2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

SRPO: Self-Reflection Policy Optimization for Stable and Robust Autonomous Driving

Dejin Wang[1] and Seyede Fatemeh Ghoreishi[2]

***Abstract*— Autonomous driving demands reinforcement learning (RL) agents that are not only performant, but also stable, sample-efficient, and robust to uncertainty. However, conventional policy optimization methods often suffer from unstable convergence, sensitivity to reward scaling, and limited generalization in safety-critical or out-of-distribution scenarios. We propose Self-Reflection Policy Optimization (SRPO), a principled, model-free framework that introduces policy-level self-evaluation by benchmarking each policy iteration against its own historical performance. This self-reflection yields a reward-shaping signal based on relative improvement, which is redistributed across trajectory steps using a rank-based credit assignment mechanism. This design emphasizes informative steps, eliminates dependence on absolute reward magnitudes, and improves stability in practice. We theoretically show that a bounds-based variant of SRPO preserves policy optimality and convergence. Empirically, we evaluate SRPO on both Highway-env and the high-fidelity CARLA simulator under adversarial perturbations and out-of-distribution driving conditions. SRPO consistently improves training efficiency, robustness, and policy performance compared to the baseline techniques. These results position SRPO as a promising and theoretically grounded approach to more reliable decision-making for autonomous driving. The source code is available at: `https://github.com/dejin-wang/SRPO_anonymous_code`.**

I. INTRODUCTION

Reinforcement learning (RL) has emerged as a powerful framework for autonomous driving, offering the ability to learn adaptive decision-making strategies directly from interaction with complex, high-dimensional environments [1]–[4]. Despite this promise, practical deployment remains challenging: RL training often exhibits unstable convergence, poor sample efficiency, and brittle behavior under distribution shift. These issues often arise from sparse or delayed rewards, sensitivity to reward scaling, and mismatches between training and deployment conditions. These limitations pose critical risks to robustness and safety in autonomous driving systems.

In human learning, progress is often accelerated through *self-reflection*, the ability to evaluate current performance relative to prior achievements, reinforcing successful strategies and discarding ineffective ones [5]. This principle suggests an RL agent should not only seek high returns in absolute terms, but also assess whether it is improving relative to its own history. Motivated by this principle, we propose *Self-Reflection Policy Optimization (SRPO)*, a reinforcement learning framework that equips a policy with an internal self-evaluation signal. At each training iteration, SRPO compares the current policy's performance to a historical reference and generates a self-guided reward-shaping signal that amplifies improvements and attenuates regressions. To localize this signal within trajectories, SRPO employs a *rank-based step-level credit assignment* scheme that distributes the shaping reward according to each timestep's relative contribution. This yields (i) invariance to heterogeneous or shifting reward scales and (ii) fine-grained guidance that accelerates learning. Conceptually, SRPO performs *single-agent self-comparison*, avoiding adversaries or engineered curricula, while remaining simple to integrate with standard policy optimization.

From a theoretical standpoint, we show that a *bounds-based* variant of SRPO preserves policy optimality and convergence, providing formal grounding for the self-reflection principle. In practice, we implement SRPO via a rank-based step-level approach, which delivers stable shaping without requiring tuning to absolute reward magnitudes. We evaluate SRPO in the Highway-env environment and the high-fidelity CARLA simulator under adversarial disturbances and out-of-distribution conditions, observing consistent gains in training stability, sample efficiency, and robustness compared to the baseline techniques. Our main contributions are as follows:

- We propose SRPO, a self-reflective RL framework that benchmarks a policy against a historical reference to produce informative, self-guided reward shaping.
- We introduce a rank-based step-level credit assignment strategy that provides fine-grained, scale-invariant reward redistribution for stable and robust learning.
- We provide theoretical guarantees that a bounds-based SRPO variant preserves policy optimality and convergence.
- We demonstrate empirical improvements in training stability, sample efficiency, and robustness in both standard and challenging driving scenarios.

II. RELATED WORK

Deep reinforcement learning (RL) has been widely studied for tactical decision-making and closed-loop control in autonomous driving. On-policy methods such as PPO [6] and TRPO [7] offer stability through trust-region constraints and advantage estimation, while off-policy methods like SAC [8] and TD3 [9] improve sample efficiency. Beyond task-specific case studies, recent benchmarks emphasize realistic closed-loop evaluation and generalization. For instance, CARLA

*This research was supported by the Army Research Office award W911NF-23-2-0207.

[1]Dejin Wang is with the College of Engineering, Northeastern University, Boston, MA 02115, USA. wang.dej@northeastern.edu

[2]Seyede Fatemeh Ghoreishi is with the College of Engineering and Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115, USA. f.ghoreishi@northeastern.edu

Leaderboard [10], NoCrash [11], and nuPlan [12] highlight the need for closed-loop validation over handcrafted metrics. Scalable multi-agent simulation platforms such as SMARTS [13] and GIGAFLOW [14], and accelerated, data-driven simulators such as Waymax [15] enable interactive training at scale [16]. Despite this progress, instability under stochastic dynamics, sensitivity to reward scale, and brittleness under distribution shift remain persistent obstacles in RL-based driving.

A. Robustness, Self-Play, and Scenario Generation

To improve robustness, adversarial training methods like Robust Adversarial RL (RARL) [17] train policies against learned disturbance agents to improve worst-case behavior. Adaptive Stress Testing (AST) casts failure-finding as an RL problem to uncover likely collision trajectories for safety evaluation [18], [19]. Very recent work scale self-play for driving: asymmetric self-play uses teacher-student scenario generation to target long-tail events [20], while large-scale self-play with batched simulation demonstrates robust, naturalistic driving policies emerging without human data [21], [22]. Such approaches improve robustness but typically require multi-agent infrastructure, engineered adversaries, or heavy scenario generators. In contrast, SRPO introduces a *single-agent, self-referential* signal that stabilizes and accelerates learning without external opponents or synthetic scenario optimization.

B. Reward Shaping and Credit Assignment

Reward shaping is a longstanding tool for guiding RL, but improper shaping can alter optimal behavior. Potential-based shaping methods [23] preserve policy invariance but often require hand-crafted potentials. Adaptive normalization (e.g., PopArt [24]) improves stability by rescaling value targets during non-stationary training. More relevant to our work are methods that exploit past experience to bootstrap learning. Self-Imitation Learning (SIL) [25] biases updates toward high-return rollouts, while Ranked Reward (R2) [26] uses percentile-based baselines to normalize learning signals in sparse domains. Our method differs in three key ways: (i) we benchmark against a dynamically updated *reference policy* rather than percentile filtering, (ii) we distribute credit *at the step level* using a rank-based allocation scheme, and (iii) we analyze a *bounds-based* variant that guarantees convergence and optimality. Recent advances in credit assignment and value decomposition, such as OPAL [27], DEUP [28], and offline RL with uncertainty estimation [29], highlight the importance of stable, fine-grained attribution. SRPO complements these approaches with a scale-invariant and general-purpose mechanism for localizing shaping rewards.

