World Models as Adversaries: Multi-Agent Self-Play Fine-Tuning for Robust Motion Planning

Tong Nie[1,2*] **Yuewen Mei**[1*] **Junlin He**[2] **Yihong Tang**[3,4]
Jian Sun[1†] **Wei Ma**[2]

[1]Tongji University, Shanghai, China
[2]The Hong Kong Polytechnic University, Hong Kong SAR, China
[3]McGill University, Montreal, QC, Canada [4]Mila-Quebec AI Institute, Montreal, QC, Canada

Abstract

Robust motion planning in dense traffic requires autonomous vehicles to interact in rare and safety-critical scenarios that are underrepresented in naturalistic driving data. Although adversarial training offers a feasible solution, existing methods often rely on external scenario generators, heuristic perturbations, or simulator-heavy rollouts, which makes them difficult to integrate with modern autoregressive planners. Here, we cast adversarially robust planner learning as a constrained min-max game and propose *Adversarial World Modeling (AWM)*, a theoretically grounded multi-agent self-play fine-tuning framework. Since solving the exact game is intractable, AWM introduces a principled decoupled solver. In the inner minimization, the planner's predictive world model is converted into a role-conditioned adversary that learns sparse, scene-adaptive attack coalitions via counterfactual credit assignment. In the outer maximization, the ego planner optimizes a regret-aware robust best response against the frozen AWM, utilizing tail-risk weighting and reference-anchored trust regions to improve hard-case recovery while preserving nominal driving behavior. Experiments on the nuPlan and InterPlan benchmarks demonstrate that our method generates transferable adversarial interactions and yields a robust planner that achieves competitive closed-loop performance in both nominal and highly interactive long-tail scenarios. Theoretical analysis justifies the decoupled solver and the main optimization components.

1 Introduction

Developing robust motion planners for closed-loop autonomous driving remains a central open problem [45, 24]. Recently, the leading paradigm has shifted from rule-based systems toward generative motion modeling [39, 34, 47, 54, 50]. Drawing inspiration from Large Language Models (LLMs), modern discrete architectures [47, 55, 44, 53, 49] treat map elements and motion primitives as discrete tokens, exhibiting remarkable performance when trained on large-scale human driving logs. However, these autoregressive planners are primarily optimized against nominal, naturalistic data distributions. Consequently, they remain brittle on interactive long-tail cases that are rare but safety-critical, such as aggressive cut-ins, coordinated blocking, and collision-inducing maneuvers from surrounding vehicles [12, 26]. To expose and rectify these vulnerabilities, adversarial training offers a compelling solution by actively searching hard scenarios to robustify the planner [16].

Despite the promise of adversarial training, existing frameworks are not well aligned to modern autoregressive architecture. Many existing adversarial frameworks rely on external adversary agents, heuristic scenario perturbations, or simulator-intensive rollouts [52, 30, 27, 41, 31]. These attacks are often designed around low-dimensional control policies, fixed objects of interest, or handcrafted

*Equal contribution.
†Corresponding author: Jian Sun (sunjian@tongji.edu.cn).

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

rules. As a result, they are infeasible to scale to the high-dimensional tokenized generation paradigm. Rather than forcing incompatible external attackers into this paradigm, we argue that an endogenous solution lies in the environment simulator itself. We observe that *behavioral World Models* [47, 53], which inherently learn multi-agent traffic evolution as sequential token predictions, are naturally suited to bridge this gap. Operating on the same autoregressive architecture, motion vocabulary, and scene representation as the ego planner, they can be repurposed to parameterize both naturalistic and adversarial traffic within a unified motion manifold. Consequently, formulating the environment as a learnable generative process enables the integration of adversarial interactions directly into the closed-loop training rollout, bypassing the need for a separate handcrafted attacker.

However, translating this intuition into a trainable framework poses three significant optimization challenges. First, coordinating synergistic adversaries in a multi-agent environment can introduce a combinatorial credit assignment ambiguity issue [2, 13]. Isolating the fine-grained marginal contribution of individual agents within a world model is difficult in high-dimensional action spaces. Standard strategies such as reward-sharing [33, 32] struggle to resolve *who* should initiate the attack, *how* the attack emerges, and *whether* multiple agents should coordinate. Second, simultaneous min-max optimization of two large autoregressive models is notoriously unstable [29, 9]. Updating a planning policy against a continuously shifting adversary model often leads to gradient divergence due to the non-stationary adversarial distributions. Third, unconstrained adversarial fine-tuning can degrade nominal driving performance. Without explicit behavioral regularization, a planner may become overly conservative on hard cases while sacrificing driving skills in normal traffic [14].

To bridge these gaps, we propose *Adversarial World Modeling* (AWM), a multi-agent self-play fine-tuning framework that formulates adversarially robust planner training as a constrained min-max game over a shared tokenized rollout structure. To instantiate a tractable solver, we approximate the ideal game through two principled stages. In the inner minimization, we introduce a scene-adaptive coalition learning mechanism. Rather than relying on external heuristics, we convert the benign world model into an adversary via role-conditioning. Crucially, we introduce a counterfactual credit assignment approach via role switching within the same model, which efficiently resolves the multi-agent reward ambiguity and identifies sparse, synergistic adversarial coalitions. In the outer maximization, the ego planner is optimized as a constrained robust best response to the AWM's calibrated risk distribution. We formulate a regret-aware objective that prioritizes informative tail vulnerabilities, coupled with reference-anchored trust-region regularization to preserve nominal driving performance. Our main contributions are summarized as follows:

- We present the first self-play adversarial fine-tuning paradigm that casts robust autoregressive motion planning as a constrained min-max game. With a shared rollout, it bridges principled adversarial world modeling and constrained planner adaptation without relying on external simulators.
- We design a tractable decoupled solver to instantiate the game. Leveraging role-conditioned counterfactual credit assignment, the AWM first discovers a sparse, scene-adaptive adversarial risk distribution. Utilizing this calibrated risk, the ego planner then optimizes a regret-aware objective to mitigate adversarial tail-risk while preserving nominal driving via trust-region constraints.
- We provide provable theoretical justifications and comprehensive empirical validation. Theoretical analysis bounds the local approximation error of the decoupled solver. Evaluations on nuPlan and InterPlan benchmarks demonstrate that AWM discovers transferable multi-agent attacks and yields a robust planner excelling in both naturalistic and highly interactive long-tail scenarios.

2 Preliminary and Problem Formulation

2.1 Autoregressive Motion Generation

Tokenized Autoregressive Rollout. Recent generative driving models formulate multi-agent traffic evolution as sequential next-token prediction over a learned discrete motion vocabulary [39, 34, 47, 53]. Consider a scene s with ego agent e, non-ego agents $\mathcal{V}_s = \{1, \dots, N_s\}$, and context c_s (e.g., map, history). At autoregressive step $t \in \{1, \dots, H\}$, each agent $i \in \{e\} \cup \mathcal{V}_s$ emits a discrete motion token $z_{i,t} \in \mathcal{T}$, and the joint multi-agent token set is $\boldsymbol{z}_t = \{z_{i,t}\}_{i \in \{e\} \cup \mathcal{V}_s}$. The generated prefix $\boldsymbol{z}_{<t}$ updates the scene rollout through a deterministic transition $x_{t+1} = \Gamma(x_t, \boldsymbol{z}_t)$, so the partially generated state can be written as $x_t = \Phi(c_s, \boldsymbol{z}_{<t})$. Thus, the Markovian generative process is: $p(\boldsymbol{z}_{1:H} \mid c_s) = \prod_{t=1}^{H} p(\boldsymbol{z}_t \mid x_t)$. This formulation yields an interactive rollout where each generated token immediately alters the spatiotemporal state observed by all agents in future steps.

Planning-Prediction Decomposition. Directly modeling the full joint token distribution over the entire horizon is computationally intractable and entangles ego decision-making with environmental

dynamics. Following decoupled planning-prediction paradigms [44, 50], we factorize the step-wise joint probability into individual marginals and instantiate them with two parameterized policies:

$$p(\boldsymbol{z}_{1:H} \mid c_s) = \prod_{t=1}^{H} \Big[p(z_{e,t} \mid x_t) \prod_{i \in \mathcal{V}_s} p(z_{i,t} \mid x_t) \Big] \approx \prod_{t=1}^{H} \Big[\underbrace{\pi_{\text{plan}}(z_{e,t} \mid x_t)}_{\text{ego planner}} \prod_{i \in \mathcal{V}_s} \underbrace{\pi_{\text{pred}}(z_{i,t} \mid x_t)}_{\text{non-ego world model}} \Big]. \tag{1}$$

π_{pred} is learned from driving logs and represents naturalistic traffic. Eq. 1 is the interface used throughout the paper: the ego factor represents the object optimized by the planner, while the non-ego factors represent the environment dynamics. However, optimizing solely against π_{pred} often leaves π_{plan} vulnerable to rare but critical out-of-distribution events, requiring robust adversarial modeling.

2.2 Problem Formulation: Robust Planning as a Constrained Game

We evaluate a rollout by a scalarized utility that trades off progress and safety. Let r_t denote the progress reward and $c_t^{(m)}$ denote the m-th safety cost term (e.g., collision, off-road) at step t. The step-wise utility and scene-level cumulative utility are $u_t(x_t, \boldsymbol{z}_t) = \lambda_R r_t - \sum_{m=1}^{M} \lambda_m c_t^{(m)}$ and $U_s(\pi_{\text{plan}}, \pi_{\text{pred}}) = \mathbb{E}_{\tau \sim (\pi_{\text{plan}}, \pi_{\text{pred}})}[\sum_{t=1}^{H} \gamma^{t-1} u_t(x_t, \boldsymbol{z}_t)]$, respectively. The standard planning objective is to maximize U_s under the benign prior π_{pred}. Robust planning requires the planner to perform well when some non-ego agents behave adversarially, while still preserving normal driving ability. We therefore replace the benign environment with a controllable **Adversarial World Model** (AWM) π_{awm} and formulate the ideal robust objective as a *constrained two-player zero-sum game*:

$$\begin{aligned} \max_{\pi_{\text{plan}}} \min_{\pi_{\text{awm}}, \{\mathcal{A}_s\}} \quad & \mathbb{E}_{s \sim \mathcal{D}} \left[U_s(\pi_{\text{plan}}, \pi_{\text{awm}}, \mathcal{A}_s) \right] \\ \text{s.t.} \quad & |\mathcal{A}_s| \le K_{\max}, \mathcal{A}_s \subseteq \mathcal{V}_s, \quad \forall s \in \mathcal{D} \quad \text{(Attack Budget)} \\ & \mathbb{E}_{x \sim \mathcal{D}} \left[\mathcal{D}_{\text{trust}} \left(\pi_{\text{plan}}(\cdot \mid x) \,\|\, \pi_{\text{plan}}^{\text{nom}}(\cdot \mid x) \right) \right] \le \epsilon \quad \text{(Trust Region Constraint)}, \end{aligned} \tag{2}$$

where the inner player minimizes U_s by optimizing the generative policy π_{awm}, and the outer player maximizes its expected utility against the risk distribution induced by π_{awm}. We further introduce two constraints: (1) a budgeted subset of active adversaries $\mathcal{A}_s$ to ensure sparse and reasonable attack; (2) a trust region around a nominal reference planner $\pi_{\text{plan}}^{\text{nom}}$ to prevent behavioral degradation.

However, Eq. 2 is not directly tractable. First, the inner step involves a combinatorial search over the high-dimensional multi-agent action space, suffering from severe credit assignment ambiguities [2]. Second, simultaneous non-convex min-max updates are notoriously unstable [29, 9], as the planner struggles to converge against a non-stationary distribution. Third, directly enforcing behavioral constraints under adversarial rollouts can yield an averaged or overly conservative policy with degraded normal performance. To overcome these issues, we introduce a principled solver in Section 3 as a decoupled approximation.

3 Self-Play Fine-tuning with Adversarial World Models

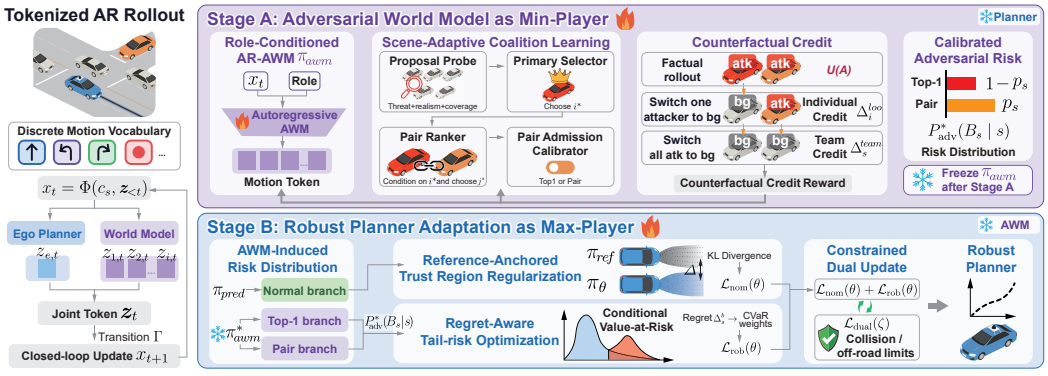


Figure 1: Decoupled self-play fine-tuning with AWM. The full pipeline is described in Algorithm 1.

We instantiate the idealized game in Eq. 2 as a decoupled self-play fine-tuning framework over the shared autoregressive factorization in Eq. 1. We use a single decoupled self-play round, $A_1 \to B_1$: Stage A learns the traffic opponent against the reference planner (Section 3.2), and Stage B fine-tunes

the planner against the resulting frozen opponent (Section 3.3). Thus, adversarial training data are generated by interactive joint rollouts within the same token space and transition dynamics, rather than by external heuristic perturbations or a separate simulator. Appendix B gives the implementation details and Appendix C provides the theoretical analysis. Figure 1 summarizes the two-stage pipeline.

3.1 Decoupled Self-Play Approximation of the Constrained Game

Stage A: Adversarial Coalition Learning. We first anchor the planner to a frozen reference $\pi_{\text{plan}}^{\text{nom}}$ and approximate the inner minimization over the non-ego factors $\prod \pi_{\text{pred}}(\cdot)$. Instead of enumerating all subsets $\mathcal{A}_s$, AWM learns a role-conditioned policy and a scene-adaptive coalition distribution:

$$\pi_{\text{awm}}^*, \mathcal{P}_{\text{adv}}^* = \arg \min_{\pi_{\text{awm}},\, \{\mathcal{A}_s\}} \mathbb{E}_{s\sim\mathcal{D}} \left[U_s(\pi_{\text{plan}}^{\text{nom}}, \pi_{\text{awm}}, \mathcal{A}_s) \right] \quad \text{s.t.} \quad |\mathcal{A}_s| \le K_{\max}. \tag{3}$$

This stage yields a frozen adversarial policy π_{awm}^* with a calibrated risk distribution $\mathcal{P}_{\text{adv}}^*$ over sparse multi-agent attacks. Proposition C.2 bounds the local error incurred by freezing the adversary, and Corollary C.3 analyzes the local gap in robust value associated with the update anchored to the reference policy.

