Preference-Guided Adversarial Policy Optimization for Long-Tail Robust Driving

Tong Nie[1,2*] **Yihong Tang**[3*] **Junlin He**[1] **Yuewen Mei**[2]
Jie Sun[2] **Lijun Sun**[3] **Wei Ma**[1†] **Jian Sun**[2†]

[1]The Hong Kong Polytechnic University, Hong Kong SAR, China
[2]Tongji University, Shanghai, China
[3]McGill University, Montreal, QC, Canada

Abstract

Deploying autonomous driving systems requires robustness against long-tail scenarios that are rare but safety-critical. While adversarial training offers a promising solution, existing methods typically decouple scenario generation from policy optimization and rely on heuristic surrogates. This leads to objective misalignment and fails to capture the shifting failure modes of evolving policies. This paper presents `ADV-0`, a closed-loop min-max optimization framework that treats the interaction between driving policy (defender) and adversarial agent (attacker) as a zero-sum Markov game. By aligning the attacker's utility directly with the defender's objective, we reveal the optimal adversary distribution. To make this tractable, we cast dynamic adversary evolution as iterative preference learning, efficiently approximating this optimum and offering an algorithm-agnostic solution to the game. Theoretically, our analysis characterizes the idealized regularized game and derives a lower bound that connects adversarial training performance with real-world long-tail generalization. Experiments indicate that `ADV-0` effectively exposes diverse safety-critical failures and greatly enhances the generalizability of both learned policies and motion planners against unseen long-tail risks.

1 Introduction

Deploying autonomous driving (AD) systems in the open world faces a crucial bottleneck: the inability to anticipate and handle rare but safety-critical long-tail scenarios. While massive naturalistic driving data provide a basis for model development, they are dominated by normal logs where high-risk events like aggressive cut-ins appear with negligible frequency [Liu and Feng, 2024, Xu et al., 2025b]. The growing literature has introduced sampling and generative methods to accelerate the discovery of rare events [Feng et al., 2023, Ding et al., 2023]. However, they are often confined to stress testing or performance validation, failing to actively target long-tail risks. Effectively leveraging these synthetic data to improve the generalizability of AD policies in the long tail remains an open question.

Closed-loop adversarial training offers an avenue to address this challenge by exposing the training policy to synthetic risks. This paradigm can naturally be formulated as a min-max bi-level optimization problem via a zero-sum Markov game, involving an adversary that generates challenges and a defender that optimizes the policy [Pinto et al., 2017]. Despite its theoretical elegance, direct application in AD and robotics has been hindered by nontrivial optimization issues. First, end-to-end solutions via gradient descent are often computationally intractable due to the non-differentiable nature of physical simulators and the difficulty of propagating gradients through long-horizon rollouts to provide learning signals. Second, the zero-sum interaction between two players is prone to instability and mode collapse [Zhang et al., 2020b], where the adversary converges to unrealistic attack patterns, limiting the scalability of this framework.

*Equal contribution.
†Corresponding authors: Wei Ma (`wei.w.ma@polyu.edu.hk`) and Jian Sun (`sunjian@tongji.edu.cn`).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To bypass computational difficulties, existing methods typically decouple the min-max objective into separate sub-problems: generating scenarios via fixed priors or heuristics, then training the agent against this static distribution [Zhang et al., 2023, 2024, Stoler et al., 2025]. However, this decoupled paradigm introduces notable limitations: (1) *Misaligned*. It creates a misalignment between the goals of the two players. While the defender optimizes a comprehensive reward for safety, efficiency, and comfort, the attacker solely targets collisions, relying on heuristic surrogates such as collision probability. First, this discrepancy renders the adversarial objective ill-defined, often resulting in an overly aggressive attacker that overwhelms the defender and destabilizes training [Zhang et al., 2020b]. Second, the divergence in gradient directions prevents the attacker from identifying non-collision failures like off-road violations and providing meaningful learning signals. Thus, the defender can overfit to specific collision modes while remaining vulnerable to broader risks [Vinitsky et al., 2020], without acquiring generalized robustness. (2) *Nonstationary*. Decoupled methods with fixed attack modes fail to uncover the nonstationary vulnerability frontier—the shifting boundary of scenarios where the current policy remains prone to failure. As the defender evolves, its failure modes shift further into the long-tail distribution, becoming increasingly rare under the initial prior. Fixed adversaries with static priors are insufficient to track these shifting weaknesses, causing fragile generalization to unseen risks. (3) *Uncertified*. Training against such static priors or heuristics is an empirical trial-and-error process, lacking theoretical safety guarantees. This lack of certified performance bounds contrasts sharply with the rigorous safety demands of real-world deployment, where agents must generalize to unbounded unknown long-tail scenarios [Brunke et al., 2022].

This work revisits the min-max formulation and introduces `ADV-O`, a closed-loop policy optimization framework that enables end-to-end training for generalizable adversarial learning. `ADV-O` solves the Markov game by directly aligning the adversarial utility with the training objective of the defender. To address tractability and stability issues, we propose an online iterative preference learning algorithm, casting adversarial evolution as preference optimization. This allows the attacker to continuously track the shifting vulnerability frontier of the improving defender. By coupling the evolution, `ADV-O` tailors the distribution towards the long tail of the current players, forcing the defender to learn generalized robustness rather than overfitting to heuristics. Importantly, `ADV-O` is algorithm-agnostic, applicable to both RL agents and motion planning models, providing a theoretically grounded pathway from adversarial generation to policy improvement. Our contributions are threefold:

- `ADV-O` is a closed-loop training framework for long-tail problems of AD that couples adversarial generation and policy optimization in an end-to-end way.
- We propose a preference-based solution to the zero-sum game, a stable and efficient realization supporting on-policy interaction and algorithm-agnostic evolution.
- Theoretically, we characterize the idealized game and establish a lower bound that links adversarial training to real-world long-tail performance. Empirically, `ADV-O` not only exposes diverse safety-critical events but also enhances the generalizability of policies against unseen long-tail risks.

2 Preliminary and Problem Formulation

Task description. We model the safe AD task as a Markov Decision Process (MDP), defined by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, T)$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces. The state $s_t \in \mathcal{S}$ includes raw sensor inputs, high-level commands, and kinematic status. The action $a_t \in \mathcal{A}$ represents continuous low-level control signals (e.g., steering, acceleration). The environment dynamics $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ describe the transition probabilities of the traffic scene. We initialize scenarios using real-world driving logs, where the ego vehicle is controlled by a policy π_θ parameterized by θ. Background traffic participants are initially governed by naturalistic behavior priors, such as log-replay or traffic models like Intelligent Driver Model (IDM). The reward function $\mathcal{R}(s, a)$ is designed to balance task progress with safety: it encourages route completion and velocity tracking, while imposing heavy penalties for safety violations, such as collisions or off-road events. The goal of the ego agent is to learn an optimal policy π_θ^* that maximizes the expected cumulative return $J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta, \mathcal{P}}[\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t)]$ within T. Unlike imitation learning which assumes a fixed data distribution, we focus on online RL to enable the agent to recover from adversarial perturbations in closed-loop interactions.

Min-max formulation. Relying solely on naturalistic scenarios often fails to expose the ego agent to low-probability but high-risk events residing in the long tail of the distribution. To ensure the robustness of the policy against long-tail risks, adversarial training frames the problem as a robust optimization task via a **two-player zero-sum game** between the ego agent and an adversary. Here, the behaviors of background agents are governed by a parameterized adversarial policy $\psi \in \Psi$ altering

transition dynamics from naturalistic $\mathcal{P}$ to adversarial $\mathcal{P}_\psi$. This yields the min-max objective:

$$\max_{\theta} \min_{\psi \in \Psi} \left[J(\pi_\theta, \mathcal{P}_\psi) = \mathbb{E}_{\tau \sim (\pi_\theta, \mathcal{P}_\psi)} [\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t)] \right], \tag{1}$$

where Ψ represents the feasible set of adversarial configurations that remain physically plausible. The outer loop maximizes the ego's performance, while the inner loop seeks an adversary minimizing the current ego's reward. Directly optimizing the bi-level objective via gradient descent is often computationally intractable due to the non-differentiable nature of physical simulators and the difficulty of propagating gradients through long-horizon rollouts. Existing methods often decouple Eq. 1 into separate problems: (1) *Adversarial generation*: generating a static set of hard scenarios via surrogate objectives J_{adv} (e.g., collision probability); (2) *Policy optimization*: then optimizing π_θ against this fixed distribution. However, this decoupled paradigm introduces objective misalignment and fails to capture the non-stationary vulnerability frontier of the evolving policy.

3 The ADV-0 Framework

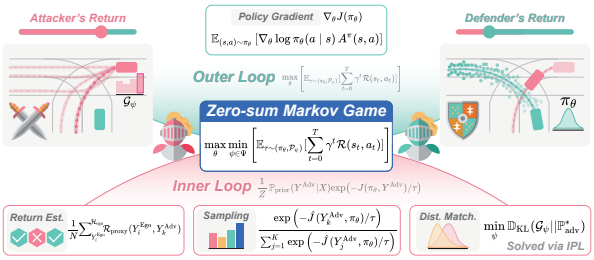

Figure 1: Illustration of ADV-0. It alternates between an *Inner Loop* where the adversary ψ evolves via IPL to track the failure frontier, and an *Outer Loop* where the ego π_θ optimizes policy gradients against the induced adversarial distribution.

We introduce ADV-0, a closed-loop adversarial training framework to solve the min-max optimization problem in Eq. 1. Our approach treats the interaction between the driving agent (*defender*) and the traffic environment (*attacker*) as a dynamic *zero-sum game*: the defender minimizes the expected risk, while the attacker continuously explores the long-tail distribution to identify and exploit the ego's evolving weaknesses. Due to the nonstationarity of the driving environment and the rarity of critical events, relying on static datasets or heuristic adversarial priors is insufficient. Instead, we seek a Nash Equilibrium where the ego policy π_θ remains robust to the worst-case distributions generated by a continuously evolving adversary policy ψ. At this equilibrium, the performance of π_θ can theoretically be related to other distributions within the trust region. This section details this end-to-end bi-level optimization scheme.

3.1 End-to-End Framework for Min-Max Optimization

To enable tractability and stationarity, we propose an iterative end-to-end training pipeline inspired by Robust Adversarial Reinforcement Learning (RARL, Pinto et al. [2017]). The core of ADV-0 lies in efficiently propagating gradients from the ego's objective J to the adversary ψ. The training alternates between two phases: (1) an *inner loop* updating ψ to track the theoretical optimal attack distribution; and (2) an *outer loop* optimizing policy π_θ against the current induced distribution (Figure 1).

Inner loop. Let X denote the context (e.g., map topology and initial traffic state) and Y^{Adv} denote future adversarial trajectories. We define critical event $\mathcal{E}$ as the set of scenarios where the ego's performance drops below a safety threshold ϵ: $\mathcal{E} := \{Y^{\text{Adv}} \mid J(\pi_\theta, Y^{\text{Adv}}) \leq \epsilon\}$. Then the attacker's goal is to find an optimal adversarial distribution $\mathbb{P}_{\text{adv}}$ maximizing likelihood of $\mathcal{E}$ while constrained within the trust region of the naturalistic prior $\mathbb{P}_{\text{prior}}$. This is equivalent to a constrained optimization:

$$\max_{\mathbb{P}_{\text{adv}}} \mathbb{E}_{X \sim \mathcal{D}} \left[\mathbb{E}_{Y^{\text{Adv}} \sim \mathbb{P}_{\text{adv}}(\cdot|X)} [\log \mathbb{P}(\mathcal{E})] \right], \quad \text{s.t. } \mathbb{D}_{\text{KL}}(\mathbb{P}_{\text{adv}} || \mathbb{P}_{\text{prior}}) \leq \delta. \tag{2}$$

Directly solving Eq. 2 involves optimizing a hard indicator function $\mathbb{I}(Y^{\text{Adv}} \in \mathcal{E})$, suffering from vanishing gradients and event sparsity. To enable end-to-end gradient optimization, we relax the hard constraint into a soft energy function. We view the negative return $-J(\pi_\theta, Y^{\text{Adv}})$ as the unnormalized log-likelihood of the critical event. The objective is thus relaxed to maximizing the expected adversarial utility $\mathbb{P}(\mathcal{E}|Y^{\text{Adv}}) \approx \mathbb{E}_{Y^{\text{Adv}} \sim \mathbb{P}_{\text{adv}}} \left[-J(\pi_\theta, Y^{\text{Adv}}) \right]$ within the trust region. By applying Lagrange multipliers, the optima $\mathbb{P}^*_{\text{adv}}$ can be derived in closed form as the Gibbs distribution:

$$\underbrace{\mathbb{P}^*_{\text{adv}}(Y^{\text{Adv}}|X)}_{\text{Posterior}} = \frac{1}{Z} \underbrace{\mathbb{P}_{\text{prior}}(Y^{\text{Adv}}|X)}_{\text{Traffic prior}} \underbrace{\exp\big(-J(\pi_\theta, Y^{\text{Adv}})/\tau\big)}_{\text{Generalized adversarial utility}}, \tag{3}$$

where Z is the partition function and τ is the temperature. Eq. 3 reveals that the theoretically optimal adversary re-weights the traffic prior based on the generalized adversarial utility (GAU). Traffic prior ensures plausibility, and GAU is the likelihood that Y^{Adv} causes the current policy π_θ to fail. However, directly sampling from $\mathbb{P}^*_{\text{adv}}(\cdot|X)$ is intractable due to the unknown partition function. `ADV-0` solves for this via a two-step approximation: (1) **Sampling (Sec. 3.2)**: We approximate expectations over $\mathbb{P}^*_{\text{adv}}$ using importance sampling from the current prior; and (2) **Learning (Sec. 3.3)**: We update the parameterized adversary ψ to approximate the theoretical optimum $\mathbb{P}^*_{\text{adv}}$ via preference learning.

Outer loop. With the adversary ψ_{k+1} fixed, the ego updates its policy to maximize the expected return under the induced adversarial distribution using standard RL methods:

$$\theta^{k+1} \leftarrow \arg\max_{\theta^k} J(\pi^k_\theta, \mathcal{P}_{\psi_{k+1}}). \tag{4}$$

Crucially, this general framework is agnostic to the ego's specific RL method. Since the adversary interacts with the ego solely through generated trajectories, the outer loop supports both on-policy and off-policy algorithms, by adjusting the synchronization schedule between the two. Moreover, `ADV-0` can be applied to π_θ either outputs continuous control signals (e.g., acceleration) or future trajectory plans (e.g., multi-modal trajectories with scores). Training proceeds by fixing one player while updating the other. This coupled iteration ensures that the attacker dynamically tracks the defender's vulnerability frontier, while the defender learns to generalize against an increasingly sophisticated attacker. The complete training procedure is summarized in Algorithm 1 in Appendix D.2.

3.2 Reward-guided Adversarial Sampling & Alignment

To approximate the optimal distribution $\mathbb{P}^*_{\text{adv}}$ without computationally expensive MCMC, we adopt a generate-and-resample paradigm. Instead of generating trajectories from scratch, we sample from a pretrained multi-modal trajectory generator $\mathcal{G}_\psi$ that approximates $\mathbb{P}_{\text{prior}}(\cdot|X)$. Given a context X, the current $\mathcal{G}_\psi$ produces K candidates $\{Y^{\text{Adv}}_k\}_{k=1}^K \sim \mathcal{G}_\psi(\cdot|X)$ with prior probabilities. We then re-weight these candidates to approximate samples from $\mathbb{P}^*_{\text{adv}}$ using GAU.

Direct objective alignment. Prior works [Zhang et al., 2022, 2023] simplify the GAU via heuristic surrogates, e.g., collision probability. However, this introduces a misalignment: the defender optimizes comprehensive reward (safety, efficiency, comfort), while the attacker solely targets collisions. This discrepancy in gradient direction prevents the attacker from identifying non-collision failures (e.g., off-road) and allows the defender to overfit to specific collision modes while remaining vulnerable to other risks. In contrast, `ADV-0` directly aligns the GAU with the defender's objective by *setting the energy function as the negation of ego's cumulative return* (Eq. 3). By targeting exactly what the ego optimizes, the attacker offers holistic supervision signals across the entire reward space, whether they are safety violations or efficiency drops.

Efficient return estimation. Evaluating the exact return $\mathbb{E}[J(\pi_\theta, Y^{\text{Adv}}_k)]$ for all K candidates via closed-loop simulation requires fully rolling out the policy, which is computationally prohibitive. To address this, we propose a *Proxy Reward Evaluator* that estimates expected return via a lightweight function query. Since the ego's response to Y^{Adv} is stochastic during training, we treat J as a random variable. We maintain a context-aware dynamic buffer of the ego's recent responses $\mathcal{H}_{\text{ego}}(X) = \{Y^{\text{Ego}}_i|X\}_{i=1}^N$ containing the N most recent trajectories generated by π_θ in context X. For a new context with an empty buffer, we perform a single warm-up rollout using π_θ against the replay log to initialize the buffer. We treat $\mathcal{H}_{\text{ego}}$ as an empirical approximation of current policy distribution. The expected return for a candidate Y^{Adv}_k is estimated via Monte Carlo integration against the history:

$$\hat{J}(Y^{\text{Adv}}_k, \pi_\theta) \approx \frac{1}{N} \sum_{Y^{\text{Ego}}_i \in \mathcal{H}_{\text{ego}}} \mathcal{R}_{\text{proxy}}(Y^{\text{Ego}}_i, Y^{\text{Adv}}_k), \tag{5}$$

where $\mathcal{R}_{\text{proxy}}$ is a vectorized function that computes geometric interactions (e.g., progress, collision overlap) between the adversary path and the cached ego paths without stepping the physics engine. This rule-based proxy provides an efficient and sufficiently accurate gradient direction for adversarial sampling (see Appendix D.2.1 for implementation).

Temperature-scaled sampling. Finally, to select the adversarial trajectory for training, we implement the Gibbs distribution (Eq. 3) over the finite set of K candidates. The probability of selecting

candidate k is given by the scaled softmax distribution over negative estimated returns:

$$\mathbb{P}(Y_k^{\text{Adv}}) = \frac{\exp\left(-\hat{J}(Y_k^{\text{Adv}}, \pi_\theta)/\tau\right)}{\sum_{j=1}^{K} \exp\left(-\hat{J}(Y_j^{\text{Adv}}, \pi_\theta)/\tau\right)}. \tag{6}$$

This serves as an *importance sampling* step, re-weighting the proposal distribution $\mathcal{G}_\psi$ towards the theoretical optimum $\mathbb{P}^*_{\text{adv}}$ (see Appendix B.1). τ balances exploration and exploitation: $\tau \to 0$ selects the worst case, while larger τ retains diversity from the prior. It ensures that the defender is exposed to diverse challenging scenarios from the long tail, rather than collapsing into a single worst case.

3.3 Iterative Preference Learning in the Long Tail

While the sampling strategy in Section 3.2 identifies hard cases within the support of $\mathcal{G}_\psi$, this fixed proposal bounds its efficacy. As the defender π_θ improves, its weakness shifts into the long tail where the prior $\mathcal{G}_{\text{ref}}$ has negligible mass. Relying solely on static sampling becomes inefficient. To track the shifting frontier, we update ψ to match the distribution of generator $\mathcal{G}_\psi$ and optimal target $\mathbb{P}^*_{\text{adv}}$.

Implicit reward optimization via preferences. Formally, our goal is to minimize the KL-divergence between $\mathcal{G}_\psi$ and $\mathbb{P}^*_{\text{adv}}$. Recall the definition in Eq. 3, we have:

$$\begin{aligned}
&\min_\psi \mathbb{D}_{\text{KL}}(\mathcal{G}_\psi || \mathbb{P}^*_{\text{adv}}) = \min_\psi \mathbb{E}_{Y\sim\mathcal{G}_\psi}\left[\log \mathcal{G}_\psi(Y|X)/\mathbb{P}^*_{\text{adv}}(Y|X)\right] \\
&= \min_\psi \mathbb{E}_{Y\sim\mathcal{G}_\psi}\left[\log \frac{\mathcal{G}_\psi(Y|X)}{\mathcal{G}_{\text{ref}}(Y|X)} + \frac{1}{\tau} J(\pi_\theta, Y)\right] + \text{const.} \\
&\iff \max_\psi \mathbb{E}_{Y\sim\mathcal{G}_\psi}\left[-J(\pi_\theta, Y)\right] - \tau \mathbb{D}_{\text{KL}}(\mathcal{G}_\psi || \mathcal{G}_{\text{ref}}).
\end{aligned} \tag{7}$$

Eq. 7 reveals a standard RL objective: maximizing the expected adversarial reward subject to a KL-divergence constraint. However, directly solving it via policy gradient is notoriously unstable in this context due to the high variance of gradients. The action space of continuous trajectories is high-dimensional, and the zero-sum interaction often leads to mode collapse. Instead of explicit RL, we cast the problem as preference learning. Preference learning updates the generator of adversarial traffic trajectories $\mathcal{G}_\psi$ using candidate orderings induced by the current defender's cumulative return. Within each context, the candidate with lower estimated defender return is preferred, and these labels are refreshed as the defender evolves. Following Rafailov et al. [2023], the optimal policy for the KL-constrained objective satisfies a specific preference ordering, which is equivalent to optimizing an *implicit reward*. This allows us to update ψ using a supervised loss on preference pairs, bypassing the need for an explicit value function or unstable reward maximization.

Online iterative evolution. Standard preference learning methods are often offline with static preference datasets. Instead, `ADV-O` operates in a nonstationary game where the preference labels depend on evolving players. Therefore, we propose online *Iterative Preference Learning* (IPL) in the inner loop. IPL generates preference data on-the-fly, conditioned on the current attacker and labels it using the current defender. This process proceeds on-policy: **(1) Sampling**: For a given context X, a series of candidates are generated $\{Y_k^{\text{Adv}}\}_{k=1}^K$ from the *current* attacker $\mathcal{G}_\psi$. **(2) Labeling**: Each candidate is evaluated using the proxy reward evaluator $\hat{J}(\cdot, \pi_\theta)$. Crucially, this evaluation uses the *latest* history of the defender. **(3) Pairing**: A preference dataset $\mathcal{D}_{\text{pref}}$ is curated by pairing (Y_w, Y_l) from the candidates, where Y_w is preferred over Y_l if $\hat{J}(Y_w, \pi_\theta) < \hat{J}(Y_l, \pi_\theta)$. To prevent trivial comparisons, we apply a reward margin δ and a spatial diversity filter ξ: $\mathcal{D}_{\text{pref}} = \left\{(Y_w, Y_l) \mid \hat{J}(Y_l) - \hat{J}(Y_w) > \delta \ \wedge \ \|Y_w - Y_l\|_2 > \xi\right\}$. This reduces noise in the proxy, making preferences distinguishable between structurally different attacks.

Objective. We update the adversarial policy $\mathcal{G}_\psi$ by minimizing the negative log-likelihood of the preferred trajectories. To handle high-variance heterogeneous traffic scenarios without high cost of a massive replay buffer, we employ a streaming gradient accumulation strategy to stabilize the training. We process a stream of scenarios sequentially, accumulating gradients over a mini-batch of M scenarios before performing a parameter update. The loss function $\mathcal{L}_{\text{IPL}}(\mathcal{G}_\psi)$ for a mini-batch $\mathcal{B}$ of generated pairs is: $\frac{-1}{|\mathcal{B}|}\sum_{(Y_w,Y_l)\in\mathcal{B}} \log\sigma\Big(\tau\big[\log\frac{\mathcal{G}_\psi(Y_w|X)}{\mathcal{G}_{\text{ref}}(Y_w|X)} - \log\frac{\mathcal{G}_\psi(Y_l|X)}{\mathcal{G}_{\text{ref}}(Y_l|X)}\big]\Big)$. (8). This reduces per-scenario update variance while maintaining on-policy data generation. Here, the reference model $\mathcal{G}_{\text{ref}}$ remains frozen as the pretrained prior. This streaming evolution ensures that the adversary

continuously adapts to the defender's evolving capabilities. By pushing the generator's distribution towards the theoretical Gibbs optimum $\mathbb{P}^*_{\text{adv}}$, the defender continuously trains against the most pertinent long-tail risks, effectively mitigating the distribution shift and forcing robust generalization.

3.4 Theoretical Analysis

We provide a theoretical analysis of the convergence properties of ADV-0 and establish a generalization bound that characterizes the agent's performance in real-world long-tail distributions. Derivations and proofs are provided in Appendix B.

Convergence to Nash Equilibrium. The interaction between the defender and the attacker is modeled as a regularized zero-sum Markov game. Building on the finding that optimizing Eq. 8 recovers the optimal adversary solved for a Gibbs distribution, we prove that iterative updates constitute a contraction mapping on the value function space.

Theorem 3.1 (Convergence to Nash Equilibrium)**.** *Under exact inner optimization and policy improvement, the idealized regularized updates converge to a unique fixed point corresponding to the Nash Equilibrium* (π^*, ψ^*) *of the game. It satisfies the saddle-point inequality* $\mathcal{J}_\tau(\pi, \mathcal{G}_{\psi^*}) \leq \mathcal{J}_\tau(\pi^*, \mathcal{G}_{\psi^*}) \leq \mathcal{J}_\tau(\pi^*, \mathcal{G}_\psi)$ *for all feasible policies, where* $\mathcal{J}_\tau$ *is the regularized objective.*

Generalization to real-world long tail. A core concern is whether robustness against a generated adversary $\mathcal{P}_\psi$ translates to safety in the real-world long-tail distribution $\mathcal{P}_{\text{real}}$. We model real dynamics as an unknown distribution lying within the trust region of the traffic prior. We derive a lower bound on expected return by measuring the discrepancy between two induced transition dynamics.

