RACER: Epistemic Risk-Sensitive RL Enables Fast Driving with Fewer Crashes

Kyle Stachowicz, Sergey Levine
UC Berkeley
kstachowicz@berkeley.edu, svlevine@eecs.berkeley.edu

***Abstract*—Reinforcement learning provides an appealing framework for robotic control due to its ability to learn expressive policies purely through real-world interaction. However, this requires addressing real-world constraints and avoiding catastrophic failures during training, which might severely impede both learning progress and the performance of the final policy. In many robotics settings, this amounts to avoiding certain "unsafe" states. The high-speed off-road driving task represents a particularly challenging instantiation of this problem: a high-return policy should drive as aggressively and as quickly as possible, which often requires getting close to the edge of the set of "safe" states, and therefore places a particular burden on the method to avoid frequent failures. To both learn highly performant policies and avoid excessive failures, we propose a reinforcement learning framework that combines risk-sensitive control with an adaptive action space curriculum. Furthermore, we show that our risk-sensitive objective automatically avoids out-of-distribution states when equipped with an estimator for epistemic uncertainty. We implement our algorithm on a small-scale rally car and show that it is capable of learning high-speed policies for a real-world off-road driving task. We show that our method greatly reduces the number of safety violations during the training process, and actually leads to higher-performance policies in both driving and non-driving simulation environments with similar challenges.**

I. Introduction

Reinforcement learning (RL) can in principle allow robots to perform complex and delicate behaviors, such as driving at high speed over rough terrain, while adapting to the particular environment in which they are trained. However, instantiating such methods while training directly in real-world environments presents a unique set of challenges. The training process is no longer free of consequences, and catastrophic failures during training can impede learning progress, damage the robot, and require costly manual intervention where a person needs to reset the robot. This makes standard RL methods most difficult to apply in precisely the high-performance settings where they might be most beneficial.

In this paper, we study this challenge in the context of high-speed off-road driving. While learning to drive quickly over uneven terrain, high-speed crashes and rollover events can both damage the vehicle and disrupt the learning process, harming final performance. However, when the primary objective is to maximize driving speed, there is a tension between safety and performance: safety requires that the robot stays within a safe region, while achieving maximum performance requires the robot to operate at the edge of this set (as depicted in Fig. 2).

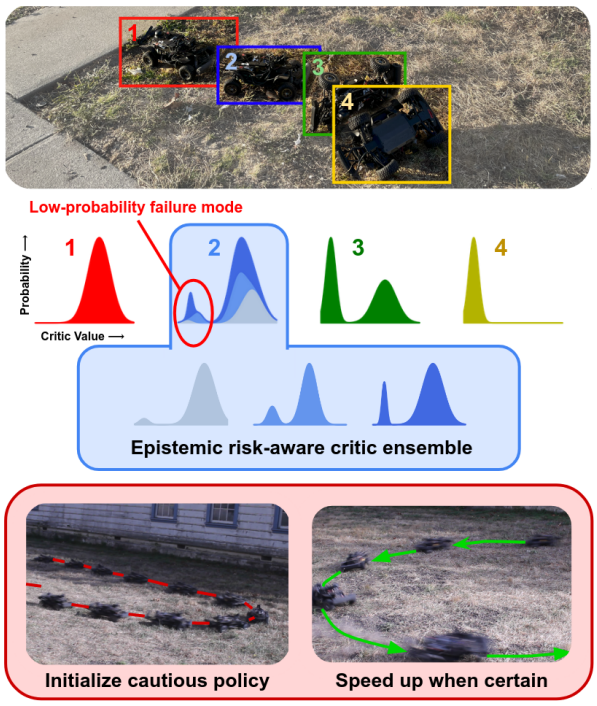

Fig. 1: Our method enables high-speed driving with fewer crashes during training. Rare failure events (such as crashes or rollovers) often appear in the return distribution as a low-probability, low-return mode that do not contribute heavily to the expected value of the return. By applying a risk-sensitive actor objective (CVaR) to a distributional critic that incorporates epistemic uncertainty and can reason about these rare events, our method simultaneously modulates the robot's action limits and learns a risk-sensitive policy.

Many existing approaches to safety in RL [44, 47] consider safety at the *end* of training, once the policy has converged and the return is maximized. However, when training in the real world, it is also crucial to consider safety *during* online training, before convergence [8]. During this period, the robot often encounters unfamiliar states during exploration as well as states that are not yet well fit by the training algorithm. The performance of the final learned policy depends critically on the robot's ability to avoid catastrophic failures consistently enough to learn high-performance policies.

We therefore aim to develop a reinforcement learning method that enables real-world robotic systems to learn high-performance behaviors (e.g., high-speed driving) while minimizing failures during training. Our key insight is twofold: firstly, to avoid rare failure events like rollovers or crashes,

the agent must both model and effectively respond to low-probability outlier events, even when they are uncertain. Secondly, in many high-performance settings, it is relatively easy to obtain robust low-performance behavior (e.g., low-speed driving), and increase performance over time as the agent's becomes more certain about the risky high-performance regime.

The primary contribution of this paper is **R**isk-sensitive **A**ctor **C**ritic with **E**pistemic **R**obustness (RACER), a method for imbuing model-free RL agents with risk sensitivity to account for uncertainty over returns. We build upon distributional RL [7], which models the full distribution of returns rather than a mean point estimate. However, unlike the standard distributional RL setting, we explicitly model two types of uncertainty: *aleatoric* uncertainty, which refers to the irreducible uncertainty in the returns (e.g. due to stochastic environment dynamics) and *epistemic* uncertainty, corresponding to lack of knowledge of the true return distribution due to incomplete data and transient underfitting of newer data that has not yet been fit by the training process.

We model the return distribution as an ensemble of independently trained distributional neural networks. Each individual ensemble member models aleatoric uncertainty in its distributional output, and epistemic uncertainty is measured by the ensemble as a whole. We propose to optimize the conditional value at risk (CVaR) of the distributional critics, which considers an expectation over the α worst-case distribution. We show that this naturally results in an agent that avoids taking actions leading to high *epistemic* risk, as well as the avoidance of actions with highly stochastic returns (aleatoric uncertainty).

We also propose a risk-sensitive mechanism for scheduling exploration in this setting. RACER starts by using only a small (cautious) subset of the allowable action space, and then slowly increases the range of allowable actions over time according to a similar CVaR-based objective to that optimized by the actor. We increase the action limits only when the critics are confident that actions near the existing limits are safe. This combination of risk-sensitive control and adaptive action space bounds provides for cautious exploration in unfamiliar situations, and avoids high-risk situations in familiar settings, leading to fewer crashes and better final performance.

We test RACER on a real-world tenth-scale autonomous vehicle performing aggressive off-road maneuvers, and show that our method allows the robot to reach $>10\%$ higher speeds at convergence while cutting failures during the course of training by more than half, and almost entirely eliminating high-speed failures. We find that RACER compares favorably to several baselines in both driving and non-driving tasks, and demonstrate the importance of each component of our algorithm in reducing failures during training via ablation studies.

II. Related Work

Risk-sensitive RL. Several recent algorithmic developments aim to introduce risk-sensitive metrics to RL and control [12, 17]. Tang et al. [49] proposes a similar CVaR-based objective for training agents that are robust to *aleatoric* uncertainty about their environment. While these works demonstrate risk-sensitive training in simulation with on-policy methods, we propose a novel integration of CVaR with efficient off-policy algorithms in place of policy gradients [38] or model-based methods [27], enabling rapid training directly in the real world.

Yang et al. [55] propose a CVaR-based actor-critic algorithm to act as a safety critic, but their method is restricted to a Gaussian distribution representing aleatoric uncertainty only. RACER suppports a flexible critic distribution – allowing representation of long-tailed failures that would be ignored with a Gaussian [2] – and handles epistemic uncertainty directly (see Section IV-B), allowing the resulting system to minimize safety violations *during* training, rather than just at policy convergence.

