexible, data-driven estimate of tail risks. We adopt it as the practical CVaR baseline throughout our experiments.

Fig. 8(a) and Fig. 8(b) show that both models achieve similar rewards, but SCVaR achieves lower costs and consistently satisfies the safety threshold. To further understand this difference, both models were evaluated under identical random seeds and environment settings. Their estimated SCVaR (CVaR) values and the means of the GMM density functions are presented in Fig. 8(c) and Fig. 8(d). While the two methods yield similar tail risk estimates close to the discounted threshold in Fig. 8(c), SCVaR produces a smaller mean in Fig. 8(d), indicating that it enforces a stronger focus on the worst-case tail.

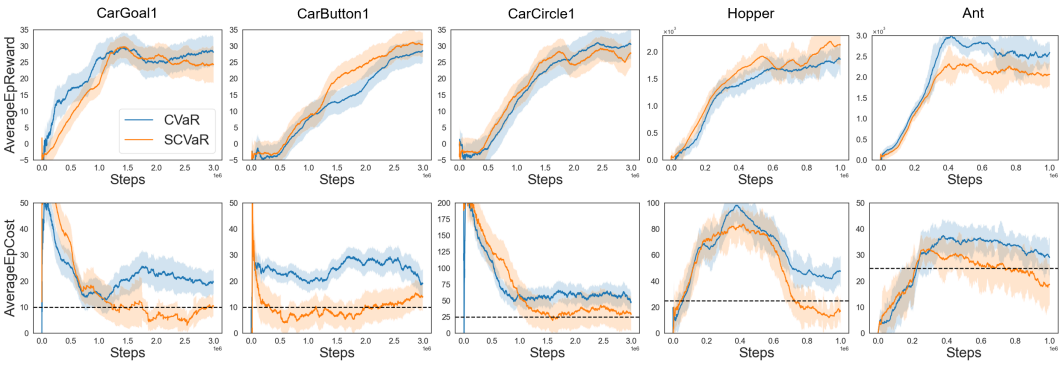


Figure 9: Additional experimental results for all environments.

Overall, these results demonstrate that SCVaR places greater emphasis on rare but extreme risks than Monte Carlo CVaR, leading to superior cost reduction even when such events occur with low probability. Consistent improvements across additional settings are reported in Fig. 9.

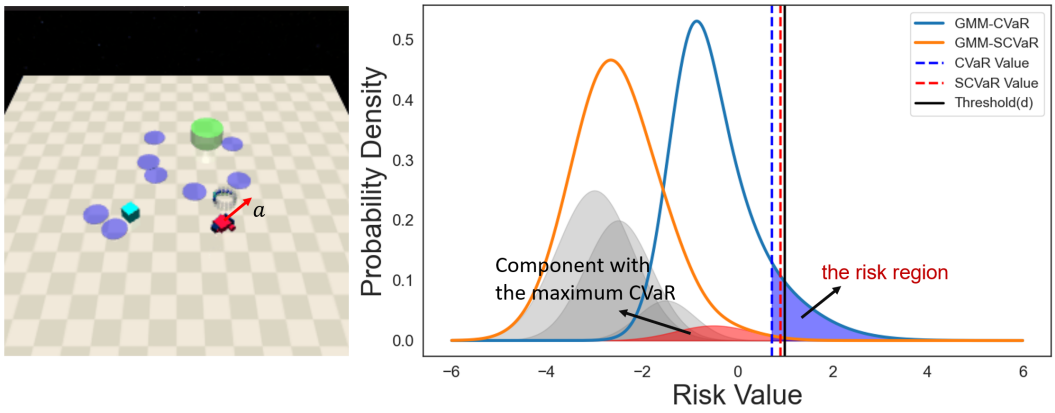


Figure 10: Visualization of the GMM cost distribution output by the Safety Critics in the SCVaR and CVaR models.