III. Background

We model autonomous driving as a discounted Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, R, \gamma, \rho_0)$. Here, $\mathcal{S}$ is the state space (e.g., ego and surrounding-vehicle kinematics, map context, sensor features), $\mathcal{A}$ is the action space (e.g., continuous steering, throttle, braking), $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition kernel capturing stochastic dynamics, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the per-step reward encoding objectives such as safety, efficiency, and comfort, $\gamma \in (0, 1)$ discounts future rewards, and ρ_0 is the initial-state distribution. A (stochastic) policy $\pi_\theta(a \mid s)$ maps states to a distribution over actions. Trajectories $\tau = (s_0, a_0, s_1, a_1, \ldots)$ are generated by $s_0 \sim \rho_0$, $a_t \sim \pi_\theta(\cdot \mid s_t)$, and $s_{t+1} \sim P(\cdot \mid s_t, a_t)$. Let $r_t := R(s_t, a_t)$ denote the environment reward at time t. The (discounted) return is $G_0 = \sum_{t=0}^{\infty} \gamma^t r_t$ (for episodic tasks the sum terminates upon reaching a terminal state). The learning objective is to find a policy that maximizes the expected return:

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim (\rho_0, \pi_\theta, P)} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]. \tag{1}$$

In the remainder of this paper, we optimize (1) using policy-gradient methods and introduce a self-reflective learning signal that reshapes the per-trajectory rewards during training.

IV. Self-Reflection Policy Optimization (SRPO)

SRPO equips a learning agent with an internal, policy-level self-evaluation signal. At each training iteration, the current policy is benchmarked against a historical reference, and the resulting (scalar) self-reflection score is redistributed to timesteps within each trajectory via a rank-based step credit rule. The shaping is injected into the raw reward, yielding informative guidance that is invariant to the absolute scale of environment rewards and requires no adversarial opponents or engineered curricula.

A. Self-Reflection Mechanism

SRPO augments learning with a *policy-level self-evaluation* that measures whether the current policy improves over its own history and turns this signal into guidance for the update.

a) Current vs. reference performance: Let π_t be the current policy at training iteration t and π_{ref} a slowly updated *reference* policy (hard copy every K iterations, or via an Exponential Moving Average (EMA)). For a batch $\mathcal{B}_t = \{\tau_i\}_{i=1}^{M}$ collected with π_t, define discounted returns:

$$G_{t,i} = \sum_{u=0}^{T_i - 1} \gamma^u r_{i,u}. \tag{2}$$

We maintain a corresponding estimate $\widehat{J}_{\text{ref}}$ from a small buffer of recent returns of π_{ref} (e.g., the batch that created the reference, optionally refreshed by short evaluation rollouts). To ensure *scale robustness*, we track a running dispersion statistic $\sigma_J > 0$ over recent returns (e.g., standard deviation via EMA).

For each trajectory, the raw improvement is:

$$\Delta_{t,i} = G_{t,i} - \widehat{J}_{\text{ref}}, \tag{3}$$

which we convert into a *self-reflection score*:

$$g_{t,i} = \text{clip}\left(\frac{\Delta_{t,i}}{\sigma_J + \varepsilon}, -g_{\max}, g_{\max} \right), \qquad \varepsilon > 0, \tag{4}$$

with $g_{\max} \in [1,3]$ limiting outliers. Thus $g_{t,i} > 0$ indicates improvement over the historical baseline, $g_{t,i} < 0$ indicates regression. Normalizing by σ_J avoids divisions by small $|\widehat{J}_{\text{ref}}|$ and stabilizes the signal across tasks with different reward scales or signs.

b) Injecting the signal at update time: Rather than rewriting environment rewards (which can introduce non-Markovian effects), SRPO injects shaping *only at update time*. We augment $r_{i,u}$ with a localized shaping term:

$$r_{i,u}^{\text{SRPO}} = r_{i,u} + \lambda\, g_{t,i}\, w_{i,u}, \qquad \lambda \geq 0, \tag{5}$$

where $\{w_{i,u}\}_{u=0}^{T_i-1}$ are nonnegative step weights that sum to one for each trajectory τ_i (see Sec. IV-B for our rank-based construction). This formulation applies to both on-policy (PPO/TRPO/A2C) and off-policy (SAC/TD3) methods by modifying the raw reward.

c) Reference update (hard copy or EMA): We set $\pi_{\text{ref}} \leftarrow \pi_t$ every K iterations, or update parameters via EMA:

$$\theta_{\text{ref}} \leftarrow \beta\,\theta_{\text{ref}} + (1-\beta)\,\theta_t, \qquad \beta \in (0,1),$$

where larger β (or larger K) yields a more stable, slower-moving benchmark. Typical β values are 0.9–0.999.

Some practical notes: (i) $G_{t,i}$ is computed from the same trajectories used for the update; $\widehat{J}_{\text{ref}}$ is held fixed within the iteration. (ii) For off-policy training, $g_{t,i}$ can be computed from short on-policy probe rollouts to avoid replay-induced bias. (iii) For additional stability, we optionally clip $\lambda g_{t,i} w_{i,u}$ to a small multiple of the typical advantage magnitude.

The mechanism above yields a single, well-conditioned scalar $g_{t,i}$ that captures *relative improvement* independent of absolute reward scale. Its step-level allocation $w_{i,u}$ (Sec. IV-B) provides fine-grained guidance without requiring reward bounds or adversarial opponents.