Constrained RL. The constrained RL approach casts safety as a *constrained MDP* [3]. Prior work [1, 5, 47, 50, 51] largely considers the problem of learning a policy to minimize test-time failures after the end of training. Safety critics train a discriminator model from offline datasets [8] or online (unsafe) interactions enabled via simulation [45].

Safe RL. Alternative approaches have focused on system-level safety [24] by merging classical techniques such as control barrier functions [14] or min-max robustness [48] with reinforcement learning. However, these control-theoretic approaches typically require a high-quality model of the system with simple parametric uncertainty [15], or require building an explicit model of how actions affect safety [20]. In contrast, our method belongs to the class of model-free reinforcement learning algorithms and therefore makes no assumptions about the structure of the dynamics.

High-Speed Driving. Substantial prior work considers high-speed driving in the classical control setting [25]. These approaches require deriving a dynamics model from first principles and fitting parameters according to physically measured quantities or system identification. Unfortunately it is difficult to handcraft a dynamics model for off-road driving. Model-predictive control methods with learned models [53] have been applied with some success in these domains. However, MPC-based methods have difficulty handling high-dimensional observations, as planning suffers from compounding error [18]. Because of these limitations, these works largely consider a simplified planar state-space. In the on-road driving setting, several works have considered the problem of modeling and preventing rollover, primarily in the quasistatic [32] case or by assuming a flat ground plane with constant wheel contact [29], which cannot be assumed in the off-road setting. In comparison, our model-free approach aims to handle complex environments where an accurate hand-designed model might be difficult or impossible to obtain due

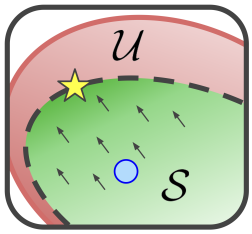

Fig. 2: In high-speed driving, maximizing speed (arrows) requires operating on the boundary of the safety set $\mathcal{S}$ (optimal policy: star) to avoid unsafe states $\mathcal{U}$. Enforcing strict safety yields an overly conservative policy (blue).

to complex or chaotic system dynamics. In these settings a true safety *guarantee* would out of necessity require a very conservative policy. Nevertheless, we find that our method greatly reduces failures during the training process and results in a highly performant policy.

Reinforcement learning has also been applied to high-speed driving in simulation [11, 21]. The closest prior work on a real robot is [46], which also learns a fast driving policy through real-world interactions. However, unlike Stachowicz et al. [46], RACER is able to explicitly consider safety during the training process. We present a real-world comparison between the two methods in Sec. V.

III. Preliminaries

We first introduce standard notation and definitions in reinforcement learning and risk-sensitive optimization.

A. Reinforcement learning

Consider a Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$ defined by a state space, an action space, a transition density, and a reward function respectively. We specifically consider the case of *non-episodic* RL, in which the robot does not receive external state resets to some stationary initial distribution, except in the case of catastrophic failure, and these external resets are very expensive. This is reflective of the real world, where external resets may be unavailable or require human supervision. We wish to learn a high-quality policy while minimizing the number of catastrophic failures (and hence, expensive external resets) across the course of training.

Q-learning methods approach the RL problem by learning a mapping $Q(s, a)$ from state-action pairs to the expected discounted return of the current policy. $Q(s, a)$ can be "bootstrapped" by iteratively performing Bellman updates:

$$Q^{k+1}(s, a) \leftarrow \mathop{\mathbb{E}}_{r, s'} \left[r + \gamma \max_{a'} Q^k(s', a') \right]$$

Actor-critic methods extend Q-learning to continuous actions by sampling a' from some function approximator trained to maximize $Q(s, a)$. These methods have proven relatively sample efficient [41], motivating their use in learning directly in the real world rather than performing sim-to-real transfer [56]. In such methods, we learn a policy π and a critic function $Q^\pi(s, a)$ representing the expected discounted return from taking action a for one step and then following the policy π. In deep RL, we typically parameterize both the actor and the critic with neural networks π_θ and Q_ϕ.

B. Distributional RL

In the standard actor-critic setting we learn a critic that represents the *expected* future discounted return and optimize an actor with respect to the critic. The *distributional* RL perspective [7, 19] instead learns the distribution of the returns $Z(s, a)$ and then maximizes its expectation $Q(s, a) = \mathbb{E}\, Z(s, a)$.

The simplest approach to distributional RL assumes the return distribution is Gaussian and parametrizes $Z(s, a)$ by its mean and a variance [34]. However, it is often desirable to represent Z with a more flexible distribution. Two popular choices are a categorical distribution, in which the critic is optimized by minimizing KL divergence [7], and a quantile distribution with an L1-type loss [19]. These methods tend to enable better feature learning and are a key component of state-of-the-art RL algorithms in discrete domains [28].

In distributional RL, we can no longer minimize MSE to perform our approximate Bellman backup (as our critic is now a probability distribution rather than a point estimate). Instead, we typically minimize the KL-divergence to our target distribution $Z_{\text{targ}} = r + \gamma Z(s', a')$. When Z is represented as a categorical distribution the KL-divergence may be undefined because the atoms of Z and the target distribution are not aligned. The backup is approximated by projecting Z_{targ} onto the atoms of Z; see [7] for further details.

C. Conditional Value at Risk

When considering safety, is desirable to be conservative in the face of uncertainty: all else being equal, we should prefer the certain outcome over an uncertain one with the same expected value. Particularly when there exists some binary indicator representing failure, returns will tend to follow a multimodal distribution: there is a high-probability, high-return mode corresponding to nominal behavior, and a low-probability, low-return mode corresponding to failure. Because failures require costly intervention, may damage the robot, and can harm the learning process, it is important to avoid this low-return mode, motivating a risk-sensitive formulation.

During training and exploration it is particularly important to be conservative with respect to *epistemic* uncertainty; that is, risk corresponding to missing data or to a model that has not yet converged. Ideally our notion of risk should effectively handle both epistemic and aleatoric uncertainty (stochasticity). While it is possible to be sensitive to aleatoric risk by increasing the penalty for negative events [52], considering epistemic risk requires applying a risk metric to the actual (epistemic) return distribution.

CVaR is one such instantiation of risk-sensitivity [2]. Let Z be some scalar random variable. We can write CVaR as:

$$\text{CVaR}_\alpha(Z) = \mathbb{E}[Z | Z < \text{VaR}_\alpha(Z)],$$

where $\text{VaR}(Z)$ (the *value at risk*) is the α-worst percentile value of Z. In our case we consider the CVaR of the return distribution, which reflects the discounted return distribution of a state-action pair under the current policy.

CVaR has many favorable properties as an optimization objective. It is less pessimistic than the worst-case value (the minimum value attainable, no matter how unlikely), which can be extremely noisy and even unbounded for stochastic systems. However, it tends to represent risk more effectively than value-at-risk [2], while being more stable with respect to the underlying probability distribution [16].

Additionally, CVaR is a *convex* risk measure in the sense that $\text{CVaR}_\alpha(\mu Y + \nu Z) \leq \mu \text{CVaR}_\alpha(Y) + \nu \text{CVaR}_\alpha(Z)$ for $\mu + \nu = 1$ and $\mu, \nu \geq 0$. We present a short proof in the appendix; see Pflug [37] for a detailed derivation.

We aim to optimize the CVaR of Z, which is difficult to compute in dynamic settings because of *time inconsistency* [4, 9]. This property prevents optimization via dynamic programming, as the true optimal policy is non-Markov and depends on accrued rewards. Instead, we consider the return distribution and optimize $\text{CVaR}_\alpha(Z)$ under a Markov policy. While previous works take advantage of a closed-form expression for CVaR in the case of Gaussian return distribution [55] or require use of on-policy (policy gradient-based), we use a more flexible categorical model of Z and show that its CVaR can still be optimized using an efficient off-policy algorithm.