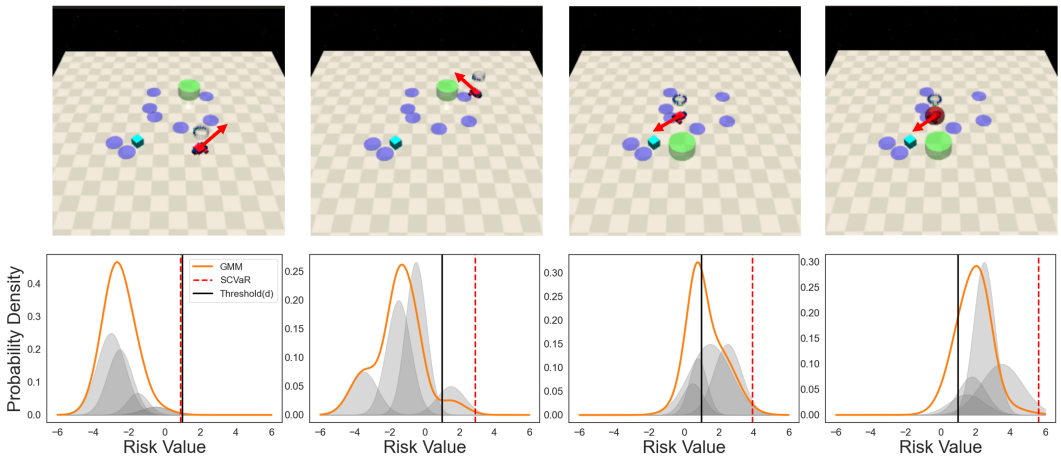


Figure 11: Cost distribution visualization for four scenarios.

E.1.1 Visualization of Cost Distribution

We input the same state-action pair into the Safety Critics of both models and visualize the GMM distributions they output, as shown in Fig. 10. Both models determine that the state-action pair is safe (SCVaR or CVaR $< d$). However, from the distribution curves, it is clear that SCVaR is much more conservative. It only considers the state-action pair safe when the high-risk region has almost no probability, and continuously adjusts its distribution during training to guide the policy network toward safer actions. In contrast, CVaR considers the state-action pair safe even when there remains a significant portion of high-risk areas, suggesting that this Safety Critic is not sufficiently "reliable."

Fig. 11 visualizes the cost distribution for four cases, representing safe, less safe, less dangerous, and dangerous scenarios. The SCVaR values accurately reflect risk levels, increasing with danger and exhibiting heavier upper tails. Note that the agent's policy is not optimal, as a fully converged policy would rarely encounter dangerous situations.

E.2 GMM-based Variant in Baselines

We extend GMM-based safety critics to existing SafeRL baselines by replacing their original cost critics with GMM-based variants. Specifically, we implement GMM variants for SAC-Lag and CAL. Since SAC lacks a safety critic and WC-SAC with GMM + SCVaR already forms the basis of GMM-SSAC, no further variants are constructed for these algorithms.

We evaluate the modified baselines on two representative tasks—CarGoal1 from Safety-Gymnasium and Ant-Velocity from MuJoCo. As shown in Fig. 12, both SAC-Lag and CAL benefit significantly from the incorporation of GMM-based critics. In particular, the GMM variants yield higher reward performance while maintaining lower cumulative cost, indicating improved safety-reward trade-offs. This improvement is consistent across both tasks, suggesting that GMM critics can serve as a plug-in module to enhance the performance of diverse SafeRL frameworks.

These findings demonstrate that the use of GMM-based safety critics is not limited to our proposed method but can generalize to other constrained RL algorithms, offering a principled and effective alternative to conventional cost estimators.

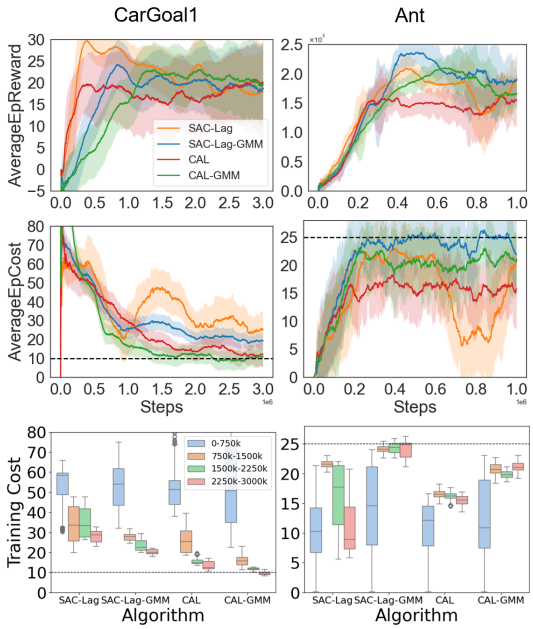


Figure 12: Performance comparison of SAC-Lag and CAL before and after substituting their cost critics with GMM-based variants.