B. Rank-Based Step-Level Credit Assignment

The trajectory-level self-reflection score $g_{t,i}$ in (4) summarizes improvement or regression relative to the reference policy, but it must be *localized* to individual decisions to be useful for credit assignment. SRPO achieves this by distributing $g_{t,i}$ across timesteps within each trajectory via nonnegative weights $\{w_{i,u}\}_{u=0}^{T_i-1}$ that sum to one; these weights determine how strongly the additive shaping term $\lambda\, g_{t,i}\, w_{i,u}$ in (9) influences the policy gradient at step u.

a) Why ranks (and not magnitudes)? Allocations based on absolute reward/value magnitudes require bounds or careful normalization and are brittle under non-stationary scales (including sign flips). Ranking within a trajectory is invariant to any strictly monotone transformation of the underlying signal, making it naturally robust to heterogeneous, sparse, or negatively shifted rewards.

b) Per-trajectory proxy and ranks: For each trajectory τ_i of length T_i, let $s_{i,u}$ denote a per-step *proxy score* used only to order timesteps. By default $s_{i,u} = r_{i,u}$ (raw reward); in high-variance settings one may instead use the return-to-go $\sum_{v\geq u} \gamma^{v-u} r_{i,v}$ or an advantage estimate $\widehat{A}_{i,u}$. SRPO is agnostic to this choice, and any strictly monotone transform of $s_{i,u}$ leaves the ranking unchanged. Define $\text{rank}_{i,u} \in \{1,\ldots,T_i\}$ as the *ascending* rank within τ_i (1 = lowest, T_i = highest), with ties assigned their average rank. Ranks are computed independently per trajectory (no cross-trajectory normalization) and require only a sort of $\{s_{i,u}\}_{u=0}^{T_i-1}$ (cost $O(T_i \log T_i)$).

c) Weights for improvement vs. regression: When the trajectory has improved ($g_t > 0$), SRPO emphasizes high-ranked steps; when it has regressed ($g_t < 0$), it emphasizes low-ranked steps to encourage correction. We define a family of *rank weights* using a nondecreasing function $\rho : \mathbb{N} \to \mathbb{R}_+$:

$$w_{i,u}^{(+)} = \frac{\rho\big(\text{rank}_{i,u}\big)}{\sum_{v=0}^{T_i-1} \rho\big(\text{rank}_{i,v}\big)}, \qquad \text{if } g_t > 0, \tag{6}$$

$$w_{i,u}^{(-)} = \frac{\rho\big(T_i - \text{rank}_{i,u} + 1\big)}{\sum_{v=0}^{T_i-1} \rho\big(T_i - \text{rank}_{i,v} + 1\big)}, \qquad \text{if } g_t < 0, \tag{7}$$

and set $w_{i,u} = 1/T_i$ when $g_t = 0$. The default choice $\rho(k) = k$ (linear) yields the simple closed forms

$$w_{i,u}^{(+)} = \frac{\text{rank}_{i,u}}{\frac{T_i(T_i+1)}{2}}, \qquad w_{i,u}^{(-)} = \frac{T_i - \text{rank}_{i,u} + 1}{\frac{T_i(T_i+1)}{2}}. \tag{8}$$

A temperature-like knob $\rho(k) = k^p$ with $p > 0$ sharpens ($p > 1$) or flattens ($p < 1$) the allocation.

d) Properties and cost: (i) *Budget preservation:* $\sum_u w_{i,u} = 1$ so the total shaping per trajectory equals $\lambda\, g_{t,i}$. (ii) *Sign consistency:* the additive term $\lambda g_{t,i} w_{i,u}$ has the same sign as $g_{t,i}$. (iii) *Scale/shift invariance:* replacing $s_{i,u}$ by any strictly monotone transform leaves ranks, and hence $w_{i,u}$, unchanged. (iv) *Computation:* weights require only sorting within a trajectory ($O(T_i \log T_i)$); no extra environment interaction is needed.

e) Integration with SRPO: With $g_{t,i}$ from (4) and $w_{i,u}$ from (8) (or (6)–(7)), SRPO augments raw reward as in (9):

$$\widehat{r}_{i,u}^{\text{SRPO}} = \widehat{r}_{i,u} + \lambda\, g_{t,i}\, w_{i,u}, \qquad \lambda \geq 0, \tag{9}$$

providing fine-grained, scale-robust guidance without requiring reward bounds or adversarial opponents. A schematic overview of the proposed framework is shown in Fig. 1, and Algorithm.1 shows the procedural details of the proposed SRPO.

V. Theoretical Guarantees

We provide guarantees for a *bounds-based* variant of SRPO that is analytically tractable and *policy invariant*, and then relate it to the practical rank-based realization used in our experiments. Throughout, assume rewards are bounded: $R_{\min} \leq r_t \leq R_{\max}$, and $\gamma \in (0,1)$. Define $b_{\min} = R_{\min}/(1-\gamma)$, $b_{\max} = R_{\max}/(1-\gamma)$, and let $c := J(\pi_{\text{ref}})$ denote the fixed expected return of the reference policy.

A. A policy-invariant bounds-based variant

Given a current policy π, let $J(\pi) = \mathbb{E}\big[\sum_{t\geq 0} \gamma^t r_t\big]$ and $\Delta(\pi) := J(\pi) - c$. Define the following *per-trajectory, step-*

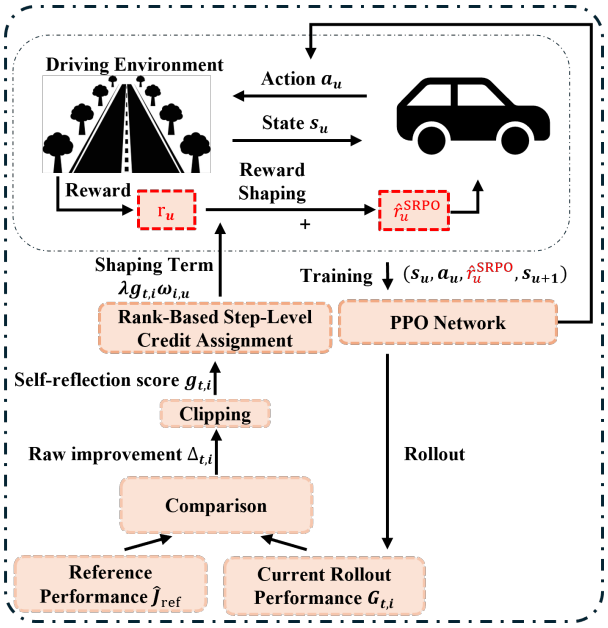

Fig. 1: Schematic representation of the SRPO framework.

local shaping using the exact $J(\pi)$:

$$\tilde{r}_t^{(+)} = r_t + \lambda\,(r_t - R_{\min})\,\frac{\Delta(\pi)}{J(\pi) - b_{\min}} \qquad \text{if } \Delta(\pi) > 0, \tag{10}$$

$$\tilde{r}_t^{(-)} = r_t + \lambda\,(R_{\max} - r_t)\,\frac{\Delta(\pi)}{b_{\max} - J(\pi)} \qquad \text{if } \Delta(\pi) < 0, \tag{11}$$

and $\tilde{r}_t = r_t$ if $\Delta(\pi) = 0$. Intuitively, (10) emphasizes larger-than-min rewards when the current policy improves, whereas (11) emphasizes smaller-than-max rewards when it regresses. Crucially, the normalizers make the shaping *budget preserving* in expectation.