IV. Risk-Sensitive Exploration with RACER

RACER (Fig. 3) is composed of three primary components: a risk-sensitive actor trained with a CVaR-based objective, a distributional critic function incorporating both epistemic and aleatoric uncertainty, and an adaptive action limit that starts by restricting the policy to take "cautious" actions but becomes less restrictive over time. Both the actor and the adaptive action limits are updated based off of the distributional critic, and therefore the critic's epistemic and aleatoric uncertainty. The adaptive action limits act as a post-processing step on the actor network. We denote the distributional critic as Z_ϕ, the (pre-limits) actor as $\bar{\pi}_\theta$, the adaptive limits as L_ψ, and the limited policy (combining the actor and the limits) as π_θ.

A. Risk-Sensitive Deep RL with CVaR

We optimize the CVaR objective via actor-critic RL, similarly to Yang et al. [55]. To provide a flexible critic distribution, we learn a *categorical* distributional value function $Z(s,a)$ [7] with distributional Bellman backups, where each iteration is a sample-based approximation to the update

$$Z^{k+1} \leftarrow \arg\min_Z \mathbb{E}\left[\text{KL}(Z(s,a)||r(s,a)+\gamma Z^k(s',a'))\right],$$

where $a' \sim \pi^k(s')$. Rather than optimizing the actor to maximize $Q(s,a) = \mathbb{E}Z(s,a)$, we optimize the CVaR with respect to the actor parameters θ over $a \sim \pi^k_{\theta\psi}(s)$:

$$\pi^k \leftarrow \max_\theta \mathop{\mathbb{E}}_{a\sim\pi_{\theta\psi}(s)} \text{CVaR}_\alpha(Z^k(s,a)).$$

In practice, the updates above are implemented approximately, using one step of stochastic gradient descent on the objective with respect to θ at each iteration:

$$\mathcal{L}_\pi = -\mathop{\mathbb{E}}_{a\sim\pi_\theta} \text{CVaR}_\alpha(Z_\phi(s,a)),$$

following the standard procedure for implementing off-policy (distributional) deep actor-critic methods [6, 26, 33].

We represent $Z(s,a)$ as a neural network outputting a categorical distribution over discrete bins. Unlike prior work which represented Z as Gaussian [55], there is not a simple closed-form solution for the CVaR in this context. Instead, we provide a differentiable procedure for computing CVaR of a categorical distribution in Sec. IV-D, which we then optimize to perform gradient descent on the actor.

B. Learning Distributional Epistemic Critics

Because CVaR is sensitive to the worst-case outcomes, its use as a policy objective implicitly penalizes states or actions resulting in highly uncertain distributions over returns. This means that CVaR will implicitly try to avoid states and actions with a high degree of uncertainty. There are two types of uncertainty: *aleatoric* uncertainty, or irreducible stochasticity in the environment, and *epistemic* uncertainty, which represents unknown quantities. While at convergence aleatoric uncertainty dominates, since the state distribution eventually stops changing as the policy converges, epistemic uncertainty is the more significant contributor to catastrophic failures during training, when the robot does not *know* for sure whether or not a failure might occur. In light of this, we consider two possible approaches to incorporating epistemic uncertainty in distributional RL.

1) Ensembled critics: rather than a single predictor, an ensemble of independent neural networks can be applied to capture epistemic uncertainty [23, 31, 39]. We train each ensemble member Z_{ϕ_i} individually with loss:

$$\mathcal{L}_{Z_{\phi_i}} = \mathop{\mathbb{E}}_{\substack{s,a,s'\sim\mathcal{D}\\ a'\sim\pi}} \left[\text{KL}\left(r+\gamma Z_{\phi_i'}(s',a')||Z_{\phi_i}(s,a)\right)\right].$$

By maintaining a collection of models with varied parameters, which are only ever trained to agree on in-distribution data, we expect the ensembles to disagree on out-of-distribution data [54].

Theorem 1. *Let Z_i be real-valued random variables with density $p_i(z)$. Denote the random variable with density $\hat{p}(z) = \frac{1}{N}\sum_i p(z)$ as $\hat{Z}$. Then for $\alpha > 0$:*

$$\textit{CVaR}_\alpha(P) \le \frac{1}{N}\sum_i \textit{CVaR}_\alpha(P_i) \tag{1}$$

*We call the positive difference $\frac{1}{N}\sum_i \textit{CVaR}_\alpha(Z_i) - \textit{CVaR}_\alpha Z_i$ the **ensemble CVaR gap**. (Proof in Appendix A)*

Theorem 2. *Assume Z_i has finite first moment. Then we have:*

$$\frac{1}{N}\sum_i \textit{CVaR}_\alpha(Z_i) - \textit{CVaR}_\alpha(\hat{Z}) \le \frac{1}{N}\sum_i \left\|\mathcal{T}_\alpha\hat{Z} - \mathcal{T}_\alpha Z_i\right\|_{\text{EMD}}, \tag{2}$$

*where $\mathcal{T}_\alpha X$ is the **tail** of the distribution X and $\|X-Y\|_{\text{EMD}}$ is the **earth mover's distance**. (Proof in Appendix A)*

In practice we find that this bound is relatively tight (Appendix B). This suggests that CVaR is conservative with respect to epistemic uncertainty represented as disagreement between individual members and the overall ensemble (Fig. 4).

2) Explicit entropy maximization: we draw inspiration from conservative Q-learning [30], in which out-of-distribution (OOD) actions are explicitly penalized by minimizing the critic on randomly-sampled actions. In our case, we wish to make OOD actions *more uncertain* rather than decreasing their mean. We use entropy as a proxy for uncertainty, reducing the entropy of the return distribution Z on seen datapoints and maximizing for sampled OOD actions. The modified critic loss

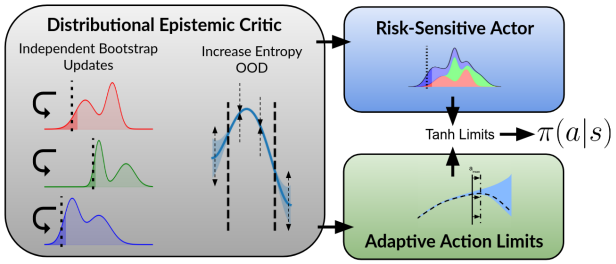

Fig. 3: RACER and its three main components. A distributional critic captures epistemic uncertainty via ensembling and explicit entropy maximization beyond action limits. A risk-sensitive actor and adaptive action limits use the distributional critic to increase speed over time while reducing failures during training.

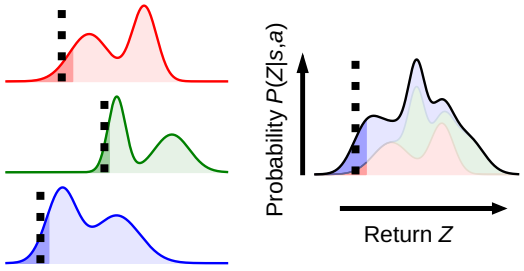

Fig. 4: CVaR naturally accounts for epistemic uncertainty when applied to the mixture distribution output of an ensemble. When the ensemble disagrees about the distribution, the CVaR of their mixture prioritizes more pessimistic ensemble members.

for a single ensemble member using this form of epistemic uncertainty is then as follows with new terms in blue:

$$\mathcal{L}_Z = \mathop{\mathbb{E}}_{\substack{s,a,s'\sim\mathcal{D}\\ a'\sim\pi_\theta}} \Big[\mathrm{KL}\left(r + \gamma Z_{\phi'}(s',a') || Z_\phi(s,a)\right) + \mathop{\mathbb{E}}_{a'\sim\pi_\theta} \mathcal{H}(Z_\phi(s,a')) - \mathop{\mathbb{E}}_{a'\sim\bar{\pi}_\theta} \mathcal{H}(Z_\phi(s,a'))\Big], \tag{3}$$

where ϕ' is a delayed copy of ϕ.