E.3 Component-Level Interpretability

We analyze whether GMM-based SCVaR components align with specific safety violations, as shown in Fig. 13. The CarButton1 environment was selected for its clear delineation of different safety risks, with three distinct violation types: **Gremlins** (contact-based penalties), **Wrong Buttons** (costs from incorrect button presses), and **Hazards** (proximity-based risks). This setup allows us to evaluate the relationship between GMM components and different violation types.

In total, we evaluate 10,000 (s, a) samples, with 2,500 samples per category: Safe, Gremlins, Wrong Buttons, and Hazards. Fig. 13(a) shows the normalized proportion of each SCVaR component activated under these violation types, while Fig. 13(b) visualizes the sample counts with a heatmap. The results show clear alignment between components and violation types: Component 3 predominantly captures Gremlins, Component 2 specializes in Wrong Buttons with some overlap with Gremlins, Component 1 is mainly associated with Hazards, and Component 0 is more evenly distributed across

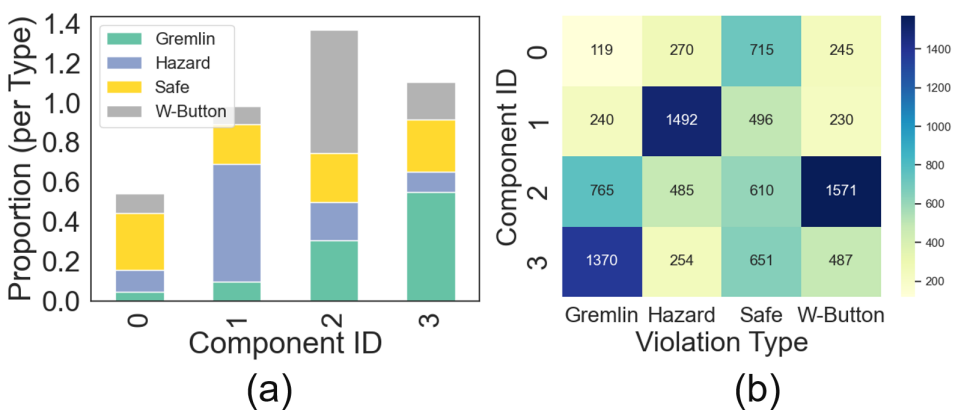


Figure 13: SCVaR component activation across safety violation types. (a) Normalized stacked bar chart of component dominance per violation type; (b) Heatmap of sample counts per component-violation pair. Components 3/2/1 correspond to Gremlins, Wrong Buttons, and Hazards respectively, while Component 0 serves as a fallback across mixed cases.

all types, acting as a fallback in ambiguous cases. These findings highlight that GMM-based SCVaR naturally separates distinct violation types without explicit supervision, demonstrating its potential for uncovering structured risk semantics in safety-critical environments. Further visualization are provided in Fig. 14.

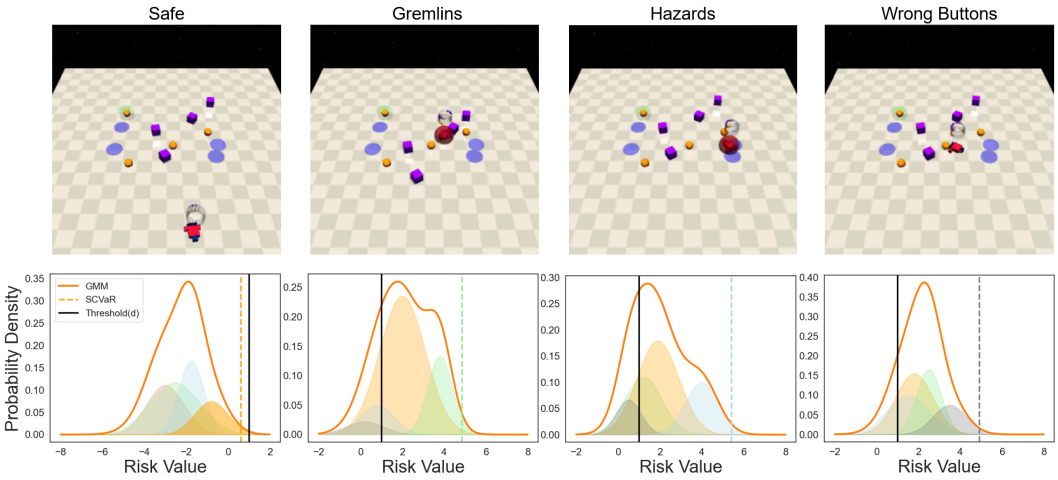


Figure 14: Cost distribution and GMM component visualization for four representative states in the CarButton1 environment. Each GMM component is assigned a consistent color across subplots (e.g., components 0–3 are colored orange, blue, gray, and green, respectively), and grouped according to their output order. The figure reveals that different types of safety-violating states are primarily associated with distinct GMM components, as indicated by color-coded contributions to SCVaR. This highlights the ability of GMM-based satety critics to disentangle different risk patterns via distinct components.