Theorem 1 (Policy-order preservation)**.** *Let $\tilde{J}(\pi)$ denote the expected return under the shaped rewards* (10)–(11)*. Then, for any $\lambda \geq 0$,*

$$\tilde{J}(\pi) = (1+\lambda)\,J(\pi) - \lambda\,c.$$

Consequently, $\arg\max_\pi \tilde{J}(\pi) = \arg\max_\pi J(\pi)$; the shaping preserves the set of optimal policies.

Proof. Write $G = \sum_{t\geq 0}\gamma^t r_t$. If $\Delta(\pi) > 0$, summing (10) over time gives $\tilde{G} = G + \lambda\,\Delta(\pi)\,(G - b_{\min})/(J(\pi) - b_{\min})$. Taking expectations, $\tilde{J}(\pi) = J(\pi) + \lambda\,\Delta(\pi)\,\mathbb{E}[G - b_{\min}]/(J(\pi) - b_{\min}) = J(\pi) + \lambda\,\Delta(\pi) = (1+\lambda)J(\pi) - \lambda c$. The case $\Delta(\pi) < 0$ is analogous using (11) and $\mathbb{E}[b_{\max} - G] = b_{\max} - J(\pi)$. If $\Delta(\pi) = 0$, the identity is trivial. Since $(1+\lambda) > 0$, the transformation is strictly order-preserving. □

Theorem 2 (Convergence of policy evaluation)**.** *Fix any policy π and define the Bellman operator under the shaped rewards, $(T^\pi_{shape}V)(s) = \mathbb{E}\big[\tilde{r}_t + \gamma\,V(s_{t+1}) \,\big|\, s_t = s, a_t \sim \pi\big]$. If rewards are bounded, then T^π_{shape} is a γ-contraction in the sup-norm and admits a unique fixed point V^π_{shape}.*

Algorithm 1 Self-Reflection Policy Optimization (SRPO)

Input: Initial policy π_0, learning rate α, reflection coefficient λ, update interval K, clipping threshold $g_{\max}$
Output: Optimized policy π

Initialize $\pi_{\text{ref}} \leftarrow \pi_0$ and calculate reference $\widehat{J}_0$.
for iteration $t = 1, 2, \ldots$ **do**
  Collect batch $\mathcal{B}_t$ with π_t.
  for each trajectory τ_i **do**
    Compute return $G_{t,i}$, improvement $\Delta_{t,i}$, and self-reflection score $g_{t,i}$ using Eq.(2), (3), (4).
  end for
  for each trajectory τ_i **do**
    for each step u in trajectory τ_i **do**
      Compute rank $\text{rank}_{i,u}$ and weight $w_{i,u}$.
      Define shaped reward $r_{i,u}^{\text{SRPO}} = r_{i,u} + \lambda g_{t,i} w_{i,u}$.
    end for
  end for
  Update policy $\pi_t \to \pi_{t+1}$ using shaped rewards.
  if $t \bmod K = 0$ **then**
    Update $\pi_{\text{ref}} \leftarrow \pi_t$ (or EMA update)
    Calculate reference $\widehat{J}_{\text{ref}}$.
  end if
end for

Proof. For any V, W, $\|T^\pi_{\text{shape}}V - T^\pi_{\text{shape}}W\|_\infty = \sup_s \big|\gamma\,\mathbb{E}[V(s') - W(s') \mid s]\big| \leq \gamma\|V - W\|_\infty$. Banach's fixed-point theorem yields existence, uniqueness, and convergence of value (or policy) iteration. □

a) Remarks: (i) Theorems 1–2 require the exact $J(\pi)$ in (10)–(11) to hold *exactly*. In practice, one can plug in sufficiently accurate estimates; as long as denominators are bounded away from zero (i.e., $J(\pi) \not\approx b_{\min}$ or $b_{\max}$) and estimation error is small, $\tilde{J}(\pi)$ remains a strictly increasing function of $J(\pi)$ in a neighborhood, preserving the optimizer. (ii) The bounds-based shaping distributes a trajectory-level budget $\lambda\,\Delta(\pi)$ across steps while keeping the policy order unchanged; it formalizes the *policy-invariant* part of SRPO.

B. Why Self-Reflection Improves Learning

SRPO changes *only the learning signal used at update time* by adding a trajectory-level score $g_{t,i}$ (derived from $\Delta_{t,i}$ in Eq. (4)) and localizing it to steps via rank weights $w_{i,u}$ (Eq. (8)). Specifically, we form the shaped per-step signal $r_{i,u}^{\text{SRPO}} = r_{i,u} + \lambda\,g_{t,i}\,w_{i,u}$ as in Eq. (9), while environment interaction remains unchanged.

a) Two-term gradient decomposition: In Monte-Carlo/REINFORCE form (or any baseline-invariant policy gradient), using $r_{i,u}^{\text{SRPO}}$ replaces the trajectory return $G_{t,i} = \sum_u \gamma^u r_{i,u}$ with

$$\tilde{G}_{t,i} = \sum_u \gamma^u r_{i,u}^{\text{SRPO}} = G_{t,i} + \lambda\,g_{t,i}\sum_u \gamma^u w_{i,u}.$$

Hence the per-trajectory gradient estimator becomes

$$\underbrace{\sum_u \nabla_\theta \log \pi_\theta(a_{i,u}\,|\,s_{i,u})\, G_{t,i}}_{\text{base PG}} \;+\; \underbrace{\lambda\, g_{t,i} \sum_u \nabla_\theta \log \pi_\theta(a_{i,u}\,|\,s_{i,u})\, \bar{w}_{i,u}}_{\text{SRPO shaping}}, \tag{12}$$

where we define $\bar{w}_{i,u} := \gamma^u w_{i,u}$ (the rank weights filtered by discounting). If an advantage baseline is used, the same decomposition holds with $G_{t,i}$ replaced by the baseline-corrected advantage and $\bar{w}_{i,u}$ replaced by its correspondingly filtered version.

b) Directional amplification at the step level: When the trajectory improves over the reference ($g_{t,i} > 0$), the rank rule assigns larger $w_{i,u}$ to higher-quality steps (Sec. IV-B). For two steps u, v in the same trajectory with $w_{i,u} > w_{i,v}$, the shaped signal increases the margin between their contributions to the update by $\lambda\, g_{t,i}\,(\bar{w}_{i,u} - \bar{w}_{i,v}) > 0$. Thus SRPO *widens the preference* for better decisions within the trajectory. If $g_{t,i} < 0$, the emphasis flips and the update pushes the policy away from poorly ranked steps, providing automatic self-correction.