The out-of-distribution actions are drawn from any policy that samples actions outside of the action limits. We use $\bar{\pi}_\theta$, the *pre-limit* policy; that is, the policy distribution induced by π *before* applying adaptive action limits.

C. Adaptive Action Limits

In many real-world problems, including autonomous off-road driving, we have a strong prior in that "cautious" actions (for example, low speeds) tend to be safe. For example, for autonomous driving, our action space has two components: a servo command for the robot's steering in range $[-1, 1]$ and a velocity target for a low-level motor controller bounded by $[v^-, v^+]$, with the velocity target corresponding directly to cautious/risky actions. To adaptively adjust these bounds across the course of training, we want to increase our bounds whenever we are sufficiently certain about the return distribution $Z(s,a)$ for a *outside* (or at the edge) of those boundaries.

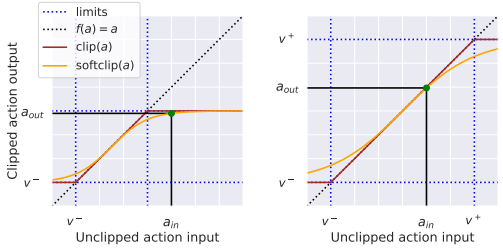

Fig. 5: RACER applies a shifted $\tanh$ to the output of the actor to enforce a particular set of limits. As the limits are adjusted (left → right), the mapped values of actions already within the limits change very little, while outputs for outside values change significantly.

Following [46], we use a shifted $\tanh$ function to soft-clip the actor's output into the desired range:

$$\mathrm{softclip}(a, v^-, v^+) = \eta \tanh\left(\frac{1}{\eta}(a-\mu)\right) + \mu; \quad \eta = \frac{v^+ - v^-}{2}, \mu = \frac{v^+ + v^-}{2} \tag{4}$$

This parameterization approximates $\mathrm{clip}(a, v^-, v^+)$, meaning that the bounds can be adjusted on the fly and for $a \in [v^-, v^+]$ the soft-clip will have minimal effect.

This differentiable approximation to clip allows us to adjust the action limits using the same risk-sensitive objective we use to optimize the policy. Because out-of-distribution actions (e.g., actions not appearing in any collected data) are subject to high epistemic uncertainty, maximizing the CVaR objective will only grow the limits for as long as the critic is certain. We thus adaptively adjust the limit it over the course of training by taking gradient steps to maximize:

$$\mathcal{L}_L(v^+) = -\mathop{\mathbb{E}}_{s\sim\mathcal{D}, a\sim\pi_\theta}\left[\mathrm{CVaR}_\alpha Z(s, \mathrm{softclip}(a, v^-, v^+))\right].$$

D. Implementation Details

For real-world training we apply sample-efficient RL by applying many more gradient steps on the critic than the number of steps we take in the environment [13, 22]. As observed in the literature [13, 22, 41], this type of sample-efficient RL algorithm often requires regularization to avoid overfitting to early samples from the non-stationary data distribution. Our method already uses ensembling [13]; we additionally apply weight decay to the critic network.

We compute the CVaR of a categorical distribution in a differentiable fashion. As shown in Algorithm 1, we compute CVaR by finding the cumulative distribution of the distribution $W = P(Z|Z \le \mathrm{VaR}_\alpha(Z))$, which is equivalent to the CDF of Z clipped to α. The CDF is then differentiated to obtain W, and $\mathrm{CVaR}_\alpha(Z)$ is computed as its expectation.

We provide a reference implementation of RACER in JAX. Additionally; we compile an extensive list of hyperparameters and architectural choices, as well as pseudocode and a reference implementation of RACER, on our website: https://sites.google.com/view/racer-epistemic-rl.

Algorithm 1 Compute CVaR

Input: $0 \leq \alpha < 1$, distribution Z via PDF P_i, atoms Z_i
Intermediates: CDF C_i, worst-cases PDF $\hat{P}_i$, CDF $\hat{C}_i$
Output: $\text{CVaR}_\alpha(Z)$
procedure COMPUTECVAR(P_i, Z_i, α)
    $C_i \leftarrow \sum_{i'=1}^{i}(P_i)$ ▷ Compute CDF (`np.cumsum`)
    $\hat{C}_i \leftarrow \frac{\min(C_i, 1-\alpha)}{1-\alpha}$ ▷ Worst-cases CDF
    $\hat{P}_i \leftarrow C_i - C_{i-1}$ ▷ Worst-cases PDF (`np.diff`)
    return $\sum_i \hat{P}_i Z_i$
end procedure

V. EXPERIMENTS

Our experimental evaluation aims to study the performance of our proposed risk-sensitive RL algorithm, both on a real-world robotic platform and in a simulated environment that provides a more controlled setting for rigorous comparisons. The primary goal of our method is to train the fastest possible driving policy that can traverse an outdoor course with uneven terrain while avoiding catastrophic failure in the form of high-speed collisions or rollover events. We hypothesize that minimizing failures during training is instrumental to achieving high performance, though of course minimizing the number of failures is also inherently desirable to avoid damage to the hardware. We therefore report both the cumulative number of failures and the final speed for each experiment. We do *not* aim to eliminate failures entirely, which is exceedingly difficult when learning from scratch.

A. Real-World Experiments

Our real-world evaluation of RACER uses a 1/10-scale remote-controlled car based on the F1TENTH platform [35]. Following prior work [46], we specify a course as a sparse sequence of checkpoints $\{c_i\}$, and define the reward function as "speed-made-good" $\vec{v} \cdot \vec{g}$, where $\vec{v}$ is the robot's velocity and $\vec{g}$ is the unit vector towards the next goal checkpoint. The robot's observation space consists of proprioceptive measurements (local velocity and IMU measurements) and a sequence of the next two goal vectors, generated using a GPS-based state estimator. Whenever one checkpoint $\{c_i\}$ is reached (measured via the robot's onboard GPS measurement), the goals are updated to reflect the next checkpoints.

We compare to FastRLAP [46], a sample-efficient autonomous learning system for the high-speed driving setting, which uses an RL method based on SAC [26]. As shown in Table I, RACER learns high-speed policies while largely avoiding failures during the training process.

Real-world learning curves are shown in Fig .6. Please see the supplemental materials for videos of high-speed driving behavior learned by RACER.

B. Simulated Comparisons and Ablation Studies

Our simulated experiments are designed to evaluate the individual design choices in our method, as well as compare it to representative previously proposed approaches for both high-speed navigation and risk-aware/safe RL.

Algorithm	# Fails ↓	# Fails >2m/s ↓	Lap time ↓	Speed (m/s) ↑
Vanilla SAC [26]	4	2	5.9	2.92
Ours	**2**	**0**	**5.2**	**3.32**

TABLE I: Results from real-world experiments. Our algorithm accrues fewer failures over the course of training while reaching comparable or better final performance (measured by lap time). Because the policy begins with a cautious action limit, its early failures occur at lower speeds. By smoothly adapting action limits over time RACER transfers its low-speed knowledge to higher speeds, avoiding high-speed failures (>2m/s) that could cause damage to the robot during training.

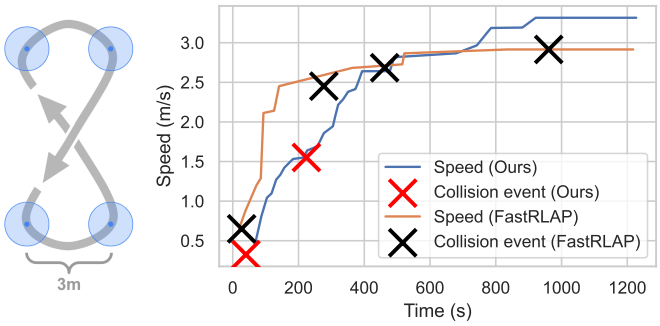

Fig. 6: Real-world experimental results with a figure-eight driving course (left). Learning curves (right) show the robot's average speed-made-good over a single lap (equivalently, total lap length divided by lap time). RACER experiences fewer crashes during training, especially later in training when the robot is operating at high speeds.