Theorem 3.2 (Generalizability)**.** *Let* $V_{\max}$ *be the maximum of the value function,* $\mathcal{P}_{real}$ *be the real dynamics, and* π_θ *is trained under the adversarial dynamics* $\mathcal{P}_\psi$ *induced by* $\mathcal{G}_\psi$*. The performance of* π_θ *under* $\mathcal{P}_{real}$ *is bounded by:*

$$J(\pi_\theta, \mathcal{P}_{real}) \geq J(\pi_\theta, \mathcal{P}_\psi) - \frac{\gamma V_{\max}\sqrt{2}}{1-\gamma}\sqrt{\mathbb{E}[D_{KL}(\mathcal{G}_\psi \| \mathcal{G}_{ref})]}.$$

Theorem 3.2 implies that optimizing against the generated adversary improves a lower bound on the expected return in the real world. The outer loop maximizes the robust return J, while the inner loop optimizes a KL-regularized adversary objective that controls deviation from the naturalistic prior.

4 Experiments

We empirically evaluate ADV-0 to answer three core questions: (1) Can ADV-0 generate plausible yet long-tailed scenarios that effectively expose the vulnerabilities of driving agents? (2) Does the training process yield a robust policy that generalizes to diverse adversarial attacks? (3) Can the safety improvements observed in simulation transfer to real-world long-tailed events? All experiments are performed in MetaDrive simulator using scenarios converted from WOMD.

4.1 Generating Safety-Critical Scenarios

Main results. We first evaluate ADV-0 in generating safety-critical scenarios against various ego policies. As presented in Table 1, ADV-0 consistently outperforms competing baselines in exposing system vulnerabilities, especially for reactive policies such as IDM and RL. A detailed ablation in Table 9 further highlights two findings: (1) The proposed energy-based sampling strategy (Eq. 6) is effective compared to the standard logit-based scheme. (2) IPL-based fine-tuning, which is agnostic to the underlying RL algorithm, further refines the adversary. By fine-tuning the generator against specific ego policies, the adversary learns to exploit specific weaknesses.

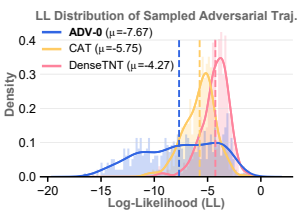

Figure 2: LL distribution of different adversarial generators.

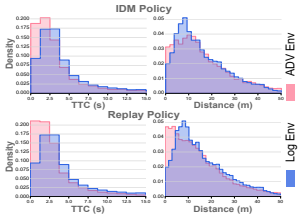

Figure 3: Scene-level distributions of B2B distance and TTC.

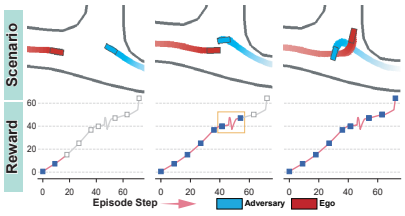

Figure 4: Examples of reward-reduced adversarial scenarios from ADV-0. Additional cases are shown in Figure 15.

Table 1: **Safety-critical scenario generation**. *CR* denotes the collision rate (%) of the ego in generated scenarios, and *ER* is the ego's cumulative return. Ego agents include Replay, IDM, and trained PPO. `ADV-0` entries are macro means across six fine-tuned generator variants; Table 9 provides the corresponding dispersion statistics.

Ego	Metric	Adversary								
		Replay	Heuristic	CAT	KING	AdvTrajOpt	GOOSE	SAGE	SEAL	**ADV-0 (ours)**
Replay	CR (%) ↑	-	100.00	90.08	23.28	69.64	20.46	74.53	59.06	**91.10**
	ER ↓	-	0.00	1.03	47.67	3.12	48.52	2.57	8.58	**0.99**
IDM	CR (%) ↑	19.03	74.70	43.13	24.49	26.92	24.48	36.50	36.70	**45.83**
	ER ↓	51.75	32.12	43.47	49.41	45.36	49.45	43.81	43.75	**40.03**
RL Agent	CR (%) ↑	16.80	69.64	36.84	21.26	28.95	13.88	35.40	37.63	**40.68**
	ER ↓	51.65	24.90	42.26	49.32	45.10	51.80	42.87	41.99	**39.13**

Table 2: **Performance validation of learned policies**. Each environment reports the macro mean across six RL algorithms (GRPO, PPO, PPO-Lag, SAC, SAC-Lag, TD3) ± the mean of the standard deviations computed over five seeds for each algorithm. Avg. averages these statistics across the five environments. Full results are shown in Tables 10, 11, 12, 13, 14, 15.

Val. Env.	ADV-0 (w/ IPL)			
	RC ↑	Crash ↓	Reward ↑	Cost ↓
Replay	0.742 ± 0.011	0.159 ± 0.019	49.56 ± 1.05	0.480 ± 0.013
ADV-0	0.695 ± 0.011	0.289 ± 0.029	44.60 ± 0.83	0.598 ± 0.022
CAT	0.704 ± 0.011	0.271 ± 0.025	45.68 ± 1.16	0.585 ± 0.025
SAGE	0.699 ± 0.013	0.263 ± 0.025	45.19 ± 1.54	0.567 ± 0.028
Heuristic	0.710 ± 0.016	0.217 ± 0.021	45.41 ± 2.08	0.552 ± 0.035
Avg.	**0.710 ± 0.012**	**0.240 ± 0.024**	**46.09 ± 1.33**	**0.556 ± 0.025**

Val. Env.	ADV-0 (w/o IPL)				CAT			
	RC ↑	Crash ↓	Reward ↑	Cost ↓	RC ↑	Crash ↓	Reward ↑	Cost ↓
Replay	0.719 ± 0.020	0.167 ± 0.024	46.41 ± 1.97	0.540 ± 0.023	0.720 ± 0.014	0.183 ± 0.026	46.52 ± 1.15	0.528 ± 0.015
ADV-0	0.664 ± 0.018	0.317 ± 0.030	41.03 ± 1.87	0.657 ± 0.020	0.660 ± 0.018	0.332 ± 0.043	40.70 ± 0.97	0.660 ± 0.020
CAT	0.677 ± 0.017	0.299 ± 0.035	42.30 ± 1.68	0.643 ± 0.030	0.667 ± 0.017	0.313 ± 0.032	41.30 ± 1.48	0.652 ± 0.018
SAGE	0.672 ± 0.024	0.270 ± 0.028	42.17 ± 2.38	0.610 ± 0.038	0.660 ± 0.021	0.307 ± 0.025	41.41 ± 2.09	0.628 ± 0.017
Heuristic	0.679 ± 0.022	0.239 ± 0.041	41.78 ± 2.08	0.612 ± 0.040	0.676 ± 0.021	0.261 ± 0.034	41.72 ± 1.69	0.605 ± 0.023
Avg.	**0.683 ± 0.020**	**0.258 ± 0.031**	**42.74 ± 2.00**	**0.613 ± 0.030**	**0.676 ± 0.018**	**0.279 ± 0.032**	**42.33 ± 1.47**	**0.614 ± 0.019**

Val. Env.	Heuristic				Replay			
	RC ↑	Crash ↓	Reward ↑	Cost ↓	RC ↑	Crash ↓	Reward ↑	Cost ↓
Replay	0.682 ± 0.032	0.201 ± 0.018	42.75 ± 3.04	0.592 ± 0.037	0.692 ± 0.035	0.209 ± 0.030	42.93 ± 1.32	0.588 ± 0.025
ADV-0	0.637 ± 0.047	0.331 ± 0.021	39.09 ± 2.12	0.678 ± 0.028	0.622 ± 0.038	0.374 ± 0.038	36.80 ± 2.48	0.700 ± 0.020
CAT	0.650 ± 0.021	0.311 ± 0.020	40.11 ± 2.38	0.668 ± 0.033	0.642 ± 0.040	0.368 ± 0.037	38.26 ± 1.83	0.683 ± 0.030
SAGE	0.642 ± 0.042	0.312 ± 0.019	39.53 ± 2.58	0.658 ± 0.025	0.625 ± 0.040	0.339 ± 0.040	37.68 ± 1.84	0.660 ± 0.032
Heuristic	0.641 ± 0.030	0.279 ± 0.016	38.81 ± 2.61	0.655 ± 0.028	0.638 ± 0.031	0.310 ± 0.034	37.80 ± 2.41	0.672 ± 0.015
Avg.	**0.650 ± 0.034**	**0.287 ± 0.019**	**40.06 ± 2.54**	**0.650 ± 0.030**	**0.644 ± 0.037**	**0.320 ± 0.036**	**38.70 ± 1.97**	**0.661 ± 0.024**

Distribution of the long tail. `ADV-0` can navigate the long-tail distribution of scenarios. Figure 2 illustrates the log-likelihood (LL) distribution of the sampled adversarial trajectories. Compared to CAT and pretrained prior, the distribution of `ADV-0` is shifted towards lower likelihood and exhibits a wider variance. This indicates that it uncovers rare but plausible, *behavior-level* events that are typically ignored by standard priors. Figure 3 shows the *scene-level* statistics. `ADV-0` produces notably lower Time-to-Collision (TTC) and closer Bumper-to-Bumper (B2B) distances compared to the naturalistic data. Crucially, this aggressiveness does not compromise physical plausibility. As shown in Figure 9, `ADV-0` maintains a comparable realism penalty.

Beyond collision. `ADV-0` directly targets the ego's return, allowing for the discovery of diverse failure modes beyond crashes. As shown in Figure 4 and Figure 15, the adversary can force the ego into abnormal behaviors and non-collision failures, such as stalling at intersections or deviating from reference paths. These behaviors result in a drastic drop in accumulated reward (e.g., lack of progress or discomfort penalties) even if a collision is avoided, which are critical performance risks often ignored by collision-centric attacks.

Table 3: **Cross-validation of** `ADV-0`. Results are macro means across six RL algorithms ± the mean of their standard deviations over five seeds. Decrease and Improvement report percentage changes from adding IPL to the adversary and agent, respectively.

Agent/Adversary	Reward (↑)			Cost (↓)		
	Adversary (w/o IPL)	Adversary (w/ IPL)	Decrease	Adversary (w/o IPL)	Adversary (w/ IPL)	Decrease
Agent w/o IPL	41.03 ± 1.87	39.01 ± 0.97	-4.92%	0.657 ± 0.020	0.685 ± 0.016	$+4.26\%$
Agent w/ IPL	44.60 ± 0.83	43.47 ± 1.40	-2.53%	0.598 ± 0.022	0.615 ± 0.025	$+2.84\%$
Improvement	$+8.70\%$	$+11.43\%$	–	-8.98%	-10.22%	–

4.2 Learning Generalizable Driving Policies

Generalization to unseen adversaries. A core challenge in adversarial RL is overfitting to a specific adversary, which limits generalization to unseen risks. We conduct a cross-validation where agents trained via different methods are tested across a spectrum of environments. Table 2 reports the performance averaged across six RL algorithms. We observe that agents trained with `ADV-0` (w/ IPL) consistently achieve the best results across all metrics. While baselines often perform well against their own attacks, they degrade when applied to unseen adversarial distributions. In contrast, `ADV-0` maintains consistent generalizability. The comparison between `ADV-0` (w/o IPL) and CAT indicates the benefit of directly aligning the GAU with the ego's objective. By actively exploring the long tail via IPL, `ADV-0` enables generalized robustness to handle unseen risks.

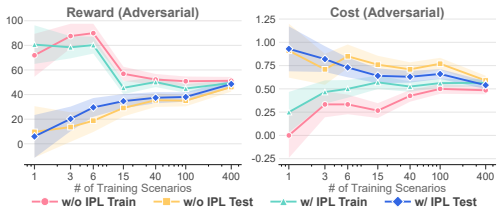

Figure 5: Generalization gap over different budgets.

Impact of IPL. To isolate the gain provided by IPL, we compare the performance of agents and adversaries trained with and without IPL in Table 3. The inclusion of IPL in the adversary creates a more challenging environment, evidenced by the drop in agent rewards. However, the agent trained with full `ADV-0` becomes more robust when facing the stronger adversary. This confirms that the dynamic evolution of the adversary via IPL forces the defender to cover broader vulnerabilities. Qualitative evidence of this dynamic evolution is shown in Figure 13. In the TD3 setting, `ADV-0` also achieves the highest route completion and return and the lowest crash rate and cost among the five methods compared in Appendix C.2. Finally, we study the sample efficiency and overfitting risks regarding the training budget. As shown in Figure 5, the gap between training and testing performance narrows with IPL, indicating that active discovery of diverse failures effectively mitigates the risk of overfitting to limited training data. Detailed learning curves and results are provided in Figures 10-11 and Tables 10-15.

Table 4: **Robustness on mined real-world long-tailed sets.** Average agent performance on long-tail scenario categories filtered by: *Critical TTC* ($\min \text{TTC} < 0.4s$), *Critical PET* ($\text{PET} < 1.0s$), *Hard Dynamics* (Acc. $< -4.0m/s^2$ or $|\text{Jerk}| > 4.0m/s^3$), and *Rare Cluster* (topologically sparse trajectory clusters). Metrics assess average *Safety Margin* (higher indicates earlier risk detection), *Stability & Comfort* (lower Jerk indicates smoother control), *Defensive Driving*, quantified by Near-Miss Rate (proximity without collision), RDP Violation (percentage of time requiring deceleration $> 6m/s^2$ to avoid collision). Full results are shown in Table 18.

	Reactive Traffic	Safety Margin		Stability & Comfort		Defensive Driving	
		Avg Min-TTC (↑)	Avg Min-PET (↑)	Mean Abs Jerk (↓)	95% Jerk (↓)	Near-Miss Rate (↓)	RDP Violation Rate (↓)
ADV-0 (w/ IPL)	✓	$\mathbf{0.993 \pm 0.189}$	$\mathbf{1.150 \pm 0.317}$	$\mathbf{1.653 \pm 0.169}$	$\mathbf{5.053 \pm 0.885}$	$\mathbf{63.45\% \pm 4.34\%}$	$\mathbf{33.70\% \pm 3.77\%}$
	×	1.527 ± 0.182	$\mathbf{1.103 \pm 0.313}$	$\mathbf{1.617 \pm 0.204}$	$\mathbf{5.177 \pm 0.953}$	$\mathbf{60.97\% \pm 3.94\%}$	$\mathbf{36.45\% \pm 2.85\%}$
CAT	✓	0.825 ± 0.192	1.017 ± 0.317	1.875 ± 0.229	5.422 ± 0.833	$83.04\% \pm 5.99\%$	$44.80\% \pm 6.12\%$
	×	1.415 ± 0.462	0.980 ± 0.327	1.979 ± 0.193	6.002 ± 0.931	$75.36\% \pm 7.86\%$	$51.20\% \pm 4.74\%$
Heuristic	✓	0.864 ± 0.200	$\mathbf{1.150 \pm 0.467}$	2.129 ± 0.237	6.542 ± 0.942	$74.46\% \pm 5.98\%$	$48.79\% \pm 3.65\%$
	×	$\mathbf{1.601 \pm 0.538}$	1.093 ± 0.450	2.199 ± 0.218	6.778 ± 1.066	$68.82\% \pm 9.05\%$	$52.10\% \pm 1.77\%$
Replay	✓	0.587 ± 0.185	0.683 ± 0.217	2.779 ± 0.431	8.265 ± 1.718	$89.53\% \pm 4.30\%$	$69.73\% \pm 5.27\%$
	×	0.978 ± 0.462	0.730 ± 0.233	2.720 ± 0.383	8.104 ± 1.537	$85.13\% \pm 4.58\%$	$68.50\% \pm 4.68\%$

4.3 Improving Policy Robustness in the Long Tail



Figure 6: Visualization of improved safe driving ability after being trained with ADV-0. More examples are shown in Figure 14.

Main results. While ADV-0 has demonstrated robustness against generated adversaries, it is crucial to verify whether this performance generalizes to real-world, naturally occurring long-tail events. To this end, we curate unbiased held-out sets mined from real-world WOMD logs, including only high-risk scenarios categorized by extreme safety criticality (e.g., critical TTC, PET) and semantic rarity (e.g., rare behaviors). As shown in Table 4 (detailed in Table 18), agents trained via ADV-0 demonstrate superior zero-shot robustness compared to baselines. It achieves the highest safety margins and stability scores, significantly reducing the rates of near misses and RDP violations, indicating that the agent learns to anticipate risks before they become critical, rather than merely reacting to emergencies. Examples in Figures 6 and 14 further visualize these defensive behaviors: ADV-0 agent proactively yields to aggressive cut-ins and navigates sudden occlusions where baseline agents fail.

4.4 Algorithmic Analysis

Applications to motion planners. To show the generality of ADV-0, we extend our evaluation beyond RL agents to two kinds of SOTA learning-based trajectory planners: PlanTF [Cheng et al., 2024] (multimodal scoring) and SMART [Wu et al., 2024] (autoregressive generation). As shown in Tables 5 and 16, adversarial fine-tuning via ADV-0 yields consistent improvements for both architectures. We further analyze the internal behavior of the planners using the trajectory-level breakdown in Table 17. The results indicate that fine-tuned models learn to prioritize safety constraints significantly more than pretrained priors. This safety improvement comes with a trade-off in efficiency. Interestingly, we observe that the performance of these planners remains slightly lower than RL agents discussed previously. We attribute this to two factors: (1) the covariate shift in behavior cloning models [Karkus et al., 2025], where they struggle to recover when the adversary forces it into out-of-distribution states; and (2) the latency introduced by the re-planning horizon. Unlike end-to-end RL policies that output immediate control actions, they generate a future trajectory executed by a controller. This delayed reaction limits the ability to react instantaneously to aggressive attacks.

Table 5: **Application to learning-based planners**. Performance comparisons of two SOTA trajectory planning models before and after fine-tuning using ADV-0 (GRPO). See Table 16 for details.

Metric	PlanTF			SMART		
	Base	+ ADV-0	Rel. Change	Base	+ ADV-0	Rel. Change
RC ↑	0.628 ± 0.025	0.674 ± 0.016	**+7.46%**	0.587 ± 0.029	0.631 ± 0.015	**+7.57%**
Crash ↓	0.357 ± 0.031	0.263 ± 0.025	**-26.23%**	0.396 ± 0.034	0.305 ± 0.023	**-22.88%**
Reward ↑	35.85 ± 2.51	41.98 ± 1.62	**+17.11%**	32.66 ± 2.85	37.85 ± 1.70	**+15.86%**
Cost ↓	1.04 ± 0.04	0.77 ± 0.02	**-25.68%**	1.15 ± 0.05	0.92 ± 0.02	**-20.31%**

Impact of temperature parameter. We study the sensitivity of the sampling temperature τ in Eq. 6, which modulates the trade-off between adversarial exploitation and exploration. Figure 7 illustrates an obvious trade-off: (1) At extremely low temperatures ($\tau \to 0$), the sampling degenerates into a deterministic hard mode. This leads to suboptimal performance, likely because an overly aggressive attacker overwhelms the defender early in training, without learning generalized robustness. (2) Conversely, high values ($\tau = 5.0$) reduce the adversarial signal, degrading ADV-0 to domain randomization. The attacker fails to consistently expose weaknesses. The results indicate that a moderate value ($\tau = 0.1$) achieves the best balance. It introduces sufficient stochasticity to cover diverse long-tail distributions while maintaining enough focus to prioritize high-risk regions, effectively forming an automatic regularization.

Ablation on return estimator. The inner loop relies on the quality of the return estimator $\hat{J}$ used to label preferences. We compare our rule-based proxy against three baselines: *GTReward* (oracle simulation), *Experience* (retrieval from history), and *RewardModel* (learnable neural network). Figure 8 reports the IPL training curves. Surprisingly, the rule-based proxy achieves low and stable

preference loss comparable to the oracle, outperforming other estimators. Figure 12 measures a strong Spearman correlation of $\rho = 0.77$ between the proxy estimates and oracle returns. This suggests that the geometric proxy provides a high-fidelity and low-variance ranking signal, which is sufficient for adversarial sampling. Future work may explore using the Q-network from an Actor-Critic architecture for value estimation to handle more complex interactions. To test candidate selection against a reactive ego policy, we additionally evaluate 12,800 oracle rollouts across five seeds. The proxy achieves 0.80 pairwise accuracy and 0.84 recall of the oracle choice among the top four candidates; its selected attacks yield an ego return of 29.9, compared with 27.6 for oracle selection (Appendix C.1).

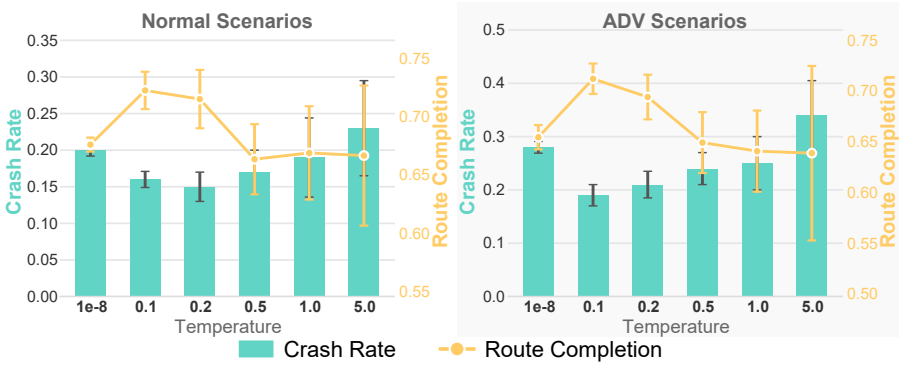

Figure 7: Impacts of the temperature parameter in sampling.

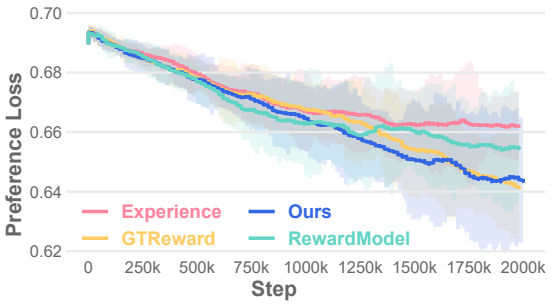

Figure 8: Impacts of different reward calculator schemes.

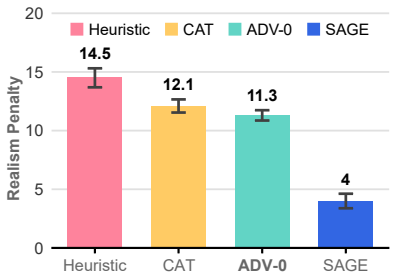

Figure 9: Realism penalty comparison.

5 Conclusion

This paper presents ADV-0, a closed-loop min-max policy optimization framework designed to enhance the robustness of AD models against long-tail risks. By formulating the problem as a zero-sum game and solving it via an alternating end-to-end training pipeline, we bridge the gap between adversarial generation and robust policy learning. Our theoretical analysis characterizes the idealized regularized game and relates the practical training objective to a lower bound on real-world long-tail performance. Empirical results suggest that ADV-0 not only generates effective safety-critical scenarios but also improves the generalizability of both RL agents and motion planners against diverse long-tail risks. Despite the promising results, several limitations exist. (1) The reliance on high-fidelity simulators for online RL training limits scalability. (2) Extending ADV-0 to vision-based sensor inputs may require differentiable neural rendering, which remains a non-trivial challenge. Future work could explore offline RL techniques [Karkus et al., 2025] to improve training efficiency and vision-based adversarial generation.

Acknowledgments and Disclosures

This work was supported by the National Natural Science Foundation of China (524B2164, 52125208, and 52232015) and the Otto Poon Charitable Foundation Smart Cities Research Institute, The Hong Kong Polytechnic University (CD06). The authors declare no competing interests.

References

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

Abdollah Amirkhani, Mohammad Parsa Karimi, and Amin Banitalebi-Dehkordi. A survey on adversarial attacks and defenses for object detection and their applications in autonomous vehicles. *The Visual Computer*, 39(11):5293–5307, 2023.

Luca Anzalone, Paola Barra, Silvio Barra, Aniello Castiglione, and Michele Nappi. An end-to-end curriculum learning approach for autonomous driving scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):19817–19826, 2022.

Adith Boloor, Xin He, Christopher Gill, Yevgeniy Vorobeychik, and Xuan Zhang. Simple physical adversarial examples against end-to-end autonomous driving models. In *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*, pages 1–7. IEEE, 2019.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.

Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. Model-free deep reinforcement learning for urban autonomous driving. In *2019 IEEE intelligent transportation systems conference (ITSC)*, pages 2765–2771. IEEE, 2019.

Keyu Chen, Wenchao Sun, Hao Cheng, and Sifa Zheng. Rift: Closed-loop rl fine-tuning for realistic and controllable traffic simulation. *arXiv preprint arXiv:2505.03344*, 2025.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking imitation-based planners for autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14123–14130. IEEE, 2024.

Hojun Chung, Junseo Lee, Minsoo Kim, Dohyeong Kim, and Songhwai Oh. Adversarial environment design via regret-guided diffusion models. *Advances in Neural Information Processing Systems*, 37:63715–63746, 2024.