Stage B: Constrained Robust Best Response. With $(\pi_{\text{awm}}^*, \mathcal{P}_{\text{adv}}^*)$ frozen, the planner solves the outer maximization within a reference trust region. Rather than optimizing raw adversarial utility, we compare the current planner with a frozen reference planner under the same induced attack. This yields the reference-relative regret over $\mathcal{P}_{\text{adv}}^*$. The planner's objective is given by:

$$\begin{aligned} \pi_{\text{plan}}^* = \arg\max_{\pi_{\text{plan}}} \quad & -\mathbb{E}_{s\sim\mathcal{D}} \left[\mathfrak{R}_\tau^{A_s\sim\mathcal{P}_{\text{adv}}^*(\cdot|s)} \left(\Delta_s^{\text{adv}}(\pi_{\text{plan}}, \mathcal{A}_s) \right) \right] \\ \text{s.t.} \quad & \mathcal{D}_{\text{ref}}\left(\pi_{\text{plan}}, \pi_{\text{ref}}\right) \le \epsilon, \quad \mathcal{V}_{\text{safe}}\left(\pi_{\text{plan}}; \pi_{\text{awm}}^*, \mathcal{P}_{\text{adv}}^*\right) \le \kappa, \end{aligned} \tag{4}$$

where the reference π_{ref} is initialized from $\pi_{\text{plan}}^{\text{nom}}$, and $\Delta_s^{\text{adv}}(\pi_{\text{plan}}, A_s) = U_s(\pi_{\text{ref}}, \pi_{\text{awm}}^*, A_s) - U_s(\pi_{\text{plan}}, \pi_{\text{awm}}^*, A_s)$. Here, $\mathfrak{R}_\tau$ denotes a generic tail-risk functional over the regret distribution. The abstract constraint $\mathcal{D}_{\text{ref}}$ enforces reference-anchored retention, while $\mathcal{V}_{\text{safe}}$ represents admissible safety-violation constraints. Importantly, Lemma C.1 justifies this local update by showing finite-horizon utility is smooth when the planner remains within a token-level trust region.

3.2 Adversarial World Modeling via Multi-Agent Coalition Learning

This section instantiates the inner minimization (Eq. 3). Directly optimizing an unstructured set of attackers is computationally prohibitive and suffers from credit assignment issues. Moreover, effective attacks are sparse and scene-dependent: a single agent may be sufficient in some scenes, whereas in others the worst failure requires coordinated behavior. AWM converts the inner search into a sequence: role conditioning defines *how* an agent attacks, coalition learning decides *who* should attack, counterfactual credit trains the attackers, and calibration decides *whether* a paired attacker should be admitted. This amortizes the worst coalition while remaining on the same motion manifold.

Role-Conditioned Adversarial Generation. To keep adversarial rollouts within the same autoregressive structure, AWM changes the behavioral distribution of selected non-ego agents without changing the token vocabulary or transition model. Each non-ego agent receives a latent role assignment $\rho_i \in \{\texttt{bg}, \texttt{atk}\}$, where $\rho_i = \texttt{atk}$ iff $i \in \mathcal{A}_s$. At each generation step t, for each non-ego agent $i \in \mathcal{V}_s$ the actions are generated by the shared AWM:

$$z_{i,t} \sim \pi_{\text{awm}}(\cdot \mid x_t, \rho_i), \quad \forall i \in V_s. \tag{5}$$

The ego token $z_{e,t}$ is still sampled from π_{plan}. The role conditioning ρ_i is explicitly injected into both internal agent and decoder representations. Let h_i be the contextual feature of agent i and $k_{i,t}$ be its decoding state. It modifies output behavioral logits $\ell_{i,t}$ in the generation process via:

$$\tilde{h}_i = h_i + E_{\text{agent}}(\rho_i), \quad \ell_{i,t} = D(k_{i,t} + E_{\text{dec}}(\rho_i)), \tag{6}$$

where E_{agent} and E_{dec} are learnable embeddings. This makes role switching a behavioral intervention rather than a loss reweighting trick. It also enables counterfactual evaluations within the same AWM, which is later used for credit assignment. Architectural details are provided in Appendix B.1.

Sparse Coalition Learning. Given the conditioned generator, the next problem is to determine the sparse coalition A_s. AWM avoids exhaustive subset search through a structured sequence: *proposal probing, primary selection, conditional pairing, and pair admission*. A background probe first rolls out all agents with $\rho_i = \texttt{bg}$ and scores each agent by threat, realism, and residual coverage (the exact

probe signals and coverage metrics are given in Appendix B.2.1). The resulting sparse proposal set P_s removes benign agents while retaining complementary attack potentials. A learned primary selector then chooses $i_s^* = \arg\max_{i \in P_s} f_{\text{pri}}(x_i)$, where x_i combines probe features and proposal evidence. Conditional on i_s^* acting as `atk`, a pair ranker scores relational features $\phi(i_s^*, j)$ and selects $\hat{j}_s = \arg\max_{j \in P_s \setminus \{i_s^*\}} f_{\text{pair}}(\phi(i_s^*, j))$. This converts the combinatorial subset search into a tractable sequential approximation. Corollary C.4 gives an interpretation of the approximation error: the coverage gap decreases when this pipeline captures more of the adversarial support that matters under the full worst-case adversary. Details for f_{pri} and f_{pair} are in Appendix B.2.2.

Counterfactual Credit Assignment. The coalition selected above is trained from scene-level utility, but the optimization signal in Eq. 3 needs to be attributed to individual attackers and to their joint behavior. Directly isolating and evaluating member-level contribution to reducing the ego utility is challenging, as the credit space scales with the joint action tree. To achieve fine-grained attribution without enumerating this exploding space, we leverage the role-conditioned nature of the AWM to perform *counterfactual self-play* credit assignment. Let $\mathcal{A}_s^{\text{fact}}$ be the factual active coalition in a training rollout, and write $U_s(\mathcal{A})$ for the planner utility under $\mathcal{A}$. For each active attacker $i \in \mathcal{A}_s^{\text{fact}}$, we compute a leave-one-out marginal contribution against counterfactual scenarios:

$$\Delta_i^{\text{loo}} = U_s(\mathcal{A}_s^{\text{fact}} \setminus \{i\}) - U_s(\mathcal{A}_s^{\text{fact}}). \tag{7}$$

A partner might be weak in isolation but critical when synergizing with the primary attacker. To preserve non-additive cooperation, we also define the counterfactual team contribution:

$$\Delta_s^{\text{team}} = U_s(\text{bg}(\mathcal{A}_s^{\text{fact}})) - U_s(\mathcal{A}_s^{\text{fact}}), \tag{8}$$

where $\text{bg}(A_s^{\text{fact}})$ denotes switches the roles of all attackers to $\rho = \texttt{bg}$. Then the attacker's reward is

$$\Delta_i^{\text{hyb}} = \omega_{\text{loo}} \Delta_i^{\text{loo}} + \omega_{\text{team}} \Delta_s^{\text{team}}. \tag{9}$$

Proposition C.6 shows that this hybrid counterfactual reward is gradient-aligned with the adversarial coalition objective, while Lemma C.7 interprets the role-switched terms as action-independent baselines for variance reduction. The full AWM training objective is given in Appendix B.2.3.

Scene-Adaptive Calibration. The final output of Stage-A decides not only who the best pair is, but also whether the pair is worth using in the current scene. To finalize the opponent distribution, we introduce a decisive scene-dependent calibrator. Given the primary attacker i_s^* and candidate partner $\hat{j}_s$, we construct a scene-level representation $\psi_s = \text{SceneEnc}(i_s^*, \hat{j}_s)$ and define the pair gain $G_s^{\text{pair}} = U_s(\{i_s^*\}) - U_s(\{i_s^*, \hat{j}_s\})$. A binary calibrator $q_\eta(\psi_s) = \sigma(f_\eta^{\text{cal}}(\psi_s))$ is trained with cross-entropy on $\mathbb{I}(G_s^{\text{pair}} > 0)$ to predict the probability p_s of a positive marginal improvement. At inference, the pair is admitted only when p_s exceeds a threshold. Proposition C.5 shows that this thresholding rule is the optimal binary admission policy under calibrated pair-gain probabilities. Thus, the coalition size is scene-adaptive, preventing attack quality degradation or redundancy.

3.3 Regret-Aware Constrained Robust Planner Optimization

With the adversary (π_{awm}^* and q_η) frozen, Stage B instantiates the constrained best-response problem in Eq. 4. Exposing the planner to generated attacks via standard adversarial training often leads to performance degradation due to two issues: (1) it forces the planner to adopt overly conservative behaviors while sacrificing nominal driving; and (2) it is risk-insensitive and optimizes an average adversarial loss, diluting the gradients of rare but critical failures. To overcome these issues, we first convert the learned AWM into a conditional risk distribution. Then the planner optimizes adversarial tail-risk using a regret-aware robust objective over the induced distribution. To enforce the behavioral regularization in a tractable way, we optimize a reference-anchored dual surrogate: a trajectory-level constraint and a normal-preserving penalty. Safety constraints are handled by dual variables.

AWM-Induced Adversarial Risk Distribution. For each scene, the frozen AWM defines three rollout branches. The normal branch uses the benign world model π_{pred}. The top-1 branch assigns $\rho_{i_s^*} = \texttt{atk}$ and keeps all other non-ego agents as background. The pair branch assigns $\rho_i = \texttt{atk}$ for $i \in \{i_s^*, \hat{j}_s\}$. Let $\alpha_s^{\text{cal}} = f_\eta^{\text{cal}}(\psi_s)$ be the calibrator logit and $p_s = \sigma(\alpha_s^{\text{cal}}/\tau_{\text{risk}})$ be the temperature-scaled probability of admitting the pair branch. AWM induces the discrete branch distribution:

$$P(B_s = \texttt{top1}) = 1 - p_s, \qquad P(B_s = \texttt{pair}) = p_s. \tag{10}$$

This converts the coalition output from Stage A into the calibrated opponent distribution $\mathcal{P}_{\text{adv}}^*$ used by Stage B. Instead of an unstructured attack buffer, the planner replays each scene and optimizes against a scene-conditional self-play distribution. Branch construction details are in Appendix B.3.1.

Reference-Anchored Trust Region Regularization. Robust adaptation should not degrade normal driving. We instantiate the reference constraint in Eq. 4 on the normal branch, where the environment remains π_{pred}. Let $U_s^{\text{norm}}(\pi_\theta)$ denote the normal scene-level utility. We use two complementary retention surrogates. First, to preserve nominal driving performance, we define the normal utility margin $m_s^{\text{norm}}(\theta) = U_s^{\text{norm}}(\pi_\theta) - U_s^{\text{norm}}(\pi_{\text{ref}})$ and impose an asymmetric degradation penalty: $\psi_s^{\text{norm}}(\theta) = [-\epsilon_{\text{norm}} - m_s^{\text{norm}}(\theta)]_+^2$. Second, to prevent large behavioral drift even when the scalar utility remains acceptable, we also regularize the token distribution along the normal rollout as: $\mathcal{D}_{\text{KL}}^{\text{norm}}(s;\theta) = \frac{1}{H}\sum_{t=1}^{H} D_{\text{KL}}\left(\pi_\theta(\cdot \mid x_{s,t}^{\text{norm}}) \,\middle\|\, \pi_{\text{ref}}(\cdot \mid x_{s,t}^{\text{norm}})\right)$. The normal-branch objective is:

$$\mathcal{L}_{\text{nom}}(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\lambda_{\text{pg}}\ell_s^{\text{norm}}(\theta) + \lambda_{\text{kl}}\mathcal{D}_{\text{KL}}^{\text{norm}}(s;\theta) + \lambda_{\text{margin}}\psi_s^{\text{norm}}(\theta)\right]. \tag{11}$$

Here ℓ_s^{norm} is the normal-branch policy-gradient loss. These terms provide a tractable surrogate for the trust-region constraint. Proposition C.12 proves that they imply a nominal-performance bound, ensuring a conservative update to obtain robustness. Implementation details are in Appendix B.3.1.

Adversarial Tail-Risk Optimization. Absolute adversarial utility can be dominated by intrinsic scene difficulty rather than the planner's vulnerabilities. We therefore propose to optimize reference-relative regret under each adversarial branch to realize the robust objective in Eq. 4:

$$\Delta_s^b(\theta) = U_s^b(\pi_{\text{ref}}) - U_s^b(\pi_\theta), \quad \text{for } b \in \{\text{top1}, \text{pair}\}. \tag{12}$$

A positive regret indicates that π_θ underperforms the baseline under the specific attack b. For each scene, CVaR operates on the distribution of regret relative to the reference planner, with two support points induced by the top-1 and pair attack modes and probabilities $1 - p_s$ and p_s:

$$\mathcal{D}_s^\theta = \left\{\left(\Delta_s^{\text{top1}}(\theta), 1-p_s\right), \left(\Delta_s^{\text{pair}}(\theta), p_s\right)\right\}. \tag{13}$$

Proposition C.9 shows that regret preserves the branchwise gradient direction of utility maximization while removing intrinsic difficulty variation. Standard expected loss would dilute the gradients of these rare but critical attacks by their low probabilities. Instead, we apply a tail-focused risk operator $\mathfrak{R}_\tau$ at tail level $\tau \in (0, 1]$ and instantiate $\mathfrak{R}_\tau$ by analytically solving the Conditional Value-at-Risk (CVaR) allocation [36] over the induced two-point regret distribution $\mathcal{D}_s^\theta$. Since the distribution has only two support points, Proposition C.11 gives closed-form weights $w_s^{\text{top1}}(\tau)$ and $w_s^{\text{pair}}(\tau)$. These weights amplify low-probability but high-regret pair attacks when they occupy the adversarial tail. Let $\ell_s^{\texttt{top1}}(\theta)$ and $\ell_s^{\texttt{pair}}(\theta)$ be the policy gradient losses, the adversarial tail-risk objective is:

$$\mathcal{L}_{\text{rob}}(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[w_s^{\text{top1}}(\tau)\ell_s^{\text{top1}}(\theta) + w_s^{\text{pair}}(\tau)\ell_s^{\text{pair}}(\theta)\right]. \tag{14}$$

Implementations are detailed in Appendix B.3.2. Regret-CVaR allocates tail weights to the top-1 and pair modes according to their regrets and admission probabilities.

Safety-constrained dual update. Finally, to prevent the planner from exploiting the objective by violating traffic rules (e.g., evading an attacker by driving off-road), we enforce explicit safety constraints in Eq. 4. Let $v_{s,m}^b \in [0, 1]$ be the m-th safety violation indicator under branch b. The expected violation rate is $\bar{v}_{s,m} = (1 - p_s)v_{s,m}^{\texttt{top1}} + p_s v_{s,m}^{\texttt{pair}}$. Given admissible thresholds κ_m, we update Lagrangian dual variables ζ to dynamically adjust the cost multipliers $\lambda_m(\zeta)$: $\mathcal{L}_{\text{dual}}(\zeta) = \sum_{m=1}^{M}\lambda_m(\zeta)\left(\mathbb{E}_{s\sim\mathcal{D}}[\bar{v}_{s,m}] - \kappa_m\right)$. Training alternates between minimizing $\mathcal{L}_{\text{planner}}(\theta) = \mathcal{L}_{\text{nom}}(\theta) + \mathcal{L}_{\text{rob}}(\theta)$ over planner parameters and maximizing $\mathcal{L}_{\text{dual}}$ over ζ. Proposition C.13 shows that the softmax-parameterized dual behaves as a smooth maximum-violation penalty, concentrating weight on the most violated safety channel while keeping the reward scale bounded. The multiplier parameterization is provided in Appendix B.3.3.