We conduct an comparison study in simulation to compare our work against prior methods and measure the effect of each of our design choices on two primary metrics: speed of the converged policy, measured by the robot's average speed at the end of training, and the cumulative number of safety violations incurred during training.

We consider two settings: an offroad driving setting and a classical locomotion task from the OpenAI gym. In the offroad driving setting, we use a simulated all-wheel-drive vehicle with suspension and Ackermann steering in two simulated environments: `Offroad-Flat`, which is an infinite flat plane, and `Offroad-Bumpy`, consisting of an infinite world procedurally generated using multi-scale Perlin noise [36]. Similar to the real-world experimental setup, the robot's objective is to drive towards the goal point as quickly as possible, with rewards defined by speed-made-good towards the goal, but we select a random goal from a normal distribution around the robot's current location each time a checkpoint is achieved. We are interested in the cumulative number of failures across training, which we define by counting the cumulative number of rollover events (when the car is upside down, terminating the episode with zero reward). A rollover event can happen as a result of bumpy terrain or when the applied steering inputs are too aggressive at high speeds. Comparisons are averaged over 5 seeds each containing 250,000 timesteps.

The `Cheetah` environment consists of OpenAI Gym's `HalfCheetah-v4`, with the addition of a safety condition: the cheetah should remain upright, and failure to do so terminates an episode with zero return.

In addition to standard **SAC**, we compare against several baselines: **SAC-Constrain** [1, 5] penalizes failure via a

Algorithm	Offroad-Flat		Offroad-Bumpy		Cheetah	
	# Fails ↓	Speed ↑	# Fails ↓	Speed ↑	# Fails ↓	Speed ↑
SAC [26]	287 ± 35	5.87 ± 0.31	363 ± 56	4.23 ± 0.23	152 ± 31	7.67 ± 1.70
PPO [40]	2750 ± 812	2.55 ± 0.09	$16e3 \pm 4e3$	1.79 ± 0.44	-	-
SAC-Constrain [1]	273 ± 39	5.93 ± 0.44	315 ± 14	4.24 ± 0.21	80 ± 38	7.45 ± 1.05
Safety critic [45]	245±34	5.91±0.73	265±86	3.99±0.34	63 ± 68	5.12 ± 1.13
WCSAC [55]	253±71	2.33±3.03	445±59	3.85±1.35	14 ± 16	0.03 ± 0.40
MPPI* [53]	1000 ± 275*	4.21 ± 0.07*	2570 ± 920*	3.60 ± 0.08*	1±1	9.61±1.17
Ours	**69±4**	***6.81±0.64***	**100±9**	***4.59±0.30***	**11±12**	**9.61±1.17**
no epistemic	122±17	6.1±0.41	165±29	***4.60±0.36***	23±19	**9.49±1.61**
no adaptive limits	108±9	**6.92±0.46**	152 ± 35	**4.67±0.13**	26±6	**9.35±1.71**

TABLE II: Ablation study in simulation comparing RACER (blue) with ablations (gray) as well as several prior works. We consider the total number of safety violations across 250k steps of training (# Fails) and the average speed of the final converged policy in m/s. The risk-sensitive objective greatly reduces safety violations over the course of training. Adaptive action limits and epistemic uncertainty also significantly reduce the number of failures. MPPI results (*) are extrapolated from convergence at 25k steps. Standard deviations are computed across 5 training runs; **bold** results are **significant** ($p < 0.05$) in comparison to all prior work; ***bold italic*** results are ***weakly significant*** ($p < 0.1$).

Lagrange multiplier that is simultaneously adapted to keep a particular constraint violation rate. **WCSAC** implements distributional RL by parametrizing Z as a normal distribution with learned mean and variance and optimizes a closed-form CVaR expression [55]. **Safety Critics** jointly learn a policy and a Q-function yielding the probability of remaining safe [8, 45], then perform rejection sampling on actions to find an action with $Q(s, a) < \epsilon$. We include **PPO** [40] for completeness, though it is not well-suited for our setting due to sample inefficiency and does not achieve high performance. SAC-based baselines use the learned maximum-entropy policy directly for exploration, with the exception of safety critics which additionally performs rejection sampling on policy actions.

We also perform several ablations to consider the effects of several design choices: use of CVaR as a risk measure (Sec. IV-A), epistemic uncertainty in the critic (Sec. IV-B), and adaptive action-space limits (Sec. IV-C).

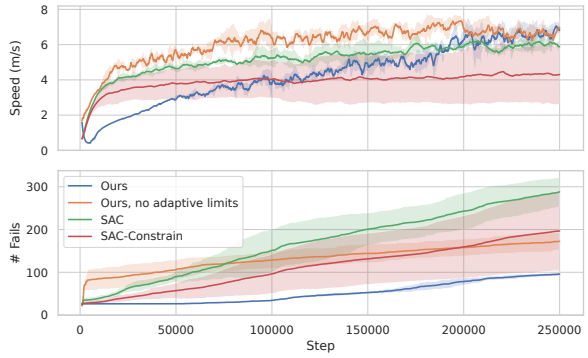

Fig. 7: Average policy speed (top) and cumulative safety violations (bottom) for selected methods in the `Flat` environment. RACER learns a high-speed policy with few safety violations. Comparable methods (e.g. SAC-Constrain) reduces safety violations to an extent but result in an overly conservative policy.

We present our results in Tab. II. RACER exhibits far fewer total of violations during training than all baselines considered, while exhibiting slightly *better* final performance. We hypothesize that when fewer failures occur during training, less of the model's representational capacity is wasted expressing high-loss outlier events, allowing for better estimation of the return earlier in the training process.

Perhaps surprisingly, non risk-sensitive methods tend to learn less performant final policies. They achieve relatively slow driving behavior in addition to a much higher failure rate. This is likely due to function approximation error in the critic $Q(s, a)$: MSE minimization tends to be very sensitive to outlier events (e.g. failures).

While safety critics do slightly reduce failures across training, the impact is relatively small in the driving setting, and negatively impacts performance in the locomotion setting. We note that Srinivasan et al. [45] suggests to train the policy on *unfiltered* (unsafe) actions, which is not possible in the purely online setting where prior unsafe data is not available. We instead apply it to the online setting, which may limit its ability to accurately classify safe actions. The Gaussian critic model in WCSAC [55] exhibited unstable training and poor performance.

Ablating the adaptive control limiting mechanism presented in Sec. IV-C results in similar final policy performance to the full implementation RACER but incurs many more failures early in training Fig. 7. Removing the explicit handling of epistemic uncertainty described in Sec. IV-B also causes increased failures and lower final speed.

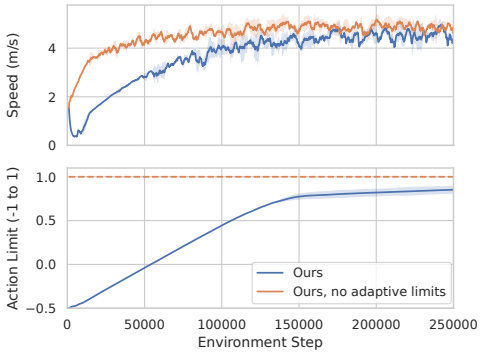

Fig. 8: Comparison of RACER with and without adaptive limits. (Top) comparison of speed: standard CVaR formulation quickly learns a fast driving policy but fails frequently in the early stages of training. (Bottom) the CVaR+limits formulation slowly increases the maximum action over time, tapering off at the high end.