Yao Deng, Tiehua Zhang, Guannan Lou, Xi Zheng, Jiong Jin, and Qing-Long Han. Deep learning-based autonomous driving systems: A survey of attacks and defenses. *IEEE Transactions on Industrial Informatics*, 17(12):7897–7912, 2021.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

Wenhao Ding, Baiming Chen, Minjun Xu, and Ding Zhao. Learning to collide: An adaptive safety-critical scenarios generating method. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2243–2250. IEEE, 2020.

Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. A survey on safety-critical driving scenario generation—a methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):6971–6988, 2023.

Shiyu Fang, Yiming Cui, Haoyang Liang, Chen Lv, Peng Hang, and Jian Sun. Corevla: A dual-stage end-to-end autonomous driving framework for long-tail scenarios via collect-and-refine. *arXiv preprint arXiv:2509.15968*, 2025.

Shuo Feng, Xintao Yan, Haowei Sun, Yiheng Feng, and Henry X Liu. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1):748, 2021.

Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953): 620–627, 2023.

Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15303–15312, 2021.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

Niklas Hanselmann, Katrin Renz, Kashyap Chitta, Apratim Bhattacharyya, and Andreas Geiger. King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients. In *European Conference on Computer Vision*, pages 335–352. Springer, 2022.

Xiangkun He, Haohan Yang, Zhongxu Hu, and Chen Lv. Robust lane change decision making for autonomous vehicles: An observation adversarial reinforcement learning approach. *IEEE Transactions on Intelligent Vehicles*, 8(1):184–193, 2022.

David Isele, Reza Rahimi, Akansel Cosgun, Kaushik Subramanian, and Kikuo Fujimura. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 2034–2039. IEEE, 2018.

Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning. *arXiv preprint arXiv:2503.07608*, 2025.

Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. *Advances in Neural Information Processing Systems*, 33:8127–8138, 2020.

Peter Karkus, Maximilian Igl, Yuxiao Chen, Kashyap Chitta, Jef Packer, Bertrand Douillard, Ran Tian, Alexander Naumann, Guillermo Garcia-Cobo, Shuhan Tan, et al. Beyond behavior cloning in autonomous driving: a survey of closed-loop training techniques. *Authorea Preprints*, 2025.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6):4909–4926, 2021.

W Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (mis) design for autonomous driving. *Artificial Intelligence*, 316:103829, 2023.

Sampo Kuutti, Saber Fallah, and Richard Bowden. Training adversarial agents to exploit weaknesses in deep control policies. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 108–114. IEEE, 2020.

Derun Li, Jianwei Ren, Yue Wang, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Zhongpu Xia, Peng Jia, Xianpeng Lang, et al. Finetuning generative trajectory model with reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*, 2025a.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022.

Yue Li, Meng Tian, Dechang Zhu, Jiangtong Zhu, Zhenyu Lin, Zhiwei Xiong, and Xinhai Zhao. Drive-r1: Bridging reasoning and planning in vlms for autonomous driving with reinforcement learning. *arXiv preprint arXiv:2506.18234*, 2025b.

Zihao Li, Xinyuan Cao, Xiangbo Gao, Kexin Tian, Keshu Wu, Mohammad Anis, Hao Zhang, Keke Long, Jiwan Jiang, Xiaopeng Li, et al. Simulating the unseen: Crash prediction must learn from what did not happen. *arXiv preprint arXiv:2505.21743*, 2025c.

Henry X Liu and Shuo Feng. Curse of rarity for autonomous vehicles. *nature communications*, 15 (1):4808, 2024.

Yuxin Liu, Zhenghao Peng, Xuanhao Cui, and Bolei Zhou. Adv-bmt: Bidirectional motion transformer for safety-critical traffic scenario generation. *arXiv preprint arXiv:2506.09485*, 2025.

Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. *arXiv preprint arXiv:1807.03858*, 2018.

Xiaobai Ma, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Improved robustness and safety for autonomous vehicle control with adversarial reinforcement learning. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1665–1671. IEEE, 2018.

Yuewen Mei, Tong Nie, Jian Sun, and Ye Tian. Bayesian fault injection safety testing for highly automated vehicles with uncertainty. *IEEE Transactions on Intelligent Vehicles*, 2024.

Yuewen Mei, Tong Nie, Jian Sun, and Ye Tian. Llm-attacker: Enhancing closed-loop adversarial scenario generation for autonomous driving with large language models. *arXiv preprint arXiv:2501.15850*, 2025.

Tong Nie, Yuewen Mei, Yihong Tang, Junlin He, Jie Sun, Haotian Shi, Wei Ma, and Jian Sun. Steerable adversarial scenario generation through test-time preference alignment. *arXiv preprint arXiv:2509.20102*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Xinlei Pan, Daniel Seita, Yang Gao, and John Canny. Risk averse robust adversarial reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8522–8528. IEEE, 2019.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International conference on machine learning*, pages 2817–2826. PMLR, 2017.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Joshua Ransiek, Johannes Plaum, Jacob Langner, and Eric Sax. Goose: Goal-conditioned reinforcement learning for safety-critical scenario generation. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2651–2658. IEEE, 2024.

Dhruv Mauria Saxena, Sangjae Bae, Alireza Nakhaei, Kikuo Fujimura, and Maxim Likhachev. Driving in dense traffic with model-free reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5385–5392. IEEE, 2020.

Bruno Scherrer. Approximate policy iteration schemes: A comparison. In *International Conference on Machine Learning*, pages 1314–1322. PMLR, 2014.

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

Benjamin Stoler, Ingrid Navarro, Jonathan Francis, and Jean Oh. Seal: Towards safe autonomous driving via skill-enabled adversary learning for closed-loop scenario generation. *IEEE Robotics and Automation Letters*, 10(9):9320–9327, 2025.

Yihong Tang, Haicheng Liao, Tong Nie, Junlin He, Ao Qu, Kehua Chen, Wei Ma, Zhenning Li, Lijun Sun, and Chengzhong Xu. E3ad: An emotion-aware vision-language-action model for human-centric end-to-end autonomous driving. *arXiv preprint arXiv:2512.04733*, 2025.

Aleksandar Taranovic, Andras Gabor Kupcsik, Niklas Freymuth, and Gerhard Neumann. Adversarial imitation learning with preferences. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=bhfp5GlDtGe`.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pages 6215–6224. PMLR, 2019.

Ran Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. *arXiv preprint arXiv:2407.00959*, 2024.

Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7153–7162, 2020.

James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13716–13725, 2020.

James Tu, Huichen Li, Xinchen Yan, Mengye Ren, Yun Chen, Ming Liang, Eilyan Bitar, Ersin Yumer, and Raquel Urtasun. Exploring adversarial robustness of multi-sensor perception systems in self driving. *arXiv preprint arXiv:2101.06784*, 2021.

Cumhur Erkan Tuncali, Georgios Fainekos, Hisahiro Ito, and James Kapinski. Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 1555–1562. IEEE, 2018.

Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre Bayen. Robust reinforcement learning using adversarial populations. *arXiv preprint arXiv:2008.01825*, 2020.

Akifumi Wachi. Failure-scenario maker for rule-based agent using multi-agent adversarial reinforcement learning and its application to autonomous driving. *arXiv preprint arXiv:1903.10654*, 2019.

Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9909–9918, 2021.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025a.

Yuping Wang, Shuo Xing, Cui Can, Renjie Li, Hongyuan Hua, Kexin Tian, Zhaobin Mo, Xiangbo Gao, Keshu Wu, Sulong Zhou, et al. Generative ai for autonomous driving: Frontiers and opportunities. *arXiv preprint arXiv:2505.08854*, 2025b.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. Smart: Scalable multi-agent real-time motion generation via next-token prediction. *Advances in Neural Information Processing Systems*, 37: 114048–114071, 2024.

Shuo Xing, Hongyuan Hua, Xiangbo Gao, Shenzhe Zhu, Renjie Li, Kexin Tian, Xiaopeng Li, Heng Huang, Tianbao Yang, Zhangyang Wang, et al. Autotrust: Benchmarking trustworthiness in large vision language models for autonomous driving. *arXiv preprint arXiv:2412.15206*, 2024.

Chejian Xu, Aleksandr Petiushko, Ding Zhao, and Bo Li. Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8797–8805, 2025a.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Ekaterina Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. *arXiv preprint arXiv:2510.26125*, 2025b.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in neural information processing systems*, 33:21024–21037, 2020a.

Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452*, 2021.

Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15459–15469, 2024.

Kaiqing Zhang, Bin Hu, and Tamer Basar. On the stability and convergence of robust adversarial reinforcement learning: A case study on linear quadratic systems. *Advances in Neural Information Processing Systems*, 33:22056–22068, 2020b.

Linrui Zhang, Zhenghao Peng, Quanyi Li, and Bolei Zhou. Cat: Closed-loop adversarial training for safe end-to-end driving. In *Conference on Robot Learning*, pages 2357–2372. PMLR, 2023.

Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, and Z Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15159–15168, 2022.

Weitao Zhou, Zhong Cao, Yunkang Xu, Nanshan Deng, Xiaoyu Liu, Kun Jiang, and Diange Yang. Long-tail prediction uncertainty aware trajectory planning for self-driving vehicles. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1275–1282. IEEE, 2022.

Appendix

Appendix Contents

A Related Work **16**

B Theoretical Analysis **17**
- B.1 Convergence Analysis . . . 17
  - B.1.1 Preliminaries: ADV-0 as a regularized zero-sum Markov game . . . 18
  - B.1.2 Inner loop optimality via implicit reward . . . 18
  - B.1.3 Global convergence to Nash Equilibrium . . . 19
- B.2 Generalization Bound and Safety Guarantees . . . 21
  - B.2.1 Discrepancy under shifted dynamics . . . 22
  - B.2.2 Adversarial generalization bound . . . 23
  - B.2.3 Safety guarantee in the long tail . . . 24
- B.3 Discussion on Theoretical Assumptions and Practical Implementation . . . 25

C Supplementary Tables and Figures **27**
- C.1 Evaluation with a Reactive Ego Policy . . . 33
- C.2 Comparisons of Preference Learning Methods . . . 33

D Detailed Experimental Setups **34**
- D.1 Environment and Dataset . . . 34
- D.2 Detailed Implementation . . . 34
  - D.2.1 Proxy reward estimator . . . 34
  - D.2.2 Evaluation and benchmark . . . 36
  - D.2.3 Adaptability to different RL algorithms . . . 37
  - D.2.4 Application to learning-based motion planning models . . . 38
- D.3 Baselines . . . 40
- D.4 Hyperparameters . . . 41

The appendix provides rigorous theoretical foundations, supplementary experimental results, and detailed implementation specifications that support the main text. We organize the contents as follows: Appendix A discusses related work and positions our work in the literature. Appendix B presents the complete derivations and proofs for the theoretical discussion provided in Section 3.4. We include additional qualitative visualizations and extended quantitative results in Appendix C to complement the main paper. Finally, Appendix D details the experimental setups, including the dataset, environment, baselines, implementations, and hyperparameters for reproducibility.

A Related Work

This section reviews the literature relevant to our approach, with a particular focus on autonomous driving (AD). We position our work at the intersection of three interleaving pathways: RL, long-tailed scenario handling, and adversarial learning.

Reinforcement learning. RL has been widely studied in AD to enable closed-loop decision-making and address the covariate shift inherent in supervised imitation [Kiran et al., 2021, Chen et al., 2024, Karkus et al., 2025]. Traditional approaches have largely focused on motion planning and continuous control using vectorized state representations [Isele et al., 2018, Saxena et al., 2020], or vision-based end-to-end driving within high-fidelity simulators [Chen et al., 2019, Toromanoff et al., 2020]. These methods typically employ actor-critic algorithms, such as PPO and SAC [Schulman et al., 2017, Haarnoja et al., 2018], to maximize cumulative returns based on handcrafted reward functions. However, reward specification and value estimation in complex driving scenarios remain notoriously difficult [Knox et al., 2023, Chen et al., 2024]. To address the difficulties, recent research has shifted toward alignment techniques emerging from large language models (LLMs). This includes learning from human preferences or feedback to recover reward functions from demonstrations [Ouyang

et al., 2022]. More notably, critic-free methods, such as Direct Preference Optimization (DPO) [Rafailov et al., 2023] and Group Relative Policy Optimization (GRPO) [Guo et al., 2025], are gaining increasing attention. These methods optimize policies directly against preference data or outcomes across rollouts without training unstable value functions, showing promising results in training end-to-end driving autonomy [Jiang et al., 2025, Li et al., 2025a,b].

Long-tailed scenario. Handling long-tailed events remains a longstanding challenge for AD deployment and system trustworthiness [Feng et al., 2021, Liu and Feng, 2024, Xing et al., 2024, Chen et al., 2024, Wang et al., 2025b]. To mitigate the scarcity of such data in naturalistic driving and provide high-value testing samples, significant effort has been devoted to safety-critical scenario generation [Ding et al., 2023, Li et al., 2025c]. Representative approaches range from rule-based [Tuncali et al., 2018, Zhang et al., 2024, Mei et al., 2024], optimization-based [Wang et al., 2021, Hanselmann et al., 2022, Zhang et al., 2022, 2023, Nie et al., 2025, Mei et al., 2025], to learning-based methods [Ding et al., 2020, Kuutti et al., 2020, Feng et al., 2023, Xu et al., 2025a, Liu et al., 2025]. Despite their success in identifying failures, effectively integrating these adversarial generation pipelines into closed-loop training remains an open question: the primary goal of existing methods is often to stress-test the system rather than to improve it. Conversely, a parallel line of research endeavors to enhance the robustness of decision-making in rare events by designing specialized architectures or leveraging the reasoning capabilities of pretrained LLMs/VLMs [Zhou et al., 2022, Tian et al., 2024, Fang et al., 2025, Xu et al., 2025b, Wang et al., 2025a, Tang et al., 2025]. However, adversarial scenario generation and policy improvement are seldom unified in a holistic framework. Consequently, the generalizability of these methods to unseen, open-world long-tailed scenarios remains under-investigated.

Adversarial learning for AD. Adversarial training offers a principled framework for improvement-targeted generation. In the context of robotics and autonomy, adversarial RL has been extensively discussed for control tasks in constrained settings [Pinto et al., 2017, Pan et al., 2019, Tessler et al., 2019, Zhang et al., 2020a,b, Vinitsky et al., 2020, Kamalaruban et al., 2020, Zhang et al., 2021]. However, they typically prioritize theoretical analysis within simplified simulation environments with controlled noise, which differs significantly from the complexity of real-world driving. Within the AD domain, adversarial methods have mainly targeted the robustness of perception and detection modules against observation perturbations [Boloor et al., 2019, Tu et al., 2020, 2021, Deng et al., 2021, He et al., 2022, Amirkhani et al., 2023]. In contrast, adversarial training for decision-making, particularly regarding long-tailed scenarios, remains underexplored [Ma et al., 2018, Wachi, 2019, Anzalone et al., 2022, Zhang et al., 2023, 2024]. Even among the few works addressing this, the generation and training phases are often decoupled. Crucially, they are typically confined to specific policy types or tested against handcrafted adversarial scenarios, which poses significant challenges to their generalizability across diverse and evolving corner cases.

Environment design and regret-guided generators. `ADV-0` is also related to unsupervised environment design and regret-guided environment generation, where the environment distribution is adapted as the learner improves [Tessler et al., 2019, Dennis et al., 2020, Chung et al., 2024]. The connection is conceptual: both lines seek informative training environments rather than passive evaluation sets. However, `ADV-0` differs in its object of optimization and regularization. Instead of designing generic task parameters or using regret as the adversarial signal, `ADV-0` optimizes a distribution over naturalistic traffic trajectories conditioned on real driving logs, regularizes this distribution toward a frozen traffic prior $\mathcal{G}_{\text{ref}}$, and uses the defender's closed-loop return as the adversarial utility. This distinction is important in autonomous driving, where generated scenarios must remain physically plausible while continuously tracking the failure modes of the current policy.

B Theoretical Analysis

B.1 Convergence Analysis

In this section, we provide a theoretical analysis of the convergence behavior of the idealized `ADV-0` game. We formulate the interaction between the ego agent π_θ and the adversary $\mathcal{G}_\psi$ as a regularized two-player Zero-Sum Markov Game (ZSMG). Our analysis proceeds in two steps: (1) We first prove that the inner loop optimization via IPL is mathematically equivalent to solving for the soft optimal

adversarial distribution subject to a KL-divergence constraint (Lemma B.1). (2) We then elaborate that the alternating soft updates of the defender and the attacker constitute a contraction mapping in the idealized setting, yielding convergence to the Nash Equilibrium of the game (Theorem B.3).

B.1.1 Preliminaries: ADV-0 as a regularized zero-sum Markov game

Formally, we define the discounted ZSMG between the ego and the adversary by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{Y}, \Pi, \Psi, \mathcal{P}, \mathcal{R}, \gamma)$, where:

- The *defender* (ego agent) chooses a policy $\pi_\theta \in \Pi : \mathcal{S} \to \Delta(\mathcal{A})$ to maximize its expected return.
- The *attacker* (adversary) chooses a generative policy $\mathcal{G}_\psi \in \Psi : \mathcal{X} \to \Delta(\mathcal{Y})$ to produce adversarial trajectories $Y^{\text{Adv}} \in \mathcal{Y}$, which in turn perturbs the transition dynamics $\mathcal{P}_\psi$. The attacker's goal is to minimize the defender's return.

Let the value function $V^{\pi,\psi}(s)$ represent the expected return of the ego agent under the dynamics induced by adversary. The robust optimization objective (Eq. 1) with the KL-constraint (Eq. 2, to maintain naturalistic priors) is formulated as finding the saddle point of the regularized value function:

$$\max_{\pi} \min_{\mathcal{G}_\psi} \mathcal{J}(\pi, \mathcal{G}_\psi) := \mathbb{E}_X \left[\mathbb{E}_{\tau \sim \pi, \mathcal{P}_\psi} \left[\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t) \right] + \tau \mathbb{D}_{\text{KL}}(\mathcal{G}_\psi(\cdot|X) || \mathcal{G}_{\text{ref}}(\cdot|X)) \right], \quad (9)$$

where $\mathcal{P}_\psi$ denotes the transition dynamics modulated by the adversary's trajectory $Y \sim \mathcal{G}_\psi(\cdot|X)$, $\tau > 0$ controls the regularization strength. Note that the adversary seeks to minimize the ego's return subject to staying close to the prior $\mathcal{G}_{\text{ref}}$.

The ultimate goal is to find converged policies (π^*, ψ^*) satisfy the saddle-point inequality condition of a Nash Equilibrium:

$$\mathcal{J}(\pi_\theta, \mathcal{G}_{\psi^*}) \leq \mathcal{J}(\pi^*, \mathcal{G}_{\psi^*}) \leq \mathcal{J}(\pi^*, \mathcal{G}_\psi), \quad \forall \pi_\theta \in \Pi, \mathcal{G}_\psi \in \Psi, \quad (10)$$

implying that neither the defender nor the attacker can unilaterally improve their objective.

B.1.2 Inner loop optimality via implicit reward

We first analyze the inner loop of Algorithm 1, where the adversary's policy $\mathcal{G}_\psi$ is updated while the ego policy π_θ is held fixed. The core of the inner loop is the IPL objective. We will show that minimizing the IPL loss (Eq. 8) is equivalent to finding the optimal adversarial policy that solves the KL-regularized reward maximization problem.

Consider the inner loop objective defined in Eqs. 2 and 7. For a fixed ego policy π_θ, the adversary seeks to find an optimal policy ψ^* that maximizes the expected risk (minimizes ego return) while remaining close to the reference prior $\mathcal{G}_{\text{ref}}$. Let the reward for the adversary be defined as $r(Y) = -J(\pi_\theta, Y)$. The objective is:

$$\max_{\psi} \mathcal{J}_{\text{inner}}(\mathcal{G}_\psi) = \mathbb{E}_{Y \sim \mathcal{G}_\psi(\cdot|X)} \left[r(Y) - \tau \log \frac{\mathcal{G}_\psi(Y|X)}{\mathcal{G}_{\text{ref}}(Y|X)} \right]. \quad (11)$$

Lemma B.1 (Closed-form optimality of the Gibbs adversary)**.** *For a fixed defender π_θ and a reference adversary $\mathcal{G}_{ref}$, the global optimum $\mathcal{G}^*$ of the KL-constrained objective in Eq. 11 is given by the Gibbs distribution:*

$$\mathcal{G}^*(Y|X) = \frac{1}{Z(X)} \mathcal{G}_{ref}(Y|X) \exp\left(-\frac{1}{\tau} J(\pi_\theta, Y) \right), \quad (12)$$

where $Z(X)$ is the partition function. Furthermore, minimizing the IPL loss $\mathcal{L}_{IPL}$ (Eq. 8) is equivalent to performing maximum likelihood estimation on this optimal policy $\mathcal{G}^*$.

Proof. The proof consists of two parts. First, we derive the closed-form optimal adversarial policy. Let $r(Y) = -J(\pi_\theta, Y)$ be the reward for the adversary. Following the derivation in Rafailov et al.

[2023], we express the objective using the Gibbs inequality. The objective can be rewritten as maximizing:

$$\begin{aligned}\mathcal{J}_{\text{inner}}(\mathcal{G}) &= \mathbb{E}_{Y\sim\mathcal{G}}\left[r(Y) - \tau\log\frac{\mathcal{G}(Y|X)}{\mathcal{G}_{\text{ref}}(Y|X)}\right] \\ &= \tau\mathbb{E}_{Y\sim\mathcal{G}}\left[\log\exp(\frac{1}{\tau}r(Y)) + \log\mathcal{G}_{\text{ref}}(Y|X) - \log\mathcal{G}(Y|X)\right] \\ &= \tau\mathbb{E}_{Y\sim\mathcal{G}}\left[\log\left(\mathcal{G}_{\text{ref}}(Y|X)\exp\left(\frac{r(Y)}{\tau}\right)\right) - \log\mathcal{G}(Y|X)\right]. \end{aligned} \tag{13}$$

Let $Z(X) = \int \mathcal{G}_{\text{ref}}(y|X)\exp(r(y)/\tau)dy$ be the partition function. We can introduce $\log Z(X)$ into the expectation:

$$\begin{aligned}\mathcal{J}_{\text{inner}}(\mathcal{G}) &= \tau\mathbb{E}_{Y\sim\mathcal{G}}\left[\log\left(\frac{1}{Z(X)}\mathcal{G}_{\text{ref}}(Y|X)\exp\left(\frac{r(Y)}{\tau}\right)\right) - \log\mathcal{G}(Y|X) + \log Z(X)\right] \\ &= -\tau\mathbb{D}_{\text{KL}}(\mathcal{G}(\cdot|X)||\mathcal{G}^*(\cdot|X)) + \tau\log Z(X), \end{aligned} \tag{14}$$

where $\mathcal{G}^*(Y|X) = \frac{1}{Z(X)}\mathcal{G}_{\text{ref}}(Y|X)\exp\left(\frac{r(Y)}{\tau}\right)$. Since $\mathbb{D}_{\text{KL}} \geq 0$, the objective is maximized when $\mathcal{G} = \mathcal{G}^*$, proving the first part of the lemma. This identifies the population-level optimal adversarial distribution. However, explicitly evaluating $Z(X)$ and sampling exactly from this distribution are intractable in high-dimensional trajectory space.

We next show that minimizing the IPL loss $\mathcal{L}_{\text{IPL}}$ with respect to ψ is consistent with maximizing the likelihood of the preference data generated by the optimal adversary $\mathcal{G}^*$, without the need to estimate the partition function. We show the equivalence following Rafailov et al. [2023]. We can invert the optimal policy equation to rewrite the reward as:

$$r(Y) = \tau\log\frac{\mathcal{G}^*(Y|X)}{\mathcal{G}_{\text{ref}}(Y|X)} + \tau\log Z(X). \tag{15}$$

Under the Bradley-Terry preference model, the probability that trajectory Y_w is preferred over Y_l (i.e., Y_w induces lower ego return) is given by $P(Y_w \succ Y_l) = \sigma(r(Y_w) - r(Y_l))$. Substituting the reparameterized reward into this model:

$$\begin{aligned}P(Y_w \succ Y_l) &= \sigma\left(\left(\tau\log\frac{\mathcal{G}^*(Y_w|X)}{\mathcal{G}_{\text{ref}}(Y_w|X)} + \tau\log Z\right) - \left(\tau\log\frac{\mathcal{G}^*(Y_l|X)}{\mathcal{G}_{\text{ref}}(Y_l|X)} + \tau\log Z\right)\right) \\ &= \sigma\left(\tau\log\frac{\mathcal{G}^*(Y_w|X)}{\mathcal{G}_{\text{ref}}(Y_w|X)} - \tau\log\frac{\mathcal{G}^*(Y_l|X)}{\mathcal{G}_{\text{ref}}(Y_l|X)}\right). \end{aligned} \tag{16}$$

The partition function $Z(X)$ cancels out. The IPL loss (Eq. 8) is exactly the negative log-likelihood of this probability with the parameterized policy $\mathcal{G}_\psi$ approximating $\mathcal{G}^*$. Thus, minimizing $\mathcal{L}_{\text{IPL}}$ is equivalent to fitting the optimal adversarial policy $\mathcal{G}^*$ consistent with the observed preferences. This proves that the inner loop of `ADV-0` effectively solves the constrained optimization problem in Eq. 2. □

B.1.3 Global convergence to Nash Equilibrium

Having established in Lemma B.1 the population-level optimal adversarial distribution $\mathcal{G}_{\psi^*}$, we now analyze the convergence of the global alternating optimization. We show that the idealized `ADV-0` game can be viewed as optimizing a specific robust Bellman Operator, which yields convergence to a unique Nash Equilibrium under exact updates.