4 Experiments

We organize the empirical study around three questions. First, does adversarial self-play improve the final closed-loop planner on standard and interaction-heavy benchmarks? Second, does the learned AWM expose transferable planner vulnerabilities when it is used as a simulator-side traffic model? Third, which components of the two-stage optimization account for the observed gains? Experimental protocols, metric definitions, and implementation details are provided in Appendix D.

4.1 Closed-Loop Planning and Adversarial Testing

Closed-Loop Planning Performance. We first evaluate whether the robust best-response update improves ordinary closed-loop planning performance. Table 1 reports benchmark results on `nuPlan` [3], `InterPlan` [18], and `InterPlan-LongTail` (Appendix D.3). Relative to the Plan-R1

Table 1: Closed-loop planning results on `nuPlan` and `InterPlan` benchmarks. Higher is better. NR/R: non-reactive/reactive mode. Among learning-based planners, the best result is in **bold** and the second best result is underlined. is the reference baseline (Plan-R1), indicates an improvement over the reference, while indicates no improvement or degradation.

Type	Planner	Val14		Test14-hard		Test14-random		InterPlan		
		NR ↑	R ↑	NR ↑	R ↑	NR ↑	R ↑	Full ↑	InterPlan10 ↑	InterPlanLT ↑
Expert	Log-Replay	93.53	80.32	85.96	68.80	94.03	75.86	14.76	–	–
Rule	IDM / IDM+Mobil	75.60	77.33	56.15	62.26	70.39	72.42	47.07	31.28	38.49
	PDM-Closed* [10]	92.84	92.12	65.08	75.19	90.05	91.64	69.64	41.81	45.93
	PDM-Hybrid* [10]	92.77	92.11	65.99	76.07	90.10	91.28	41.61	26.12	27.79
Learning	UrbanDriver [37]	68.57	64.11	50.40	49.95	51.83	67.15	5.56	3.62	7.75
	PDM-Open [10]	53.53	54.24	33.51	35.83	52.81	57.23	26.22	24.67	17.51
	GameFormer [21]	13.32	8.69	7.08	6.69	11.36	9.31	11.07	9.12	8.13
	PlanTF [8]	84.27	76.95	69.70	61.61	85.62	79.58	47.72	33.00	32.38
	PLUTO [7]	88.89	78.11	70.03	59.74	89.90	78.62	57.74	42.87	38.46
	Diff. Planner$_{\text{DPM}}$ [54]	89.87	82.80	75.99	69.22	89.19	82.93	50.07	24.75	29.96
	Diff. Planner$_{\text{DDIM}}$ [54]	89.81	82.94	76.01	68.18	89.14	82.63	49.86	24.11	28.98
	Flow Planner [43]	**90.43**	83.31	76.47	70.42	89.88	82.93	**61.82**	35.42	31.06
	PlannerRFT [25]	89.96	84.46	77.16	72.21	90.76	85.80	–	–	–
	Plan-R1 [44]	88.98	87.69	77.45	77.20	**91.23**	90.04	56.64	40.95	34.89
Ours	AWM-Planner	89.72 (+0.74)	**88.44** (+0.75)	**80.05** (+2.60)	**78.48** (+1.28)	91.22 (-0.01)	**90.71** (+0.67)	58.74 (+2.10)	**45.92** (+4.97)	**40.70** (+5.81)

reference, AWM-Planner (ours) improves five of the six `nuPlan` settings, with the largest gains on Test14-hard NR and R ($+2.60$ and $+1.28$). The Test14-random NR score is essentially unchanged (-0.01), while Test14-random R increases by $+0.67$, indicating that the hard-case improvement is not obtained by broadly sacrificing normal driving behavior. The interaction benchmarks provide a more targeted assessment of the intended mechanism. AWM-Planner improves Plan-R1 by $+2.10$ on full `InterPlan`, $+4.97$ on InterPlan10, and $+5.81$ on `InterPlan-LongTail`. The per-template breakdown in Table 6 shows improvements on seven of eight designed templates, with the strongest gains in close straight-driving and medium-density yielding cases. These results suggest that AWM primarily improves recoverable interaction failures exposed by structured adversarial traffic.

AWM as a Transferable Adversarial Simulator. We next test whether the learned adversary captures planner-agnostic interaction stress. Table 2 deploys AWM as the non-ego sim-agent traffic model for seven planners. AWM reduces the sim-agent closed-loop score for every planner, decreasing the mean score from 70.24 to 67.46. Although Plan-R1 has the largest drop, rule-based, optimization-based, diffusion-based, and transformer-based planners are all affected, indicating that the adversarial behavior generalizes beyond the planner used in Stage A. The same evaluation also separates adversarial strength from planner robustness after fine-tuning. AWM-Planner starts from a higher score under the normal host than Plan-R1 (77.67 vs. 75.86) and suffers a smaller AWM-induced degradation (-3.14 vs. -4.15). The submetric pattern is consistent with controlled interaction stress: the largest changes occur in progress and TTC, whereas drivable-area, speed-limit, and driving-direction scores remain nearly unchanged. The simulator diagnostics in Table 7 further show that AWM utilizes nearly the full two-agent budget while remaining constrained by feasibility gates, indicating that the score drop is not a result of an unconstrained simulator collapse.

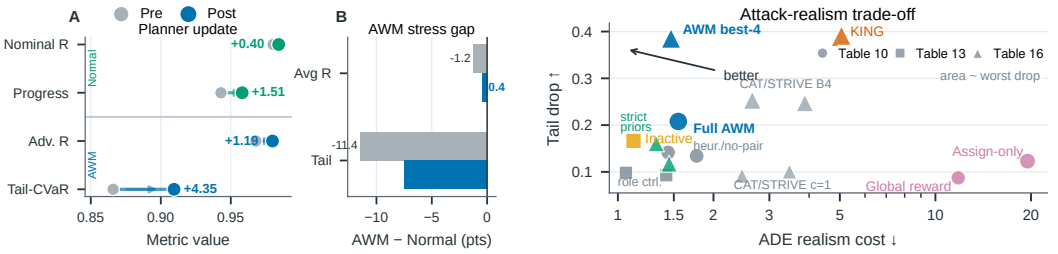
4.2 Algorithmic Analysis and Discussion

Planner-World Cross-Validation. Closed-loop scores couple the planner policy with the simulator dynamics. To isolate the effect of planner adaptation, we evaluate pre- and post-training planners under matched normal and adversarial world branches. Figure 2(a) summarizes this matrix, and Table 8 provides the full values. The post-trained planner improves nominal reward and progress under the normal world, while the larger gains occur under the AWM world: adversarial reward increases by 0.0119, and Tail-CVaR increases from 0.8660 to 0.9095. This asymmetry matches the objective of Stage B, which should lift the adversarial lower tail while preserving nominal rollout quality. Table 9 further decomposes the AWM host into Top-1, Pair, and Adaptive branches. Pair and

Table 2: Cross-planner Test14-hard sim-agent evaluation. NORMAL is the learned non-adversarial sim-agent baseline, while a held-out AWM uses the adversarial host as the simulator-side non-ego policy. Thus, the R-Score is an AWM/Normal sim-agent closed-loop score, not an ordinary planner closed-loop score. Δ_R is computed as $R_{\text{AWM}} - R_{\text{NORMAL}}$, where negative values indicate the R-Score drop induced by the adversarial host. Prog., Coll., TTC, Driv., Speed, and Dir. denote progress, collision, time-to-collision, drivable-area, speed-limit, and driving-direction submetrics. mark the corresponding NORMAL reference R-Score, and mark the AWM-induced R-Score change.

Planner	World	R-Score ↑	Δ_R	Prog. ↑	Coll. ↑	TTC ↑	Driv. ↑	Comfort ↑	Speed ↑	Dir. ↑
Diffusion Planner [54]	Normal	70.81	–	88.06	84.71	73.33	95.41	86.20	96.50	98.96
	AWM	68.10	−2.71	84.25	83.64	69.49	94.85	84.56	96.48	98.90
IDM+Mobil	Normal	55.64	–	67.55	80.51	65.44	86.40	88.97	96.93	98.53
	AWM	53.21	−2.43	62.96	80.15	62.13	86.03	88.24	96.97	98.53
PDM-Closed [10]	Normal	65.45	–	73.97	87.68	70.96	94.85	83.09	99.54	99.08
	AWM	64.20	−1.25	70.37	88.97	69.49	94.85	83.46	99.54	99.08
PLUTO [7]	Normal	76.63	–	81.64	92.50	85.39	96.57	89.02	97.04	96.44
	AWM	73.60	−3.03	75.28	92.10	81.99	96.69	87.50	96.99	96.69
PlanTF [8]	Normal	69.62	–	83.52	86.21	78.68	94.49	89.34	97.03	96.69
	AWM	66.89	−2.73	79.20	86.95	74.63	94.12	83.09	97.06	96.69
Plan-R1 [44]	Normal	75.86	–	87.93	89.15	79.41	95.22	99.26	99.07	96.69
	AWM	71.71	−4.15	82.83	88.24	73.16	94.85	98.16	99.08	97.06
AWM-Planner (Ours)	Normal	**77.67**	–	83.60	93.38	80.51	96.69	95.22	99.47	98.90
	AWM	**74.53**	−3.14	79.07	91.54	77.94	96.69	95.96	99.47	98.90

Adaptive branches produce larger lower-tail stress than the single-agent Top-1 branch, and the adaptive branch remains less aggressive than forced Pair because the calibrator admits two-agent coalitions only when they are predicted to be useful. Thus, the planner gains in Table 1 indicate a targeted policy adaptation that specifically improves the response to structured adversarial interactions.



(a) Planner-world cross-validation **(b)** Attack and realism trade-off

Figure 2: (a) Cross-validation shows that robust fine-tuning improves the adversarial tail while preserving nominal rollout quality (full values in Table 8). (b) The attack and realism plot summarizes the AWM ablations in Tables 10, 11, 12, 13 and the comparison in Table 16. AWM achieves a balanced operating point, whereas aggressive search increases attack strength at a clear realism cost.

Mechanistic Analysis of AWM. The AWM is useful for planner training only if its adversarial rollouts are both informative and plausible. The AWM-side ablations in Tables 10–13 support this requirement. Role-conditioned counterfactual credit produces substantially stronger lower-tail degradation than global reward broadcast or assignment-only variants, while avoiding the large ADE/FDE collapse observed in coarse credit baselines. This indicates that role switching provides a more reliable training signal for coordinated adversarial behavior than assigning the same scene-level reward to all agents. The coalition ablations show a complementary effect. Forced $K{=}2$ attacks can increase raw stress, but they move the generator toward a less realistic mode. Instead, the adaptive calibrated AWM preserves most of the controlled attack effect while keeping pair usage sparse and well calibrated. Figure 2(b) gives a compact view of this attack and realism trade-off (full results in Table 16). Matched best-4 search and the KING-style [19] aggressive bound can produce

larger raw disruption, but they also incur larger realism costs and background-collision proxies. In contrast, single-candidate CAT* [52] and STRIVE* [35] remain closer to ordinary AWM generation but produce weaker tail stress. AWM therefore provides the intended middle regime: sufficient adversarial stress for robust training without becoming a purely destructive test-time attacker.

Table 3: Training paradigm comparison. Open-loop rewards are on the planner's rollout utility scale; CL scores are ordinary nuPlan closed-loop scores. Relative GPU-hours are normalized to the decoupled two-stage pipeline, and plateau updates denote thousands of planner update steps until validation Tail-CVaR stops improving. Stable seeds count runs that finish without divergence or failure, and HARD STD. is the seed-wise standard deviation of Test14-hard NR score.

Training paradigm	Open-loop rollout			Planner CL		Computation and stability			
	Nom. R ↑	Adv. R ↑	Tail ↑	Hard NR ↑	Random R ↑	Rel. GPU-h ↓	Plateau k ↓	Stable seeds ↑	Hard std. ↓
Planner-only refinement	0.9804	0.9685	0.8467	77.45	90.04	0.62	5.0	3/3	0.55
Simultaneous alternating	0.9808	0.9726	0.8663	78.30	89.30	2.35	18.5	2/3	1.65
Alternating warmup + fixed AWM	0.9812	0.9744	0.8810	79.30	90.20	1.45	12.0	3/3	0.85
Decoupled Stage-A/B (ours)	**0.9818**	**0.9756**	**0.8919**	**80.05**	**90.71**	1.00	9.0	3/3	**0.40**

Planner Adaptation Ablations. We finally examine whether the planner's gain depends on the proposed decoupled optimization and regret-aware objective. Table 3 compares planner-only refinement, simultaneous alternating self-play, alternating warmup followed by fixed refinement, and the proposed decoupled pipeline. Our framework achieves the best adversarial Tail-CVaR, the highest closed-loop scores, and the lowest variability. Simultaneous alternating updates improve over planner-only refinement, but they require substantially more computation and exhibit weaker stability. This supports the design of first learning an opponent and then optimizing a constrained robust best response against the frozen distribution. Table 4 isolates the objective of Stage-B. Expected-risk and no-CVaR variants preserve competitive averages, but they lose adversarial Tail-CVaR and Test14-hard NR score, which is consistent with the need to emphasize rare severe risks. Removing normal retention strengthens adversarial reward but degrades nominal progress and score. Additional opponent sensitivity and hyperparameter results in Tables 14 and 15, with Figure 5, show that the planner is stable under moderate host changes while following the expected robustness and nominal performance trade-off.

Table 4: Planner objective ablation with a robustness and nominal performance trade-off. Offline rewards are the planner's rollout utility; CL scores and metrics are ordinary nuPlan Test14 simulation.

Variant	Open-loop rollout				CL: Test14-hard NR			CL: Test14-random R		
	Nom. R	Nom. prog.	Adv. R	Tail	Score	Prog.	TTC	Score	Coll.	TTC
Full	**0.9818**	0.9545	0.9756	0.8919	80.05	90.16	**81.99**	**90.71**	**98.08**	**95.40**
Expected only	0.9815	**0.9579**	0.9681	0.8640	76.80	86.20	78.36	90.60	97.88	95.20
w/o CVaR	0.9816	0.9530	0.9670	0.8501	77.40	87.00	78.50	90.55	97.90	95.15
w/o normal retention	0.9790	0.9390	**0.9762**	**0.8950**	**80.20**	**91.00**	81.20	88.60	96.40	92.05

4.3 Case Studies

The examples in Figures 3 and 4 illustrate the two mechanisms quantified above. In the recovery cases, AWM-Planner reacts earlier than the original Plan-R1 and avoids collision or TTC failures in the same adversarial scenes. Additional recovery cases, `InterPlan-LongTail` examples, and $K{=}2$ cooperative attacks are shown in Figures 7-9. The simulator-side example in Figure 4 shows how the learned adversary creates stress without destroying the scene. Under the normal host, the ego can proceed through the local interaction. Under the $K{=}2$ attack, two selected non-ego agents jointly constrain the ego corridor and induce a collision or TTC failure. This behavior matches the results in Table 2: AWM mainly reduces progress and TTC, while rule-compliance metrics remain nearly unchanged.