c) Scale/shift invariance and robustness: Because $w_{i,u}$ depends only on *ranks* of a per-step proxy (reward, return-to-go, or advantage), the allocation is invariant to strictly monotone rescalings or offsets of that proxy. Meanwhile, $g_{t,i}$ is normalized by a running dispersion σ_J (Eq. (4)), producing a score that is well-conditioned across tasks with different reward magnitudes or signs. Together, these properties remove a common source of brittleness in driving tasks: drifting reward scales.

d) Bounded variance contribution: The added term in (12) is uniformly bounded by $\lambda |g_{t,i}|$ since $\sum_u \bar{w}_{i,u} = \sum_u \gamma^u w_{i,u} \le \sum_u \gamma^u \le 1/(1-\gamma)$ and $|g_{t,i}| \le g_{\max}$ (by clipping). Because $g_{t,i}$ is computed from trajectory returns and normalized by σ_J, its variance concentrates with batch size, adding a low-variance global direction that stabilizes training (observed empirically as tighter confidence bands).

C. Connection to the practical rank-based SRPO

Bounds-based allocation requires explicit global reward bounds ($R_{\min}$ or $R_{\max}$), which are typically obtained only after the training phase, or, in some cases, estimated initially from theoretical limits, though such limits may not always exist. Our implemented SRPO replaces the bounds-based allocation with *rank-based* weights $w_{i,u}$ and uses the trajectory-level, normalized self-reflection score $g_{t,i}$ from (4). This yields two desirable properties:

- *Bounded perturbation of the policy gradient.* With $w_{i,u} \ge 0$, $\sum_u w_{i,u} = 1$, and $|g_t| \le g_{\max}$, the additive term is uniformly bounded by $\lambda\, g_{\max}$. For small λ (and standard trust-region/clipping), the shaped update remains within the usual policy-improvement regime.
- *Scale and shift invariance.* Because $w_{i,u}$ depends only on *ranks* within a trajectory, it is invariant to any strictly monotone transform of the per-step proxy (reward, return-to-go, or advantage). This avoids failure modes tied to reward rescaling or sign flips.

The bounds-based variant provides clean policy-invariance and evaluation convergence guarantees (Thms. 1–2). The practical rank-based SRPO preserves these stability intuitions while trading exact invariance for a robust, scale-free credit assignment that performs well empirically.

VI. Simulation Experiments

We evaluate the proposed SRPO framework in two autonomous driving environments with increasing levels of fidelity and complexity. The first, Highway-env [30], enables controlled testing of policy learning, credit assignment, and sample efficiency in tactical driving scenarios. The second, CARLA [31], is a high-fidelity urban simulator used to assess SRPO's robustness, adaptability, and generalization under more realistic vehicle dynamics. Together, these environments allow us to examine SRPO across key axes of interest: training stability, reward shaping efficacy, robustness to adversarial perturbations, and out-of-distribution (OOD) generalization.

A. Experiments in Highway-env

We first evaluate SRPO on the `highway-v0` task in Highway-env, which models multi-lane tactical driving. The ego vehicle must continuously adjust its lane and speed in response to surrounding traffic to maximize safety and driving efficiency. A schematic of the task is shown in Fig. 2.

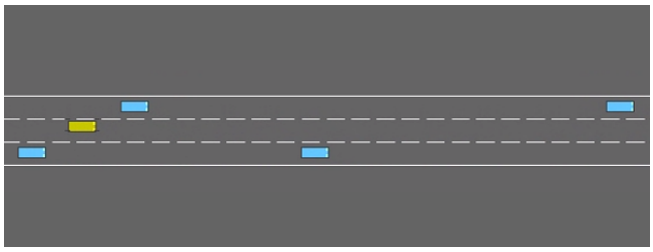

Fig. 2: Schematic of the `highway-v0` task in Highway-env.

The observation at each timestep is a matrix $O_t \in \mathbb{R}^{5\times 4}$, encoding absolute positions and velocities of the ego and four nearest vehicles. If fewer than four vehicles are nearby, the matrix is zero-padded. The agent selects a continuous control input $u_t = [a_t, \delta_t]^\top \in [-6, 6] \times [-0.2, 0.2]$, where a_t is the longitudinal acceleration and δ_t the lateral (steering) command. The reward is composed of six components:

$$R_t = R_{\text{vel}} + R_{\text{head}} + R_{\text{imp}} + R_{\text{risk}} + R_{\text{lane}} + R_{\text{crash}},$$

each promoting safe, efficient, and smooth driving as:

$$R_{\text{vel}} = -\big[|v_{\min} - v_t|\,\mathbf{1}_{\{v_t < v_{\min}\}} + |v_t - v_{\max}|\,\mathbf{1}_{\{v_t > v_{\max}\}}\big],$$
$$R_{\text{head}} = -(T_{\text{des}} - \text{THW}_t)\,\mathbf{1}_{\{\text{THW}_t \le T_{\text{des}}\}}, \quad R_{\text{imp}} = -|a_t - a_{t-1}|,$$
$$R_{\text{risk}} = -(d_{\text{safe}} - d_t)\,\mathbf{1}_{\{d_t < d_{\text{safe}}\}}, \quad R_{\text{lane}} = -c_{\text{lane}}\,\mathbf{1}_{\{\text{lane change}\}} - c_\delta |\delta_t|,$$
$$R_{\text{crash}} = \begin{cases} -r_{\text{crash}}, & \text{if collision or off-road,} \\ +r_{\text{surv}}, & \text{otherwise.} \end{cases}$$

The coefficients are set as: $v_{\min} = 25\,\text{m/s}$, $v_{\max} = 30\,\text{m/s}$, $T_{\text{des}} = 2\,\text{s}$, $d_{\text{safe}} = 30\,\text{m}$, $c_{\text{lane}} = 0.5$, $c_\delta = 5$, $r_{\text{crash}} = 5$, $r_{\text{surv}} = 2$.

We adopt Proximal Policy Optimization (PPO) [32] as our primary baseline, and include Soft Actor-Critic (SAC) [33] for comparison. To assess robustness, we also evaluate against Robust Adversarial Reinforcement Learning (RARL) [34], which improves policies through adversarially generated perturbations. Finally, we include a trajectory-level shaping variant based on Eq. (4) as an ablation to assess the effect of step-level credit assignment.