Before beginning the formal derivation, we first establish the following Lemma:

Lemma B.2 (Non-expansiveness of Soft-Min)**.** *Let $f_\tau(X) \triangleq -\tau\log\mathbb{E}_Y[\exp(-X(Y)/\tau)]$ be the Soft-Min operator over a random variable Y with temperature $\tau > 0$. For any two bounded functions X_1, X_2, the following inequality holds:*

$$|f_\tau(X_1) - f_\tau(X_2)| \leq \max_Y |X_1(Y) - X_2(Y)|. \tag{17}$$

Proof. Let $\Delta = \max_Y |X_1(Y) - X_2(Y)|$. By definition, for all Y, the difference is bounded by:

$$X_2(Y) - \Delta \leq X_1(Y) \leq X_2(Y) + \Delta. \tag{18}$$

Multiplying by $-1/\tau$ and exponentiating yields:

$$\exp\left(\frac{-X_2(Y) - \Delta}{\tau}\right) \leq \exp\left(\frac{-X_1(Y)}{\tau}\right) \leq \exp\left(\frac{-X_2(Y) + \Delta}{\tau}\right). \tag{19}$$

Taking the expectation $\mathbb{E}_Y$ preserves the inequality. We can factor out the constant terms $\exp(\pm\Delta/\tau)$:

$$e^{-\Delta/\tau}\mathbb{E}_Y[e^{-X_2(Y)/\tau}] \leq \mathbb{E}_Y[e^{-X_1(Y)/\tau}] \leq e^{\Delta/\tau}\mathbb{E}_Y[e^{-X_2(Y)/\tau}]. \tag{20}$$

Next, we apply the strictly decreasing function $-\tau\log(\cdot)$ to all sides. We have:

$$-\tau\log\left(e^{\Delta/\tau}\mathbb{E}[e^{-X_2/\tau}]\right) \leq -\tau\log\mathbb{E}[e^{-X_1/\tau}] \leq -\tau\log\left(e^{-\Delta/\tau}\mathbb{E}[e^{-X_2/\tau}]\right). \tag{21}$$

Using the definition of f_τ and expanding the terms:

$$f_\tau(X_2) - \Delta \leq f_\tau(X_1) \leq f_\tau(X_2) + \Delta. \tag{22}$$

Subtracting $f_\tau(X_2)$ from all sides, we obtain:

$$-\Delta \leq f_\tau(X_1) - f_\tau(X_2) \leq \Delta. \tag{23}$$

This is equivalent to $|f_\tau(X_1) - f_\tau(X_2)| \leq \Delta$. □

Theorem B.3 (Contraction and convergence to Nash Equilibrium)**.** *Let $\mathcal{V}$ be the space of bounded value functions equipped with the L_∞-norm. The Soft-Robust Bellman Operator $\mathcal{T}_{rob} : \mathcal{V} \to \mathcal{V}$, defined as:*

$$(\mathcal{T}_{rob}V)(s) \triangleq \max_{a\in\mathcal{A}} \min_{\mathcal{G}_\psi\in\Psi} \left(\mathcal{R}(s,a) + \gamma\mathbb{E}_{\substack{Y\sim\mathcal{G}_\psi \\ s'\sim\mathcal{P}(\cdot|s,a,Y)}}[V(s')] + \tau\mathbb{D}_{KL}(\mathcal{G}_\psi||\mathcal{G}_{ref})\right), \tag{24}$$

is a γ-contraction mapping. Specifically, for any two value functions $V, U \in \mathcal{V}$, the following inequality holds:

$$\|\mathcal{T}_{rob}V - \mathcal{T}_{rob}U\|_\infty \leq \gamma\|V - U\|_\infty. \tag{25}$$

Consequently, the idealized iterative updates converge to a unique fixed point V^*. This fixed point corresponds to the value of the unique Nash Equilibrium $(\pi^*, \mathcal{G}_{\psi^*})$ of the regularized zero-sum game, satisfying the saddle-point inequality:

$$\mathcal{J}_\tau(\pi, \mathcal{G}_{\psi^*}) \leq \mathcal{J}_\tau(\pi^*, \mathcal{G}_{\psi^*}) \leq \mathcal{J}_\tau(\pi^*, \mathcal{G}_\psi), \quad \forall\pi \in \Pi, \forall\mathcal{G}_\psi \in \Psi, \tag{26}$$

where $\mathcal{J}_\tau(\pi, \mathcal{G}_\psi) \triangleq \mathbb{E}_{\pi,\mathcal{G}_\psi}[\sum\gamma^t\mathcal{R}(s_t, a_t)] + \tau\mathbb{E}_\pi[\sum\gamma^t\mathbb{D}_{KL}(\mathcal{G}_\psi||\mathcal{G}_{ref})]$ is the regularized cumulative objective.

Proof. First, we define the soft-robust Bellman operator $\mathcal{T}_{\text{rob}}$ acting on the value function $V \in \mathbb{R}^{|\mathcal{S}|}$:

$$(\mathcal{T}_{\text{rob}}V)(s) = \max_{a\in\mathcal{A}} \min_{\mathcal{G}_\psi} \mathbb{E}_{Y\sim\mathcal{G}_\psi}\left[\mathcal{R}(s,a) + \gamma\mathbb{E}_{s'\sim\mathcal{P}(s,a,Y)}[V(s')] + \tau\log\frac{\mathcal{G}_\psi(Y|X)}{\mathcal{G}_{\text{ref}}(Y|X)}\right]. \tag{27}$$

This operator represents one step of optimal decision-making by the ego agent against a worst-case adversary that is regularized by the KL-divergence. Note that the inner minimization in $\mathcal{T}_{\text{rob}}$ corresponds exactly to the dual of the maximization problem in Lemma B.1 (due to the zero-sum sign flip).

Next, let $V_1, V_2 \in \mathbb{R}^{|\mathcal{S}|}$ be two arbitrary bounded value functions. We aim to show $\|\mathcal{T}_{\text{rob}}V_1 - \mathcal{T}_{\text{rob}}V_2\|_\infty \leq \gamma\|V_1 - V_2\|_\infty$. We simplify the inner minimization problem. Let $\mathcal{Q}_V(s,a,Y) = \mathcal{R}(s,a) + \gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a,Y)}[V(s')]$. Using the closed-form solution derived in Lemma B.1, the inner minimization over $\mathcal{G}_\psi$ is equivalent to a `LogSumExp` (Soft-Min) function. We define the smoothed value $\Omega_V(s,a)$ as:

$$\begin{aligned}\Omega_V(s,a) &\triangleq \min_{\mathcal{G}_\psi} \mathbb{E}_{Y\sim\mathcal{G}_\psi}\left[\mathcal{Q}_V(s,a,Y) + \tau\log\frac{\mathcal{G}_\psi(Y|X)}{\mathcal{G}_{\text{ref}}(Y|X)}\right] \\ &= -\tau\log\mathbb{E}_{Y\sim\mathcal{G}_{\text{ref}}}\left[\exp\left(-\frac{\mathcal{Q}_V(s,a,Y)}{\tau}\right)\right].\end{aligned} \tag{28}$$

Thus, the operator simplifies to $(\mathcal{T}_{\text{rob}}V)(s) = \max_a \Omega_V(s,a)$. Consider the difference for any state s:

$$\begin{aligned}|(\mathcal{T}_{\text{rob}}V_1)(s) - (\mathcal{T}_{\text{rob}}V_2)(s)| &= |\max_a \Omega_{V_1}(s,a) - \max_a \Omega_{V_2}(s,a)| \\ &\leq \max_a |\Omega_{V_1}(s,a) - \Omega_{V_2}(s,a)|. \end{aligned} \tag{29}$$

where Eq. 29 follows from the non-expansiveness of the max operator (i.e., $|\max f - \max g| \leq \max|f-g|$). Next, we apply Lemma B.2 to the soft-min term Ω_V. By identifying $X(Y)$ with $Q_V(s,a,Y)$, we obtain:

$$\begin{aligned}|\Omega_{V_1}(s,a) - \Omega_{V_2}(s,a)| &= \left|-\tau \log \frac{\mathbb{E}_Y[\exp(-\mathcal{Q}_{V_1}(s,a,Y)/\tau)]}{\mathbb{E}_Y[\exp(-\mathcal{Q}_{V_2}(s,a,Y)/\tau)]}\right| \\ &\leq \max_Y \left|-\tau\left(-\frac{\mathcal{Q}_{V_1}(s,a,Y)}{\tau} + \frac{\mathcal{Q}_{V_2}(s,a,Y)}{\tau}\right)\right| \\ &= \max_Y |\mathcal{Q}_{V_1}(s,a,Y) - \mathcal{Q}_{V_2}(s,a,Y)|. \end{aligned} \tag{30}$$

Substituting the definition of Q_V and expanding the expectation:

$$\begin{aligned}&= \max_Y \left|\gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a,Y)}[V_1(s') - V_2(s')]\right| \\ &\leq \gamma \max_Y \mathbb{E}_{s'}[|V_1(s') - V_2(s')|] \quad \text{(Jensen's inequality)} \\ &\leq \gamma \|V_1 - V_2\|_\infty. \quad \text{(Definition of } L_\infty\text{-norm)} \end{aligned} \tag{31}$$

The last two steps utilize the convexity of the absolute value function and bound the local error by the global supremum norm. Since $\gamma \in (0,1)$, $\mathcal{T}_{\text{rob}}$ is a γ-contraction mapping. By Banach's Fixed Point Theorem, there exists a unique value V^* such that $\mathcal{T}_{\text{rob}}V^* = V^*$.

Finally, we connect this to the alternating updates in Algorithm 1 under the idealized setting of exact inner optimization and exact policy improvement. The population-level procedure performs generalized policy iteration. The inner loop solves for the optimal soft adversary under the KL-regularized objective, effectively evaluating $\Omega_{V^\pi}(s,a)$. The outer loop performs policy improvement on the induced robust value function. Tessler et al. [2019] (Theorem 3) and Scherrer [2014] demonstrate that soft policy iteration converges to the optimal value if the policy improvement step is a contraction. Since $\mathcal{T}_{\text{rob}}$ is a contraction, the idealized sequence of policies (π_k, ψ_k) converges to the Nash Equilibrium (π^*, ψ^*) where π^* is optimal against the worst-case regularized adversary ψ^*. □

B.2 Generalization Bound and Safety Guarantees

In this section, we provide a theoretical justification of `ADV-0` in its generalizability and safety guarantees. We aim to answer a fundamental question: *Does optimizing the policy against a generated adversarial distribution guarantee performance and safety in the real-world long-tail distribution?* Different from the previous section, we now model the interaction between the ego policy π_θ and the adversarial environment as a problem of policy optimization under dynamical uncertainty. Our goal is to show that optimizing the policy under the adversarial dynamics $\mathcal{P}_\psi$ (subject to the KL-constraint in the inner loop) improves a lower bound on the performance in the target real-world long-tail distribution $\mathcal{P}_{\text{real}}$, leading to a **generalization bound** (Theorem B.6) and a **safety guarantee** (Theorem B.8). Our analysis builds upon the *Simulation Lemma* from Luo et al. [2018] and trust-region bounds from Achiam et al. [2017].

Preliminaries. Simply let $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma)$ be the shared components of the MDP. We consider two transition dynamics:

- $\mathcal{P}_{\text{real}}(s'|s,a)$: The true, unknown long-tail dynamics of the real world.
- $\mathcal{P}_\psi(s'|s,a)$: The adversarial dynamics induced by the generator $\mathcal{G}_\psi(\cdot|X)$.

In our context, the state transition is deterministic given the background traffic trajectories. Let Y denote the joint trajectory of background agents. The transition function can be written as $s' = f(s,a,Y)$. Thus, the stochasticity in dynamics comes entirely from the distribution of Y. Let $V^{\pi,\mathcal{P}}(s) = \mathbb{E}_{\tau\sim\pi,\mathcal{P}}[\sum_{t=0}^\infty \gamma^t \mathcal{R}(s_t,a_t)|s_0 = s]$ be the value function. The expected return is

$J(\pi, \mathcal{P}) = \mathbb{E}_{s_0 \sim \rho_0}[V^{\pi,\mathcal{P}}(s_0)]$. Similarly, let $J_C(\pi, \mathcal{P})$ denote the expected cumulative safety cost, where $C(s, a) \in [0, C_{\max}]$ is a safety cost function (e.g., collision indicator). We assume the reward is bounded by $R_{\max}$, implying the value function is bounded by $V_{\max} = \frac{R_{\max}}{1-\gamma}$.

B.2.1 Discrepancy under shifted dynamics

To analyze performance generalization, we first quantify the discrepancy in expected return resulting from the shift from real dynamics to adversarial dynamics. We invoke the Simulation Lemma [Achiam et al., 2017] and adapt here to quantify the gap between the adversarial training environment and the real world.

Lemma B.4 (Value difference under dynamics shift)**.** *For any fixed policy π and two transition dynamics $\mathcal{P}$ and $\mathcal{P}'$, the difference in expected return is:*

$$J(\pi, \mathcal{P}) - J(\pi, \mathcal{P}') = \gamma \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{s_t \sim \pi, \mathcal{P}, a_t \sim \pi} \Big[\mathbb{E}_{s' \sim \mathcal{P}(\cdot|s_t, a_t)}[V^{\pi,\mathcal{P}'}(s')] - \mathbb{E}_{s' \sim \mathcal{P}'(\cdot|s_t, a_t)}[V^{\pi,\mathcal{P}'}(s')] \Big]. \tag{32}$$

Proof. We use the telescoping sum technique, following Lemma 4.3 in Luo et al. [2018], Let $V^{\mathcal{P}'}$ denote $V^{\pi,\mathcal{P}'}$ for brevity. Remind that $-V^{\mathcal{P}'}(s_0) = \sum_{t=0}^{\infty} \gamma^t (\gamma V^{\mathcal{P}'}(s_{t+1}) - V^{\mathcal{P}'}(s_t)) - \lim_{T \to \infty} \gamma^T V^{\mathcal{P}'}(s_T)$, we expand the difference as follows:

$$\begin{aligned}
J(\pi, \mathcal{P}) - J(\pi, \mathcal{P}') &= \mathbb{E}_{s_0}[V^{\pi,\mathcal{P}}(s_0) - V^{\pi,\mathcal{P}'}(s_0)] \\
&= \mathbb{E}_{\tau \sim \pi, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \right] - \mathbb{E}_{s_0}[V^{\mathcal{P}'}(s_0)] \\
&= \mathbb{E}_{\tau \sim \pi, \mathcal{P}} \Bigg[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) + \sum_{t=0}^{\infty} \gamma^t (\gamma V^{\mathcal{P}'}(s_{t+1}) - V^{\mathcal{P}'}(s_t)) \\
&\qquad - \lim_{T \to \infty} \gamma^T V^{\mathcal{P}'}(s_T) \Bigg] \\
&= \mathbb{E}_{\tau \sim \pi, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t \Big(\mathcal{R}(s_t, a_t) + \gamma V^{\mathcal{P}'}(s_{t+1}) - V^{\mathcal{P}'}(s_t) \Big) \right].
\end{aligned} \tag{33}$$

Recall the Bellman equation for $V^{\mathcal{P}'}$ is $V^{\mathcal{P}'}(s) = \mathbb{E}_{a \sim \pi}[\mathcal{R}(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}'}[V^{\mathcal{P}'}(s')]]$. Substituting $\mathcal{R}(s_t, a_t) = V^{\mathcal{P}'}(s_t) - \gamma \mathbb{E}_{s' \sim \mathcal{P}'(\cdot|s_t, a_t)}[V^{\mathcal{P}'}(s')]$ into the summation, the term $V^{\mathcal{P}'}(s_t)$ cancels out:

$$\begin{aligned}
J(\pi, \mathcal{P}) - J(\pi, \mathcal{P}') &= \mathbb{E}_{\tau \sim \pi, \mathcal{P}} \Bigg[\sum_{t=0}^{\infty} \gamma^t \Big(V^{\mathcal{P}'}(s_t) - \gamma \mathbb{E}_{s' \sim \mathcal{P}'}[V^{\mathcal{P}'}(s')] \\
&\qquad + \gamma V^{\mathcal{P}'}(s_{t+1}) - V^{\mathcal{P}'}(s_t) \Big) \Bigg] \\
&= \mathbb{E}_{\tau \sim \pi, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t \Big(\gamma V^{\mathcal{P}'}(s_{t+1}) - \gamma \mathbb{E}_{s' \sim \mathcal{P}'(\cdot|s_t, a_t)}[V^{\mathcal{P}'}(s')] \Big) \right] \\
&= \sum_{t=0}^{\infty} \gamma^{t+1} \mathbb{E}_{\substack{s_t \sim \pi, \mathcal{P} \\ a_t \sim \pi}} \Big[\mathbb{E}_{s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}[V^{\mathcal{P}'}(s_{t+1})] \\
&\qquad - \mathbb{E}_{s' \sim \mathcal{P}'(\cdot|s_t, a_t)}[V^{\mathcal{P}'}(s')] \Big].
\end{aligned} \tag{34}$$

Adjusting the index of summation yields the lemma statement. □

Lemma B.4 suggests that the performance differences depend on the differences of transition dynamics, but in `ADV-O` we optimize $\mathcal{G}_\psi$, not $\mathcal{P}_\psi$ directly. To fill this gap, we establish the connection between the dynamics divergence and the generator divergence in the following lemma.

Lemma B.5 (Divergence bound of the generator)**.** *Let $\mathcal{P}$ and $\mathcal{P}'$ be the transition dynamics induced by the trajectory generators $\mathcal{G}$ and $\mathcal{G}'$, respectively. Specifically, the next state is obtained by a deterministic simulator $s' = F(s, a, Y)$, where Y is the adversarial trajectory sampled from the generator. For any value function $V^{\pi,\mathcal{P}'}$ bounded by $V_{\max} = \frac{R_{\max}}{1-\gamma}$, the difference in expected next-state value is bounded by the Total Variation (TV) divergence of the generators:*

$$\left|\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^{\pi,\mathcal{P}'}(s')] - \mathbb{E}_{s'\sim\mathcal{P}'(\cdot|s,a)}[V^{\pi,\mathcal{P}'}(s')]\right| \leq 2V_{\max} \cdot \mathbb{E}_X[D_{TV}(\mathcal{G}(\cdot|X)\|\mathcal{G}'(\cdot|X))]. \tag{35}$$

Proof. We start by explicitly writing the expectation over the next state s' as an expectation over the generated trajectories Y, i.e., $\mathbb{E}_{s'\sim\mathcal{P}}[V(s')] = \int \mathcal{G}(Y|X) \cdot V(F_{\text{phy}}(s, a, Y))\, dY$. Let $p(Y|X)$ and $q(Y|X)$ denote the probability density functions of the generators $\mathcal{G}$ and $\mathcal{G}'$ conditioned on context X. Since $s' = F(s, a, Y)$, the expectation can be rewritten via the change of variables:

$$\begin{aligned}
\Delta_V &= \left|\mathbb{E}_{s'\sim\mathcal{P}}[V^{\pi,\mathcal{P}'}(s')] - \mathbb{E}_{s'\sim\mathcal{P}'}[V^{\pi,\mathcal{P}'}(s')]\right| \\
&= \left|\int p(Y|X)V^{\pi,\mathcal{P}'}(F(s,a,Y))\, dY - \int q(Y|X)V^{\pi,\mathcal{P}'}(F(s,a,Y))\, dY\right|, \qquad (36)\\
&= \left|\int (p(Y|X) - q(Y|X)) \cdot V^{\pi,\mathcal{P}'}(F(s,a,Y))\, dY\right|. \qquad (37)
\end{aligned}$$

Next, we apply the integral form of Hölder's inequality ($|\int f(x)g(x)dx| \leq \int |f(x)||g(x)|dx \leq \|f\|_1\|g\|_\infty$). Here, we treat the probability difference as the measure and the value function as the bounded term:

$$\begin{aligned}
\Delta_V &\leq \int |p(Y|X) - q(Y|X)| \cdot \left|V^{\pi,\mathcal{P}'}(F(s,a,Y))\right| \, dY \\
&\leq \underbrace{\left(\int |p(Y|X) - q(Y|X)|\, dY\right)}_{\|p-q\|_1} \cdot \underbrace{\sup_Y \left|V^{\pi,\mathcal{P}'}(F(s,a,Y))\right|}_{\|V\|_\infty}. \qquad (38)
\end{aligned}$$

We then use two key properties: 1. The L_1 norm of the difference between two probability distributions is twice the Total Variation distance: $\|p - q\|_1 = 2D_{\text{TV}}(p, q)$. 2. The value function is bounded by the maximum possible cumulative return: $\|V\|_\infty \leq V_{\max} = R_{\max}/1 - \gamma$. Substituting these back into the inequality, we obtain:

$$\Delta_V \leq 2D_{\text{TV}}(\mathcal{G}(\cdot|X)\|\mathcal{G}'(\cdot|X)) \cdot V_{\max}. \tag{39}$$

Taking the expectation over the context distribution X completes the proof. This lemma serves as a crucial bridge in our analysis: it formally translates the divergence in the high-level trajectory generator space (which we explicitly optimize and constrain via IPL) into the divergence in the low-level state transition space, thereby allowing us to bound the error. □

B.2.2 Adversarial generalization bound

Finally, we present the main theorem. We show that the policy π_θ trained by `ADV-O` maximizes a lower bound on the performance in real-world distribution. IPL loop enforces a KL-constraint between the adversary $\mathcal{G}_\psi$ and the prior $\mathcal{G}_{\text{ref}}$.

Theorem B.6 (Lower bound of adversarial generalization)**.** *Let π_θ be the policy trained under adversarial dynamics $\mathcal{P}_\psi$. The expected return of π_θ under the real-world dynamics $\mathcal{P}_{real}$ is lower-bounded by:*

$$J(\pi_\theta, \mathcal{P}_{real}) \geq J(\pi_\theta, \mathcal{P}_\psi) - \frac{\gamma V_{\max}\sqrt{2}}{1-\gamma}\sqrt{\mathbb{E}_X[D_{KL}(\mathcal{G}_\psi(\cdot|X)\|\mathcal{G}_{ref}(\cdot|X))]}. \tag{40}$$

Proof. The proof utilizes the established lemmas above. We first invoke Lemma B.4 with $\mathcal{P} = \mathcal{P}_{\text{real}}$ and $\mathcal{P}' = \mathcal{P}_\psi$:

$$|J(\pi, \mathcal{P}_{\text{real}}) - J(\pi, \mathcal{P}_\psi)| \leq \gamma \sum_{t=0}^{\infty} \gamma^t \left|\mathbb{E}_{s_t,a_t}\left[\mathbb{E}_{s'\sim\mathcal{P}_{\text{real}}}[V^{\pi,\mathcal{P}_\psi}(s')] - \mathbb{E}_{s'\sim\mathcal{P}_\psi}[V^{\pi,\mathcal{P}_\psi}(s')]\right]\right|. \tag{41}$$

Given that the reference generator $\mathcal{G}_{\text{ref}}$ is trained on large-scale naturalistic driving logs (WOMD), we assume that the *dynamics induced by $\mathcal{G}_{ref}$ serve as a high-fidelity approximation of the real-world dynamics $\mathcal{P}_{real}$*. Consequently, by applying Lemma B.5 with $\mathcal{G}' = \mathcal{G}_{\text{ref}}$, we bound the inner term:

$$\left|\mathbb{E}_{\mathcal{P}_{\text{real}}}[V^{\pi,\mathcal{P}_\psi}] - \mathbb{E}_{\mathcal{P}_\psi}[V^{\pi,\mathcal{P}_\psi}]\right| \leq 2V_{\max}\mathbb{E}_X[D_{\text{TV}}(\mathcal{G}_{\text{ref}}(\cdot|X)\|\mathcal{G}_\psi(\cdot|X))]. \tag{42}$$

Substituting this bound back into the summation and using the geometric series sum $\sum_{t=0}^{\infty}\gamma^t = \frac{1}{1-\gamma}$:

$$J(\pi,\mathcal{P}_{\text{real}}) \geq J(\pi,\mathcal{P}_\psi) - \frac{2\gamma V_{\max}}{1-\gamma}\mathbb{E}_X[D_{\text{TV}}(\mathcal{G}_{\text{ref}}\|\mathcal{G}_\psi)]. \tag{43}$$

Finally, we utilize the symmetry of the Total Variation distance, i.e., $D_{\text{TV}}(P\|Q) = D_{\text{TV}}(Q\|P)$, to equate $D_{\text{TV}}(\mathcal{G}_{\text{ref}}\|\mathcal{G}_\psi) = D_{\text{TV}}(\mathcal{G}_\psi\|\mathcal{G}_{\text{ref}})$. We then apply Pinsker's Inequality $D_{\text{TV}}(P\|Q) \leq \sqrt{\frac{1}{2}D_{\text{KL}}(P\|Q)}$ alongside Jensen's inequality $\mathbb{E}[\sqrt{Z}] \leq \sqrt{\mathbb{E}[Z]}$:

$$\mathbb{E}_X[D_{\text{TV}}(\mathcal{G}_{\text{ref}}\|\mathcal{G}_\psi)] = \mathbb{E}_X[D_{\text{TV}}(\mathcal{G}_\psi\|\mathcal{G}_{\text{ref}})] \leq \sqrt{\frac{1}{2}\mathbb{E}_X[D_{\text{KL}}(\mathcal{G}_\psi\|\mathcal{G}_{\text{ref}})]}. \tag{44}$$

Combining these yields the theorem. □

Remark B.7. Theorem B.6 theoretically motivates the practical objective of `ADV-O`. The outer loop maximizes the first term $J(\pi_\theta,\mathcal{P}_\psi)$ (robustness), while the IPL inner loop uses a KL-regularized adversary objective to control the divergence term appearing in the bound. Thus, the practical solver should be interpreted as a tractable surrogate for improving the lower bound, rather than as direct optimization of the closed-form bound itself.
Crucially, $\mathcal{P}_{\text{real}}$ here represents any target distribution within the δ-trust region of the prior, specifically including the real-world long-tail scenarios. Since the adversary $\mathcal{P}_\psi$ is optimized to be challenging within this region, the theorem bounds the gap between performance on synthetic adversarial cases and performance on unseen real-world critical events.