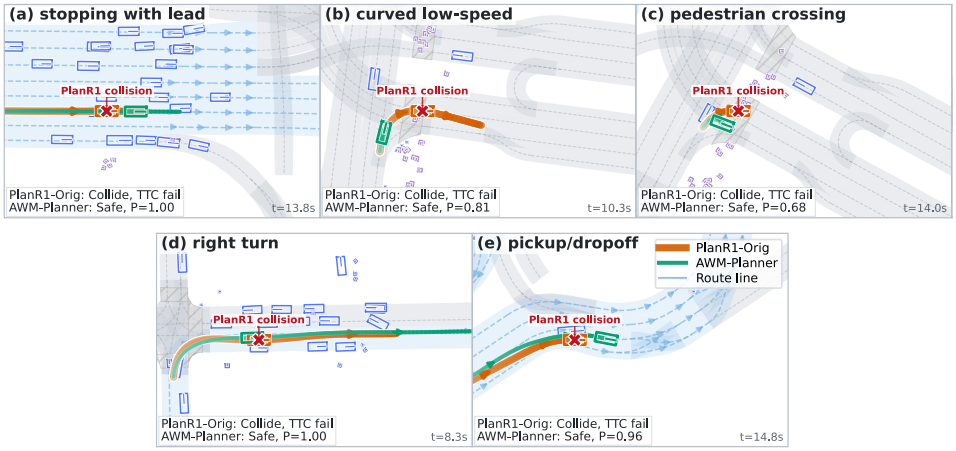


Figure 3: Test14-hard qualitative cases. Orange denotes the original Plan-R1 planner and green denotes the AWM-trained planner. Each panel overlays the full closed-loop trajectory in the same adversarial scene; red markers identify the original planner's collision or TTC failure point.

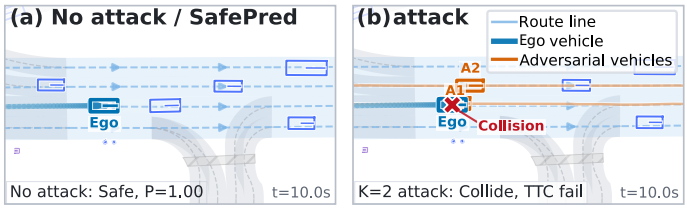


Figure 4: K=2 cooperative attack in sim-agent evaluation. Ego is shown in blue, while A1/A2 are the two adversarial agents selected by AWM.

5 Conclusion

This paper introduced AWM, a multi-agent self-play fine-tuning framework leveraging predictive world models as structured adversaries for robust motion planning. To overcome the intractability of high-dimensional adversarial training, we formulated the problem as a constrained min-max game and proposed a theoretically grounded decoupled solver. By integrating scene-adaptive coalition learning with counterfactual credit assignment, AWM efficiently exposes sparse, synergistic interaction failures within the planner's inherent autoregressive interface. Subsequently, a regret-aware robust best response, with reference-anchored trust region constraints, fine-tunes the planner against the induced tail risk. Theoretical analysis and empirical evaluations demonstrate that our framework significantly improves planner robustness in rare, complex interactions while maintaining nominal performance. Cross-planner sim-agent evaluation and ablations further support the transferability of the learned adversarial traffic and the necessity of fine-grained credit assignment, learnable adversarial coalition, and tail-focused learning. However, AWM remains limited by the fidelity of the tokenized world model and lacks exhaustive scenario coverage or formal closed-loop safety guarantees.

Broader Impact. AWM supports robustness evaluation and training for autonomous driving in simulation. Adversarial scenarios may be misused or create false confidence beyond the evaluated settings. We constrain generation through feasibility and realism checks and distinguish attack strength from planner performance.

Acknowledgments and Disclosures

This work was supported by the National Natural Science Foundation of China (52125208, 524B2164, and 52232015).

The authors declare no competing interests.

References

[1] Rohun Agrawal, Nimit Kalra, Arjun Parthasarathy, Yann LeCun, Oumayma Bounou, Pavel Izmailov, and Micah Goldblum. Closing the train-test gap in world models for gradient-based planning, 2025. URL `https://arxiv.org/abs/2512.09929`.

[2] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[3] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

[4] Yulong Cao, Boris Ivanovic, Chaowei Xiao, and Marco Pavone. Reinforcement learning with human feedback for realistic traffic simulation. In *2024 IEEE international conference on robotics and automation (ICRA)*, pages 14428–14434. IEEE, 2024.

[5] Wei-Jer Chang, Francesco Pittaluga, Masayoshi Tomizuka, Wei Zhan, and Manmohan Chandraker. Safe-sim: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries. In *European conference on computer vision*, pages 242–258. Springer, 2024.

[6] Keyu Chen, Wenchao Sun, Hao Cheng, and Sifa Zheng. Rift: Group-relative rl fine-tuning for realistic and controllable traffic simulation. *arXiv preprint arXiv:2505.03344*, 2025.

[7] Jie Cheng, Yingbing Chen, and Qifeng Chen. Pluto: Pushing the limit of imitation learning-based planning for autonomous driving. *arXiv preprint arXiv:2404.14327*, 2024.

[8] 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.

[9] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

[10] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, pages 1268–1281. PMLR, 2023.

[11] Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *IEEE Robotics and Automation Letters*, 11(1): 226–233, 2025.

[12] 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.

[13] 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.

[14] Shuo Feng, Haojie Zhu, Haowei Sun, Xintao Yan, Linxuan He, Jingxuan Yang, Guangzhen Su, Boqi Li, Shu Li, Ling Wang, et al. Breaking through safety performance stagnation in autonomous vehicles with dense learning. *Nature Communications*, 2026.

[15] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[16] Yuan Gao, Mattia Piccinini, Yuchen Zhang, Dingrui Wang, Korbinian Moller, Roberto Brusnicki, Baha Zarrouki, Alessio Gambi, Jan Frederik Totz, Kai Storms, et al. Foundation models in autonomous driving: A survey on scenario generation and scenario analysis. *IEEE Open Journal of Intelligent Transportation Systems*, 2026.

[17] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov): 1471–1530, 2004.

[18] Marcel Hallgarten, Julian Zapata, Martin Stoll, Katrin Renz, and Andreas Zell. Can vehicle motion planning generalize to realistic long-tail scenarios? In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5388–5395. IEEE, 2024.

[19] 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.

[20] Yihan Hu, Siqi Chai, Zhening Yang, Jingyu Qian, Kun Li, Wenxin Shao, Haichao Zhang, Wei Xu, and Qiang Liu. Solving motion planning tasks with a scalable generative model. In *European Conference on Computer Vision*, pages 386–404. Springer, 2024.

[21] Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3903–3913, 2023.

[22] Zhiyu Huang, Xinshuo Weng, Maximilian Igl, Yuxiao Chen, Yulong Cao, Boris Ivanovic, Marco Pavone, and Chen Lv. Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3445–3451. IEEE, 2025.

[23] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274, 2002.

[24] 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.

[25] Hongchen Li, Tianyu Li, Jiazhi Yang, Haochen Tian, Caojun Wang, Lei Shi, Mingyang Shang, Zengrong Lin, Gaoqiang Wu, Zhihui Hao, et al. Plannerrft: Reinforcing diffusion planners through closed-loop and sample-efficient fine-tuning. *arXiv preprint arXiv:2601.12901*, 2026.

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

[27] 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.

[28] Yuewen Mei, Tong Nie, Jian Sun, and Ye Tian. Llm-attacker: Enhancing closed-loop adversarial scenario generation for autonomous driving with large language models. *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[29] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *Advances in neural information processing systems*, 30, 2017.

[30] 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.

[31] Tong Nie, Yihong Tang, Junlin He, Yuewen Mei, Jie Sun, Lijun Sun, Wei Ma, and Jian Sun. Adv-0: Closed-loop min-max adversarial training for long-tail robustness in autonomous driving. *arXiv preprint arXiv:2603.15221*, 2026.

[32] Muleilan Pei, Shaoshuai Shi, and Shaojie Shen. Advancing multi-agent traffic simulation via r1-style reinforcement fine-tuning. *arXiv preprint arXiv:2509.23993*, 2025.

[33] Zhenghao Peng, Wenjie Luo, Yiren Lu, Tianyi Shen, Cole Gulino, Ari Seff, and Justin Fu. Improving agent behaviors with rl fine-tuning for autonomous driving. In *European Conference on Computer Vision*, pages 165–181. Springer, 2024.

[34] Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. *arXiv preprint arXiv:2312.04535*, 2023.

[35] Davis Rempe, Jonah Philion, Leonidas J Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17305–17315, 2022.

[36] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000.

[37] Oliver Scheel, Luca Bergamini, Maciej Wolczyk, Błażej Osiński, and Peter Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Conference on Robot Learning*, pages 718–728. PMLR, 2022.

[38] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[39] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023.

[40] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[41] 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.

[42] Qiao Sun, Shiduo Zhang, Danjiao Ma, Jingzhe Shi, Derun Li, Simian Luo, Yu Wang, Ningyi Xu, Guangzhi Cao, and Hang Zhao. Large trajectory models are scalable motion predictors and planners. *arXiv preprint arXiv:2310.19620*, 2023.

[43] Tianyi Tan, Yinan Zheng, Ruiming Liang, Zexu Wang, Kexin Zheng, Jinliang Zheng, Jianxiong Li, Xianyuan Zhan, and Jingjing Liu. Flow matching-based autonomous driving planning with advanced interactive behavior modeling. *arXiv preprint arXiv:2510.11083*, 2025.

[44] Xiaolong Tang, Meina Kan, Shiguang Shan, and Xilin Chen. Plan-r1: Safe and feasible trajectory planning as language modeling. *arXiv preprint arXiv:2505.17659*, 2025.

[45] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, et al. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 8(6):3692–3711, 2023.

[46] 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.

[47] 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.

[48] 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, 2025.

[49] Bowen Ye, Bin Zhang, and Hang Zhao. Dap: A discrete-token autoregressive planner for autonomous driving. *arXiv preprint arXiv:2511.13306*, 2025.

[50] Dongkun Zhang, Jiaming Liang, Ke Guo, Sha Lu, Qi Wang, Rong Xiong, Zhenwei Miao, and Yue Wang. Carplanner: Consistent auto-regressive trajectory planning for large-scale reinforcement learning in autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17239–17248, 2025.

[51] Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, et al. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27220–27230, 2025.

[52] 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.

[53] Zhejun Zhang, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5422–5432, 2025.

[54] Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. *arXiv preprint arXiv:2501.15564*, 2025.

[55] Zikang Zhou, Haibo Hu, Xinhong Chen, Jianping Wang, Nan Guan, Kui Wu, Yung-Hui Li, Yu-Kai Huang, and Chun J Xue. Behaviorgpt: Smart agent simulation for autonomous driving with next-patch prediction. *Advances in Neural Information Processing Systems*, 37:79597–79617, 2024.

Appendix

Appendix Contents

A Related Work **16**

B Extended Architectural and Algorithmic Details **17**
B.1 Discrete Autoregressive Planning Architecture and Role Conditioning . . . 17
B.1.1 Unified Autoregressive Motion Generation Interface . . . 17
B.1.2 Model Instantiation and Role Conditioning Mechanism . . . 18
B.2 Algorithmic Details of Adversarial Coalition Learning . . . 18
B.2.1 Probe-Time Threat Signals and Residual Proposal Construction . . . 18
B.2.2 Feature Representations for Primary Selector and Pair Ranker . . . 19
B.2.3 Multi-Stage Training Objective for AWM . . . 20
B.3 Details of Regret-Aware Constrained Planner Optimization . . . 20
B.3.1 Reference-Anchored Nominal Retention . . . 20
B.3.2 Analytical Regret-CVaR Weighting . . . 21
B.3.3 Safety-Constrained Dual Update Parameterization . . . 22
B.4 AWM as a Closed-Loop Sim-Agent . . . 22

C Theoretical Analysis **24**
C.1 Notation and Autoregressive Local Stability . . . 24
C.2 Local Stackelberg Justification of the Decoupled Stage-A/B Solver . . . 25
C.3 Scene-Adaptive Pair Admission as a Bayes Gate . . . 27
C.4 Role-Switched Counterfactual Credit as a Difference-Reward Estimator . . . 28
C.5 Regret as a Reference-Anchored Control Variate . . . 30
C.6 Closed-Form CVaR on the AWM-Induced Two-Point Risk Distribution . . . 31
C.7 Nominal Retention as a Local Performance Floor . . . 32
C.8 Safety Dual Variables as a Projected Max-Violation Surrogate . . . 33
C.9 Summary and Theory-to-Practice Gap . . . 34

D Detailed Experimental Setups **36**
D.1 Training Data, Computation, and Hyperparameters . . . 36
D.2 Evaluation Setup and Protocol . . . 36
D.2.1 Closed-Loop and Sim-Agent Evaluation . . . 37
D.2.2 Dual-Rollout Open-Loop Evaluation . . . 37
D.2.3 AWM Mechanism Ablations . . . 37
D.2.4 Planner Mechanism Ablations . . . 38
D.2.5 Baseline Adversarial Generator Adaptation . . . 38
D.3 Construction of the `InterPlan-LongTail` Benchmark . . . 38

E Supplementary Results **40**
E.1 Closed-Loop Long-Tail Breakdown . . . 40
E.2 Adversarial Sim-Agent Diagnostics . . . 40
E.3 Planner-World Cross-Validation Details . . . 40
E.4 AWM Mechanism Ablations . . . 41
E.5 Additional Planner Ablations . . . 43
E.6 Adapted Prior Adversarial Generators . . . 43
E.7 Supplementary Visualizations . . . 44
E.7.1 `InterPlan-LongTail` Examples . . . 44
E.7.2 Time-Unrolled Robust-Recovery Cases . . . 44
E.7.3 K=2 Cooperative Attack Cases . . . 47

A Related Work

Autoregressive Motion Generation. Autoregressive modeling has recently emerged as a dominant paradigm for both multi-agent scenario simulation and ego-motion planning in autonomous driving. Early sequence modeling methods primarily operate in continuous spaces and leverage Transformer-based architectures [11, 42] or diffusion models [51, 54] to refine continuous waypoints iteratively. However, inspired by the success of Large Language Models (LLMs), discrete next-token prediction modeling has gained significant traction due to its scalability and temporal causality. Agent simulation models such as Trajeglish [34], SMART [47], and BehaviorGPT [55] tokenize maps and trajectories to progressively decode joint multi-agent futures, achieving realistic and interactive rollouts. Building upon this motion-as-language concept [39], recent efforts have extended this discrete representation to closed-loop ego planning [20, 49, 44], casting ego decision-making as motion token generation. Most relevant to our work are architectures that explicitly decouple predictive scene evolution (the world model) from ego planning (the policy), such as the planning-prediction dual-branch design in [44, 49] and the policy-transition decomposition in [50]. This predictive branch allows the planner to efficiently interact with reactive non-ego responses during rollouts without relying on a simulator. However, prior methods treat the world model as a frozen or passive behavioral prior that merely replicates logged data, lacking a structured mechanism to generate customized non-ego behaviors within the same tokenized framework.