The α parameter controls the conservatism of the CVaR metric. Fig. 9 shows that as $\alpha \to 1$, the learned policy becomes more conservative and experiences fewer safety violations but somewhat decreased performance. However, RACER retains high performance for a wide range of α.

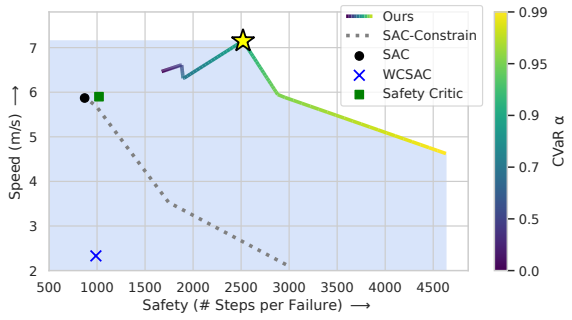

Fig. 9: The parameter α trades off between safety and performance. While performance decreases dramatically at extreme risk-sensitivity ($\alpha \geq 0.95$), for moderate $\alpha \approx 0.9$ it actually results in *increased* performance compared to risk-neutral settings ($\alpha = 0$) while accruing far fewer safety violations. The blue region represents points that are dominated by RACER in both metrics for some value of α.

C. Analysis of the Learned Critic

To better understand why risk-sensitive RL appears to be particularly helpful in the aggressive driving problem studied, we analyze the behavior of the distributional critic learned by RACER in risky states.

Fig. 10 shows the critic outputs in a case in which the policy experiences a failure. We highlight that only some members of the critic ensemble successfully model the low-probability mode corresponding to failure. Although the model knows that a failure is possible, the mean of the return distribution Z actually remains relatively high. However, the CVaR objective only considers the tail of the distribution and thus decreases sharply in the face of ensemble disagreement, as in Theorem 2. This indicates that the CVaR-based actor should be more responsive to these uncertain events.

Fig. 11 shows recovery behavior demonstrated by the learned policy. The critic still identifies the low-probability failure mode, but in this case is able to recover by steering in the "downhill" direction, a technique applied by human offroad driving experts [10].

VI. Discussion, Limitations, and Future Work

We presented RACER, a method for risk-sensitive real-world reinforcement learning applied to high-speed navigation that can acquire effective high-speed policies for driving on uneven terrain while minimizing the number of *training time* failures. The core of our method is a novel extension of the CVaR principle to deep RL with epistemic uncertainty, which constitutes to our knowledge the first CVaR-based deep RL algorithm that makes use of full distributional critics, accounts for epistemic uncertainty during training (which is important for avoiding risky behavior in unfamiliar states), and works well for real-world robotic control tasks. Our experiments demonstrate that our approach leads to fewer training-time failures and actually enables both real-world and simulated robots to attain faster driving speeds.

Our method has a number of limitations. Firstly, we require the existence of some "safe" region of action space. While this assumption is reasonable in many cases – often a "do-nothing"

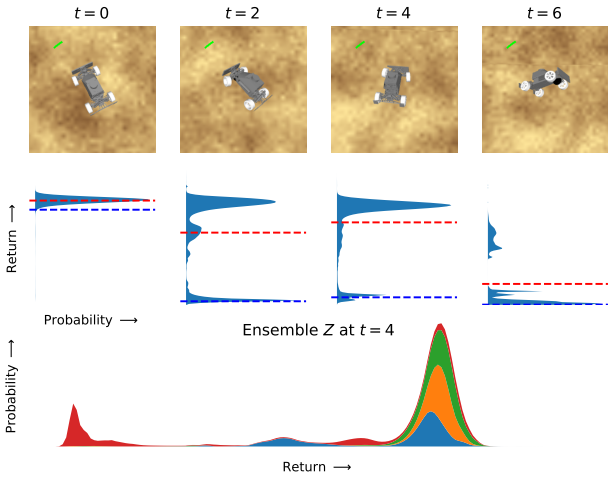

Fig. 10: Sequence of simulated states ending in a failure (rollover). Below each image the distributional critic $Z(s, a)$ is displayed with its mean in red and $\text{CVaR}_{\alpha=0.9}$ in blue. While the mean of the distribution remains high even when the critic has identified the low-probability failure mode (frame 4), the risk-sensitive CVaR metric appropriately penalizes the failure mode immediately as soon as it is detected. The final plot shows the individual ensemble member distributions at $t = 4$. In this case only one ensemble member identifies the risky failure mode (red), highlighting the importance of handling epistemic uncertainty to avoid overconfidence.

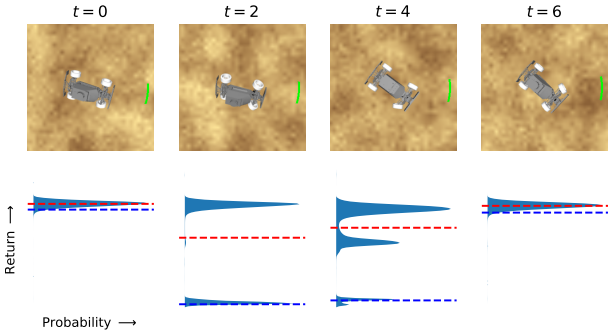

Fig. 11: The risk-sensitive policy successfully recovers from a likely failure ($t = 2$) by steering in the direction of the roll as soon as the critic detects a possible rollover. Goal direction indicated in green.

policy is quite easy to construct – it is not applicable to all settings. While RACER does effectively reduce the number of failures, it does not eliminate them entirely in all cases. Indeed, our aim is not to provide a true "safety constraint" (as would be necessary, for example, for full-scale autonomous driving systems), but only to reduce the number of failures enough so as to obtain the best final policy performance. Eliminating failures entirely likely requires additional domain knowledge, and studying this direction further is an important topic for future work. However, our results do show that our approach reduces the number of failures by a large margin and, perhaps surprisingly, that reducing the number of failures during *training* actually leads to more performant *final* policies. We hope for this reason that our approach will provide an important stepping stone toward RL algorithms that are practical to use in the real world for safety-critical applications.

Acknowledgements

This research was supported by DARPA ASNR, DARPA Assured Autonomy, and NSF IIS-2150826. The authors would like to thank Qiyang Li and Laura Smith for feedback on a draft version of this paper.

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017. 2, 6, 7

[2] Gordon J Alexander and Alexandre M Baptista. A comparison of var and cvar constraints on portfolio selection with the mean-variance model. *Management science*, 50(9):1261–1273, 2004. 2, 3

[3] Eitan Altman. *Constrained Markov Decision Processes*, volume 7. CRC Press, 1999. 2

[4] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999. 4

[5] Yarden As, Ilnura Usmanova, Sebastian Curi, and Andreas Krause. Constrained policy optimization via bayesian world models. *arXiv preprint arXiv:2201.09802*, 2022. 2, 6

[6] Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva Tb, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018. 4

[7] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pages 449–458. PMLR, 2017. 2, 3, 4

[8] Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and Animesh Garg. Conservative safety critics for exploration. *arXiv preprint arXiv:2010.14497*, 2020. 1, 2, 7

[9] Kang Boda and Jerzy A Filar. Time consistent dynamic risk measures. *Mathematical Methods of Operations Research*, 63:169–186, 2006. 4

[10] Marina Bruce. Anatomy of a rollover, 2018. URL https://outdooruae.com/outdoor-activity/off-road/anatomy-of-a-rollover/. 8

[11] Peide Cai, Xiaodong Mei, Lei Tai, Yuxiang Sun, and Ming Liu. High-speed autonomous drifting with deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2):1247–1254, apr 2020. doi: 10.1109/lra.2020.2967299. URL https://doi.org/10.1109%2Flra.2020.2967299. 3

[12] Margaret P Chapman, Riccardo Bonalli, Kevin M Smith, Insoon Yang, Marco Pavone, and Claire J Tomlin. Risk-sensitive safety analysis using conditional value-at-risk. *IEEE Transactions on Automatic Control*, 67(12):6521–6536, 2021. 2