B.2.3 Safety guarantee in the long tail

Finally, we derive the formal safety guarantee. We denote $J_C(\pi,\mathcal{P})$ the expected cumulative cost (e.g., collision risk).

Theorem B.8 (Worst-case safety certificate)**.** *Let $C_{\max}$ be the maximum instantaneous cost (upper bound of the per-step cost). If the policy π_θ satisfies the safety constraint $J_C(\pi_\theta,\mathcal{P}_\psi) \leq \delta$ under the adversarial dynamics, then the safety violation in the real environment is bounded by:*

$$J_C(\pi_\theta,\mathcal{P}_{real}) \leq \delta + \frac{\gamma C_{\max}\sqrt{2}}{(1-\gamma)^2}\sqrt{\mathbb{E}_X[D_{KL}(\mathcal{G}_\psi(\cdot|X)\|\mathcal{G}_{ref}(\cdot|X))]}. \tag{45}$$

Proof. The proof follows similar logic as Lemma B.4 and Theorem B.6. We explicitly extend Corollary 2 of Achiam et al. [2017] (which bounds cost performance under policy shifts) to the setting of adversarial dynamics shifts. Formally, we apply Lemma B.4 to the cost value function $V_C^{\pi,\mathcal{P}}$.

$$\begin{aligned} J_C(\pi,\mathcal{P}_{\text{real}}) &\leq J_C(\pi,\mathcal{P}_\psi) + |J_C(\pi,\mathcal{P}_{\text{real}}) - J_C(\pi,\mathcal{P}_\psi)| \\ &\leq \delta + \frac{\gamma}{1-\gamma}\sum_{t=0}^{\infty}(1-\gamma)\gamma^t\left|\mathbb{E}_{s,a}[\mathbb{E}_{\mathcal{P}_{\text{real}}}[V_C] - \mathbb{E}_{\mathcal{P}_\psi}[V_C]]\right|. \end{aligned} \tag{46}$$

Using Lemma B.5 with $\|V_C\|_\infty \leq \frac{C_{\max}}{1-\gamma}$:

$$J_C(\pi,\mathcal{P}_{\text{real}}) \leq \delta + \frac{\gamma}{(1-\gamma)}\cdot\frac{2C_{\max}}{1-\gamma}\mathbb{E}_X[D_{\text{TV}}(\mathcal{G}_{\text{ref}}\|\mathcal{G}_\psi)]. \tag{47}$$

Applying Pinsker's inequality again completes the proof. □

Remark B.9. Theorem B.8 provides a formal safety certificate. It implies that if `ADV-O` successfully trains the agent to be safe ($\leq \delta$) against a risk-seeking adversary $\mathcal{P}_\psi$ that remains physically plausible, then the safety cost in the naturalistic environment is bounded up to a margin controlled by the KL divergence in IPL.

B.3 Discussion on Theoretical Assumptions and Practical Implementation

The theoretical results established in Sections B.1 and B.2 provide the formal motivation for `ADV-0`, which characterizes the ideal behavior of the ZSMG. In practice, our implementation involves necessary approximations to ensure computational tractability. Here, we discuss the validity of these principled approximations and how they connect to the theoretical results. In general, `ADV-0` solves the theoretical ZSMG via efficient approximations. The theoretical analysis identifies *what* to optimize (min-max objective with KL regularization) and *why* (improving a lower bound on real-world performance), while the practical implementation provides a tractable *how* (IPL with finite sampling).

Ideal regularized game and practical solver. The convergence result in Theorem B.3 concerns the idealized KL-regularized game, where the robust Bellman operator includes the regularization term inside the value backup. The implementation in Algorithm 1 does not add this KL term to the ego reward used by the outer RL algorithm. Instead, the outer loop optimizes the original driving return under the adversarial dynamics induced by $\mathcal{G}_\psi$, while the inner loop keeps the adversary close to the naturalistic prior through the KL-regularized objective. This separation is intentional: the ideal game characterizes the regularized adversarial target, whereas the implemented solver is aligned with the lower-bound form in Theorem B.6, whose first term is the pure environmental return and whose second term penalizes generator divergence.

Minimax duality in distribution space. For a fixed value function V and context X, the inner adversary can be analyzed over trajectory distributions rather than over raw trajectory samples. Let

$$F_V(\pi, \mu) = \mathbb{E}_{a\sim\pi, Y\sim\mu}[\mathcal{Q}_V(s, a, Y)] + \tau D_{\mathrm{KL}}(\mu \| \mathcal{G}_{\mathrm{ref}}), \tag{48}$$

where $\mathcal{Q}_V(s, a, Y) = \mathcal{R}(s, a) + \gamma \mathbb{E}[V(s') \mid s, a, Y]$. Under the boundedness and compactness assumptions used in Appendix B, F_V is affine in the defender distribution π and convex in the adversarial trajectory distribution μ. Sion's minimax theorem therefore yields the saddle-point interchange needed by the robust operator. Moreover, the KL term gives a unique inner best response,

$$\mu^*_{V,a}(Y|X) \propto \mathcal{G}_{\mathrm{ref}}(Y|X) \exp(-\mathcal{Q}_V(s, a, Y)/\tau), \tag{49}$$

which reduces the inner player to the Gibbs form used in Lemma B.1. After this reduction, the remaining robust Bellman operator is a γ-contraction, giving the fixed-point argument used in Theorem B.3.

Linear KL surrogate and square-root bound. The lower bound in Theorem B.6 contains a square-root KL term because the proof first bounds dynamics mismatch in total variation and then applies Pinsker's inequality together with Jensen's inequality. This bound is not used as a directly optimized inner-loop objective. The practical adversary instead solves the Lagrangian form of a KL-constrained problem,

$$\min_q J_q \quad \text{s.t.} \quad D_{\mathrm{KL}}(q \| \mathcal{G}_{\mathrm{ref}}) \le \delta, \qquad \mathcal{L}(q) = J_q + \tau D_{\mathrm{KL}}(q \| \mathcal{G}_{\mathrm{ref}}). \tag{50}$$

The linear KL form preserves the Gibbs closed form and makes IPL tractable without estimating the partition function. If the inner loop controls the divergence within $\delta(\tau)$, then the square-root penalty in Theorem B.6 is correspondingly bounded by the controlled trust-region radius.

Finite sample approximation of the Gibbs adversary. Lemma B.1 derives the optimal adversarial distribution as a Gibbs distribution $\mathcal{G}^*(Y|X) \propto \mathcal{G}_{\text{ref}}(Y|X) \exp(-J(\pi_\theta, Y)/\tau)$. In the theoretical analysis, the expectation is taken over the entire continuous trajectory space $\mathcal{Y}$. In our implementation (Eq. 6), this distribution is approximated through proposal-based finite resampling rather than exact sampling from the frozen prior. The frozen $\mathcal{G}_{\text{ref}}$ defines the naturalistic trust region, while the current generator $\mathcal{G}_\psi$ serves as an adaptive proposal initialized from $\mathcal{G}_{\text{ref}}$ and updated online by IPL. At each iteration, we sample a finite set of K candidates from $\mathcal{G}_\psi$ and apply temperature-scaled resampling over this support. Thus the practical step should be interpreted as finite-support adversarial resampling followed by amortized posterior fitting of $\mathcal{G}_\psi$, rather than exact importance sampling from $\mathcal{G}_{\text{ref}}$. While finite sampling introduces variance and support limitation, the online proposal update allows the generator to track the defender's moving failure frontier while the KL term preserves plausibility relative to the naturalistic prior.

Proxy reward and bias. The inner loop optimization relies on the proxy reward estimator $\hat{J}$ rather than the exact rollout return J. This could introduce bias in the gradient direction. However, we note that the effectiveness of IPL depends primarily on the *ranking* accuracy rather than the precision of *absolute value*. Under the Bradley-Terry model, the probability of preference depends on the value difference: $P(Y_w \succ Y_l) = \sigma(\hat{J}(Y_l) - \hat{J}(Y_w))$. The convergence requires that the proxy estimator preserves the relative ordinality of the true objective, i.e., $J(Y_a) < J(Y_b) \implies \mathbb{E}[\hat{J}(Y_a)] < \mathbb{E}[\hat{J}(Y_b)]$. This means that as long as the proxy estimator preserves the relative ordering of safety-critical events (e.g., correctly identifying that a collision is worse than a near-miss), the gradient direction for ψ remains consistent with the theoretical objective.

Trust region assumption in generalization. Theorem B.6 and B.8 rely on the assumption that the real-world dynamics $\mathcal{P}_{\text{real}}$ lie within a trust region support of the traffic prior $\mathcal{G}_{\text{ref}}$. While this is a strong assumption, it is a necessary condition for data-driven simulation approaches. This formalizes the requirement that the long tail consists of rare but plausible events, rather than out-of-domain anomalies. In the context of `ADV-0`, this assumption is a constraint enforced by the traffic prior. The term $\mathbb{E}[D_{\text{KL}}(\mathcal{G}_\psi || \mathcal{G}_{\text{ref}})]$ in the generalization bound is controlled by the regularization term in the IPL loss (Eq. 8). By constraining this KL-divergence during training, `ADV-0` improves the lower bound on performance within the δ-neighborhood of the naturalistic prior. The guarantee therefore applies when the target long-tail events remain within the physical plausibility modeled by the pretrained generator.

C Supplementary Tables and Figures

Additional diagnostic results. We report additional compact diagnostics that clarify the behavior of the online adversary, the proxy-based preference signal, and the computational cost of the training loop. These results complement the main ablations by isolating the mechanisms that are most relevant to long-tail coverage, estimator robustness, and min-max stability. Table 6 examines long-tail coverage, Table 7 reports sensitivity to preference construction, and Table 8 summarizes compute and stability diagnostics.

Table 6: **Long-tail coverage of the adversarial proposal.** The IPL-updated generator places more probability mass on rare but plausible failure modes than fixed-prior energy resampling, while retaining the same naturalistic prior constraint.

Adversary	Critical TTC Share	Rare-Cluster Share	Ego Collision
Pretrained + resampling	54%	14%	36.3%
`ADV-O` w/ IPL	**69%**	**25%**	**40.7%**

Table 7: **Sensitivity analyses for preference construction.** The proxy estimator mainly needs to preserve the ordering of adversarial candidates; performance degrades gracefully under mild noise and with nearby IPL filtering thresholds.

Proxy used for IPL	ρ	Return	Crash
Collision-only	0.58	42.8	0.270
Progress + collision	0.69	45.0	0.254
Current + 10% noise	0.73	45.6	0.243
Rollout-return oracle	1.00	**46.3**	**0.237**
Current proxy	0.77	46.1	0.240

IPL setting	Valid Pairs	Return	Crash
Margin = 0	7.6	44.9	0.255
Margin = 5	4.0	**46.1**	**0.240**
Margin = 8	2.5	45.6	0.244
Spatial thr. = 1 m	5.0	45.7	0.246
Spatial thr. = 2 m	4.3	**46.1**	**0.240**

Table 8: **Computational cost and long-run stability.** Runtime and memory are measured on the two NVIDIA RTX A6000 platform used for the experiments. The 2M-step diagnostic indicates that the online adversary does not introduce training collapse in these runs.

Method	Wall Time	Steps/h ($\times 10^3$)	VRAM	FLOPs
CAT	4.0 h	250.0	18.9 GB	5.1×10^{15}
`ADV-O` w/o IPL	4.5 h	222.2	20.7 GB	5.5×10^{15}
`ADV-O` w/ IPL	5.9 h	169.5	23.4 GB	5.9×10^{15}

Method @ 2M	Return Std	Crash Std	Grad p95	KL Drift	Div.
CAT	2.11	0.034	1.90	0.11	0/5
`ADV-O` w/o IPL	1.78	0.022	1.87	0.18	0/5
`ADV-O` w/ IPL	2.02	0.028	1.94	0.27	0/5

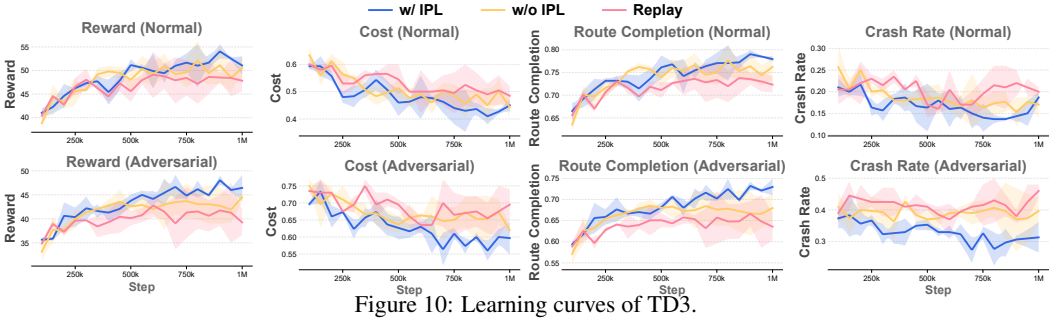

Figure 10: Learning curves of TD3.

Table 9: Detailed results of safety-critical scenario generation using `ADV-0`. Each fine-tuned variant reports mean ± standard deviation over five seeds; Avg. reports the arithmetic means of these means and standard deviations across six variants.

ADV-0 Variation	Replay		IDM		RL Agent	
	CR ↑	Reward ↓	CR ↑	Reward ↓	CR ↑	Reward ↓
Pretrained (logit-based sampling)	$31.72\% \pm 1.06\%$	43.31 ± 1.10	$18.84\% \pm 1.48\%$	50.66 ± 2.22	$12.49\% \pm 1.06\%$	52.12 ± 2.03
Pretrained (energy-based sampling)	$85.09\% \pm 1.13\%$	1.89 ± 0.20	$40.14\% \pm 1.06\%$	45.72 ± 0.47	$36.30\% \pm 0.77\%$	41.99 ± 0.33
GRPO Finetuned	$92.61 \pm 0.62\%$	0.87 ± 0.06	$46.34 \pm 0.73\%$	39.60 ± 0.13	$41.95 \pm 0.47\%$	38.45 ± 0.42
PPO Finetuned	$90.93 \pm 0.69\%$	1.01 ± 0.07	$45.09 \pm 0.63\%$	40.69 ± 0.22	$39.12 \pm 0.38\%$	39.63 ± 0.43
SAC Finetuned	$91.88 \pm 0.86\%$	1.03 ± 0.08	$46.09 \pm 0.19\%$	39.62 ± 0.24	$41.09 \pm 0.59\%$	39.17 ± 0.48
TD3 Finetuned	$89.54 \pm 0.62\%$	1.07 ± 0.06	$46.08 \pm 0.42\%$	39.38 ± 0.11	$40.35 \pm 0.84\%$	39.68 ± 0.72
PPO-Lag Finetuned	$90.08 \pm 0.42\%$	1.01 ± 0.07	$45.74 \pm 0.22\%$	40.50 ± 0.09	$41.30 \pm 0.50\%$	38.77 ± 0.39
SAC-Lag Finetuned	$91.54 \pm 0.24\%$	0.95 ± 0.04	$45.61 \pm 0.31\%$	40.38 ± 0.06	$40.28 \pm 0.73\%$	39.07 ± 0.37
Avg.	$\mathbf{91.10 \pm 0.57\%}$	$\mathbf{0.99 \pm 0.06}$	$\mathbf{45.83 \pm 0.42\%}$	$\mathbf{40.03 \pm 0.14}$	$\mathbf{40.68 \pm 0.59\%}$	$\mathbf{39.13 \pm 0.47}$

Table 10: Cross-validation performances of driving agents learned by GRPO. Within each environment, entries report mean ± standard deviation over five seeds; Avg. averages the means and standard deviations across the five environments.

	Val. Env.	RC ↑	Crash ↓	Reward ↑	Cost ↓
ADV-0 (w/ IPL)	Replay	0.713 ± 0.016	0.177 ± 0.019	45.55 ± 2.21	0.55 ± 0.01
	ADV-0	0.687 ± 0.005	0.273 ± 0.025	42.71 ± 0.98	0.64 ± 0.01
	CAT	0.695 ± 0.004	0.250 ± 0.022	44.50 ± 0.63	0.60 ± 0.02
	SAGE	0.670 ± 0.015	0.270 ± 0.016	42.49 ± 2.16	0.60 ± 0.02
	Heuristic	0.687 ± 0.007	0.207 ± 0.012	42.87 ± 1.25	0.59 ± 0.02
	Avg.	$\mathbf{0.690 \pm 0.009}$	$\mathbf{0.235 \pm 0.019}$	$\mathbf{43.62 \pm 1.45}$	$\mathbf{0.60 \pm 0.02}$
ADV-0 (w/o IPL)	Replay	0.694 ± 0.027	0.183 ± 0.025	42.20 ± 3.21	0.61 ± 0.04
	ADV-0	0.667 ± 0.014	0.280 ± 0.024	40.13 ± 1.55	0.67 ± 0.02
	CAT	0.667 ± 0.019	0.267 ± 0.026	39.99 ± 2.01	0.67 ± 0.03
	SAGE	0.661 ± 0.028	0.240 ± 0.008	40.28 ± 3.31	0.63 ± 0.04
	Heuristic	0.666 ± 0.016	0.217 ± 0.048	39.03 ± 1.62	0.64 ± 0.02
	Avg.	$\mathbf{0.671 \pm 0.021}$	$\mathbf{0.237 \pm 0.026}$	$\mathbf{40.33 \pm 2.34}$	$\mathbf{0.64 \pm 0.03}$
CAT	Replay	0.684 ± 0.015	0.197 ± 0.005	41.33 ± 1.32	0.59 ± 0.02
	ADV-0	0.641 ± 0.022	0.293 ± 0.042	37.84 ± 1.29	0.69 ± 0.00
	CAT	0.646 ± 0.020	0.280 ± 0.036	38.62 ± 1.36	0.68 ± 0.02
	SAGE	0.630 ± 0.024	0.313 ± 0.034	38.17 ± 1.55	0.67 ± 0.01
	Heuristic	0.630 ± 0.029	0.273 ± 0.033	36.57 ± 1.75	0.67 ± 0.01
	Avg.	$\mathbf{0.646 \pm 0.022}$	$\mathbf{0.271 \pm 0.030}$	$\mathbf{38.51 \pm 1.45}$	$\mathbf{0.66 \pm 0.01}$
Heuristic	Replay	0.694 ± 0.031	0.190 ± 0.010	41.45 ± 3.01	0.61 ± 0.02
	ADV-0	0.682 ± 0.068	0.323 ± 0.009	41.45 ± 2.26	0.67 ± 0.04
	CAT	0.684 ± 0.022	0.310 ± 0.021	41.89 ± 2.18	0.68 ± 0.04
	SAGE	0.661 ± 0.073	0.357 ± 0.014	40.20 ± 1.98	0.67 ± 0.01
	Heuristic	0.674 ± 0.029	0.250 ± 0.013	39.72 ± 1.77	0.66 ± 0.02
	Avg.	$\mathbf{0.679 \pm 0.045}$	$\mathbf{0.286 \pm 0.013}$	$\mathbf{40.94 \pm 2.24}$	$\mathbf{0.66 \pm 0.03}$
Replay	Replay	0.680 ± 0.029	0.180 ± 0.044	39.26 ± 0.90	0.61 ± 0.01
	ADV-0	0.635 ± 0.042	0.308 ± 0.050	36.64 ± 3.20	0.69 ± 0.01
	CAT	0.645 ± 0.055	0.319 ± 0.038	37.48 ± 1.06	0.66 ± 0.01
	SAGE	0.616 ± 0.062	0.319 ± 0.032	35.72 ± 1.25	0.69 ± 0.03
	Heuristic	0.637 ± 0.047	0.264 ± 0.047	35.77 ± 3.46	0.69 ± 0.02
	Avg.	$\mathbf{0.643 \pm 0.047}$	$\mathbf{0.278 \pm 0.042}$	$\mathbf{36.97 \pm 1.97}$	$\mathbf{0.67 \pm 0.02}$

Table 11: Cross-validation performances of driving agents learned by PPO. Within each environment, entries report mean ± standard deviation over five seeds; Avg. averages the means and standard deviations across the five environments.

	Val. Env.	RC ↑	Crash ↓	Reward ↑	Cost ↓
ADV-0 (w/ IPL)	Replay	0.707 ± 0.018	0.193 ± 0.015	45.56 ± 0.87	0.56 ± 0.02
	ADV-0	0.681 ± 0.021	0.270 ± 0.019	42.68 ± 0.69	0.62 ± 0.03
	CAT	0.690 ± 0.010	0.260 ± 0.029	43.65 ± 1.54	0.60 ± 0.04
	SAGE	0.678 ± 0.008	0.268 ± 0.023	43.60 ± 1.37	0.59 ± 0.03
	Heuristic	0.676 ± 0.006	0.250 ± 0.016	41.63 ± 1.59	0.61 ± 0.03
	Avg.	$\mathbf{0.686 \pm 0.013}$	$\mathbf{0.248 \pm 0.020}$	$\mathbf{43.42 \pm 1.21}$	$\mathbf{0.60 \pm 0.03}$
ADV-0 (w/o IPL)	Replay	0.696 ± 0.010	0.188 ± 0.018	44.80 ± 1.01	0.59 ± 0.01
	ADV-0	0.673 ± 0.006	0.280 ± 0.021	42.10 ± 1.34	0.67 ± 0.01
	CAT	0.677 ± 0.009	0.275 ± 0.047	42.62 ± 2.00	0.67 ± 0.01
	SAGE	0.670 ± 0.013	0.263 ± 0.013	42.96 ± 1.80	0.63 ± 0.03
	Heuristic	0.658 ± 0.002	0.265 ± 0.017	39.65 ± 1.10	0.68 ± 0.02
	Avg.	$\mathbf{0.675 \pm 0.008}$	$\mathbf{0.254 \pm 0.023}$	$\mathbf{42.43 \pm 1.45}$	$\mathbf{0.65 \pm 0.02}$
CAT	Replay	0.717 ± 0.013	0.211 ± 0.041	45.40 ± 1.05	0.57 ± 0.03
	ADV-0	0.666 ± 0.023	0.330 ± 0.071	40.52 ± 1.85	0.66 ± 0.03
	CAT	0.679 ± 0.021	0.319 ± 0.028	42.29 ± 1.53	0.64 ± 0.01
	SAGE	0.653 ± 0.008	0.310 ± 0.010	40.53 ± 0.75	0.65 ± 0.01
	Heuristic	0.676 ± 0.012	0.270 ± 0.030	41.49 ± 0.70	0.62 ± 0.01
	Avg.	$\mathbf{0.678 \pm 0.015}$	$\mathbf{0.288 \pm 0.036}$	$\mathbf{42.05 \pm 1.18}$	$\mathbf{0.63 \pm 0.02}$
Heuristic	Replay	0.608 ± 0.038	0.210 ± 0.020	37.08 ± 0.91	0.66 ± 0.02
	ADV-0	0.577 ± 0.051	0.303 ± 0.050	33.53 ± 0.80	0.74 ± 0.01
	CAT	0.593 ± 0.010	0.278 ± 0.015	35.79 ± 2.10	0.70 ± 0.02
	SAGE	0.593 ± 0.008	0.270 ± 0.012	36.02 ± 2.00	0.70 ± 0.02
	Heuristic	0.563 ± 0.021	0.289 ± 0.020	31.72 ± 1.77	0.72 ± 0.01
	Avg.	$\mathbf{0.587 \pm 0.026}$	$\mathbf{0.270 \pm 0.023}$	$\mathbf{34.83 \pm 1.52}$	$\mathbf{0.70 \pm 0.02}$
Replay	Replay	0.697 ± 0.052	0.229 ± 0.014	44.26 ± 0.55	0.60 ± 0.02
	ADV-0	0.629 ± 0.035	0.420 ± 0.030	38.32 ± 1.33	0.73 ± 0.02
	CAT	0.663 ± 0.025	0.380 ± 0.048	41.26 ± 1.67	0.68 ± 0.04
	SAGE	0.638 ± 0.020	0.360 ± 0.045	41.01 ± 1.90	0.63 ± 0.03
	Heuristic	0.620 ± 0.019	0.381 ± 0.032	36.63 ± 1.00	0.72 ± 0.02
	Avg.	$\mathbf{0.649 \pm 0.030}$	$\mathbf{0.354 \pm 0.034}$	$\mathbf{40.30 \pm 1.29}$	$\mathbf{0.67 \pm 0.03}$

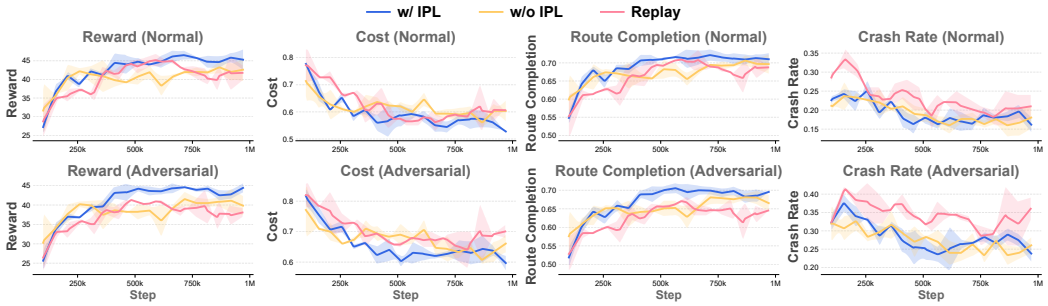

Figure 11: Learning curves of GRPO.