We begin by evaluating the training performance of SRPO using the episode return on the `highway-v0` task. As shown in Fig. 3, SRPO converges faster and achieves the highest return among all methods. This reflects its improved sample efficiency and learning stability, driven by the self-reflection mechanism and rank-based credit assignment. Trajectory-level shaping offers modest improvements over PPO and SAC but lacks fine-grained guidance due to its uniform credit allocation. RARL performs worst in this nominal setting, consistent with its robustness-oriented training that sacrifices early convergence speed.

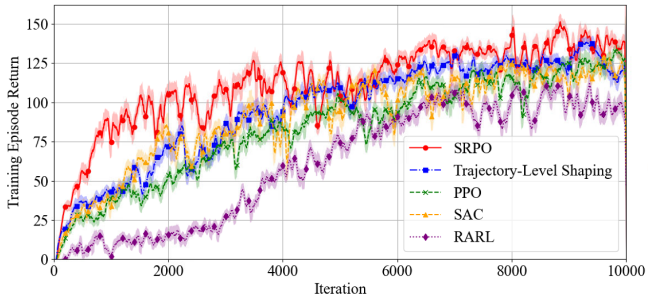

Fig. 3: Training episode return of the proposed method, trajectory-level shaping (4), PPO [32], SAC [33], RARL [34] for the `highway-v0`. SRPO converges faster and achieves the highest final performance.

To assess robustness, we evaluate all methods in a perturbed test environment where: (i) surrounding vehicles are injected with zero-mean Gaussian noise, and (ii) traffic density is increased from 8 to 20 vehicles. Specifically, for all non-ego IDM/MOBIL agents, we add Gaussian noise to their control actions at each step: $\sigma_{\text{long}} = 2\,\text{m/s}^2$ in the longitudinal direction and $\sigma_{\text{lat}} = 0.2\,\text{rad/s}$ in lateral steering. These disturbances simulate unmodeled real-world variability.

We report per-step violations on four safety and control metrics:

- *Velocity Violation* ($v_{\text{violation}}$): deviation from the target speed range,
- *Time Headway Violation* ($\text{THW}_{\text{violation}}$): insufficient time gap to the lead vehicle,
- *Impulsivity* ($a_{\text{impulsivity}}$): magnitude of acceleration changes, indicating jerky control,
- *Distance Violation* ($d_{\text{violation}}$): proximity below the safe following distance.

Results in Table I show that RARL achieves the lowest violation rates across all metrics, consistent with its robustness-driven training objective. SRPO consistently ranks second, outperforming PPO, SAC, and trajectory-level shaping in every category. This suggests that self-reflection and rank-based shaping confer strong robustness even without explicit adversarial training, by directing gradient updates toward consistently high-return behaviors.

B. Experiments in CARLA

We further evaluate SRPO in a high-fidelity adaptive cruise control (ACC) scenario using the CARLA simulator (v0.9.13, Town03). The ego vehicle is tasked with following a lead vehicle at a safe distance, maintaining speed limits, and minimizing abrupt control, all under realistic road and vehicle dynamics. A schematic is shown in Fig. 4.

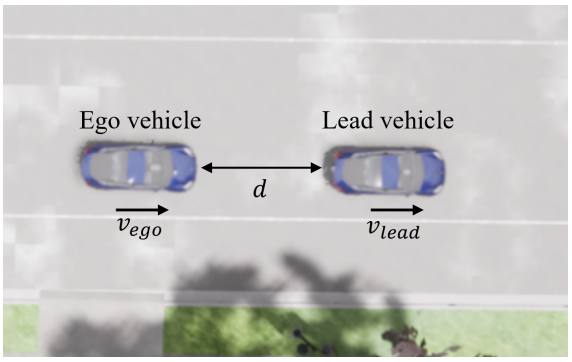

Fig. 4: Adaptive cruise control (ACC) setup in CARLA. The ego vehicle tracks a lead car using throttle/brake, while lateral control is handled separately.

The system state is $s_t = [d_t, v_t, v_t^{\text{lead}}, a_t^{\text{lead}}]$, where d_t is the gap to the lead vehicle, v_t is ego speed, and v_t^{lead}, a_t^{lead} are the velocity and acceleration of the lead vehicle. The ego agent selects a continuous action $a_t \in [-1, 1]$, interpreted as throttle (if $a_t > 0$) or brake (if $a_t < 0$). Lateral control is handled via a simple proportional controller:

$$\delta = \text{clip}\left(K_p(\psi - \psi_{\text{ref}}), -1, 1\right),$$

where $K_p = 0.1$, ψ is the current yaw, and $\psi_{\text{ref}} = 180°$ is the target heading. The lead vehicle follows a velocity-tracking controller:

$$a_t^{\text{lead}} = \text{clip}\left(K_v(v_{\text{des}} - v_t^{\text{lead}}), -1, 1\right),$$

with $K_v = 0.5$ and desired speed $v_{\text{des}} = 5\,\text{m/s}$.

The reward encourages safe, smooth, and efficient driving:

$$R_t = R_{\text{safety}} + \alpha R_{\text{eff}} + \beta R_{\text{comfort}},$$

$$R_{\text{safety}} = -(d_{\text{safe}} - d_{t+1})\,\mathbf{1}_{\{d_{t+1} < d_{\text{safe}}\}},$$

$$R_{\text{eff}} = -(v_{\min} - v_{t+1})\,\mathbf{1}_{\{v_{t+1} < v_{\min}\}} - (v_{t+1} - v_{\max})\,\mathbf{1}_{\{v_{t+1} > v_{\max}\}},$$

$$R_{\text{comfort}} = -|a_t|.$$

The reward coefficients are: $\alpha = 0.5$, $\beta = 0.3$, $d_{\text{safe}} = 10\,\text{m}$, $v_{\min} = 5\,\text{m/s}$, and $v_{\max} = 10\,\text{m/s}$.