[13] Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. *arXiv preprint arXiv:2101.05982*, 2021. 5

[14] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3387–3395, 2019. 2

[15] Jason Choi, Fernando Castaneda, Claire J Tomlin, and Koushil Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. *arXiv preprint arXiv:2004.07584*, 2020. 2

[16] Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for cvar optimization in mdps. *Advances in neural information processing systems*, 27, 2014. 3

[17] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017. 2

[18] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018. 2

[19] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3

[20] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018. 2

[21] Fabian Domberg, Carlos Castelar Wembers, Hiren Patel, and Georg Schildbach. Deep drifting: Autonomous drifting of arbitrary trajectories using deep reinforcement learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7753–7759, 2022. doi: 10.1109/ICRA46639.2022.9812249. 3

[22] Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022. 5

[23] Mudasir A Ganaie, Minghui Hu, AK Malik, M Tanveer, and PN Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115: 105151, 2022. 4

[24] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015. 2

[25] Tushar Goel, Jonathan Y. Goh, and J. Christian Gerdes. Opening new dimensions: Vehicle motion planning and control using brakes while drifting. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 560–565, 2020. doi: 10.1109/IV47402.2020.9304728. 2

[26] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018. 4, 6, 7

[27] Astghik Hakobyan, Gyeong Chan Kim, and Insoon Yang. Risk-aware motion planning and control using cvar-constrained optimization. *IEEE Robotics and Automation letters*, 4(4):3924–3931, 2019. 2

[28] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 3

[29] Milad Jalali, Ehsan Hashemi, Amir Khajepour, Shih-ken Chen, and Bakhtiar Litkouhi. Model predictive control of vehicle roll-over with experimental verification. *Control Engineering Practice*, 77:95–108, 2018. 2

[30] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020. 4

[31] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 4

[32] Xingyu Li, Bo Tang, John Ball, Matthew Doude, and Daniel W Carruth. Rollover-free path planning for off-road autonomous driving. *Electronics*, 8(6):614, 2019. 2

[33] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. 4

[34] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Parametric return density estimation for reinforcement learning. *arXiv preprint arXiv:1203.3497*, 2012. 3

[35] Matthew O'Kelly, Hongrui Zheng, Dhruv Karthik, and Rahul Mangharam. F1tenth: An open-source evaluation environment for continuous control and reinforcement learning. In Hugo Jair Escalante and Raia Hadsell, editors, *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, volume 123 of *Proceedings of Machine Learning Research*, pages 77–89. PMLR, 08–14 Dec 2020. URL https://proceedings.mlr.press/v123/o-kelly20a.html. 6

[36] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985. 6

[37] Georg Ch Pflug. Some remarks on the value-at-risk and the conditional value-at-risk. *Probabilistic constrained optimization: Methodology and applications*, pages 272–281, 2000. 3, 12

[38] LA Prashanth. Policy gradients for cvar-constrained mdps. In *International Conference on Algorithmic Learning Theory*, pages 155–169. Springer, 2014. 2

[39] Rahul Rahaman et al. Uncertainty quantification and deep ensembles. *Advances in Neural Information Processing Systems*, 34:20063–20075, 2021. 4

[40] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 7

[41] Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, pages 30365–30380. PMLR, 2023. 3, 5

[42] Archit Sharma, Kelvin Xu, Nikhil Sardana, Abhishek Gupta, Karol Hausman, Sergey Levine, and Chelsea Finn. Autonomous reinforcement learning: Formalism and benchmarking. *arXiv preprint arXiv:2112.09605*, 2021.

[43] William T. Shaw. Risk, var, cvar and their associated portfolio optimizations when asset returns have a multivariate student t distribution, 2011.

[44] Oswin So and Chuchu Fan. Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning, 2023. 1

[45] Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. Learning to be safe: Deep rl with a safety critic. *arXiv preprint arXiv:2010.14603*, 2020. 2, 7

[46] Kyle Stachowicz, Dhruv Shah, Arjun Bhorkar, Ilya Kostrikov, and Sergey Levine. Fastrlap: A system for learning high-speed driving via deep rl and autonomous practicing, 2023. 3, 5, 6

[47] Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods, 2020a. 1, 2

[48] Aviv Tamar, Huan Xu, and Shie Mannor. Scaling up robust mdps by reinforcement learning. *arXiv preprint arXiv:1306.6189*, 2013. 2

[49] Yichuan Charlie Tang, Jian Zhang, and Ruslan Salakhutdinov. Worst cases policy gradients. *arXiv preprint arXiv:1911.03618*, 2019. 2

[50] Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018. 2

[51] Lu Wen, Jingliang Duan, Shengbo Eben Li, Shaobing Xu, and Huei Peng. Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020. 2

[52] P. Whittle. Risk-sensitive linear/quadratic/gaussian control. *Advances in Applied Probability*, 13(4):764–777, 1981. ISSN 00018678. URL http://www.jstor.org/stable/1426972. 3

[53] Grady Williams, Paul Drews, Brian Goldfain, James M.

Rehg, and Evangelos A. Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018. doi: 10.1109/TRO.2018.2865891. 2, 7

[54] Donghun Yang, Kien Mai Ngoc, Iksoo Shin, Kyong-Ha Lee, and Myunggwon Hwang. Ensemble-based out-of-distribution detection. *Electronics*, 10(5), 2021. ISSN 2079-9292. doi: 10.3390/electronics10050567. URL https://www.mdpi.com/2079-9292/10/5/567. 4

[55] Qisong Yang, Thiago D. Simão, Simon H Tindemans, and Matthijs T. J. Spaan. Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10639–10646, May 2021. doi: 10.1609/aaai.v35i12.17272. URL https://ojs.aaai.org/index.php/AAAI/article/view/17272. 2, 4, 7

[56] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pages 737–744. IEEE, 2020. 3

APPENDIX A
PROOF OF CVAR THEOREMS

We begin by restating the theorems listed in Section IV-A. Let Z_i be N real-valued random variables with density $p_i(z)$ and cumulative distribution P_i. Additionally, let $\hat{p}(z) = \frac{1}{N}\sum_{i=1}^{N} p_i(z)$ be the density for the ensemble, with cumulative density function $\hat{P}(z) = \frac{1}{N}\sum_{i=1}^{N} P_i(z)$, and let $\hat{Z}$ be a random variable drawn from this distribution.

Lemma 1. *CVaR is convex, in the sense that* $\textit{CVaR}_\alpha[Z] \leq \lambda \textit{CVaR}_\alpha[X] + (1-\lambda)\textit{CVaR}_\alpha[Y]$ *when* Z *is the mixture distribution* $p_Z(\cdot) = \lambda p_X(\cdot) + (1-\lambda)p_Y(\cdot)$.