Table 12: Cross-validation performances of driving agents learned by PPO-Lag. Within each environment, entries report mean ± standard deviation over five seeds; Avg. averages the means and standard deviations across the five environments.

	Val. Env.	RC ↑	Crash ↓	Reward ↑	Cost ↓
ADV-0 (w/ IPL)	Replay	0.676 ± 0.012	0.142 ± 0.015	44.82 ± 1.20	0.53 ± 0.01
	ADV-0	0.626 ± 0.008	0.260 ± 0.027	38.17 ± 0.95	0.67 ± 0.02
	CAT	0.647 ± 0.013	0.250 ± 0.027	40.80 ± 1.11	0.64 ± 0.02
	SAGE	0.632 ± 0.014	0.271 ± 0.018	39.61 ± 1.83	0.65 ± 0.02
	Heuristic	0.638 ± 0.009	0.256 ± 0.019	38.34 ± 1.55	0.68 ± 0.02
	Avg.	**0.644 ± 0.011**	**0.236 ± 0.021**	**40.35 ± 1.33**	**0.63 ± 0.02**
ADV-0 (w/o IPL)	Replay	0.605 ± 0.022	0.178 ± 0.028	33.08 ± 2.55	0.74 ± 0.03
	ADV-0	0.603 ± 0.018	0.291 ± 0.030	32.78 ± 1.83	0.74 ± 0.02
	CAT	0.603 ± 0.023	0.272 ± 0.035	32.56 ± 2.14	0.74 ± 0.03
	SAGE	0.585 ± 0.020	0.260 ± 0.025	31.85 ± 2.84	0.72 ± 0.04
	Heuristic	0.593 ± 0.020	0.265 ± 0.040	31.80 ± 1.90	0.76 ± 0.02
	Avg.	**0.598 ± 0.021**	**0.253 ± 0.032**	**32.41 ± 2.25**	**0.74 ± 0.03**
CAT	Replay	0.625 ± 0.015	0.190 ± 0.005	36.51 ± 1.30	0.68 ± 0.02
	ADV-0	0.599 ± 0.022	0.305 ± 0.042	32.91 ± 1.25	0.75 ± 0.01
	CAT	0.608 ± 0.020	0.290 ± 0.035	33.82 ± 1.35	0.73 ± 0.02
	SAGE	0.590 ± 0.025	0.316 ± 0.035	33.19 ± 1.50	0.74 ± 0.01
	Heuristic	0.604 ± 0.028	0.286 ± 0.033	33.09 ± 1.70	0.71 ± 0.01
	Avg.	**0.605 ± 0.022**	**0.277 ± 0.030**	**33.90 ± 1.42**	**0.72 ± 0.01**
Heuristic	Replay	0.630 ± 0.030	0.202 ± 0.010	36.82 ± 3.00	0.69 ± 0.02
	ADV-0	0.590 ± 0.065	0.333 ± 0.010	33.52 ± 2.20	0.76 ± 0.04
	CAT	0.606 ± 0.022	0.324 ± 0.020	34.27 ± 2.15	0.75 ± 0.04
	SAGE	0.586 ± 0.070	0.354 ± 0.015	32.56 ± 2.00	0.76 ± 0.01
	Heuristic	0.590 ± 0.029	0.290 ± 0.015	32.80 ± 1.80	0.74 ± 0.02
	Avg.	**0.600 ± 0.043**	**0.301 ± 0.014**	**33.99 ± 2.23**	**0.74 ± 0.03**
Replay	Replay	0.620 ± 0.030	0.215 ± 0.045	34.53 ± 0.90	0.70 ± 0.01
	ADV-0	0.570 ± 0.040	0.364 ± 0.050	30.20 ± 3.20	0.78 ± 0.01
	CAT	0.585 ± 0.055	0.358 ± 0.040	31.53 ± 1.10	0.76 ± 0.01
	SAGE	0.565 ± 0.060	0.358 ± 0.035	29.80 ± 1.25	0.78 ± 0.03
	Heuristic	0.581 ± 0.045	0.320 ± 0.045	30.50 ± 3.40	0.77 ± 0.02
	Avg.	**0.584 ± 0.046**	**0.323 ± 0.043**	**31.31 ± 1.97**	**0.76 ± 0.02**

Table 13: Cross-validation performances of driving agents learned by SAC. Within each environment, entries report mean ± standard deviation over five seeds; Avg. averages the means and standard deviations across the five environments.

	Val. Env.	RC ↑	Crash ↓	Reward ↑	Cost ↓
ADV-0 (w/ IPL)	Replay	0.781 ± 0.011	0.130 ± 0.016	53.66 ± 1.02	0.41 ± 0.01
	ADV-0	0.736 ± 0.011	0.305 ± 0.022	49.70 ± 0.34	0.53 ± 0.01
	CAT	0.745 ± 0.013	0.268 ± 0.024	50.48 ± 1.02	0.52 ± 0.03
	SAGE	0.745 ± 0.016	0.250 ± 0.025	48.84 ± 1.79	0.51 ± 0.04
	Heuristic	0.758 ± 0.031	0.170 ± 0.019	50.48 ± 2.79	0.45 ± 0.06
	Avg.	**0.753 ± 0.016**	**0.225 ± 0.021**	**50.63 ± 1.39**	**0.48 ± 0.03**
ADV-0 (w/o IPL)	Replay	0.784 ± 0.020	0.160 ± 0.022	54.40 ± 1.69	0.44 ± 0.03
	ADV-0	0.706 ± 0.020	0.307 ± 0.026	45.92 ± 2.59	0.58 ± 0.03
	CAT	0.729 ± 0.011	0.287 ± 0.026	47.81 ± 1.48	0.58 ± 0.03
	SAGE	0.743 ± 0.027	0.260 ± 0.024	49.35 ± 2.73	0.50 ± 0.05
	Heuristic	0.743 ± 0.033	0.203 ± 0.034	48.98 ± 3.48	0.49 ± 0.07
	Avg.	**0.741 ± 0.022**	**0.243 ± 0.026**	**49.29 ± 2.39**	**0.52 ± 0.04**
CAT	Replay	0.775 ± 0.013	0.177 ± 0.041	54.06 ± 1.12	0.42 ± 0.00
	ADV-0	0.698 ± 0.010	0.353 ± 0.034	45.81 ± 0.57	0.60 ± 0.03
	CAT	0.704 ± 0.018	0.327 ± 0.041	45.78 ± 1.55	0.60 ± 0.02
	SAGE	0.705 ± 0.025	0.297 ± 0.029	46.65 ± 3.41	0.56 ± 0.03
	Heuristic	0.727 ± 0.024	0.247 ± 0.038	47.48 ± 2.57	0.53 ± 0.04
	Avg.	**0.722 ± 0.018**	**0.280 ± 0.037**	**47.96 ± 1.84**	**0.54 ± 0.02**
Heuristic	Replay	0.718 ± 0.030	0.200 ± 0.028	47.20 ± 4.13	0.54 ± 0.07
	ADV-0	0.665 ± 0.025	0.330 ± 0.025	42.10 ± 2.64	0.63 ± 0.02
	CAT	0.672 ± 0.024	0.320 ± 0.022	42.50 ± 2.84	0.64 ± 0.03
	SAGE	0.670 ± 0.030	0.295 ± 0.030	42.30 ± 3.77	0.60 ± 0.05
	Heuristic	0.678 ± 0.028	0.275 ± 0.015	43.10 ± 4.29	0.61 ± 0.05
	Avg.	**0.681 ± 0.027**	**0.284 ± 0.024**	**43.44 ± 3.53**	**0.60 ± 0.04**
Replay	Replay	0.710 ± 0.035	0.217 ± 0.035	45.51 ± 1.30	0.56 ± 0.02
	ADV-0	0.625 ± 0.045	0.360 ± 0.046	37.81 ± 2.50	0.67 ± 0.03
	CAT	0.652 ± 0.035	0.353 ± 0.041	39.20 ± 2.00	0.66 ± 0.04
	SAGE	0.635 ± 0.029	0.332 ± 0.040	38.54 ± 2.10	0.64 ± 0.03
	Heuristic	0.638 ± 0.027	0.316 ± 0.035	39.09 ± 2.50	0.65 ± 0.01
	Avg.	**0.652 ± 0.034**	**0.316 ± 0.039**	**40.03 ± 2.08**	**0.64 ± 0.03**

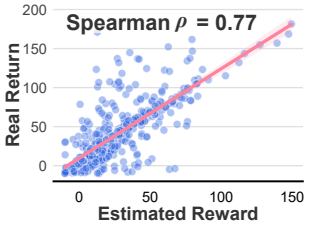

Figure 12: **Validation of proxy reward estimator.** Comparison of the estimated returns against the ground-truth. The strong Spearman correlation ($\rho = 0.77$) suggests that our rule-based proxy effectively preserves the preference ranking of adversarial candidates.

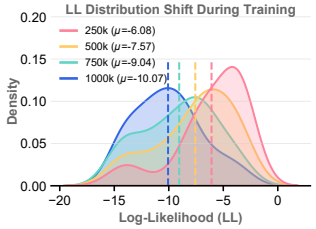

Figure 13: **Evolution of the adversary distribution**. Likelihood of adversarial trajectories at different training steps. As the ego improves, the distribution shifts towards lower values, suggesting that `ADV-0` actively identifies nonstationary failure boundary.

Table 14: Cross-validation performances of driving agents learned by SAC-Lag. Within each environment, entries report mean ± standard deviation over five seeds; Avg. averages the means and standard deviations across the five environments.

	Val. Env.	RC ↑	Crash ↓	Reward ↑	Cost ↓
ADV-0 (w/ IPL)	Replay	0.787 ± 0.006	0.127 ± 0.017	54.62 ± 0.26	0.40 ± 0.02
	ADV-0	0.729 ± 0.002	0.303 ± 0.042	48.78 ± 0.35	0.54 ± 0.03
	CAT	0.725 ± 0.011	0.297 ± 0.019	48.01 ± 1.10	0.57 ± 0.02
	SAGE	0.736 ± 0.007	0.250 ± 0.024	48.88 ± 0.58	0.52 ± 0.01
	Heuristic	0.742 ± 0.022	0.217 ± 0.045	49.25 ± 2.68	0.49 ± 0.06
	Avg.	**0.744 ± 0.010**	**0.239 ± 0.029**	**49.91 ± 0.99**	**0.50 ± 0.03**
ADV-0 (w/o IPL)	Replay	0.776 ± 0.009	0.135 ± 0.025	53.75 ± 1.29	0.42 ± 0.02
	ADV-0	0.689 ± 0.004	0.320 ± 0.020	44.41 ± 0.98	0.61 ± 0.01
	CAT	0.699 ± 0.002	0.320 ± 0.000	45.36 ± 0.43	0.61 ± 0.03
	SAGE	0.707 ± 0.009	0.260 ± 0.030	46.63 ± 0.19	0.56 ± 0.02
	Heuristic	0.730 ± 0.013	0.205 ± 0.035	47.97 ± 0.34	0.49 ± 0.03
	Avg.	**0.720 ± 0.007**	**0.248 ± 0.022**	**47.62 ± 0.65**	**0.54 ± 0.02**
CAT	Replay	0.765 ± 0.015	0.165 ± 0.036	52.80 ± 1.15	0.43 ± 0.01
	ADV-0	0.685 ± 0.012	0.345 ± 0.035	44.55 ± 0.61	0.61 ± 0.03
	CAT	0.692 ± 0.018	0.320 ± 0.047	44.78 ± 1.50	0.61 ± 0.02
	SAGE	0.695 ± 0.025	0.305 ± 0.033	45.87 ± 3.22	0.57 ± 0.03
	Heuristic	0.712 ± 0.024	0.255 ± 0.042	46.50 ± 2.52	0.54 ± 0.04
	Avg.	**0.710 ± 0.019**	**0.278 ± 0.039**	**46.90 ± 1.80**	**0.55 ± 0.03**
Heuristic	Replay	0.731 ± 0.030	0.193 ± 0.024	48.36 ± 4.17	0.53 ± 0.07
	ADV-0	0.677 ± 0.022	0.320 ± 0.024	43.47 ± 2.58	0.62 ± 0.02
	CAT	0.686 ± 0.024	0.313 ± 0.021	43.81 ± 2.86	0.63 ± 0.03
	SAGE	0.687 ± 0.027	0.283 ± 0.031	43.61 ± 3.71	0.59 ± 0.05
	Heuristic	0.691 ± 0.028	0.267 ± 0.012	44.19 ± 4.19	0.60 ± 0.05
	Avg.	**0.694 ± 0.026**	**0.275 ± 0.022**	**44.69 ± 3.50**	**0.59 ± 0.04**
Replay	Replay	0.718 ± 0.035	0.205 ± 0.034	46.25 ± 1.25	0.55 ± 0.02
	ADV-0	0.635 ± 0.047	0.355 ± 0.045	38.80 ± 2.55	0.66 ± 0.03
	CAT	0.665 ± 0.033	0.345 ± 0.040	40.15 ± 2.00	0.65 ± 0.04
	SAGE	0.645 ± 0.031	0.325 ± 0.040	39.50 ± 2.11	0.63 ± 0.03
	Heuristic	0.650 ± 0.029	0.305 ± 0.036	39.80 ± 2.54	0.64 ± 0.01
	Avg.	**0.663 ± 0.035**	**0.307 ± 0.039**	**40.90 ± 2.09**	**0.63 ± 0.03**

Table 15: Cross-validation performances of driving agents learned by TD3. Within each environment, entries report mean ± standard deviation over five seeds; Avg. averages the means and standard deviations across the five environments.

	Val. Env.	RC ↑	Crash ↓	Reward ↑	Cost ↓
ADV-0 (w/ IPL)	Replay	0.787 ± 0.002	0.187 ± 0.031	53.15 ± 0.76	0.43 ± 0.01
	ADV-0	0.711 ± 0.016	0.323 ± 0.038	45.54 ± 1.70	0.59 ± 0.03
	CAT	0.724 ± 0.016	0.300 ± 0.029	46.67 ± 1.55	0.58 ± 0.02
	SAGE	0.734 ± 0.016	0.267 ± 0.046	47.69 ± 1.52	0.53 ± 0.05
	Heuristic	0.757 ± 0.021	0.200 ± 0.014	49.87 ± 2.64	0.49 ± 0.02
	Avg.	**0.743 ± 0.014**	**0.255 ± 0.032**	**48.58 ± 1.63**	**0.52 ± 0.03**
ADV-0 (w/o IPL)	Replay	0.761 ± 0.031	0.160 ± 0.024	50.21 ± 2.09	0.44 ± 0.01
	ADV-0	0.644 ± 0.046	0.423 ± 0.057	40.86 ± 2.90	0.67 ± 0.03
	CAT	0.689 ± 0.036	0.373 ± 0.077	45.46 ± 2.04	0.59 ± 0.05
	SAGE	0.669 ± 0.049	0.337 ± 0.068	41.95 ± 3.39	0.62 ± 0.05
	Heuristic	0.685 ± 0.046	0.277 ± 0.069	43.25 ± 4.07	0.61 ± 0.08
	Avg.	**0.690 ± 0.042**	**0.314 ± 0.059**	**44.35 ± 2.90**	**0.59 ± 0.04**
CAT	Replay	0.755 ± 0.013	0.160 ± 0.029	49.03 ± 0.97	0.48 ± 0.01
	ADV-0	0.668 ± 0.017	0.367 ± 0.033	42.60 ± 0.23	0.65 ± 0.02
	CAT	0.673 ± 0.006	0.343 ± 0.005	42.46 ± 1.60	0.65 ± 0.02
	SAGE	0.687 ± 0.016	0.300 ± 0.008	43.99 ± 2.09	0.58 ± 0.01
	Heuristic	0.706 ± 0.009	0.237 ± 0.029	45.21 ± 0.88	0.56 ± 0.03
	Avg.	**0.698 ± 0.012**	**0.281 ± 0.021**	**44.66 ± 1.15**	**0.58 ± 0.02**
Heuristic	Replay	0.713 ± 0.030	0.210 ± 0.016	45.57 ± 3.01	0.52 ± 0.02
	ADV-0	0.632 ± 0.051	0.380 ± 0.010	40.47 ± 2.24	0.65 ± 0.04
	CAT	0.661 ± 0.022	0.320 ± 0.020	42.40 ± 2.18	0.61 ± 0.04
	SAGE	0.654 ± 0.045	0.315 ± 0.015	42.51 ± 1.99	0.63 ± 0.01
	Heuristic	0.652 ± 0.047	0.305 ± 0.022	41.35 ± 1.85	0.60 ± 0.02
	Avg.	**0.662 ± 0.039**	**0.306 ± 0.017**	**42.46 ± 2.25**	**0.60 ± 0.03**
Replay	Replay	0.727 ± 0.026	0.210 ± 0.010	47.77 ± 3.00	0.51 ± 0.07
	ADV-0	0.638 ± 0.019	0.435 ± 0.005	39.01 ± 2.08	0.67 ± 0.02
	CAT	0.643 ± 0.038	0.455 ± 0.015	39.96 ± 3.15	0.69 ± 0.04
	SAGE	0.654 ± 0.035	0.340 ± 0.050	41.54 ± 2.42	0.59 ± 0.04
	Heuristic	0.700 ± 0.020	0.280 ± 0.010	45.02 ± 1.55	0.56 ± 0.01
	Avg.	**0.672 ± 0.028**	**0.344 ± 0.018**	**42.66 ± 2.44**	**0.60 ± 0.04**

Table 16: Performance comparison of learning-based planners before and after adversarial fine-tuning using ADV-0 (GRPO).

Model Phase	Val. Env.	RC ↑	Crash ↓	Reward ↑	Cost ↓
PlanTF (Pretrained)	Replay	0.727 ± 0.021	0.220 ± 0.027	41.53 ± 3.10	0.76 ± 0.03
	ADV-0	0.581 ± 0.030	0.420 ± 0.035	32.11 ± 2.54	1.24 ± 0.05
	CAT	0.615 ± 0.025	0.385 ± 0.030	35.26 ± 2.16	1.09 ± 0.03
	SAGE	0.595 ± 0.028	0.407 ± 0.032	33.80 ± 2.80	1.15 ± 0.04
	Heuristic	0.620 ± 0.020	0.352 ± 0.032	36.53 ± 1.97	0.94 ± 0.03
	Avg.	**0.628 ± 0.025**	**0.357 ± 0.031**	**35.85 ± 2.51**	**1.04 ± 0.04**
PlanTF (Fine-tuned)	Replay	0.738 ± 0.015	0.199 ± 0.018	46.20 ± 1.88	0.62 ± 0.02
	ADV-0	0.655 ± 0.012	0.293 ± 0.022	40.80 ± 1.20	0.85 ± 0.00
	CAT	0.668 ± 0.010	0.273 ± 0.020	42.11 ± 1.16	0.78 ± 0.01
	SAGE	0.640 ± 0.018	0.305 ± 0.025	39.50 ± 1.56	0.88 ± 0.03
	Heuristic	0.671 ± 0.024	0.246 ± 0.038	41.28 ± 2.32	0.72 ± 0.05
	Avg.	**0.674 ± 0.016**	**0.263 ± 0.025**	**41.98 ± 1.62**	**0.77 ± 0.02**
Average Relative Change		**+7.46%**	**-26.23%**	**+17.11%**	**-25.68%**
SMART (Pretrained)	Replay	0.686 ± 0.020	0.255 ± 0.023	38.59 ± 3.51	0.83 ± 0.05
	ADV-0	0.540 ± 0.035	0.460 ± 0.049	29.80 ± 2.99	1.35 ± 0.04
	CAT	0.565 ± 0.031	0.433 ± 0.033	31.50 ± 2.43	1.23 ± 0.05
	SAGE	0.556 ± 0.032	0.448 ± 0.038	30.27 ± 3.10	1.30 ± 0.04
	Heuristic	0.586 ± 0.029	0.384 ± 0.026	33.16 ± 2.20	1.05 ± 0.07
	Avg.	**0.587 ± 0.029**	**0.396 ± 0.034**	**32.66 ± 2.85**	**1.15 ± 0.05**
SMART (Fine-tuned)	Replay	0.705 ± 0.010	0.230 ± 0.026	42.13 ± 2.11	0.77 ± 0.01
	ADV-0	0.610 ± 0.018	0.340 ± 0.022	36.51 ± 1.56	0.97 ± 0.02
	CAT	0.625 ± 0.012	0.313 ± 0.022	37.82 ± 1.40	0.95 ± 0.03
	SAGE	0.595 ± 0.020	0.352 ± 0.028	35.20 ± 1.82	1.05 ± 0.03
	Heuristic	0.620 ± 0.016	0.292 ± 0.018	37.57 ± 1.63	0.85 ± 0.03
	Avg.	**0.631 ± 0.015**	**0.305 ± 0.023**	**37.85 ± 1.70**	**0.92 ± 0.02**
Average Relative Change		**+7.57%**	**-22.88%**	**+15.86%**	**-20.31%**

Table 17: **Detailed breakdown of the trajectory-level reward model values**. Values represent the average accumulated discounted reward (over a 2.0s horizon with discount factor $\gamma = 0.99$) of the *best* planned trajectory selected by the model across all validation steps. **Reward components & weights**: **Progress** ($w_{\text{prog}} = 1.0$, longitudinal advance), **Collision** ($w_{\text{coll}} = 20.0$, penalty for overlap with objects), **Off-road** ($w_{\text{off}} = 5.0$, penalty for lane deviation), **Comfort** ($w_{\text{acc}} = 0.1$, $w_{\text{jerk}} = 0.1$, penalties for harsh dynamics), **Speed Efficiency** ($w_{\text{eff}} = 0.2$, penalty for deviating from 10m/s). Note that the negative improvement in Speed Efficiency reflects a safety-efficiency trade-off, where the fine-tuned planner adopts a more conservative velocity profile to satisfy safety constraints.

Method	Progress (+)	Collision (-)	Off-road (-)	Comfort (-)	Speed Eff. (-)	Total Score
PlanTF (Pretrained)	14.17 ± 1.53	-2.85 ± 0.92	-0.12 ± 0.05	-0.34 ± 0.08	-0.54 ± 0.12	**10.32 ± 1.79**
PlanTF (Fine-tuned)	16.45 ± 0.85	-0.89 ± 0.45	-0.05 ± 0.04	-0.24 ± 0.05	-0.64 ± 0.13	**14.63 ± 0.97**
Improvement	**+16.09%**	**+68.77%**	**+58.33%**	**+29.41%**	-18.52%	**+41.76%**
SMART (Pretrained)	13.20 ± 1.80	-3.61 ± 1.11	-0.25 ± 0.08	-1.26 ± 0.25	-0.60 ± 0.15	**7.48 ± 2.14**
SMART (Fine-tuned)	15.10 ± 1.10	-1.75 ± 0.37	-0.17 ± 0.05	-0.92 ± 0.19	-0.65 ± 0.11	**11.61 ± 1.18**
Improvement	**+14.39%**	**+51.52%**	**+32.00%**	**+26.98%**	-8.33%	**+55.21%**

Table 18: **Full results of quantitative evaluation on the unbiased long-tailed set mined from real-world data.** The benchmark consists of four long-tail scenario categories filtered by strict physical thresholds: *Critical TTC* ($\min \text{TTC} < 0.4s$), *Critical PET* ($\text{PET} < 1.0s$), *Hard Dynamics* (Longitudinal Acc $< -4.0m/s^2$ or $|\text{Jerk}| > 4.0m/s^3$), and *Rare Cluster* (topologically sparse trajectory clusters). **Reactive Traffic** denotes whether background vehicles utilize IDM/MOBIL policies to interact with the agent (✓) or strictly follow logged trajectories (×). Metrics assess **Safety Margin** (higher values indicate earlier risk detection), **Stability & Comfort** (lower Jerk indicates smoother control), and **Defensive Driving** performance, quantified by Near-Miss Rate (hazardous proximity without collision) and RDP Violation (percentage of time requiring deceleration $> 6m/s^2$ to avoid collision).