F Related Works

F.1 Safe Reinforcement Learning

Recent advances in Safe RL have introduced a variety of methods to ensure safety during RL training. Early Safe RL approaches were heavily influenced by control theory. Lyapunov functions are widely used in control theory to guarantee safety by constraining the action space during exploration

(Chow et al., 2019; Huh & Yang, 2020; Jeddi et al., 2021). However, defining suitable Lyapunov functions often requires a system model, which may not be readily available in general RL scenarios. In contrast, Lagrangian-based approaches, particularly primal-dual optimization (Paternain et al., 2022), have gained significant attention due to their flexibility and broad applicability. These methods have been shown to achieve a zero duality gap under certain conditions, providing theoretical guarantees for constraint satisfaction (Paternain et al., 2019). Among these approaches, risk-constrained primal-dual methods (Chow et al., 2018) focus on developing efficient reinforcement learning algorithms for risk-constrained MDPs, where risk is typically represented through chance constraints or constraints on the CVaR of the cumulative cost. Additionally, reward-constrained methods (Tessler et al., 2018) utilize alternative penalty signals to guide the policy towards satisfying safety constraints. Alternatively, robust MDP methods (Iyengar, 2005; Wang & Zou, 2021) aim to learn policies that perform well under worst-case transition dynamics, but they often lead to overly conservative strategies and require specifying uncertainty sets. Furthermore, the Natural Policy Gradient Primal-Dual (NPG-PD) method (Ding et al., 2020) is the first to establish non-asymptotic convergence guarantees.

Similarly, our work builds upon primal-dual optimization methods. Among the most relevant recent developments, two works are particularly aligned with our framework. WCSAC (Yang et al., 2021) estimates the risk distribution using a unimodal Gaussian and extends the SAC-Lag method by incorporating a variance estimator to enhance risk control. CAL (Wu et al., 2024), on the other hand, addresses cost underestimation by employing an upper confidence bound (UCB) for the cost value, thereby improving risk management during policy optimization. However, these methods rely on Gaussian distributions to approximate risk distributions, overlooking the inherent limitations in their expressiveness.

F.2 GMMs IN RL

GMMs have demonstrated significant potential in RL by effectively modeling complex data distributions, particularly in state-action spaces, policy representations, and value functions. Initially proposed to address challenges in continuous state-action spaces (Agostini & Celaya, 2010), GMMs model the density of observed samples in the joint space of states, actions, and Q-values. This approach provides substantial modeling flexibility. In policy optimization, GMMs have been widely used to construct mixture-based policies, where components are selected dynamically at each step to enable adaptive and robust decision-making (Baram et al., 2021; Haarnoja et al., 1861; Kim et al., 2022). To address the intractability of computing entropy in diffusion-based policies, a recent online RL algorithm estimates policy entropy using a GMM fitted to periodically sampled actions, enabling a more effective trade-off between exploration and exploitation during training (Wang et al., 2024). In value function approximation, GMM-based Q-function approximators (GMM-QFs) are integrated into the Bellman residual to enable efficient optimization (Vu & Slavakis, 2024). Additionally, in distributed RL, mixture density networks have been applied to model reward distributions. The introduction of the Jensen-Tsallis distance metric further refines distance computations between Gaussian mixture distributions, enhancing the performance of methods such as DQN and C51 (Choi et al., 2019). Our work also leverages GMMs for value function approximation but focuses specifically on safety (risk-sensitive) value functions.

G THE USE OF LARGE LANGUAGE MODELS (LLMS)

We acknowledge the use of large language models (LLMs) in preparing this paper. LLMs were employed exclusively as an assistive tool for language refinement, including improving clarity, grammar, and readability of the manuscript. They were not used for research ideation, algorithm design, experimental implementation, or result generation. All technical contributions, theoretical analyses, and experimental results are solely the work of the authors.

The authors take full responsibility for the correctness and integrity of the paper's content.