Proof. Consider the expression:

$$\inf_{a\in\mathbb{R}} \left[a + \frac{1}{1-\alpha}\,\mathbb{E}[\max(0, Z-a)] \right].$$

Assuming smoothness, the infinimum occurs when the derivative of the expression is zero with respect to a:

$$f(a, Z) = a + \frac{1}{1-\alpha}\,\mathbb{E}[\max(0, Z-a)]$$

$$\frac{d}{da} f(a, Z) = 1 - \frac{1}{1-\alpha}\int_Z 1_{Z\geq a} dp(Z) = 1 - \frac{1}{1-\alpha} P(Z \geq a)$$

Of course, this expression is equal to zero when $P(Z \geq a) = 1-\alpha$ (when a is the VaR of Z). It then follows that:

$$\inf_{a\in\mathbb{R}} \left[a + \frac{1}{1-\alpha}\,\mathbb{E}[\max(0, Z-a)] \right] = \frac{1}{1-\alpha}\,\mathbb{E}[\max(0, Z - \text{VaR}_\alpha(Z))] = \text{CVaR}_\alpha(Z)$$

. With this alternative definition of CVaR (see Pflug [37] for a more rigorous derivation), we assume a_1, a_2 be the arg-min in this definition for X and Y respectively. Then, again following Pflug [37]:

$$\begin{aligned}
\text{CVaR}_\alpha(Z) &= \text{CVaR}_\alpha(\lambda X + (1-\lambda)Y) \\
&\leq \lambda a_1 + (1-\lambda)a_2 + \frac{1}{1-\alpha}\,\mathbb{E}\max(0, \lambda X + (1-\lambda)Y - \lambda a_1 - (1-\lambda)a_2) \\
&\leq \lambda \left[a_1 + \frac{1}{1-\alpha}\,\mathbb{E}\max(0, X - a_1) \right] + (1-\lambda)\left[a_2 + \frac{1}{1-\alpha}\,\mathbb{E}\max(0, Y - a_2) \right] \\
&\leq \lambda \text{CVaR}_\alpha(X) + (1-\lambda)\text{CVaR}_\alpha(Y)
\end{aligned}$$

□

Theorem 1. *Let* Z_i *be real-valued random variables with density* $p_i(z)$. *Denote the random variable with density* $\hat{p}(z) = \frac{1}{N}\sum_i p(z)$ *as* $\hat{Z}$. *Then for* $\alpha > 0$*:*

$$\textit{CVaR}_\alpha(P) \leq \frac{1}{N}\sum_i \textit{CVaR}_\alpha(P_i)$$

We call the positive difference $\frac{1}{N}\sum_i \textit{CVaR}_\alpha(Z_i) - \textit{CVaR}_\alpha Z_i$ *the CVaR gap.*

Proof. The result follows directly from applying Jensen's inequality to $\text{CVaR}_\alpha(P)$. □

Definition 1. *Let* Z *be a real-valued random variable. Then for* $0 < \alpha < 1$ *define the* ***tail distribution*** $\mathcal{T}_\alpha Z$ *as the distribution with probability mass* $\frac{1}{1-\alpha}p(z)$ *over the support:*

$$\text{supp}(\mathcal{T}_\alpha Z) = (-\infty, \textit{VaR}_\alpha(Z)] \cap \text{supp}(Z).$$

We also denote the density function of $\mathcal{T}_\alpha Z$ *as* $\mathcal{T}_\alpha p(z)$ *and its cumulative density as* $\mathcal{T}_\alpha P(z)$.

Note that $\text{CVaR}_\alpha(Z) = \mathbb{E}[\mathcal{T}_\alpha Z]$. We now provide the proof of our first theorem:

Definition 2. *Let* X, Y *be real-valued random variables with cumulative distribution functions* Φ_X, Φ_Y *respectively. Define the* ***earth-mover's distance*** $\|X - Y\|_{\text{EMD}}$ *as:*

$$\int_{\mathbb{R}} |\Phi_X(x) - \Phi_Y(x)| dx$$

Theorem 2. *Let* Z_i *be random variables with density* $p_i(z)$. *Assume* Z_i *has finite first moment and denote the mixture distribution as* $\hat{Z}$ *with density* $\hat{p}(z)$. *Then we have:*

$$\frac{1}{N}\sum_i \textit{CVaR}_\alpha(Z_i) - \textit{CVaR}_\alpha(\hat{Z}) \leq \frac{1}{N}\sum_i \left\| \mathcal{T}_\alpha \hat{Z} - \mathcal{T}_\alpha Z_i \right\|_{\text{EMD}}$$

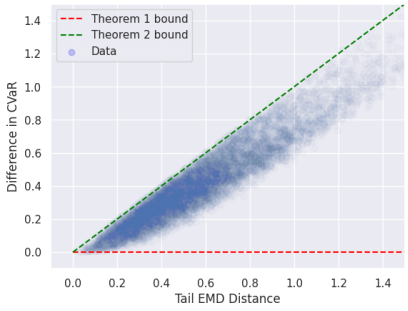

Fig. 1: Tail EMD and CVaR gap for randomly sampled mixtures of Gaussians. CVaR gap correlates very well with tail EMD, indicating that the bound provided in Theorem 2 is relatively tight.

Proof.

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{T}_\alpha Z_i - \mathcal{T}_\alpha \hat{Z}\right\|_{\text{EMD}} &= \frac{1}{N}\sum_{i=1}^{N}\int_{\mathbb{R}} |\mathcal{T}_\alpha \hat{P}(z) - \mathcal{T}_\alpha P_i(z)|dz \\ &\geq \frac{1}{N}\sum_{i=1}^{N}\int_{\mathbb{R}} \left(\mathcal{T}_\alpha \hat{P}(z) - \mathcal{T}_\alpha P_i(z)\right) dz\end{aligned}$$

Integration by parts gives that $\int_{\mathbb{R}}(P(x)-Q(x))\,dx = [x(P(x)-Q(x))]_{-\infty}^{\infty} + \int_a^b (q(x)-p(x))x\,dx$:

$$\frac{1}{N}\sum_{i=1}^{N}\left[\lim_{z\to\infty} z\left(\mathcal{T}_\alpha P_i(z) - \mathcal{T}_\alpha \hat{P}(z) + \mathcal{T}_\alpha P_i(-z) - \mathcal{T}_\alpha \hat{P}(-z)\right) + \int_{\mathbb{R}} \left(\mathcal{T}_\alpha p_i(z) - \mathcal{T}_\alpha \hat{p}(z)\right) z\,dz\right]$$

The first term vanishes because all cumulative density functions approach 0 and 1 at $\pm\infty$ respectively, and the distributions have finite moments. Then we are left with:

$$\frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{T}_\alpha Z_i - \mathcal{T}_\alpha \hat{Z}\right\|_{\text{EMD}} \geq \frac{1}{N}\sum_i \mathbb{E}\,\mathcal{T}_\alpha Z_i - \mathbb{E}\,\mathcal{T}_\alpha \hat{Z} = \frac{1}{N}\sum_i \text{CVaR}_\alpha Z_i - \text{CVaR}_\alpha \hat{Z}$$

□

Appendix B
Emperical Analysis of CVaR and Ensemble Divergence

While Theorem 2 provides an upper-bound on CVaR gap based on the tail EMD, it does not provide a lower bound stricter than that in Theorem 1. In fact, for any tail-EMD $D \geq 0$ it is possible to construct an adversarial ensemble distribution with zero CVaR gap; for example (with $\alpha = 0.5$):

$$p_1(z) = \begin{cases} 0.5 & z = D \\ 0.5 & z = D/2 \\ 0 & \text{otherwise}\end{cases} \qquad p_2(z) = \begin{cases} 0.5 & z = D \\ 0.5 & z = -D/2 \\ 0 & \text{otherwise}\end{cases} \qquad \hat{p}(z) = \begin{cases} 0.5 & z = D \\ 0.25 & z = D/2 \\ 0.25 & z = -D/2 \\ 0 & \text{otherwise}\end{cases}$$

Nonetheless, in practice we find that tail-EMD empirically correlates extremely well with the CVaR gap for typical distributions (bounded, smooth). We probe this relationship in the setting of random gaussian-mixture ensemble distributions using the following procedure:

1) Select a random "base" distribution as a Gaussian mixture with $K = 3$ components with parameters sampled randomly.
2) Create $N = 3$ ensemble members by randomly sampling parameters around the "base" parameters.
3) Find the mixture distribution of the ensemble members then compute $\frac{1}{N}\sum_i \left\|\mathcal{T}_\alpha Z_i - \mathcal{T}_\alpha \hat{Z}\right\|_{\text{EMD}}$ and the CVaR gap.

Plotting the CVaR gap and tail-EMD yields Figure 1. There is a clear relationship between the two. This indicates that pessimism in RACER correlates extremely well with ensemble divergence.