	Scenario Category	Percentage	Reactive Traffic	Safety Margin		Stability & Comfort		Defensive Driving	
				Avg Min-TTC (↑)	Avg Min-PET (↑)	Mean Abs Jerk (↓)	95% Jerk (↓)	Near-Miss Rate (↓)	RDP Violation Rate (↓)
ADV-0 (w/ IPL)	Critical TTC	7.40%	✓	0.645 ± 0.122	1.450 ± 0.50	1.853 ± 0.188	5.850 ± 1.252	65.28% ± 1.45%	15.53% ± 4.16%
	Critical PET	3.40%		0.492 ± 0.025	1.150 ± 0.20	1.623 ± 0.095	5.157 ± 0.653	74.57% ± 5.52%	58.26% ± 3.29%
	Hard Dynamics	3.20%		0.885 ± 0.157	N/A	1.685 ± 0.250	4.555 ± 0.951	55.40% ± 8.26%	29.54% ± 4.80%
	Rare Cluster	5.20%		1.951 ± 0.450	0.850 ± 0.25	1.451 ± 0.143	4.650 ± 0.684	58.55% ± 2.12%	31.48% ± 2.81%
	Avg			**0.993 ± 0.189**	**1.150 ± 0.317**	**1.653 ± 0.169**	**5.053 ± 0.885**	**63.45% ± 4.34%**	**33.70% ± 3.77%**
	Critical TTC	7.40%	×	0.658 ± 0.130	1.410 ± 0.48	1.812 ± 0.165	5.784 ± 1.102	62.15% ± 1.32%	14.80% ± 3.83%
	Critical PET	3.40%		2.550 ± 0.036	1.080 ± 0.22	1.651 ± 0.117	5.923 ± 0.823	68.40% ± 4.86%	65.13% ± 2.56%
	Hard Dynamics	3.20%		0.850 ± 0.140	N/A	1.525 ± 0.380	4.421 ± 1.259	58.11% ± 7.52%	30.23% ± 3.50%
	Rare Cluster	5.20%		2.051 ± 0.422	0.820 ± 0.24	1.480 ± 0.155	4.580 ± 0.626	55.23% ± 2.05%	35.63% ± 1.51%
	Avg			**1.527 ± 0.182**	**1.103 ± 0.313**	**1.617 ± 0.204**	**5.177 ± 0.953**	**60.97% ± 3.94%**	**36.45% ± 2.85%**
CAT	Critical TTC	7.40%	✓	0.465 ± 0.058	1.350 ± 0.45	2.123 ± 0.245	6.726 ± 1.064	81.98% ± 3.12%	24.46% ± 3.47%
	Critical PET	3.40%		0.411 ± 0.021	0.950 ± 0.25	1.653 ± 0.108	4.706 ± 0.405	91.67% ± 3.61%	73.20% ± 8.84%
	Hard Dynamics	3.20%		0.573 ± 0.137	N/A	1.889 ± 0.271	4.985 ± 0.706	85.42% ± 9.55%	41.04% ± 7.71%
	Rare Cluster	5.20%		1.850 ± 0.55	0.750 ± 0.25	1.834 ± 0.292	5.271 ± 1.155	73.08% ± 7.69%	40.48% ± 4.46%
	Avg			**0.825 ± 0.192**	**1.017 ± 0.317**	**1.875 ± 0.229**	**5.422 ± 0.833**	**83.04% ± 5.99%**	**44.80% ± 6.12%**
	Critical TTC	7.40%	×	0.469 ± 0.058	1.310 ± 0.40	2.168 ± 0.206	6.871 ± 1.149	81.08% ± 4.68%	24.82% ± 3.54%
	Critical PET	3.40%		2.850 ± 1.10	0.910 ± 0.30	1.818 ± 0.085	5.876 ± 0.869	75.00% ± 12.50%	87.97% ± 4.17%
	Hard Dynamics	3.20%		0.559 ± 0.091	N/A	1.960 ± 0.299	5.232 ± 0.886	81.25% ± 6.25%	44.03% ± 5.88%
	Rare Cluster	5.20%		1.780 ± 0.60	0.720 ± 0.28	1.969 ± 0.182	6.028 ± 0.820	64.10% ± 8.01%	47.96% ± 5.38%
	Avg			**1.415 ± 0.462**	**0.980 ± 0.327**	**1.979 ± 0.193**	**6.002 ± 0.931**	**75.36% ± 7.86%**	**51.20% ± 4.74%**
Heuristic	Critical TTC	7.40%	✓	0.513 ± 0.066	1.550 ± 0.60	2.252 ± 0.101	6.849 ± 0.698	81.98% ± 1.56%	25.73% ± 1.89%
	Critical PET	3.40%		0.404 ± 0.039	1.100 ± 0.50	1.957 ± 0.316	6.204 ± 0.577	85.42% ± 3.61%	81.71% ± 6.07%
	Hard Dynamics	3.20%		0.938 ± 0.244	N/A	2.151 ± 0.308	6.353 ± 1.384	62.50% ± 16.54%	41.41% ± 5.35%
	Rare Cluster	5.20%		1.600 ± 0.45	0.800 ± 0.30	2.155 ± 0.224	6.761 ± 1.107	67.95% ± 2.22%	46.31% ± 1.31%
	Avg			**0.864 ± 0.200**	**1.150 ± 0.467**	**2.129 ± 0.237**	**6.542 ± 0.942**	**74.46% ± 5.98%**	**48.79% ± 3.65%**
	Critical TTC	7.40%	×	0.503 ± 0.061	1.480 ± 0.55	2.229 ± 0.054	6.945 ± 0.891	79.28% ± 4.13%	26.07% ± 2.60%
	Critical PET	3.40%		3.550 ± 1.50	1.050 ± 0.45	2.103 ± 0.171	6.449 ± 0.485	75.00% ± 12.50%	88.02% ± 2.03%
	Hard Dynamics	3.20%		0.833 ± 0.190	N/A	2.098 ± 0.358	6.444 ± 1.220	64.58% ± 7.22%	43.71% ± 0.52%
	Rare Cluster	5.20%		1.520 ± 0.40	0.750 ± 0.35	2.366 ± 0.288	7.273 ± 1.668	56.41% ± 12.36%	50.61% ± 1.95%
	Avg			**1.601 ± 0.538**	**1.093 ± 0.450**	**2.199 ± 0.218**	**6.778 ± 1.066**	**68.82% ± 9.05%**	**52.10% ± 1.77%**
Replay	Critical TTC	7.40%	✓	0.355 ± 0.045	0.950 ± 0.35	2.855 ± 0.359	8.553 ± 2.101	94.28% ± 2.51%	55.69% ± 6.27%
	Critical PET	3.40%		0.282 ± 0.015	0.650 ± 0.15	2.555 ± 0.427	7.857 ± 1.259	96.86% ± 1.56%	92.54% ± 1.85%
	Hard Dynamics	3.20%		0.453 ± 0.124	N/A	2.951 ± 0.550	8.200 ± 1.655	85.49% ± 9.55%	68.23% ± 8.46%
	Rare Cluster	5.20%		1.257 ± 0.555	0.450 ± 0.15	2.753 ± 0.387	8.450 ± 1.858	81.50% ± 3.59%	62.47% ± 4.50%
	Avg			**0.587 ± 0.185**	**0.683 ± 0.217**	**2.779 ± 0.431**	**8.265 ± 1.718**	**89.53% ± 4.30%**	**69.73% ± 5.27%**
	Critical TTC	7.40%	×	0.389 ± 0.050	1.050 ± 0.40	2.653 ± 0.281	8.159 ± 1.806	91.51% ± 2.11%	52.45% ± 5.58%
	Critical PET	3.40%		1.855 ± 1.204	0.720 ± 0.18	2.481 ± 0.357	7.620 ± 0.956	88.56% ± 4.25%	90.18% ± 2.11%
	Hard Dynamics	3.20%		0.481 ± 0.112	N/A	2.825 ± 0.485	7.950 ± 1.458	82.15% ± 8.81%	65.57% ± 7.25%
	Rare Cluster	5.20%		1.188 ± 0.483	0.420 ± 0.12	2.922 ± 0.410	8.688 ± 1.928	78.29% ± 3.16%	65.82% ± 3.80%
	Avg			**0.978 ± 0.462**	**0.730 ± 0.233**	**2.720 ± 0.383**	**8.104 ± 1.537**	**85.13% ± 4.58%**	**68.50% ± 4.68%**

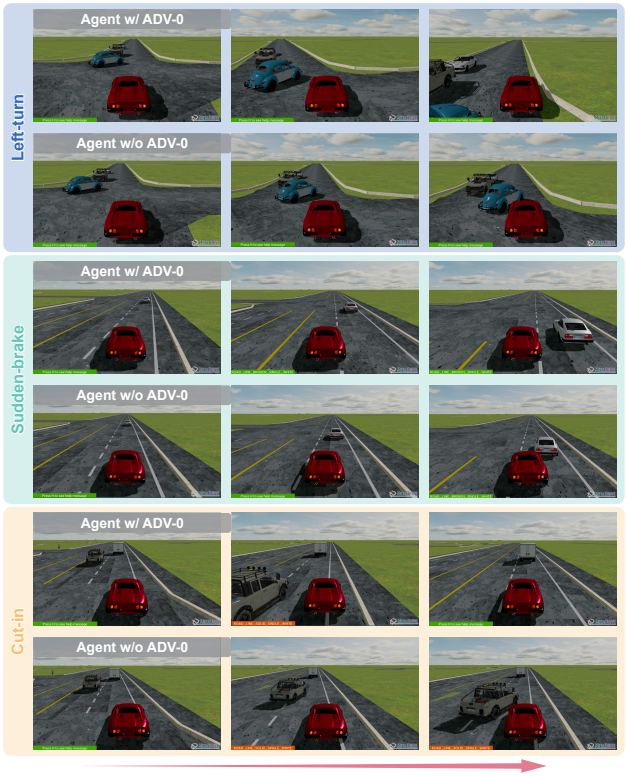

Figure 14: **Qualitative comparison of improved safe driving ability after being trained with `ADV-0`.** We showcase three typical safety-critical scenarios generated by the adversary: Left-turn, Sudden-brake, and Cut-in. In each column, the bottom row (Agent w/o `ADV-0`) shows the baseline agent failing to anticipate the aggressive behavior of the background traffic, resulting in collisions. In contrast, the top row (Agent w/ `ADV-0`) demonstrates that the agent trained with our framework learns robust defensive behaviors, such as yielding at intersections or decelerating in time, thereby successfully avoiding accidents.

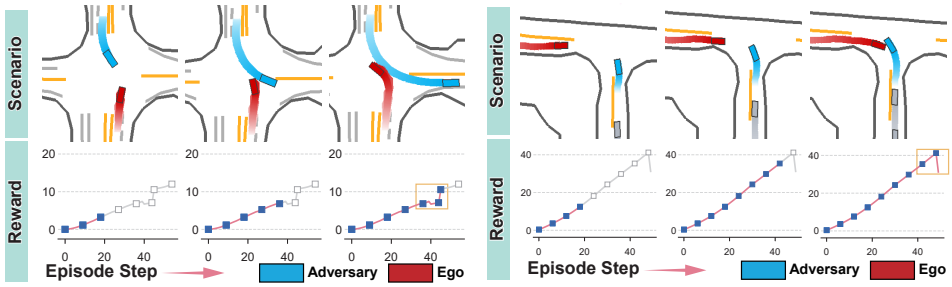

Figure 15: **Additional qualitative examples of reward-reduced adversarial scenarios from `ADV-0`.** In the first case (left), the adversary interrupts the straight-going ego, forcing it to deviate from the lane centerline to avoid a crash; this maneuver halts the ego's progress, causing the cumulative reward to stagnate. In the second case (right), a collision occurs near the end of the episode, triggering a sharp drop in the cumulative reward due to the safety penalty.

C.1 Evaluation with a Reactive Ego Policy

We evaluate candidate scoring at 250k, 500k, 750k, and 1M training steps across five random seeds. At each checkpoint, we sample 32 candidates from the current generator for each of 20 held-out contexts and execute every candidate against the current reactive ego in the simulator, producing 12,800 oracle rollouts for individual candidates. Table 19(a) compares candidate rankings and selections with these oracle returns. Pairwise accuracy uses pairs with a difference in oracle returns greater than 5; regret is the return of the selected candidate minus the lowest oracle return. Table 19(b) reports the selected candidates' closed-loop outcomes. The realism penalty is computed for each trajectory as described in Appendix D.2.2.

Table 19: Evaluation with a reactive ego policy. Results report mean $\pm$ standard deviation over five seeds. In panel (a), Top-1 and Oracle in top-4 measure whether the candidate with the lowest oracle return is selected or included among the four candidates ranked highest, respectively. Panel (b) reports the outcomes of the selected attacks.

(a) Ranking and selection against the reactive oracle

Estimator	Spearman $\rho \uparrow$	Pairwise accuracy $\uparrow$	Top-1 $\uparrow$	Oracle in top-4 $\uparrow$	Regret $\downarrow$
Replay-only passive	0.56 ± 0.05	0.69 ± 0.01	0.35 ± 0.05	0.62 ± 0.04	5.7 ± 0.8
Experience	0.64 ± 0.05	0.73 ± 0.03	0.44 ± 0.05	0.70 ± 0.04	4.4 ± 0.7
RewardModel	0.71 ± 0.04	0.76 ± 0.03	0.51 ± 0.04	0.77 ± 0.04	3.4 ± 0.6
Current proxy	0.77 ± 0.03	0.80 ± 0.03	0.61 ± 0.05	0.84 ± 0.03	2.3 ± 0.5
Reactive oracle rollout	1.00	1.00	1.00	1.00	0.0

(b) Closed-loop outcomes of selected attacks

Selection	True return $\downarrow$	Reactive crash $\uparrow$	Non-collision failure $\uparrow$	Realism penalty $\downarrow$
Random prior	43.2 ± 1.4	0.31 ± 0.04	0.07 ± 0.01	9.8 ± 0.4
Collision-only	34.6 ± 1.8	0.57 ± 0.03	0.08 ± 0.06	12.0 ± 0.6
Proxy top-1	29.9 ± 1.7	0.51 ± 0.03	0.21 ± 0.02	11.4 ± 0.5
Reactive oracle top-1	27.6 ± 1.6	0.54 ± 0.02	0.23 ± 0.03	11.5 ± 0.6

The proxy achieves 0.80 pairwise accuracy and 0.84 recall of the oracle choice among the top four candidates. The selected attacks yield an ego return of 29.9 versus 27.6 for oracle selection.

C.2 Comparisons of Preference Learning Methods

We compare five methods using the TD3 setting and five paired random seeds. Static Return-DPO constructs preference pairs once using the initial defender, receives the same generator optimization budget as `ADV-0`, and keeps the resulting generator fixed during 1M steps of ego training. AILP-Traj adapts adversarial imitation learning with preferences [Taranovic et al., 2023] to traffic trajectory generation: context X is the state, a complete traffic future Y is treated as a single action, WOMD futures provide demonstrations, and samples from the current generator provide imperfect demonstrations. The current defender's returns determine the preference ordering, and the discriminator combines demonstration matching with preference learning. The generator is updated through policy gradient over complete trajectories under the same KL constraint to $\mathcal{G}_{\text{ref}}$.

Table 20: Comparisons of preference learning methods in the TD3 setting. Results report mean performance over five random seeds.

Method	Preference mechanism	RC $\uparrow$	Crash $\downarrow$	Reward $\uparrow$	Cost $\downarrow$
Replay	None	0.672	0.344	42.66	0.600
`ADV-0` w/o IPL	Return sampling only	0.690	0.314	44.35	0.590
Static Return-DPO	Fixed offline pairs	0.718	0.378	43.14	0.597
AILP-Traj	Demo + preference AIL	0.707	0.291	45.47	0.571
`ADV-0` w/ IPL	Online pairwise	0.743	0.255	48.58	0.520

RC denotes route completion. `ADV-0` w/ IPL performs best on all four metrics (Table 20).

D Detailed Experimental Setups

In this section, we provide a comprehensive description of the experimental environment, datasets, baseline methods, model implementations, and hyperparameter configurations used in our study. Our experimental design follows the protocols established in prior works [Zhang et al., 2023, Nie et al., 2025, Stoler et al., 2025] to ensure a fair and rigorous comparison.

D.1 Environment and Dataset

Waymo Open Motion Dataset (WOMD). We utilize the WOMD as the source of real-world traffic scenarios. WOMD is a large-scale dataset containing diverse and complex urban driving environments captured in various conditions. Each scenario spans 9 seconds and is sampled at 10 Hz, capturing complex interactions between vehicles, pedestrians, and cyclists. Following the standard practices in safety-critical scenario generation, we filter and select a subset of 500 scenarios that involve interactive and complex behaviors.

MetaDrive simulator. All experiments are conducted within the MetaDrive simulator [Li et al., 2022], a lightweight and efficient platform that supports importing real-world data for closed-loop simulation. MetaDrive constructs the static map environment and replays the background traffic trajectories based on the WOMD logs. The simulation runs at a frequency of 10 Hz. The observation space consists of the ego vehicle's kinematic state (velocity, steering, heading), navigation information (relative distance and direction to references), and surrounding information (surrounding traffic, road boundaries, and road lines) encoded as a vector by a simulated 2D LiDAR with 30 lasers and a 50-meter detection range. The action space consists of low-level continuous control signals including steering, brake, and throttle. The long-tail evaluation therefore uses real-world logged scenarios in a closed-loop simulation environment, rather than on-road adversarial deployment.

Compute resources. All experiments are conducted on a computing platform equipped with two NVIDIA RTX A6000 GPUs. The computational overhead of the experiments is reported in the corresponding experimental analysis.

Reward definition. The ground-truth reward function for the ego agent is designed to balance safety and progression. It is composed of a dense driving reward and sparse terminal penalties. Formally, the reward function R_t at step t is defined as:

$$R_t = R_{\text{driving}} + 0.1v_t + R_{\text{success}} - P_{\text{collision}} - P_{\text{offroad}} \tag{51}$$

where $R_{\text{driving}} = d_t - d_{t-1}$ represents the longitudinal progress along the reference route, incentivizing the agent to move toward the destination. $R_{\text{success}} = +10$ is a sparse reward granted upon reaching the destination. Safety penalties are applied for terminal failures: $P_{\text{collision}} = 10$ for collisions with vehicles or objects, and $P_{\text{offroad}} = 10$ for driving out of the drivable road boundaries. Additionally, a small speed reward of $0.1 \times v_t$ is added to encourage movement. The episode terminates if the agent succeeds, crashes, or leaves the drivable area. For Lagrangian-based algorithms (e.g., PPO-Lag), we define a binary cost function C_t which equals 1 if a safety violation occurs and 0 otherwise.

D.2 Detailed Implementation

The overall closed-loop training procedure is summarized in Algorithm 1.

D.2.1 Proxy reward estimator

To efficiently estimate the ego's expected return $\hat{J}(Y^{\text{Adv}}, \pi_\theta)$ without executing computationally expensive closed-loop simulations during the inner loop, we implement a vectorized rule-based proxy evaluator. This module calculates the geometric interaction between a candidate adversarial trajectory Y^{Adv} and a sampled ego response Y^{Ego} from the cached history buffer. Let $Y^{\text{Ego}} = \{p_t^{\text{ego}}, \psi_t^{\text{ego}}\}_{t=1}^T$ and $Y^{\text{Adv}} = \{p_t^{\text{adv}}, \psi_t^{\text{adv}}\}_{t=1}^T$ denote the sequences of position and yaw for the ego and adversary, respectively. The proxy reward is calculated by mimicking the scheme in MetaDrive (Eq. 51):

Algorithm 1 Closed-Loop Min-Max Policy Optimization

Input: Initial ego policy π_θ, Pretrained adversary prior $\mathcal{G}_{\text{ref}}$.
Hyperparameters: Temperature τ, Reward filters δ, ξ, Learning rates α, η, Frequency N_{freq}, IPL batch size M.
Initialize: Adversary $\mathcal{G}_\psi \leftarrow \mathcal{G}_{\text{ref}}$, Ego history buffer $\mathcal{H}_{\text{ego}} \leftarrow \emptyset$, Replay buffer $\mathcal{D}$ (Off-policy) or Batch buffer $\mathcal{B}$ (On-policy).
for timestep $t = 1$ **to** $T_{\max}$ **do**
  // — Phase 1: Adversary Update (Inner Loop) —
  if $t \mod N_{\text{freq}} == 0$ **then**
    for iteration $k = 1$ **to** K_{IPL} **do**
      Sample context batch $\{X_m\}_{m=1}^{M}$.
      Generate candidates $\{Y_{m,j}^{\text{Adv}}\}_{j=1}^{K} \sim \mathcal{G}_\psi(\cdot|X_m), \forall X_m$.
      Calculate $\{\hat{J}(Y_{m,j}^{\text{Adv}}, \pi_\theta)\}_{j=1}^{K}$ using $\mathcal{H}_{\text{ego}}[X_m]$ (Eq. 5).
      Construct preference pairs $\mathcal{D}_{\text{pref}}$ based on all $\hat{J}$.
      Update ψ via IPL on $\mathcal{D}_{\text{pref}}$ (Eq. 8).
    end for
  end if
  // — Phase 2: Ego Update (Outer Loop) —
  Sample new scenario context X.
  Generate candidates $\{Y_k\} \sim \mathcal{G}_\psi(\cdot|X)$.
  Select Y^{adv} via softmax sampling (Eq. 6).
  Rollout π_θ in environment $\mathcal{P}_\psi$ to get Y^{Ego}.
  Update history: $\mathcal{H}_{\text{ego}}[X] \leftarrow \text{FIFO}(\mathcal{H}_{\text{ego}} \cup \{Y^{\text{Ego}}\})$.
  if `Algorithm` is On-Policy (e.g., PPO) **then**
    Store Y^{Ego} in Batch Buffer $\mathcal{B}$.
    if $\mathcal{B}$ is full **then**
      Update θ via Eq. 1 several steps on $\mathcal{B}$, then clear $\mathcal{B}$.
    end if
  else if `Algorithm` is Off-Policy (e.g., SAC) **then**
    Store transitions from Y^{Ego} into Replay Buffer $\mathcal{D}$.
    Sample mini-batch from $\mathcal{D}$ and update θ one step (Eq. 1).
  end if
end for

Collision detection. We approximate the vehicle geometry using Oriented Bounding Boxes (OBB) defined by the center position, yaw, length l, and width w. For each timestep t, we compute the four corner coordinates of both vehicles. We employ the Separating Axis Theorem (SAT) to determine if the two OBBs overlap. A collision penalty r_{crash} is applied if an overlap is detected at any timestep, and the evaluation terminates early.

Route progress and termination. We map the ego's position to the Frenet frame of the reference lane to obtain the longitudinal coordinate s_t and lateral deviation d_t. The evaluation terminates if: (1) *Success:* the longitudinal progress $s_t/L_{\text{total}} > 0.95$, granting a reward r_{success}. (2) *Off-road:* the lateral deviation $|d_t| > 10.0$ meters, applying a penalty r_{offroad}.

Dense reward. If no termination condition is met, a dense driving reward is accumulated based on the incremental longitudinal progress $\Delta s_t = s_t - s_{t-1}$. The total estimated return is the sum of step-wise rewards and terminal bonuses/penalties:

$$\mathcal{R}_{\text{proxy}} = \sum_{t=1}^{T_{\text{end}}} (\lambda_{\text{drive}} \cdot \Delta s_t) + \mathbb{I}_{\text{crash}} \cdot r_{\text{crash}} + \mathbb{I}_{\text{success}} \cdot r_{\text{success}} + \mathbb{I}_{\text{offroad}} \cdot r_{\text{offroad}}. \tag{52}$$

Once a terminal condition is met, the summation stops, and the cumulative value is returned. This geometric calculation is fully vectorized across the batch of candidate trajectories, allowing for rapid evaluation of potential attacks. In our experiments, we set $r_{\text{success}} = 10$, $r_{\text{crash}} = -10$, $r_{\text{offroad}} = -10$, and $\lambda_{\text{drive}} = 1.0$.

Baseline schemes. For the ablation study in Figure 8, we compare against: (1) *Experience:* We query the ego's replay buffer. If the exact scenario context exists, we use the recorded return; otherwise, we retrieve the return of the nearest neighbor scenario based on trajectory similarity. (2) *RewardModel:* We train a separate learnable reward model $\mathcal{M}(X, Y^{\text{Ego}}, Y^{\text{Adv}})$ via supervised

regression on the historical interaction dataset to predict the scalar return. (3) *GTReward:* We execute the physics engine in a parallel process to roll out the interaction between π_θ and the specific Y^{Adv} to obtain the exact return. While our results suggest that rule-based approach is effective, future work could explore integrating value function approximations (e.g., Q-networks from Actor-Critic architecture) to estimate returns for more complex reward functions.

Context-aware buffer. As noted in Section 3.2, the validity of the rule-based proxy estimator relies on the geometric consistency between the adversarial candidate Y^{Adv} and the ego response Y^{Ego}. Since different scenarios possess distinct map topologies and coordinate systems, using a global history buffer would lead to physically meaningless calculations. To address this, we implement the ego history buffer $\mathcal{H}_{\text{ego}}$ as a context-indexed dictionary, mapping each unique scenario ID to a First-In-First-Out (FIFO) queue of its historical ego trajectories: $\mathcal{H}_{\text{ego}}(X) = \{Y_i^{\text{Ego}} | X\}_{i=1}^{N}$. This ensures that the geometric interaction $R_{\text{proxy}}(Y^{\text{Ego}}, Y^{\text{Adv}})$ is always computed between trajectories residing in the same spatial environment.