Reinforcement Learning Fine-Tuning (RLFT). While large-scale behavior cloning establishes a strong traffic prior, it invariably suffers from covariate shift in closed-loop evaluation and fails to enforce safety-critical driving principles [24]. To bridge this gap, RLFT is increasingly adopted for aligning driving policies with downstream objectives. In the continuous domain, frameworks such as [22, 25] employ learned reward models and policy-guided diffusion to fine-tune generative planners for sample efficiency and alignment. Within discrete architectures, Plan-R1 [44] applies GRPO-style optimization with rule-based rewards while maintaining a frozen reactive world model for surrounding agents. Similarly, Ye et al. [49] incorporates offline SAC-BC to mitigate imitation ambiguity, and Zhang et al. [50] suggests that autoregressive structures naturally facilitate large-scale RLFT. Furthermore, RLFT has also been applied to simulation agents for better collision realism and behavior fidelity [33, 4, 6]. Despite these advances, existing RLFT pipelines primarily optimize the ego policy against the expected distribution of a naturalistic traffic prior. Due to the rarity of long-tail events, the resulting planners can learn average behaviors and remain vulnerable to rare but safety-critical risks.

Adversarial Training and Testing. Evaluating and ensuring the robustness of autonomous driving needs moving beyond naturalistic traffic scenarios to adversarial edge cases [26]. Early adversarial generation methods [46, 35, 5, 28, 48] often rely on heuristic perturbations, optimization-based attacks on continuous trajectories, or training specific adversarial agents to induce collisions for stress testing. To further improve policy robustness, closed-loop adversarial training frameworks [52, 30, 27, 41, 31] alternate between safety-critical scenario generation and ego policy updates. Recent literature emphasizes that effective adversaries must be structured and reactive rather than purely collision-seeking perturbations to ensure the generated scenarios remain realistic and useful for training. Several techniques such as skill learning [41], preference alignment [30], reverse-time token reconstruction [27] are shown to be effective for preserving realism or controllability. Despite these advances, existing methods are largely treated as external environment augmentations, heuristic resampling against a static planner, computationally heavy post-hoc optimization or test-time biasing. More importantly, existing closed-loop adversarial training frameworks usually assume model-free RL and thus rely on CPU-intensive policy rollouts on physical simulators, significantly limiting their scalability and efficiency [31]. Therefore, they often operate in a small set of scenarios (e.g., 500), making them less effective for large-scale motion planners. Agrawal et al. [1] train world models with adversarial perturbations of latent states and actions to improve gradient-based planning. Our AWM learns traffic policies conditioned on agent roles, assigns counterfactual credit to sparse coalitions, and uses the resulting interactions to fine-tune the ego planner.

B Extended Architectural and Algorithmic Details

This appendix provides technical details to supplement the method presented in Section 3. Algorithm 1 first summarizes the full decoupled self-play procedure. Section B.1 details the shared discrete autoregressive architecture and the role-injection mechanism. Section B.2 elaborates on the mathematical formulations of the scene-adaptive coalition learning pipeline, including probe statistics, feature representations, and the full training objective. Section B.3 provides the exact analytical derivations and loss formulations for the regret-aware constrained planner optimization. Finally, Section B.4 explains how the learned AWM is instantiated as a closed-loop sim-agent evaluator for transferable adversarial testing.

Algorithm 1 Decoupled Self-Play Adversarial Fine-Tuning

Require: Scenario distribution $\mathcal{D}$; nominal planner $\pi_{\text{plan}}^{\text{nom}}$; benign world model π_{pred}; coalition budget $K_{\max}$; CVaR tail mass τ; safety thresholds κ.
Ensure: Frozen adversarial host $\pi_{\text{awm}}^{\star}$, calibrated opponent distribution $\mathcal{P}_{\text{adv}}^{\star}$, robust planner $\widehat{\pi}_{\text{plan}}$.
Initialize planner and world-model branches with the shared autoregressive factorization in Eq. 1; set $\pi_{\text{ref}} \leftarrow \pi_{\text{plan}}^{\text{nom}}$.
Stage A: learn the adversarial world model for Eq. 3.
for AWM minibatches $\mathcal{B} \subset \mathcal{D}$ **do**
  for each scene $s \in \mathcal{B}$ **do**
    Background probe: assign $\rho_i = \texttt{bg}$ for all non-ego agents and roll out the role-conditioned generator in Eq. 5.
    Build a sparse proposal pool P_s using realism-penalized threat and residual coverage (Appendix B.2.1).
    if $P_s = \emptyset$ **then**
      Set $A_s^{\text{top1}} = A_s^{\text{pair}} = \emptyset$ and $p_s = 0$.
    else
      Select $i_s^{\star} = \arg\max_{i \in P_s} f_{\text{pri}}(x_i)$; condition on $\rho_{i_s^{\star}} = \texttt{atk}$ and choose $\hat{j}_s = \arg\max_{j \in P_s \setminus \{i_s^{\star}\}} f_{\text{pair}}(\phi(i_s^{\star}, j))$.
      Define $A_s^{\text{top1}} = \{i_s^{\star}\}$, $A_s^{\text{pair}} = \{i_s^{\star}, \hat{j}_s\}$, and pair-admission probability $p_s = \sigma(\alpha_s^{\text{cal}}/\tau_{\text{risk}}))$.
      Update π_{awm}, f_{pri}, and f_{pair} with role-switched counterfactual credit (Eq. 9); update the calibrator q_η using Appendix B.2.3.
    end if
  end for
end for
Freeze $\pi_{\text{awm}}^{\star} \leftarrow \pi_{\text{awm}}$ and $q_\eta^{\star} \leftarrow q_\eta$.
Define $\mathcal{P}_{\text{adv}}^{\star}(B_s = \texttt{top1} \mid s) = 1 - p_s$ and $\mathcal{P}_{\text{adv}}^{\star}(B_s = \texttt{pair} \mid s) = p_s$ as in Eq. 10.
Stage B: train the constrained planner best response in Eq. 4.
for planner minibatches $\mathcal{B} \subset \mathcal{D}$ **do**
  for each scene $s \in \mathcal{B}$ **do**
    Roll out π_θ on the benign branch π_{pred} and adversarial branches $B_s \in \{\texttt{top1}, \texttt{pair}\}$ induced by $\pi_{\text{awm}}^{\star}$.
    Compute the nominal-retention loss $\mathcal{L}_{\text{nom}}$ in Eq. 11.
    Compute adversarial regrets $\Delta_s^b(\theta) = U_s^b(\pi_{\text{ref}}) - U_s^b(\pi_\theta)$ and the two-point regret distribution in Eqs. 12–13.
    Obtain $w_s^{\text{top1}}, w_s^{\text{pair}}$ from the closed-form regret-CVaR allocation in Appendix B.3.2.
    Accumulate the adversarial tail-risk loss $\mathcal{L}_{\text{rob}}$ in Eq. 14 and safety statistics $\mathcal{L}_{\text{dual}}$.
  end for
  Update θ with $\mathcal{L}_{\text{nom}} + \mathcal{L}_{\text{rob}}$; update dual variables using thresholds κ (Appendix B.3.3).
end for
Select $\widehat{\pi}_{\text{plan}}$ by validation robustness under $\mathcal{P}_{\text{adv}}^{\star}$ subject to nominal-retention criteria.
return $\pi_{\text{awm}}^{\star}$, $\mathcal{P}_{\text{adv}}^{\star}$, and $\widehat{\pi}_{\text{plan}}$.

B.1 Discrete Autoregressive Planning Architecture and Role Conditioning

This section details the architectural choices required to implement the unified planning-prediction decomposition introduced in Section 2, and expands upon the role-conditioned generation mechanism. By casting both planning and environment simulation as discrete next-token prediction tasks [44], we ensure that adversarial interventions modify only the behavioral distribution of the agents, without violating the underlying physical transition dynamics.

B.1.1 Unified Autoregressive Motion Generation Interface

A cornerstone of our framework is that both the ego planner and the environment evolution can be formulated as discrete next-token prediction tasks. Let x_t denote the interactivep scene state at step t, which includes static map elements, historical agent trajectories, and the generated future prefix

up to t. For each agent type (e.g., vehicles, pedestrians), we maintain a discrete motion vocabulary $\mathcal{T}$ whose entries correspond to local motion primitives (e.g., discretized displacement and heading increments $(\Delta x, \Delta y, \Delta\psi)$) over a fixed autoregressive interval following previous practices [47, 39].

During the rollout, the standard two-branch architecture factorizes the joint one-step generation into a planning branch π_{plan} for the ego agent and an environment branch π_{pred} for all non-ego agents:

$$p(\boldsymbol{z}_t \mid x_t) = \pi_{\text{plan}}(z_{e,t} \mid x_t) \prod_{i \in \mathcal{V}_s} \pi_{\text{pred}}(z_{i,t} \mid x_t). \tag{15}$$

Once the step-wise token set $\boldsymbol{z}_t = \{z_{i,t}\}_{i \in \{e\} \cup \mathcal{V}_s}$ is sampled, it is decoded into continuous kinematic updates. A deterministic transition operator Γ then updates the global coordinates and advances the scene state: $x_{t+1} = \Gamma(x_t, \boldsymbol{z}_t)$. Crucially, this rollout interface is strictly preserved throughout our framework. The AWM alters *who* generates the non-ego tokens and *how* they behave, but it does not modify the underlying simulator, the token space, or the physical transition dynamics. Consequently, the normal, top-1, and pair adversarial branches share the identical state representation, transition operator, and physical motion manifold, ensuring that the planner is evaluated fairly across different attack complexities and are strictly comparable.

B.1.2 Model Instantiation and Role Conditioning Mechanism

To instantiate the policies, both the planner and the environment branches share a unified high-level network template but maintain separate parameters. Specifically, for a branch $k \in \{\texttt{pred}, \texttt{plan}\}$, the architecture comprises a map encoder E^k_{map}, an interaction backbone B^k, and a token decoder D^k:

$$g^k = E^k_{\text{map}}(x_t), \quad h^k_{i,t} = B^k(x_t, g^k), \quad \ell^k_{i,t} = D^k(h^k_{i,t}), \tag{16}$$

where $h^k_{i,t}$ is the hidden state for agent i and $\ell^k_{i,t}$ are the output logits over the vocabulary $\mathcal{T}$. The backbone B^k typically employs stacked transformer blocks over relation graphs (e.g., temporal self-attention, map-to-agent cross-attention, and agent-agent interaction) [44, 47, 55].

To construct the AWM from the pretrained predictive model π_{pred} without training a separate adversarial generator from scratch, we introduce a latent role conditioning $\rho_i \in \{\texttt{ego}, \texttt{bg}, \texttt{atk}\}$ for each agent. The role identity is injected into the network at two levels to alter the agent's interaction behavior and output distribution. First, an agent-side role embedding $E_{\text{agent}}(\rho_i)$ is added to the agent's initial hidden state before the spatial-temporal interaction blocks, allowing the role to influence how the agent attends to the map and other vehicles. Second, a decoder-side embedding $E_{\text{dec}}(\rho_i)$ is added immediately before the final token projection. The role-conditioned generation is given by:

$$\tilde{h}^{\text{awm}}_{i,t} = h^{\text{awm}}_{i,t} + E_{\text{agent}}(\rho_i), \quad \tilde{\ell}^{\text{awm}}_{i,t} = D^{\text{awm}}\left(\tilde{h}^{\text{awm}}_{i,t} + E_{\text{dec}}(\rho_i)\right). \tag{17}$$

The background role embedding $E(\texttt{bg})$ is initialized as a zero vector. Consequently, when no attacker is activated ($\rho_i = \texttt{bg}, \forall i \in \mathcal{V}_s$), the AWM perfectly recovers the naturalistic behavior of the pretrained π_{pred}. Furthermore, switching between factual and counterfactual coalitions (as required in Section 3.2) only requires modifying the role table ρ_i without altering the network parameters or the simulator. This guarantees that the calculated counterfactual utility gaps isolate the true marginal contribution of the coalition rather than architectural discrepancies, which is the cornerstone of the counterfactual credit assignment (Section 3.2).

B.2 Algorithmic Details of Adversarial Coalition Learning

This section expands upon the scene-adaptive coalition learning pipeline, detailing the mathematical formulations for the probe signals, the neural selector inputs, and the multi-stage training objectives.

B.2.1 Probe-Time Threat Signals and Residual Proposal Construction

Directly evaluating the neural primary selector on all background agents is computationally prohibitive. The objective of the proposal stage is to reduce the combinatorial multi-agent search space into a sparse potential candidate. Therefore, we execute a lightweight background probe rollout (where $\rho_i = \texttt{bg}, \forall i \in \mathcal{V}_s$) to extract heuristic threat signals and construct a sparse proposal set P_s.

For each non-ego agent i, we first compute a dense, heuristic ego-centric threat signal $r_{i,\text{dense}}(t)$ at each step t, which aggregates proximity (r_{prox}), time-to-collision (TTC) approximations (r_{cpa}), blocking behaviors (r_{block}), and cut-in maneuvers (r_{cutin}):

$$r_{i,\text{dense}}(t) = w_{\text{prox}} r_{i,\text{prox}}(t) + w_{\text{cpa}} r_{i,\text{cpa}}(t) + w_{\text{block}} r_{i,\text{block}}(t) + w_{\text{cutin}} r_{i,\text{cutin}}(t). \tag{18}$$

To capture severe safety violations, we define the raw probe signal by incorporating hard collision indicators:

$$r_i^{\text{probe}}(t) = \max\left(r_{i,\text{dense}}(t), \lambda_{\text{ttc}} \mathbb{I}_{i,\text{ttc}}(t), \lambda_{\text{col}} \mathbb{I}_{i,\text{col}}(t)\right). \tag{19}$$

To discourage the selection of agents that might execute physically invalid or trivial attacks (e.g., driving off-road to hit the ego), we define a comprehensive penalized signal:

$$\bar{r}_i^{\text{probe}}(t) = r_i^{\text{probe}}(t) - w_{\text{kin}} c_i^{\text{kin}}(t) - w_{\text{road}} c_i^{\text{road}}(t) - w_{\text{obs}} c_i^{\text{obs}}(t), \tag{20}$$

where $c_i^{(\cdot)}$ are penalty indicators for kinematic violations, off-road driving, and static obstacle collisions, respectively, weighted by constants $w_{(\cdot)}$. Using the discounted return operator $G(v)_\tau = \sum_{t=\tau}^{H} \gamma^{t-\tau} v(t)$, we obtain the scalarized pure threat score $q_i = \max_\tau G(r_i^{\text{probe}})_\tau$ and the comprehensive score $\bar{q}_i = \max_\tau G(\bar{r}_i^{\text{probe}})_\tau$. The first proposal candidate is greedily selected based on the highest $\bar{q}_i$. To ensure the proposal set P_s is diverse, subsequent candidates are added based on their *residual threat coverage*. Given a partially constructed proposal set S, the residual threat of a new agent i is $\tilde{r}_i^S(t) = \max\left(r_i^{\text{probe}}(t) - \max_{j \in S} r_j^{\text{probe}}(t), 0\right)$. The residual score is $\psi_i^S = \max_\tau G(\tilde{r}_i^S)_\tau$. We only admit agent i if its relative residual ratio $\eta_i^S = \psi_i^S / (q_i + \epsilon)$ exceeds a predefined threshold. The rationale behind this design is to ensure that newly proposed agents explain complementary spatial-temporal adversarial potential rather than merely duplicating the interaction patterns of already selected candidates, thereby preserving diversity for the pair-ranking stage.

B.2.2 Feature Representations for Primary Selector and Pair Ranker

While scalar probe scores provide a heuristic prior for efficient filtering, they lack the higher-order interaction context necessary to identify the true optimal attacker. The learnable primary selector f_{pri} and the conditional pair ranker f_{pair} bridge this gap by fusing neural hidden states with proposal statistics.