We compare SRPO against PPO [32], SAC [33], RARL [34], and a trajectory-level shaping variant (4) in the ACC task. As shown in Fig. 5, SRPO achieves the highest training return, followed by the trajectory-level shaping variant. This confirms that shaping based on relative policy improvement is beneficial even when applied uniformly at the trajectory level, but that step-level credit assignment further boosts learning efficiency. RARL performs

Metric	SRPO	Trajectory-Level Shaping	PPO	SAC	RARL
$v_{\text{violation}}$	$(3.32 \pm 0.12) \times 10^{-2}$	$(3.83 \pm 0.14) \times 10^{-2}$	$(5.23 \pm 0.15) \times 10^{-2}$	$(4.74 \pm 0.18) \times 10^{-2}$	$\mathbf{(2.22 \pm 0.10) \times 10^{-2}}$
$\text{THW}_{\text{violation}}$	$(1.41 \pm 0.08) \times 10^{-1}$	$(1.68 \pm 0.11) \times 10^{-1}$	$(1.89 \pm 0.18) \times 10^{-1}$	$(2.23 \pm 0.17) \times 10^{-1}$	$\mathbf{(1.25 \pm 0.12) \times 10^{-1}}$
$a_{\text{impulsivity}}$	$\mathbf{(4.31 \pm 0.19) \times 10^{-4}}$	$(5.33 \pm 0.18) \times 10^{-4}$	$(6.19 \pm 0.11) \times 10^{-4}$	$(6.32 \pm 0.14) \times 10^{-4}$	$(4.35 \pm 0.13) \times 10^{-4}$
$d_{\text{violation}}$	0.95 ± 0.06	1.15 ± 0.07	1.34 ± 0.15	1.27 ± 0.11	$\mathbf{0.59 \pm 0.06}$

TABLE I: Per-step violation metrics (± standard deviation) under adversarial test conditions in `highway-v0`. Lower is better. SRPO consistently improves safety and control over PPO [32], SAC [33], and trajectory-level shaping (4).

worst in this nominal setting, consistent with its robustness-oriented design that prioritizes adversarial resilience over sample efficiency. These trends mirror those observed in the `highway-v0` task and further support the general effectiveness of SRPO.

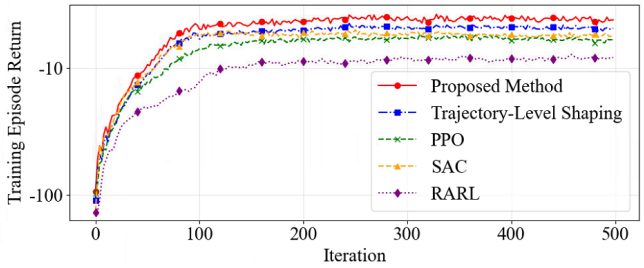

Fig. 5: Training episode return in the CARLA adaptive cruise control task. SRPO converges faster and outperforms trajectory-level shaping (4), PPO [32], SAC [33], and RARL [34].

To evaluate robustness, we test all methods under increasing levels of environmental disturbance in the CARLA ACC task. Specifically, we inject i.i.d. zero-mean Gaussian noise into the lead vehicle's acceleration, with variances of 0, 1, 2, and 4 representing zero, low, medium, and high noise regimes, respectively. The zero-noise setting serves as a nominal baseline.

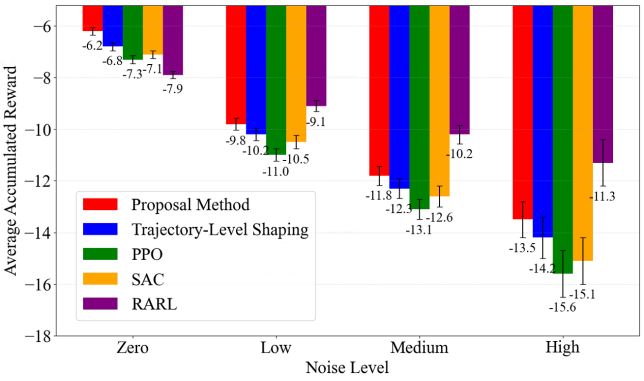

Fig. 6: Average accumulated reward (± standard deviation) across noise levels in the CARLA adaptive cruise control task. SRPO maintains strong performance across all regimes compared to trajectory-level shaping (4), PPO [32], SAC [33], and RARL [34].

As shown in Fig. 6, performance degrades across all methods as noise increases, consistent with greater uncertainty in the lead vehicle's behavior. RARL achieves the highest reward under severe perturbations, reflecting its robustness-focused training. However, its performance drops in nominal conditions, highlighting a trade-off between adversarial resilience and general policy quality. In contrast, SRPO achieves the highest return in the zero-noise setting and consistently ranks second across all noise levels. This indicates strong robustness without requiring adversarial training. We attribute this to SRPO's self-reflection mechanism and rank-based credit assignment, which together guide the policy toward consistently high-return behaviors even under stochastic disturbances.

Autonomous vehicles often operate in environments that differ significantly from training conditions due to variations in road surface, weather, and perception quality. To evaluate generalization, we test each method in out-of-distribution (OOD) scenarios using CARLA's configurable physics and perception models. We vary two environment parameters: friction level E_1 (reflecting road grip) and dryness level E_2 (affecting perception quality). Higher values correspond to more favorable conditions. The policy is trained in a nominal setting with $E_1 = 0.5$, $E_2 = 1.0$, and evaluated in three OOD configurations:

1) *Env. 1 (Low friction)*: $E_1 = 0.2$, $E_2 = 1.0$ — simulating icy or snowy roads.
2) *Env. 2 (Low perception)*: $E_1 = 0.5$, $E_2 = 0.2$ — simulating fog, rain, or degraded sensors.
3) *Env. 3 (High friction)*: $E_1 = 1.0$, $E_2 = 1.0$ — ideal driving conditions on high-grip asphalt.

Method	**Env. 1**	**Env. 2**	**Env. 3**
SRPO	$\mathbf{-7.99 \pm 0.20}$	$\mathbf{-6.89 \pm 0.11}$	$\mathbf{-5.14 \pm 0.11}$
Trajectory-Level Shaping	-8.52 ± 0.22	-7.62 ± 0.14	-5.90 ± 0.12
PPO	-8.71 ± 0.22	-8.12 ± 0.17	-6.21 ± 0.11
SAC	-8.35 ± 0.19	-7.85 ± 0.18	-6.05 ± 0.10
RARL	-9.02 ± 0.23	-9.21 ± 0.16	-7.52 ± 0.13

TABLE II: Average accumulated reward (± standard deviation) in three OOD environments. SRPO achieves the best performance in all cases compared to trajectory-level shaping (4), PPO [32], SAC [33], and RARL [34].

As shown in Table II, SRPO achieves the highest reward in all three OOD settings. In particular, it outperforms PPO and SAC by a large margin, particularly in low-friction (Env. 1) and low-perception (Env. 2) conditions. Trajectory-level shaping improves slightly over PPO but lacks the step-level granularity needed for robust generalization. RARL performs worst overall, despite its robustness in adversarial training, indicating poor transfer to OOD environments. These results highlight SRPO's generalization advantage, which we attribute to its self-reflection mechanism that promotes stable improvement over time and its rank-based shaping that preserves gradient quality across distributions.