A key challenge raised by the dynamic training process is handling newly sampled contexts X_{new} that have not yet been interacted with by the current ego policy, resulting in an empty buffer $\mathcal{H}_{\text{ego}}(X_{\text{new}}) = \emptyset$. To address this, we employ a warm-up rollout before the inner-loop adversarial update begins for a sampled batch of contexts: (1) We check the buffer status for each context X in the batch. (2) If $\mathcal{H}_{\text{ego}}(X_{\text{new}}) = \emptyset$, we execute a single inference rollout of the current ego policy π_θ in the simulator. Importantly, this rollout is conducted against the non-adversarial replay log. (3) The resulting trajectory $Y_{\text{ref}}^{\text{Ego}}$ represents the ego's baseline behavior in the absence of attacks and is added to $\mathcal{H}_{\text{ego}}(X)$. This mechanism ensures that the proxy estimator always has a valid reference to approximate the ego's vulnerability, even for unseen scenarios. In addition, the proxy estimator always calculates geometric interactions between trajectories sharing the same spatial topology. As the training progresses, new interactions under adversarial perturbations are generated and added to the buffer, gradually shifting the distribution in $\mathcal{H}_{\text{ego}}(X)$ from naturalistic responses to defensive responses against attacks.

Computational complexity. Let M denote the number of contexts in an update, K the candidates per context, N the cached ego responses, and T the trajectory horizon. Vectorized proxy scoring has time complexity $O(MKNT)$. The FIFO history requires $O(NT)$ memory per active context, and generated candidates require $O(MKT)$ temporary memory. We use $K = 32$ and $N = 5$, which remain fixed throughout training.

D.2.2 Evaluation and benchmark

Adversarial scenario generation. We follow the safety-critical scenario generation evaluation protocol used in Zhang et al. [2023]. For each scenario, the evaluation presented in Table 1 and Table 9 follows a two-stage process: (1) The environment is reset with a fixed seed and the ego agent first interacts with the log-replayed environment to generate a reference trajectory; (2) The adversarial generator conditions on the context and the ego's reference trajectory to generate adversarial trajectories for one selected adversary (which is marked as `Object-of-Interests` in WOMD). This trajectory is set as a fixed plan for the adversary traffic in the simulator. For `ADV-0`, we employ a worst-case sampling strategy for evaluation, setting the sampling temperature $\tau \to 0$ (Eq. 6) to select the trajectory with the highest estimated adversarial utility from $K = 32$ candidates. We test against three kinds of driving policies: a *Replay* policy that follows ground truth logs from WOMD, an *IDM* policy representing reactive rule-based drivers, and *RL agents* trained via standard PPO on replay logs. These systems under test execute their control loop in the environment. The primary metrics are Collision Rate (CR), defined as the percentage of episodes where the ego collides with the adversary, and Ego's Return (ER), the cumulative reward achieved by the ego. For the baselines adversarial generators, we utilize their official implementations but adapt them for our evaluation environment to ensure a fair comparison. All of them follow the same evaluation procedure. For the realism penalty metric in Figure 9, we adopt the trajectory-level measure from Nie et al. [2025], which discourages trajectories that are physically implausible or exhibit unnatural driving behavior.

Performance validation of learned agents. To rigorously evaluate the generalizability of the learned AD policies and compare different adversarial learning methods, we established a cross-validation protocol where each trained agent is tested against multiple distinct scenario generators.

For the evaluation, we utilized a held-out test set of 100 WOMD scenarios that were not seen during training. We compared five types of agents: those adversarially trained by `ADV-0` (with and without IPL), `CAT`, `Heuristic`, and a baseline trained solely on `Replay` data. All these agents are trained using 400 WOMD scenarios. Each agent was evaluated in five distinct environments with the held-out test set: `Replay`, `ADV-0`, `CAT`, `SAGE`, and `Heuristic`. Note that since `SAGE` introduces an additional scenario difficulty training curriculum, we exclude it for training methods. For each evaluation run, we used agents saved at the best validation checkpoints and recorded four metrics: Route Completion (RC), Crash Rate (percentage of episodes ending in collision), Reward (cumulative environmental reward), and Cost (safety violation penalty). Each method uses five independent random seeds. Unless specified otherwise, entries for an individual method, RL algorithm, and evaluation environment report the mean and standard deviation across five seeds. Table 2 averages the six algorithm means and the standard deviations across seeds for each algorithm. The Avg. row of Table 9 applies the same aggregation across the six fine-tuned generator variants, and Table 1 reports the corresponding `ADV-0` macro means. In Tables 2 and 10–15, Avg. rows average the means and standard deviations across the five evaluation environments. For the cross-validation in Table 3, we compare the adversary enhanced with IPL against the pretrained prior equipped with energy-based sampling (Eq. 6). Results are macro means across the six RL algorithms $\pm$ the mean of their standard deviations over five seeds.

Evaluation in long-tailed scenarios. Generated scenarios from adversarial models can be biased and cause a sim-to-real gap for AD policies. To ensure an unbiased evaluation of policy robustness, we construct a curated evaluation set by mining additional 500 held-out scene segments from the WOMD. These scenarios are not shown in the training set. We employ strict physical thresholds to identify and categorize rare, safety-critical events: (1) *Critical TTC*: scenarios containing frames where the TTC between the ego and any object drops below $0.4s$; (2) *Critical PET*: scenarios with a Post-Encroachment Time (PET) lower than 1.0s, indicating high-risk intersection crossings; (3) *Hard Dynamics*: scenarios involving aggressive behaviors, defined by longitudinal deceleration exceeding $-4.0m/s^2$ or absolute jerk exceeding $4.0m/s^3$; and (4) *Rare Cluster*: scenarios belonging to the two lowest-density clusters identified via K-Means clustering ($k = 10$) on trajectory features for all interacting objects, including curvature, velocity profiles, and displacement. During evaluation, we reproduce these scenarios in the simulator and utilize two traffic modes: a non-reactive mode where background vehicles follow logged trajectories, and a reactive mode where vehicles are controlled by IDM and MOBIL policies to simulate human-like interaction with the agent. In addition to safety margin and stability metrics, we also report *Near-Miss Rate*, defined as the percentage of episodes where TTC $< 1.0s$ or distance $< 0.5m$ without collision, and *RDP Violation*, which measures the frequency of violating the Responsibility-Sensitive Safety (RSS) Danger Priority safety distances.

D.2.3 Adaptability to different RL algorithms

This section elaborates on the implementation details regarding the integration of `ADV-0` with various RL algorithms, as mentioned in Section 3.1. We specifically address the synchronization mechanisms and the specialized credit assignment strategy developed for critic-free architectures to ensure stable convergence in safety-critical tasks.

RL algorithms. Our framework is designed to be algorithm-agnostic, treating the adversarial generator as a dynamic component of the environment dynamics $\mathcal{P}_\psi$. Consequently, the ego agent perceives the generated adversarial trajectories simply as state transitions, allowing `ADV-0` to support both on-policy and off-policy algorithms. In our experiments, we instantiate the ego policy using six distinct algorithms, covering both on-policy and off-policy paradigms, as well as Lagrangian variants for constrained optimization: GRPO, PPO, SAC, TD3, PPO-Lag, and SAC-Lag. The primary distinction in implementation lies in the data collection and update scheduling. For *on-policy* methods (e.g., PPO, GRPO), the training alternates strictly between the adversary and the defender. In the outer loop, we fix the adversary ψ and collect a batch of trajectories $\mathcal{B}$ using the current ego policy π_θ. The policy is updated using this batch, which is then discarded to ensure that the policy gradient is estimated using the data distribution induced by the current adversary. Conversely, for *off-policy* methods (e.g., SAC, TD3), we maintain a replay buffer $\mathcal{D}$. While the adversary ψ evolves periodically, older transitions in $\mathcal{D}$ technically become off-dynamics data. To mitigate the impact of non-stationarity, we employ a sliding window approach where the replay buffer has a limited capacity, ensuring that the value function estimation relies predominantly on recent interactions with the current or near-current adversary. The adversary update frequency N_{freq} is tuned such that the

off-policy agent has sufficient steps to adapt to the current risk distribution before the adversary shifts its strategy. This historical diversity acts as a natural form of domain randomization, preventing the ego from overfitting to specific attack patterns.

Credit assignment in critic-free methods. While critic-based methods (e.g., PPO) rely on a value function $V(s)$ to reduce variance, critic-free methods like GRPO typically utilize sequence-level outcome supervision. In the training of LLMs, it is common to assign the final reward of a completed sequence to all tokens. However, we identify that *directly applying this sequence-level supervision to AD tasks leads to severe credit assignment issues, particularly when addressing the long-tail distribution*. Our experiments revealed that applying standard outcome supervision leads to training instability and policy collapse after certain steps. This occurs because safety-critical failures (e.g., collisions) often happen at the very end ($t = T$) of a long-horizon episode. Assigning a low return to the entire trajectory incorrectly penalizes the correct driving behaviors exhibited in the early stages of the episode ($t \ll T$), resulting in high-variance gradients that disrupt the fine-tuning phase. On the other hand, implementing standard process supervision (e.g., via Monte Carlo value estimation across rollouts [Guo et al., 2025]) would require the physical simulator to support resetting to arbitrary intermediate states to perform multiple forward rollouts from every timestep. In complex high-fidelity simulators, this requirement introduces substantial engineering complexities regarding state serialization and incurs prohibitive computational overhead.

To resolve this, we propose a *step-aligned group advantage estimator* that provides dense step-level supervision without requiring a learned critic or simulator state resets. Specifically, for each update step, we sample a scenario context X and generate a group of G independent episodes ($G = 6$ in our experiments) starting from the exact same initial state (achieved via scenario seeding) but with different stochastic action realizations. Let $\tau_i = \{(s_t, a_t, r_t)\}_{t=0}^{T_i}$ denote the i-th transition in the group. We implement the following modifications to the advantage estimation:

1. **Calculating returns-to-go:** Instead of the total episode return, we calculate the discounted return-to-go $R_{t,i} = \sum_{k=t}^{T_i} \gamma^{k-t} r_{k,i}$ for each step t in transition i. This ensures that an action is only evaluated based on its consequences.
2. **Step-aligned group normalization:** We compute the advantage $A_{t,i}$ by normalizing $R_{t,i}$ against the returns of other trajectories in the same group at the same time step t, which uses the peer group as a dynamic baseline:

$$A_{t,i} = \frac{R_{t,i} - \mu_t}{\sigma_t + \epsilon}, \quad \text{where } \mu_t = \frac{1}{G}\sum_{j=1}^{G} R_{t,j}, \quad \sigma_t = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(R_{t,j} - \mu_t)^2}. \tag{53}$$

3. **Baseline padding:** Since episodes have varying lengths (e.g., due to early termination from crashes), the group size at step t could decrease. To maintain a low-variance baseline, we apply zero-padding to terminated trajectories. If trajectory j ends at $T_j < t$, we set $R_{t,j} = 0$. This ensures the baseline μ_t is always computed over the full group size G, correctly reflecting that survival yields higher future returns than termination.
4. **Advantage clipping:** To prevent single outliers such as rare failures from dominating the gradient and destabilizing the policy, we clip the calculated advantages as: $\hat{A}_{t,i} = \text{clip}(A_{t,i}, -C, C)$, where $C = 5.0$.

Then Eq. 53 is integrated into the standard GRPO loss and optimized via mini-batch gradient descent. This modification provides an efficient and low-variance gradient signal that correctly attributes failure to specific actions leading up to it, without the instability observed in outcome supervision or expensive state resetting.

D.2.4 Application to learning-based motion planning models

In this section, we detail the implementation of applying `ADV-O` to fine-tune trajectory planning models. Unlike standard end-to-end RL policies that output control actions directly, motion planners output future trajectories which are then executed by a low-level controller. This introduces challenges regarding re-planning and reward attribution. To address this, we decouple the planning evaluation from environmental execution, inspired by Chen et al. [2025]. Note that this framework can be applied to any RL algorithm, and we adopt GRPO as a demonstration.

Model architectures. To demonstrate the versatility of ADV-0, we apply it to two representative categories of state-of-the-art motion planning models:

1. **Autoregressive generation (SMART [Wu et al., 2024]):** SMART formulates motion generation as a next-token prediction task, analogous to LLMs. It discretizes vectorized map data and continuous agent trajectories into sequence tokens, utilizing a decoder-only Transformer to model spatial-temporal dependencies. By autoregressively predicting the next motion token, the model effectively captures complex multi-agent interactions and has demonstrated potential for motion generation and planning.
2. **Multimodal Scoring (PlanTF [Cheng et al., 2024]):** PlanTF is a Transformer-based imitation learning planner designed to address the shortcut learning phenomenon often observed in history-dependent models. It encodes the ego vehicle's current kinematic states, map polylines, and surrounding agents using a specialized attention-based state dropout encoder to mitigate compounding errors. This architecture allows the planner to generate robust closed-loop trajectories by focusing on the causal factors of the current scene rather than overfitting to historical observations. PlanTF has achieved state-of-the-art in several popular motion planning benchmarks.

Following the standard pretrain-then-finetune practice, we first apply imitation learning to train supervised policies by behavior cloning. To deploy these models in closed-loop simulation, we implement a wrapper policy that executes a re-planning cycle every $N = 10$ simulation steps (1.0s). At each cycle, the planner receives the current observation and generates a trajectory. A PID controller then tracks this trajectory to produce steering and acceleration commands for the underlying physics engine. We employ an advanced PID controller with a dynamic lookahead distance to ensure smooth tracking of the planned path.

Fine-tuning planners via ADV-0. Directly applying standard RL algorithms to fine-tune trajectory planners is inefficient due to the sparsity of rewards relative to the high-dimensional output space. In addition, the low-level controller executed during a re-planning horizon increases the difficulty of reward credit assignment. To address this, we decouple planning evaluation from execution and implement a state-wise reward model (SWRM) to provide dense supervision directly on the planned trajectories, following Chen et al. [2025]. We employ GRPO to fine-tune the planners. The process at each re-planning step t is as follows:

1. **Generation:** The planner generates a group of K candidate trajectories $\{T_1, \ldots, T_K\}$ conditioned on the current state s_t. For PlanTF, these are the multimodal outputs; for SMART, we sample K sequences via temperature sampling.
2. **Evaluation:** Instead of rolling out K trajectories in the simulator, we evaluate them immediately using SWRM. SWRM calculates an instant reward r_k for each T_k based on geometric and kinematic properties over a horizon $H = 2.0s$:
$$r(T_k) = R_{\text{prog}}(T_k) - P_{\text{coll}}(T_k) - P_{\text{off}}(T_k) - P_{\text{comf}}(T_k) - P_{\text{eff}}(T_k), \tag{54}$$
where the terms are the weighted reward and penalties defined in Table 17, accumulated over the 2.0s horizon with $\gamma = 0.99$. The comfort term includes acceleration and jerk penalties, and the speed term penalizes deviation from 10m/s.
3. **Optimization:** We compute the advantage for each candidate as $A_k = (r_k - \bar{r})/\sigma_r$, where $\bar{r}$ and σ_r are the mean and standard deviation of rewards within the group. The policy is updated to maximize the likelihood of high-advantage trajectories using the GRPO objective:
$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{K}\sum_{k=1}^{K} \min\left(\frac{\pi_\theta(T_k|s_t)}{\pi_{\text{old}}(T_k|s_t)}A_k, \text{clip}\left(\frac{\pi_\theta(T_k|s_t)}{\pi_{\text{old}}(T_k|s_t)}, 1-\epsilon, 1+\epsilon\right)A_k\right). \tag{55}$$
For PlanTF, we fine-tune the trajectory scoring head and decoder; for SMART, we fine-tune the token prediction logits.
4. **Execution:** The trajectory with the highest SWRM score is selected for execution by the PID controller to advance the environment to the next re-planning step.

This approach allows the planner to learn from the adversarial scenarios generated by ADV-0 by explicitly penalizing trajectories that the SWRM identifies as risky, without requiring dense environmental feedback.

Adversarial interaction. The ADV-0 adversary operates in the inner loop as described in the main paper. The adversary generator $\mathcal{G}_\psi$ creates challenging scenarios based on the planner's executed history, and is further updated via IPL. The planner is then fine-tuned via GRPO to propose safer trajectories in response to these generated risks.

D.3 Baselines

We compare ADV-0 against a comprehensive set of baselines, categorized into adversarial scenario generators (methods that create the environment) and closed-loop adversarial training frameworks (methods that train the ego policy).

Backbone model. Consistent with prior works [Zhang et al., 2023, Nie et al., 2025, Stoler et al., 2025], we employ DenseTNT [Gu et al., 2021] as the backbone motion prediction model for the adversarial generator. DenseTNT is an anchor-free, goal-based motion forecasting model capable of generating multimodal distributions of future trajectories, which is known for its high performance on the WOMD benchmark. We initialize the generator using the publicly available pretrained checkpoint, ensuring a fair comparison of the generation capabilities.

Adversarial generators. To evaluate the effectiveness of the generated scenarios, we compare ADV-0 against a comprehensive set of adversarial generation methods, covering optimization-based, learning-based, and sampling-based paradigms:

- **Heuristic** [Zhang et al., 2023]: A hand-crafted baseline that modifies the trajectory of the background vehicle to intercept the ego vehicle's path using Bezier curve fitting. It heuristically generates aggressive cut-ins or emergency braking maneuvers based on the ego vehicle's position. This serves as an oracle method representing worst-case physical attacks.
- **CAT** [Zhang et al., 2023]: A state-of-the-art sampling-based approach that generates adversarial trajectories by resampling from the DenseTNT traffic prior. It selects trajectories that maximize the posterior probability of collision with the ego vehicle's planned path.
- **KING** [Hanselmann et al., 2022]: A gradient-based approach that perturbs adversarial trajectories by backpropagating through a differentiable kinematic bicycle model to minimize the distance to the ego vehicle.
- **AdvTrajOpt** [Zhang et al., 2022]: An optimization-based approach that formulates adversarial generation as a trajectory optimization problem. It employs Projected Gradient Descent (PGD) to iteratively modify trajectory waypoints to induce collisions.
- **SEAL** [Stoler et al., 2025]: A skill-enabled adversary that combines a learned objective function with a reactive policy. It utilizes a scoring network to predict collision criticality and ego behavior deviation.
- **GOOSE** [Ransiek et al., 2024]: A goal-conditioned RL framework. The adversary is modeled as an RL agent that learns to manipulate the control points of Non-Uniform Rational B-Splines (NURBS) to construct safety-critical trajectories.
- **SAGE** [Nie et al., 2025]: A recent preference alignment framework that fine-tunes motion generation models using pairs of trajectories. It learns to balance adversariality and realism, allowing for test-time steerability via weight interpolation between adversarial and realistic expert models.

Adversarial training frameworks. To demonstrate the effectiveness of our closed-loop training pipeline, we compare ADV-0 with the following training paradigms. All methods use the same outer-loop ego policy training structure but differ in how the training environment is generated and how the inner and outer loops are integrated:

- **Replay (w/o Adversary):** The ego agent is trained purely on the original log-replay scenarios from WOMD without any adversarial modification. This serves as a lower bound for performance.
- **Heuristic training:** The ego agent is trained against the Heuristic rule-based generator described above.

- **CAT:** The state-of-the-art closed-loop training framework where the ego agent is trained against the CAT generator. The generator selects adversarial trajectories based on collision probability against the latest ego policy but does not update its policy via preference learning during the training loop.
- **ADV-0 w/o IPL:** An ablation variant of `ADV-0` where IPL is removed and the adversary is not fine-tuned. It relies solely on energy-based sampling from the pretrained backbone. This isolates the contribution of the evolving adversary.

For fair comparison, all adversarial training methods (CAT, `ADV-0`, etc.) utilize the same curriculum learning schedule regarding the frequency and intensity of adversarial encounters. Note that we explicitly exclude comparisons with standard adversarial RL frameworks, such as RARL [Pinto et al., 2017, Ma et al., 2018] or observation-based perturbation methods [Tessler et al., 2019, Zhang et al., 2020a] due to two key factors: (1) They primarily focus on perturbations to observations or state vectors. In contrast, ADV-0 targets behavioral robustness by altering the transition dynamics via trajectory generation. (2) Standard adversarial RL models the adversary as an agent with a low-dimensional action space. Our setting involves noisy real-world traffic data and the adversary outputs high-dimensional continuous trajectories. Training a standard RL adversary from scratch to generate effective trajectories in this noisy environment is computationally intractable and fails to converge in our preliminary experiments.

D.4 Hyperparameters

We provide the detailed hyperparameters used in our experiments to facilitate reproducibility. Table 21 lists the parameters for the various RL algorithms used to train the ego agent. Table 25 details the hyperparameters for the `ADV-0` framework, including the IPL fine-tuning process and the min-max training schedule.

Table 21: Hyperparameters for different RL algorithms used in the experiments.

Table 22: TD3

Hyper-parameter	**Value**
Discount Factor γ	0.99
Batch Size	256
Actor Learning Rate	3e-4
Critic Learning Rate	3e-4
Target Update τ	0.005
Policy Delay	2
Exploration Noise	0.1
Policy Noise	0.2
Noise Clip	0.5

Table 23: SAC & SAC-Lag

Hyper-parameter	**Value**
Discount Factor γ	0.99
Batch Size	256
Learning Rate	3e-4
Target Update τ	0.005
Entropy α	0.2 (Auto)
Cost Coefficient	0.5
SAC-Lag Specific	
Cost Limit	0.3
Lagrangian LR	5e-2

Table 24: PPO, PPO-Lag & GRPO

Hyper-parameter	**Value**
Discount Factor γ	0.99
Batch Size	256
Learning Rate	3e-5
Update Timestep	4096
Epochs per Update	10
Clip Ratio	0.2
GAE Lambda λ	0.95
Entropy Coefficient	0.01
Value Coefficient	0.5
Algorithm Specific	
Cost Limit (PPO-Lag)	0.4
Lagrangian LR (Lag)	5e-2
Group Size (GRPO)	6
KL Beta (GRPO)	0.001

Broader Impacts

This paper presents work whose goal is to advance the field of Machine Learning, specifically in the domain of safety-critical autonomous systems. Our research focuses on identifying system vulnerabilities and improving robustness against rare, long-tail events, which is a prerequisite for the safe large-scale deployment of autonomous vehicles. By providing a rigorous framework for generating and mitigating high-risk scenarios, this work contributes to reducing potential accidents and enhancing public trust in automation technologies. While adversarial generation techniques could theoretically be repurposed to identify vulnerabilities for malicious intent, our framework is designed as a defensive mechanism to patch these flaws before deployment. Beyond this potential dual-use concern, we do not identify specific negative ethical consequences or societal risks associated with this research, as the adversarial generation is strictly confined to simulation environments for system validation and improvement.

Table 25: Hyperparameters for ADV-O Framework and IPL Fine-tuning.

Module	**Parameter**	**Value**
Backbone (DenseTNT)	Hidden Size	128
	Sub-graph Depth	3
	Global-graph Depth	1
	Trajectory Modes (K)	32
	NMS Threshold	7.2
IPL Fine-tuning	Learning Rate	5e-6
	Temperature (τ)	0.05
	Optimizer	AdamW
	Scheduler	CosineAnnealing
	Gradient Accumulation Steps	16
	Pairs per Scenario	8
	Reward Margin	5.0
	Spatial Diversity Threshold	2.0 m
Min-Max Schedule (RARL)	Adversary Update Frequency	Every 5 Ego Updates
	Adversary Training Iterations	5 Epochs per Block
	Adversary Training Batch Size	32 Scenarios
	Adversarial Sampling Temperature	0.1
	Max Training Timesteps	1×10^6
	Opponent Trajectory Candidates	32
	Ego History Buffer Length	5
	Min Probability (Curriculum)	0.1

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and Section 2–Section 3 state the scope, assumptions, and contributions. The theoretical and empirical support is provided in Section 3.4 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The conclusion discusses limitations, including reliance on high-fidelity simulators for online RL training and the challenge of extending the approach to vision-based inputs.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: The main paper summarizes the theoretical results in Section 3.4. Complete derivations, assumptions, and proofs are provided in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: Appendix D describes the dataset, simulator, evaluation protocol, baselines, implementation details, and hyperparameters used for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code is not publicly available. Appendix D describes the dataset, implementation, training procedures, and evaluation protocol.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The paper specifies the WOMD scenario splits, MetaDrive environment, benchmark protocol, RL algorithms, and hyperparameters in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: The table captions and Appendix D.2.2 define standard deviations computed over five seeds and the macro averages of means and standard deviations. Table 19 reports mean $\pm$ standard deviation over five seeds; Table 20 reports means over five seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendix D states that the experiments are conducted on a computing platform with two NVIDIA RTX A6000 GPUs, and the paper reports the corresponding computational overhead in the experimental analysis.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work is evaluated in simulation and is framed as a defensive robustness method for autonomous systems. The broader impacts discussion describes safety motivations and possible misuse considerations.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

Justification: The broader impacts section describes positive safety implications and notes the possible dual-use risk of adversarial generation, while emphasizing the simulation-only defensive setting.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The manuscript does not release a high-risk model or scraped dataset. The method is evaluated within simulation for robustness analysis.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No].

Justification: The paper cites the main existing assets and datasets, such as WOMD and MetaDrive, but the current manuscript does not explicitly list their licenses or terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The paper introduces a method and experimental framework but does not release a new dataset, benchmark, or model asset in the current submission package.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The research does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

Justification: The research does not involve human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: LLMs are not used as an important, original, or non-standard component of the proposed method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.