Primary Selector. For each candidate $i \in P_s$, the primary selector f_{pri} operates on a fused feature vector x_i that concatenates the high-dimensional hidden state from the background probe with the heuristic proposal statistics:

$$x_i = \left[h_i^{\text{probe}},\ \mathbb{I}(i \in P_s),\ \frac{\text{rank}(i)}{B_{\text{prop}}},\ q_i,\ \bar{q}_i,\ \psi_i^S,\ m_i\right], \tag{21}$$

where h_i^{probe} is the final-layer hidden state of the AWM extracted from the background probe, B_{prop} is the proposal budget, and m_i is a binary stage marker. The selector outputs a confidence score $a_i = f_{\text{pri}}(x_i)$, and the primary attacker is determined as $i_s^* = \arg\max_{i \in P_s} a_i$. After selecting the primary attacker, we retain a small runner-up candidate set to serve as potential partners:

$$R_s = \text{Top}_{B_{\text{pair}}}\left(\{i \in P_s \setminus \{i_s^*\}\}; a_i\right), \tag{22}$$

where B_{pair} is the maximum partner-candidate budget and $\text{Top}_{B_{\text{pair}}}(\cdot; a_i)$ returns the candidates with the largest primary-selector scores. Thus, P_s is the proposal pool produced by the residual threat-coverage procedure, while R_s is the smaller selector-filtered candidate pool used for conditional pair ranking.

Conditional Pair Ranker and Calibrator. To evaluate the synergistic value of a secondary attacker, we execute a second conditional probe where i_s^* is assigned $\rho = \texttt{atk}$. This yields conditional hidden states h^{cond} that encode the scene's reaction to the primary attack. For any candidate partner $j \in P_s \setminus \{i_s^*\}$, the pair ranker f_{pair} takes a relational feature $\phi(i_s^*, j)$ that captures the conditional interaction dynamics:

$$\phi(i_s^*, j) = \left[h_{i_s^*}^{\text{cond}},\ h_j^{\text{cond}},\ \left|h_{i_s^*}^{\text{cond}} - h_j^{\text{cond}}\right|,\ h_{i_s^*}^{\text{cond}} \odot h_j^{\text{cond}},\ \bar{q}_{i_s^*},\ \bar{q}_j,\ \psi_j^{\{i_s^*\}},\ \Delta a_{i_s^*, j}\right], \tag{23}$$

where $\odot$ denotes element-wise multiplication, and $\Delta a_{i_s^*,j} = f_{\mathrm{pri}}(x_{i_s^*}) - f_{\mathrm{pri}}(x_j)$ represents the confidence gap from the primary selector. The pair ranker predicts the marginal coalition gain $\hat{g}_s(i_s^*, j) = f_{\mathrm{pair}}(\phi(i_s^*, j))$, and the optimal partner is $\hat{j}_s = \arg\max_j \hat{g}_s(i_s^*, j)$.

Subsequently, the scene-adaptive calibrator q_η determines whether admitting $\hat{j}_s$ is globally beneficial (should actually be admitted). The scene-level feature aggregates the selected pair representation, the predicted pair gain, the primary-selector confidence, and two cardinality-based uncertainty indicators:

$$\psi_s = \left[\phi(i_s^*, \hat{j}_s),\ \hat{g}_s(i_s^*, \hat{j}_s),\ a_{i_s^*},\ \sigma(a_{i_s^*}),\ |P_s|,\ |R_s|\right]. \tag{24}$$

Here, $|P_s|$ denotes the number of plausible attackers admitted by the proposal stage, and $|R_s|$ denotes the number of selector-retained runner-up candidates considered for pair ranking. These two scalars provide the calibrator with scene-level ambiguity information: a larger $|P_s|$ indicates a denser or more ambiguous adversarial proposal pool, while a larger $|R_s|$ indicates more plausible secondary partners after primary selection. The calibrator is trained via Binary Cross-Entropy (BCE) to predict the probability p_s that the true marginal improvement $G_s^{\mathrm{pair}} = U_s(\{i_s^*\}) - U_s(\{i_s^*, \hat{j}_s\})$ is strictly positive.

B.2.3 Multi-Stage Training Objective for AWM

To stabilize the optimization of the role-conditioned policy alongside the coalition search modules, we employ a two-stage training paradigm.

Stage I (Host and Search Optimization): We jointly train the role-conditioned AWM policy, the primary selector, and the conditional pair ranker. The objective is:

$$\mathcal{L}_{\mathrm{stage1}} = \mathcal{L}_{\mathrm{policy}}(\Delta^{\mathrm{hyb}}) + \lambda_{\mathrm{pri}}\mathcal{L}_{\mathrm{pri}} + \lambda_{\mathrm{pair}}\mathcal{L}_{\mathrm{pair}} + \Omega_{\mathrm{stab}}, \tag{25}$$

Here, $\mathcal{L}_{\mathrm{policy}}$ is the policy gradient loss driven by the hybrid counterfactual reward Δ_i^{hyb} (defined in Eq. 9). $\mathcal{L}_{\mathrm{pri}}$ and $\mathcal{L}_{\mathrm{pair}}$ supervise the selector and ranker using the observed utility gaps as regression and ranking targets. Specifically: (1) For the primary selector, the target label for candidate i is its absolute adversarial impact compared to the nominal background traffic: $y_i^{\mathrm{pri}} = U_s(\emptyset) - U_s(\{i\})$, where $\emptyset$ denotes no active attackers. (2) For the conditional pair ranker, the target label for a partner candidate j is its marginal coalition gain over the primary attacker i_s^*: $y_{i_s^*,j}^{\mathrm{pair}} = U_s(\{i_s^*\}) - U_s(\{i_s^*, j\})$. Finally, the stabilizer Ω_{stab} anchors the AWM's background behavior to the pretrained predictive prior π_{pred} via KL divergence, preventing the model from collapsing into unrealistic adversarial behaviors.

Stage II (Calibrator Optimization): We freeze the components from Stage-I and train the scene-level calibrator q_η using the BCE loss $\mathcal{L}_{\mathrm{cal}}$. This decoupling is crucial because the calibrator's objective is to act as an adaptive gate that decides *when* a multi-agent attack is statistically justified, rather than discovering new attackers. Stage-II trains the scene-level calibrator q_η to ensure unbiased admission probabilities.

B.3 Details of Regret-Aware Constrained Planner Optimization

This section details the planner-side robust optimization (Section 3.3), including the exact formulation of the trust region constraints, the closed-form derivation of the CVaR weights, and the dual update mechanism for safety constraints. It explains how the AWM is converted into a risk distribution and how the constrained optimization is solved.

B.3.1 Reference-Anchored Nominal Retention

During the training of the planner, we do not sample arbitrary attacks from a replay buffer. Instead, for each scene s, the frozen AWM executes the coalition pipeline to generate a top-1 attacker gate g_s^{top1} and a pair gate g_s^{pair} (if a valid partner exists). This yields three explicit rollout branches, evaluated by their respective utilities:

$$J_s^{\mathrm{norm}}(\pi) = U_s(\pi, \pi_{\mathrm{pred}}), \quad J_s^{\mathrm{top1}}(\pi) = U_s(\pi, \pi_{\mathrm{awm}}; g_s^{\mathrm{top1}}), \quad J_s^{\mathrm{pair}}(\pi) = U_s(\pi, \pi_{\mathrm{awm}}; g_s^{\mathrm{pair}}). \tag{26}$$

To prevent the planner from sacrificing naturalistic driving capabilities in pursuit of worst-case robustness (Section 3.3), we enforce a reference-anchored nominal retention objective on the normal branch. Let π_{ref} be the frozen baseline planner. We define the normal utility margin as $m_s^{\text{norm}}(\theta) = J_s^{\text{norm}}(\pi_\theta) - J_s^{\text{norm}}(\pi_{\text{ref}})$. To penalize degradation without forcing the planner to over-optimize already safe scenes, we introduce an asymmetric margin penalty that activates only when the planner's performance drops below an acceptable margin $\epsilon_{\text{norm}} \geq 0$:

$$\psi_s^{\text{norm}}(\theta) = [-\epsilon_{\text{norm}} - m_s^{\text{norm}}(\theta)]_+^2 , \tag{27}$$

where $[\cdot]_+ = \max(\cdot, 0)$. The total nominal retention loss combines the standard policy gradient surrogate $\ell_s^{\text{norm}}(\theta)$, a trajectory-level KL divergence constraint, and the margin penalty:

$$\mathcal{L}_{\text{nom}}(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\lambda_{\text{pg}}\ell_s^{\text{norm}}(\theta) + \lambda_{\text{kl}} D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}} \mid x_{s,1:H}^{\text{norm}}) + \lambda_{\text{margin}}\psi_s^{\text{norm}}(\theta)\right], \tag{28}$$

where $\ell_s^{\text{norm}}(\theta)$ is the standard GRPO [40] surrogate loss maximizing nominal utility. This asymmetric design penalizes performance degradation but does not force the planner to over-optimize on already safe scenes.

B.3.2 Analytical Regret-CVaR Weighting

The core of robust adaptation is to optimize the risk objective with respect to the AWM-induced regret distribution. To optimize the adversarial tail-risk, we compute the reference-relative regret $\Delta_s^b(\theta) = J_s^b(\pi_{\text{ref}}) - J_s^b(\pi_\theta)$ for $b \in \{\text{top1}, \text{pair}\}$. The AWM calibrator induces a discrete two-point risk distribution:

$$\mathcal{D}_s^\theta = \left\{\left(\Delta_s^{\text{top1}}(\theta), 1-p_s\right), \left(\Delta_s^{\text{pair}}(\theta), p_s\right)\right\}, \tag{29}$$

where p_s is the calibrated probability of admitting the pair attack. We apply the Conditional Value-at-Risk (CVaR) operator [36] at tail level $\tau \in (0, 1]$ over this specific distribution. Because AWM naturally induces a discrete two-point distribution over the attack types $b \in \{\texttt{top1}, \texttt{pair}\}$, the CVaR allocation admits an elegant *closed-form analytical solution*, avoiding the high variance typically associated with sampling-based CVaR estimation. Let $b_{(1)}$ and $b_{(2)}$ denote the two branches sorted by their regret in descending order (i.e., $\Delta_s^{b_{(1)}} \geq \Delta_s^{b_{(2)}}$), and let $\mu_s^{b_{(1)}}$ be the probability mass of the worst-case branch. For a specified tail level τ, the CVaR operator focuses strictly on the worst τ-proportion of the distribution. The exact dynamic weight $w_s^{b_{(1)}}$ for the worse branch is given by:

$$w_s^{b_{(1)}}(\tau) = \begin{cases} 1, & \text{if } \mu_s^{b_{(1)}} \geq \tau, \\ \frac{\mu_s^{b_{(1)}}}{\tau}, & \text{if } \mu_s^{b_{(1)}} < \tau. \end{cases} \tag{30}$$

The weight for the better branch is simply the residual tail mass: $w_s^{b_{(2)}}(\tau) = 1 - w_s^{b_{(1)}}(\tau)$. Intuitively, these weights dynamically route the gradient updates. If the probability of the worst-case attack exceeds τ, the planner focuses entirely on surviving that specific attack ($w = 1$) and absorbs as much of the tail mass τ as its probability allows. If the worst-case attack is extremely rare (probability $< \tau$), the planner distributes its learning capacity proportionally between both branches.

In our implementation, these weights are treated as detached constants during backpropagation. The final robust objective dynamically scales the branch-wise policy gradient losses:

$$\mathcal{L}_{\text{rob}}(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\text{sg}\left(w_s^{\text{top1}}(\tau)\right)\ell_s^{\text{top1}}(\theta) + \text{sg}\left(w_s^{\text{pair}}(\tau)\right)\ell_s^{\text{pair}}(\theta)\right], \tag{31}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator, making the planner optimize its policy rather than attempting to manipulate the risk distribution itself. This dynamic weighting reconciles the sparsity of the AWM's attacks with the planner's safety demands, ensuring the planner's gradients are guided by the scene-adaptive vulnerabilities discovered by the AWM. Because pair attacks are scene-dependent, their calibrated probability p_s is often small. Under a standard expected loss, a rare but fatal pair attack would have its gradient severely diluted by p_s, causing the planner to under-optimize for it. By applying CVaR, if the low-probability pair branch currently induces the highest regret, the operator aggressively upweights it (e.g., scaling its weight from p_s to p_s/τ or 1). This encourages the planner to defend against its most severe weakness rather than just average-case perturbations.

B.3.3 Safety-Constrained Dual Update Parameterization

To prevent the planner from exploiting the regret objective by violating hard traffic rules (e.g., evading an attacker by driving off-road), we enforce explicit safety constraints via a dual update mechanism (Section 3.3). Let $v^b_{s,m} \in [0,1]$ denote the violation indicator for the m-th safety cost under branch b. The expected branch-mixed violation rate is $\bar{v}_{s,m} = (1-p_s)v^{\texttt{top1}}_{s,m} + p_s v^{\texttt{pair}}_{s,m}$. To ensure optimization stability and guarantee that the utility multipliers remain strictly positive and bounded, we parameterize the progress reward multiplier λ_R and the M safety cost multipliers λ_m jointly using a softmax projection over unconstrained dual variables $\zeta \in \mathbb{R}^M$ and a fixed progress logit α_R:

$$[\lambda_R, \lambda_1, \ldots, \lambda_M] = \mathrm{softmax}([\alpha_R, \zeta_1, \ldots, \zeta_M]). \tag{32}$$

This parameterization ensures $\lambda_m \in (0,1)$ and $\sum \lambda = 1$. Given the admissible violation thresholds $\kappa \in \mathbb{R}^M$, the dual objective to be maximized with respect to ζ is:

$$\mathcal{L}_{\mathrm{dual}}(\zeta) = \sum_{m=1}^{M} \lambda_m(\zeta)\left(\mathbb{E}_{s\sim\mathcal{D}}[\bar{v}_{s,m}] - \kappa_m\right). \tag{33}$$

During training, the planner parameters θ and the dual variables ζ are updated via alternating gradient descent-ascent. This separation allows the planner to aggressively minimize adversarial regret while treating hard traffic rules as strict boundaries rather than easily exploitable reward shaping.

B.4 AWM as a Closed-Loop Sim-Agent

This section describes how the learned AWM is used as a simulator-side traffic world model for the transferable adversarial evaluation in Table 2. The goal is to expose a planner to adversarial but feasible non-ego reactions within the closed-loop reactive simulation. This replaces the rule-based or replay agents with our learned AWM policies. We therefore instantiate AWM as a drop-in non-ego policy: the evaluated planner still controls only the ego vehicle, the simulator still advances the world state with its standard transition and scoring logic, and AWM only changes the motion proposals of selected surrounding agents.

State synchronization and policy roles. At each simulation step t, the simulator provides the current closed-loop state x_t, including the map context, the ego state produced by the evaluated planner, the generated non-ego histories, and the generated future prefix. We encode x_t using the same tokenized autoregressive representation used in the AWM training. The normal sim-agent samples each non-ego token from the background predictive policy π_{pred}. The adversarial sim-agent instead runs the AWM coalition pipeline on the current state: the proposal module constructs a sparse candidate set, the primary selector chooses the main adversarial agent, the pair ranker proposes a complementary partner, and the calibrator decides whether the pair branch is admitted. The selected coalition receives the adversarial role $\rho_i = \texttt{atk}$, while all other agents retain the background role $\rho_i = \texttt{bg}$.