VII. Conclusion

We presented Self-Reflection Policy Optimization (SRPO), a reinforcement learning algorithm that introduces policy-level self-evaluation via performance benchmarking and reward shaping. SRPO computes a relative improvement signal and redistributes it using rank-based step-level credit assignment, yielding a scale-invariant shaping mechanism compatible with standard policy gradient methods. We provided theoretical guarantees that a bounds-based variant of SRPO preserves both policy optimality and convergence. Empirical results on Highway-env and CARLA demonstrate that SRPO improves training efficiency, robustness, and generalization under adversarial and out-of-distribution conditions. Future work includes extending SRPO to sparse-reward domains, where step-level shaping can guide exploration, and incorporating time-aware penalties to accelerate convergence and improve temporal credit allocation.

References

[1] A. Folkers, M. Rick, and C. Büskens, "Controlling an autonomous vehicle with deep reinforcement learning," in *2019 IEEE intelligent vehicles symposium (IV)*. IEEE, 2019, pp. 2025–2031.

[2] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740–759, 2020.

[3] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[4] M. Emamifar and S. F. Ghoreishi, "Physics-informed particle-based reinforcement learning for autonomy in signalized intersections," *International Journal of Intelligent Transportation Systems Research*, vol. 22, no. 2, pp. 416–430, 2024.

[5] D. A. Schön, *The reflective practitioner: How professionals think in action*. Routledge, 2017.

[6] R. Sharma and P. Garg, "Optimizing autonomous driving with advanced reinforcement learning: Evaluating dqn and ppo," in *2024 5th International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, 2024, pp. 910–914.

[7] H. Yuan, P. Li, B. Van Arem, L. Kang, H. Farah, and Y. Dong, "Safe, efficient, comfort, and energy-saving automated driving through roundabout based on deep reinforcement learning," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 6074–6079.

[8] M. Gao and D. E. Chang, "Autonomous driving based on modified sac algorithm through imitation learning pretraining," in *2021 21st international conference on control, automation and systems (ICCAS)*. IEEE, 2021, pp. 1360–1364.

[9] X. Wang, J. Zhang, D. Hou, and Y. Cheng, "Autonomous driving based on approximate safe action," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 14 320–14 328, 2023.

[10] W. Zhang, M. Elmahgiubi, K. Rezaee, B. Khamidehi, H. Mirkhani, F. Arasteh, C. Li, M. A. Kaleem, E. R. Corral-Soto, D. Sharma, *et al.*, "Analysis of a modular autonomous driving architecture: The top submission to carla leaderboard 2.0 challenge," *arXiv preprint arXiv:2405.01394*, 2024.

[11] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9329–9338.

[12] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," *arXiv preprint arXiv:2106.11810*, 2021.

[13] M. Zhou, J. Luo, J. Villella, Y. Yang, D. Rusu, J. Miao, W. Zhang, M. Alban, I. Fadakar, Z. Chen, *et al.*, "Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving," *arXiv preprint arXiv:2010.09776*, 2020.

[14] M. Cusumano-Towner, D. Hafner, A. Hertzberg, B. Huval, A. Petrenko, E. Vinitsky, E. Wijmans, T. Killian, S. Bowers, O. Sener, *et al.*, "Robust autonomy emerges from self-play," *arXiv preprint arXiv:2502.03349*, 2025.

[15] C. Gulino, J. Fu, W. Luo, G. Tucker, E. Bronstein, Y. Lu, J. Harb, X. Pan, Y. Wang, X. Chen, *et al.*, "Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research," *Advances in Neural Information Processing Systems*, vol. 36, pp. 7730–7742, 2023.

[16] D. Wang and S. F. Ghoreishi, "Entropy-regularized two-stage state-only imitation learning under unknown dynamics," in *2026 American Control Conference (ACC)*. IEEE, 2026.

[17] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *International conference on machine learning*. PMLR, 2017, pp. 2817–2826.

[18] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1–7.

[19] D. Wang and S. F. Ghoreishi, "RGDR: Reward-guided domain randomization for autonomous driving," in *2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC 2025), IEEE*, 2025.

[20] C. Zhang, S. Biswas, K. Wong, K. Fallah, L. Zhang, D. Chen, S. Casas, and R. Urtasun, "Learning to drive via asymmetric self-play," in *European Conference on Computer Vision*. Springer, 2024, pp. 149–168.

[21] D. Wang and S. F. Ghoreishi, "Robust reinforcement learning for autonomous driving in uncertain environments," in *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*. IEEE, 2025, pp. 1436–1443.

[22] N. Asadi, S. H. Hosseini, M. Imani, D. P. Aldrich, and S. F. Ghoreishi, "Privacy-preserved federated reinforcement learning for autonomy in signalized intersections," in *International Conference on Transportation and Development 2024*, 2024, pp. 390–403.

[23] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Icml*, vol. 99. Citeseer, 1999, pp. 278–287.

[24] H. P. Van Hasselt, A. Guez, M. Hessel, V. Mnih, and D. Silver, "Learning values across many orders of magnitude," *Advances in neural information processing systems*, vol. 29, 2016.

[25] J. Oh, Y. Guo, S. Singh, and H. Lee, "Self-imitation learning," in *International conference on machine learning*. PMLR, 2018, pp. 3878–3887.

[26] A. Laterre, Y. Fu, M. K. Jabri, A.-S. Cohen, D. Kas, K. Hajjar, T. S. Dahl, A. Kerkeni, and K. Beguir, "Ranked reward: Enabling self-play reinforcement learning for combinatorial optimization," *arXiv preprint arXiv:1807.01672*, 2018.

[27] A. Ajay, A. Kumar, P. Agrawal, S. Levine, and O. Nachum, "Opal: Offline primitive discovery for accelerating offline reinforcement learning," *arXiv preprint arXiv:2010.13611*, 2020.

[28] S. Lahlou, M. Jain, H. Nekoei, V. I. Butoi, P. Bertin, J. Rector-Brooks, M. Korablyov, and Y. Bengio, "DEUP: Direct epistemic uncertainty prediction," *arXiv preprint arXiv:2102.08501*, 2021.

[29] M. Emamifar and S. F. Ghoreishi, "Uncertainty-aware reinforcement learning for safe control of autonomous vehicles in signalized intersections," in *2023 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2023, pp. 81–82.

[30] E. Leurent *et al.*, "An environment for autonomous driving decision-making," 2018.

[31] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[33] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 1861–1870.

[34] A. Srinivasan, A. Paras, and A. Bera, "Adversarial agent behavior learning in autonomous driving using deep reinforcement learning," *arXiv preprint arXiv:2508.15207*, 2025.