Closed-loop execution. The sim-agent executes a mixed non-ego policy rather than replacing the entire traffic scene with adversarial behavior. For each non-ego agent i, we first generate a normal token and, if i belongs to the admitted coalition A_t, an adversarial token from the role-conditioned AWM. The token finally submitted to the simulator is

$$z^{\mathrm{sim}}_{i,t} = \begin{cases} z^{\mathrm{awm}}_{i,t} \sim \pi_{\mathrm{awm}}(\cdot \mid x_t, \rho_i = \texttt{atk}), & i \in A_t \text{ and } g_{i,t} = 1, \\ z^{\mathrm{norm}}_{i,t} \sim \pi_{\mathrm{pred}}(\cdot \mid x_t, \rho_i = \texttt{bg}), & \text{otherwise}, \end{cases} \tag{34}$$

where $g_{i,t}$ is a feasibility gate defined below. The accepted token is decoded into a short-horizon displacement and heading update, and the simulator applies the same state transition used for the normal learned sim-agent. This design keeps the evaluation controlled: the AWM changes the conditional behavior distribution of selected agents, but it does not alter the ego planner, the map, the vehicle model, the collision checker, or the closed-loop score.

Feasibility-gated adversarial replacement. Before an adversarial token is accepted, the sim-agent checks whether the decoded motion remains within the shared motion vocabulary's plausible region and the simulator's local validity constraints. The gate rejects proposals that introduce implausible

kinematic jumps, immediate map invalidity, static-object conflicts, or degenerate ego-local overlaps that would be better interpreted as simulator artifacts than as adversarial traffic. Formally, the gate can be viewed as

$$g_{i,t} = \mathbb{I}\left[c_{i,t}^{\mathrm{kin}} \le \kappa_{\mathrm{kin}},\ c_{i,t}^{\mathrm{map}} \le \kappa_{\mathrm{map}},\ c_{i,t}^{\mathrm{static}} \le \kappa_{\mathrm{static}},\ c_{i,t}^{\mathrm{ego}} \le \kappa_{\mathrm{ego}}\right], \tag{35}$$

where the four checks summarize kinematic continuity, map consistency, static-object validity, and ego-local sanity constraints. If the gate fails, the agent falls back to the normal sim-agent token for the same state and step. The fallback makes the AWM host an adversarial simulator under feasibility constraints, rather than a mechanism for injecting arbitrary invalid trajectories.

Temporal use of the coalition. The coalition is selected from the current closed-loop state, not from a fixed logged trajectory. This matters because the evaluated planner can change the future interaction geometry, and the AWM must respond to the planner's generated ego behavior. Recomputing the role assignment online lets the adversarial host adapt when the ego yields, accelerates, or changes its path. At the same time, the attack budget $K_{\max} = 2$ and the calibrator prevent the sim-agent from turning every nearby participant into an attacker. Therefore, the closed-loop host remains sparse: it stresses the planner through a small number of behaviorally meaningful agents while leaving the rest of the traffic under the normal predictive policy.

Interpretation of AWM-sim scores. The AWM sim-agent score is a closed-loop evaluation score under an adversarial non-ego traffic host, and it is distinct from both open-loop AWM rollout rewards and ordinary planner closed-loop scores reported in the main benchmark (Table 1). A lower score under the AWM host indicates that the planner is more vulnerable when nearby traffic follows the learned adversarial policy. To verify that score drops are caused by controlled adversarial interaction rather than simulator collapse, we report simulator-side diagnostics in Table 7.

C Theoretical Analysis

This section provides a theoretical justification for the decoupled solver introduced in Section 3. We do not claim an exact neural solution to the full non-convex simultaneous game in Eq. 2; such games are generally intractable and unstable even in simpler smooth two-player settings [29, 9]. Instead, we analyze the implemented two-stage procedure as a local constrained Stackelberg surrogate of the ideal robust game. In this surrogate, Section 3.2 learns a structured adversarial world model around a frozen reference planner, and Section 3.3 computes a regret-sensitive constrained best response inside a reference-anchored trust region. The analysis formalizes four claims: (1) freezing the AWM induces a controlled local approximation error determined by Stage-A optimization error, structured-class coverage, and the planner trust-region radius; (2) role-switched counterfactual rewards are valid difference-reward baselines for attacker-side score-function optimization when they are evaluated as action-independent stop-gradient counterfactuals; (3) regret-CVaR is an appropriate planner-side risk signal because regret removes intrinsic scene difficulty while CVaR emphasizes rare but severe adversarial branches; and (4) nominal retention and the projected safety dual provide conservative local safeguards against obtaining robustness by degrading ordinary driving behavior.

C.1 Notation and Autoregressive Local Stability

For a scene s, let $\Omega_s^\star$ denote the set of all admissible adversarial executions. An element $\omega \in \Omega_s^\star$ specifies both an active coalition $A_s \subseteq \mathcal{V}_s$ satisfying $|A_s| \le K_{\max}$ and the corresponding role-conditioned non-ego rollout law induced by the AWM under the shared tokenized transition operator. Let $\nu_s \in \Delta(\Omega_s^\star)$ be a scene-conditional adversarial environment distribution. For a planner π, define the expected finite-horizon utility

$$J_s(\pi, \nu_s) := \mathbb{E}_{\omega\sim\nu_s}\mathbb{E}_{\tau\sim(\pi,\omega)}\left[\sum_{t=1}^{H}\gamma^{t-1}u_t(x_t, z_t)\right], \tag{36}$$

where the rollout follows the autoregressive dynamics $x_{t+1} = \Gamma(x_t, \boldsymbol{z}_t)$ from Section 2.1. Throughout the analysis, we assume the step utility is bounded:

$$|u_t(x_t, z_t)| \le u_{\max}, \qquad \forall t \in \{1, \dots, H\}. \tag{37}$$

Define the finite-horizon sensitivity constant

$$\Lambda_{H,\gamma} := 2u_{\max}\sum_{t=1}^{H} t\gamma^{t-1} = \begin{cases} 2u_{\max}\dfrac{1-(H+1)\gamma^H + H\gamma^{H+1}}{(1-\gamma)^2}, & \gamma \in (0,1), \\ u_{\max}H(H+1), & \gamma = 1. \end{cases} \tag{38}$$

Let π_{ref} be the frozen reference planner. For the theoretical analysis, we use the stronger statewise trust region

$$\Pi_\epsilon := \left\{\pi : \sup_x D_{\mathrm{KL}}\big(\pi(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)\big) \le \epsilon\right\}. \tag{39}$$

The actual implementation uses a trajectory-level KL penalty, so Eq. 39 should be read as an idealized sufficient condition. We also implicitly restrict attention to planners that are absolutely continuous with respect to π_{ref} on the discrete token support, so that the KL divergence is finite. By Pinsker's inequality, any $\pi \in \Pi_\epsilon$ satisfies

$$\delta(\pi, \pi_{\mathrm{ref}}) := \sup_x D_{\mathrm{TV}}\big(\pi(\cdot \mid x), \pi_{\mathrm{ref}}(\cdot \mid x)\big) \le \sqrt{\epsilon/2} =: \delta_\epsilon. \tag{40}$$

Lemma C.1 (Autoregressive planner sensitivity) *For any fixed scene s, any fixed adversarial environment distribution $\nu_s \in \Delta(\Omega_s^\star)$, and any two planners π, π', the finite-horizon utility satisfies*

$$\big|J_s(\pi, \nu_s) - J_s(\pi', \nu_s)\big| \le \Lambda_{H,\gamma}\,\delta(\pi, \pi'), \tag{41}$$

where

$$\delta(\pi, \pi') := \sup_x D_{\mathrm{TV}}\big(\pi(\cdot \mid x), \pi'(\cdot \mid x)\big). \tag{42}$$

Consequently, for every $\pi \in \Pi_\epsilon$,

$$\big|J_s(\pi, \nu_s) - J_s(\pi_{\mathrm{ref}}, \nu_s)\big| \le \Lambda_{H,\gamma}\delta_\epsilon. \tag{43}$$

Proof. Couple the two rollouts of π and π' against the same sampled environment law $\omega \sim \nu_s$ and the same exogenous randomness. Conditional on identical prefixes up to a given step, the two rollouts have the same state and therefore the same non-ego conditional laws; we couple the non-ego samples identically until the ego token first differs. As long as the generated token prefixes are identical, the scene states are identical because the transition operator Γ is deterministic given the sampled token set. At any step t, conditional on identical prefixes up to $t-1$, a maximal coupling of the ego token distributions gives

$$\mathbb{P}(z_{e,t} \neq z'_{e,t}) \leq D_{\mathrm{TV}}\big(\pi(\cdot \mid x_t), \pi'(\cdot \mid x_t)\big) \leq \delta(\pi, \pi'). \tag{44}$$

Let E_t be the event that the two coupled rollouts have diverged at or before step t. By the union bound,

$$\mathbb{P}(E_t) \leq t\,\delta(\pi, \pi'). \tag{45}$$

On E_t^c, the two-step utilities are identical. On E_t, their difference is at most $2u_{\max}$. Therefore,

$$\begin{aligned} \big|J_s(\pi, \nu_s) - J_s(\pi', \nu_s)\big| &\leq \sum_{t=1}^{H} \gamma^{t-1} \cdot 2u_{\max}\mathbb{P}(E_t) \\ &\leq 2u_{\max} \sum_{t=1}^{H} t\gamma^{t-1}\delta(\pi, \pi') = \Lambda_{H,\gamma}\delta(\pi, \pi'). \end{aligned} \tag{46}$$

Eq. 43 follows from Eq. 40. □

C.2 Local Stackelberg Justification of the Decoupled Stage-A/B Solver

We now analyze the error introduced by freezing Stage A before updating the planner. The ideal adversary class $\Delta(\Omega_s^\star)$ is too large to optimize directly. Stage A restricts it to the structured class induced by the AWM pipeline: `proposal` $\rightarrow$ `primary selector` $\rightarrow$ `conditional pair ranker` $\rightarrow$ `scene calibrator`.

Let $\widehat{\Omega}_s \subseteq \Omega_s^\star$ denote the support covered by this structured pipeline, and define $\mathcal{Q}_s^\star$ as the full set of admissible adversarial environment distributions satisfying the attack budget and realism constraints. Let $\widehat{\mathcal{Q}}_s \subseteq \mathcal{Q}_s^\star$ denote the structured class achieved by our AWM pipeline:

$$\widehat{\mathcal{Q}}_s := \Delta(\widehat{\Omega}_s) \subseteq \mathcal{Q}_s^\star := \Delta(\Omega_s^\star). \tag{47}$$

For a fixed planner π, define the full and structured worst-case utilities:

$$g_s(\pi) := \inf_{\nu_s \in \mathcal{Q}_s^\star} J_s(\pi, \nu_s), \qquad \widehat{g}_s(\pi) := \inf_{\nu_s \in \widehat{\mathcal{Q}}_s} J_s(\pi, \nu_s). \tag{48}$$

Because $\widehat{\mathcal{Q}}_s \subseteq \mathcal{Q}_s^\star$, we have

$$\widehat{g}_s(\pi) \geq g_s(\pi). \tag{49}$$

The structured class may not cover the full adversarial class. We define the scene-level coverage gap as

$$\delta_{\mathrm{cov},s}(\pi) := \widehat{g}_s(\pi) - g_s(\pi) \geq 0. \tag{50}$$

Proposition C.2 (Local error of freezing the Stage-A adversary) *Assume Stage A returns a frozen structured adversary $\widehat{\nu}_s \in \widehat{\mathcal{Q}}_s$ satisfying the reference-planner suboptimality condition*

$$J_s(\pi_{\mathrm{ref}}, \widehat{\nu}_s) \leq \widehat{g}_s(\pi_{\mathrm{ref}}) + \varepsilon_{A,s}. \tag{51}$$

Then, for every planner $\pi \in \Pi_\epsilon$,

$$0 \leq J_s(\pi, \widehat{\nu}_s) - g_s(\pi) \leq \varepsilon_{A,s} + \delta_{\mathrm{cov},s}(\pi_{\mathrm{ref}}) + 2\Lambda_{H,\gamma}\delta_\epsilon. \tag{52}$$

Proof. The lower bound follows immediately from the definition of $g_s(\pi)$:

$$g_s(\pi) = \inf_{\nu_s \in \mathcal{Q}_s^\star} J_s(\pi, \nu_s) \leq J_s(\pi, \widehat{\nu}_s). \tag{53}$$

For the upper bound, add and subtract the reference-planner quantities:

$$J_s(\pi, \widehat{\nu}_s) - g_s(\pi) = \underbrace{J_s(\pi, \widehat{\nu}_s) - J_s(\pi_{\mathrm{ref}}, \widehat{\nu}_s)}_{(I)} + \underbrace{J_s(\pi_{\mathrm{ref}}, \widehat{\nu}_s) - g_s(\pi_{\mathrm{ref}})}_{(II)} + \underbrace{g_s(\pi_{\mathrm{ref}}) - g_s(\pi)}_{(III)}.$$

By Lemma C.1,

$$(I) \le \Lambda_{H,\gamma}\delta_\epsilon. \tag{54}$$

For term (II), Eq. 51 and Eq. 50 give

$$\begin{aligned}(II) &= J_s(\pi_{\mathrm{ref}}, \widehat{\nu}_s) - g_s(\pi_{\mathrm{ref}}) \\ &\le \varepsilon_{A,s} + \widehat{g}_s(\pi_{\mathrm{ref}}) - g_s(\pi_{\mathrm{ref}}) \\ &= \varepsilon_{A,s} + \delta_{\mathrm{cov},s}(\pi_{\mathrm{ref}}).\end{aligned} \tag{55}$$

For term (III), observe that $g_s(\pi)$ is the pointwise infimum of functions $J_s(\pi, \nu_s)$ that are all $\Lambda_{H,\gamma}$-Lipschitz in π by Lemma C.1. The pointwise infimum of uniformly Lipschitz functions is also Lipschitz with the same constant. Thus

$$(III) \le |g_s(\pi_{\mathrm{ref}}) - g_s(\pi)| \le \Lambda_{H,\gamma}\delta_\epsilon. \tag{56}$$

Combining the bounds on (I), (II), and (III) proves Eq. 52. □

Corollary C.3 (Approximate local robust value guarantee) *Define the ideal local robust value*

$$V_\epsilon^\star := \sup_{\pi \in \Pi_\epsilon} \mathbb{E}_{s\sim\mathcal{D}}\big[g_s(\pi)\big]. \tag{57}$$

Let Stage B return a planner $\widehat{\pi} \in \Pi_\epsilon$ that is an ε_B-approximate best response to the frozen Stage-A adversary:

$$\mathbb{E}_s[J_s(\widehat{\pi}, \widehat{\nu}_s)] \ge \sup_{\pi \in \Pi_\epsilon} \mathbb{E}_s[J_s(\pi, \widehat{\nu}_s)] - \varepsilon_B. \tag{58}$$

Let

$$\bar{\varepsilon}_A := \mathbb{E}_s[\varepsilon_{A,s}], \qquad \bar{\delta}_{\mathrm{cov}} := \mathbb{E}_s[\delta_{\mathrm{cov},s}(\pi_{\mathrm{ref}})]. \tag{59}$$

Then

$$\mathbb{E}_s[g_s(\widehat{\pi})] \ge V_\epsilon^\star - \varepsilon_B - \bar{\varepsilon}_A - \bar{\delta}_{\mathrm{cov}} - 2\Lambda_{H,\gamma}\delta_\epsilon. \tag{60}$$

Proof. Taking expectation in Proposition C.2 yields, for every $\pi \in \Pi_\epsilon$,

$$\mathbb{E}_s[g_s(\pi)] \ge \mathbb{E}_s[J_s(\pi, \widehat{\nu}_s)] - \bar{\varepsilon}_A - \bar{\delta}_{\mathrm{cov}} - 2\Lambda_{H,\gamma}\delta_\epsilon. \tag{61}$$

Applying Eq. 61 at $\pi = \widehat{\pi}$ and using Eq. 58,

$$\begin{aligned}\mathbb{E}_s[g_s(\widehat{\pi})] &\ge \mathbb{E}_s[J_s(\widehat{\pi}, \widehat{\nu}_s)] - \bar{\varepsilon}_A - \bar{\delta}_{\mathrm{cov}} - 2\Lambda_{H,\gamma}\delta_\epsilon \\ &\ge \sup_{\pi \in \Pi_\epsilon} \mathbb{E}_s[J_s(\pi, \widehat{\nu}_s)] - \varepsilon_B - \bar{\varepsilon}_A - \bar{\delta}_{\mathrm{cov}} - 2\Lambda_{H,\gamma}\delta_\epsilon \\ &\ge \sup_{\pi \in \Pi_\epsilon} \mathbb{E}_s[g_s(\pi)] - \varepsilon_B - \bar{\varepsilon}_A - \bar{\delta}_{\mathrm{cov}} - 2\Lambda_{H,\gamma}\delta_\epsilon \\ &= V_\epsilon^\star - \varepsilon_B - \bar{\varepsilon}_A - \bar{\delta}_{\mathrm{cov}} - 2\Lambda_{H,\gamma}\delta_\epsilon.\end{aligned} \tag{62}$$

□

Corollary C.4 (Support-mass interpretation of the coverage gap) *Let*

$$U_H := u_{\max} \sum_{t=1}^{H} \gamma^{t-1} \tag{63}$$

so that $|J_s(\pi, \omega)| \le U_H$ for every rollout execution ω. Assume that a full adversarial best response at the reference planner exists, and let $\nu_{s,\mathrm{full}}^\star \in \arg\min_{\nu_s \in \mathcal{Q}_s^\star} J_s(\pi_{\mathrm{ref}}, \nu_s)$. If the infimum is not attained, the same statement holds with an additional arbitrarily small optimality slack by taking an approximate minimizer. Define the omitted support mass

$$\eta_s := \nu_{s,\mathrm{full}}^\star\big(\Omega_s^\star \setminus \widehat{\Omega}_s\big). \tag{64}$$

If $\eta_s < 1$, then

$$\delta_{\mathrm{cov},s}(\pi_{\mathrm{ref}}) \le 2U_H\eta_s. \tag{65}$$

Consequently,

$$\bar{\delta}_{\mathrm{cov}} \le 2U_H\mathbb{E}_{s\sim\mathcal{D}}[\eta_s]. \tag{66}$$

Proof. For each scene s, let $\widetilde{\nu}_s$ be the renormalization of $\nu_{s,\mathrm{full}}^\star$ onto $\widehat{\Omega}_s$. Then $\widetilde{\nu}_s \in \widehat{\mathcal{Q}}_s$ and

$$D_{\mathrm{TV}}(\widetilde{\nu}_s, \nu_{s,\mathrm{full}}^\star) = \eta_s. \tag{67}$$

By the boundedness of the utility,

$$\left|J_s(\pi_{\mathrm{ref}}, \widetilde{\nu}_s) - J_s(\pi_{\mathrm{ref}}, \nu^\star_{s,\mathrm{full}})\right| \le 2U_H\eta_s. \tag{68}$$

Because $\widehat{g}_s(\pi_{\mathrm{ref}}) \le J_s(\pi_{\mathrm{ref}}, \widetilde{\nu}_s)$ and $g_s(\pi_{\mathrm{ref}}) = J_s(\pi_{\mathrm{ref}}, \nu^\star_{s,\mathrm{full}})$,

$$\begin{aligned}\delta_{\mathrm{cov},s}(\pi_{\mathrm{ref}}) &= \widehat{g}_s(\pi_{\mathrm{ref}}) - g_s(\pi_{\mathrm{ref}}) \\ &\le J_s(\pi_{\mathrm{ref}}, \widetilde{\nu}_s) - J_s(\pi_{\mathrm{ref}}, \nu^\star_{s,\mathrm{full}}) \le 2U_H\eta_s.\end{aligned} \tag{69}$$

Averaging over s proves Eq. 66. □

Interpretation. Proposition C.2 and Corollaries C.3–C.4 formalize the role of the decoupled Stage-A/B solver. The frozen AWM objective is optimistic relative to the full worst-case game because it evaluates the planner against one learned structured adversary rather than the entire adversary class. However, freezing the AWM is a local Stackelberg approximation whose error is controlled by four interpretable quantities: the Stage-A optimization error $\bar{\varepsilon}_A$, the Stage-B best-response error ε_B, the structured-class coverage gap $\bar{\delta}_{\mathrm{cov}}$, and the planner trust-region radius ϵ through the $\mathcal{O}(\sqrt{\epsilon})$ term $2\Lambda_{H,\gamma}\delta_\epsilon$. The support-mass bound further explains why the proposal–top1–pair–calibrator design matters: it should cover the low-utility adversarial executions that carry mass under the true worst-case adversary, while avoiding an intractable dense subset search.

C.3 Scene-Adaptive Pair Admission as a Bayes Gate

The selector and pair ranker decide *who* should attack. The calibrator decides *whether* the second attacker should be admitted. This distinction can be formalized as a Bayes decision problem.

For a scene s, let the candidate pair gain be

$$G^{\mathrm{pair}}_s := U_s(\{i^\star_s\}) - U_s(\{i^\star_s, \widehat{j}_s\}). \tag{70}$$

Positive G^{pair}_s means that the pair attack reduces planner utility more than the top-1 attack. Let ψ_s be the scene-level calibrator feature.

Proposition C.5 (Optimal pair-admission rule) *Among all binary admission policies $a(\psi_s) \in \{0, 1\}$, the policy maximizing expected admitted pair gain*

$$\mathbb{E}\left[a(\psi_s)G^{\mathrm{pair}}_s\right] \tag{71}$$

is

$$a^\star(\psi_s) = \mathbb{I}\left\{\mathbb{E}[G^{\mathrm{pair}}_s \mid \psi_s] > 0\right\}. \tag{72}$$

If the calibrator is calibrated for the binary label

$$Y_s := \mathbb{I}\{G^{\mathrm{pair}}_s > 0\}, \qquad p_s = \mathbb{P}(Y_s = 1 \mid \psi_s), \tag{73}$$

and if

$$m_+(\psi_s) := \mathbb{E}[G^{\mathrm{pair}}_s \mid \psi_s, Y_s = 1], \qquad m_-(\psi_s) := \mathbb{E}[-G^{\mathrm{pair}}_s \mid \psi_s, Y_s = 0], \tag{74}$$

then, on feature values where the conditional quantities are well defined and $m_+(\psi_s) + m_-(\psi_s) > 0$, Eq. 72 is equivalent to

$$p_s > \frac{m_-(\psi_s)}{m_+(\psi_s) + m_-(\psi_s)}. \tag{75}$$

When the conditional magnitudes are approximately stable over the deployment distribution, this reduces to a scalar threshold rule $p_s > \tau^\star_{\mathrm{cal}}$, which is the tractable rule used by our scene-adaptive calibrator.

Proof. For any measurable admission policy $a(\psi_s)$,

$$\mathbb{E}[a(\psi_s)G^{\mathrm{pair}}_s] = \mathbb{E}\left[a(\psi_s)\mathbb{E}[G^{\mathrm{pair}}_s \mid \psi_s]\right]. \tag{76}$$

Since $a(\psi_s)$ is binary, the pointwise maximizer admits the pair exactly when the conditional expected gain is positive, proving Eq. 72. Next,

$$\begin{aligned}\mathbb{E}[G^{\mathrm{pair}}_s \mid \psi_s] &= p_s\mathbb{E}[G^{\mathrm{pair}}_s \mid \psi_s, Y_s = 1] + (1 - p_s)\mathbb{E}[G^{\mathrm{pair}}_s \mid \psi_s, Y_s = 0] \\ &= p_s m_+(\psi_s) - (1 - p_s)m_-(\psi_s).\end{aligned} \tag{77}$$

This quantity is positive if and only if Eq. 75 holds. □

Interpretation. The calibrator is not merely a heuristic confidence score. It is a learned approximation to the Bayes gate that decides whether the additional attack dimension has positive expected value in the current scene. This explains why adaptive $K > 1$ is preferable to fixed $K = 2$: pair attacks should be admitted only when their conditional expected marginal gain is positive.

C.4 Role-Switched Counterfactual Credit as a Difference-Reward Estimator

We next justify the role-switched counterfactual rewards used to train the AWM. For this subsection, fix a scene s and a factual coalition $A = A_s^{\text{fact}}$. Let $U_s(A)$ denote the planner's utility under the factual coalition, and define the adversarial team reward

$$R_s(A) := -U_s(A). \tag{78}$$

A larger $R_s(A)$ means a stronger attack.

For an active attacker $i \in A$, let $Y_i = (z_{i,1}, \dots, z_{i,H})$ denote its sampled factual attack-token trajectory, and let

$$\Psi_i := \nabla_\phi^{(i)} \log p_\phi(Y_i \mid \mathcal{H}_i, \rho_i = \texttt{atk}) = \sum_{t=1}^{H} \nabla_\phi \log \pi_{\text{awm},\phi}\big(z_{i,t} \mid x_t, \rho_i = \texttt{atk}\big) \tag{79}$$

be the score-function term associated with attacker i; $\mathcal{H}_i$ denotes the factual pre-sampling history needed to define the conditional trajectory law. When parameters are shared across attackers, the full attacker-side policy-gradient estimator sums Eq. 79 over active attackers.

Let $\text{bg}(A)$ denote the counterfactual rollout obtained by switching all active attackers in A to the background role without removing the agents from the scene. Define the admissible baselines

$$B_i^{\text{loo}} := -U_s(A \setminus \{i\}), \qquad B^{\text{team}} := -U_s(\text{bg}(A)), \tag{80}$$

where both quantities are evaluated by the action-independent counterfactual replay described above. Then

$$\Delta_i^{\text{loo}} = U_s(A \setminus \{i\}) - U_s(A) = R_s(A) - B_i^{\text{loo}}, \tag{81}$$

and

$$\Delta_s^{\text{team}} = U_s(\text{bg}(A)) - U_s(A) = R_s(A) - B^{\text{team}}. \tag{82}$$

Proposition C.6 (Hybrid counterfactual reward is gradient-aligned) *Assume the counterfactual utilities used to form B_i^{loo} and B^{team} are evaluated with stop-gradient, are $\mathcal{F}_i^{\text{cf}}$-measurable, and are conditionally independent of the factual attacker trajectory Y_i given $\mathcal{F}_i^{\text{cf}}$. For weights $\omega_{\text{loo}}, \omega_{\text{team}} \geq 0$, define*

$$\Delta_i^{\text{hyb}} := \omega_{\text{loo}} \Delta_i^{\text{loo}} + \omega_{\text{team}} \Delta_s^{\text{team}}, \qquad W := \omega_{\text{loo}} + \omega_{\text{team}}. \tag{83}$$

Then

$$\mathbb{E}[\Psi_i \Delta_i^{\text{hyb}}] = W \, \nabla_\phi^{(i)} \mathbb{E}[R_s(A)], \tag{84}$$

where $\nabla_\phi^{(i)}$ denotes the policy-gradient contribution associated with attacker i. In particular, if $W > 0$, the hybrid counterfactual reward has the same ascent direction as the adversarial team objective up to the positive scalar W; if $W = 1$, it is an unbiased estimator of that component.

Proof. By the score-function identity applied to the factual attack trajectory,

$$\nabla_\phi^{(i)} \mathbb{E}[R_s(A)] = \mathbb{E}[\Psi_i R_s(A)]. \tag{85}$$

It remains to show that the counterfactual baselines vanish after multiplication by Ψ_i. Because B_i^{loo} is $\mathcal{F}_i^{\text{cf}}$-measurable and conditionally independent of Y_i given $\mathcal{F}_i^{\text{cf}}$,

$$\mathbb{E}[\Psi_i B_i^{\text{loo}}] = \mathbb{E}\big[B_i^{\text{loo}} \mathbb{E}[\Psi_i \mid \mathcal{F}_i^{\text{cf}}]\big]. \tag{86}$$

For the conditional score,

$$\begin{aligned} \mathbb{E}[\Psi_i \mid \mathcal{F}_i^{\text{cf}}] &= \sum_{y_i} p_\phi(y_i \mid \mathcal{F}_i^{\text{cf}}) \nabla_\phi \log p_\phi(y_i \mid \mathcal{F}_i^{\text{cf}}) \\ &= \sum_{y_i} \nabla_\phi p_\phi(y_i \mid \mathcal{F}_i^{\text{cf}}) = \nabla_\phi \sum_{y_i} p_\phi(y_i \mid \mathcal{F}_i^{\text{cf}}) = 0. \end{aligned} \tag{87}$$

Thus $\mathbb{E}[\Psi_i B_i^{\mathrm{loo}}] = 0$. The same argument gives $\mathbb{E}[\Psi_i B^{\mathrm{team}}] = 0$. Therefore,

$$\begin{aligned}\mathbb{E}[\Psi_i \Delta_i^{\mathrm{hyb}}] &= \omega_{\mathrm{loo}} \mathbb{E}[\Psi_i (R_s(A) - B_i^{\mathrm{loo}})] + \omega_{\mathrm{team}} \mathbb{E}[\Psi_i (R_s(A) - B^{\mathrm{team}})] \\ &= (\omega_{\mathrm{loo}} + \omega_{\mathrm{team}}) \mathbb{E}[\Psi_i R_s(A)] = W \nabla_\phi^{(i)} \mathbb{E}[R_s(A)].\end{aligned} \tag{88}$$

□

Lemma C.7 (Variance-optimal action-independent baseline) *Let $\mathcal{F}_{-i}$ be any sigma-field that excludes the factual attacker sample Y_i and satisfies $\mathbb{E}[\Psi_i \mid \mathcal{F}_{-i}] = 0$. Consider any $\mathcal{F}_{-i}$-measurable baseline $b_i(\mathcal{F}_{-i})$. Define the score-weighted second moment*

$$\mathcal{V}(b_i) := \mathbb{E}\left[\|\Psi_i\|_2^2 \big(R_s(A) - b_i(\mathcal{F}_{-i})\big)^2 \right]. \tag{89}$$

On events where $\mathbb{E}[\|\Psi_i\|_2^2 \mid \mathcal{F}_{-i}] > 0$, the unique minimizer among all $\mathcal{F}_{-i}$-measurable baselines is

$$b_i^\star(\mathcal{F}_{-i}) = \frac{\mathbb{E}[\|\Psi_i\|_2^2 R_s(A) \mid \mathcal{F}_{-i}]}{\mathbb{E}[\|\Psi_i\|_2^2 \mid \mathcal{F}_{-i}]}. \tag{90}$$

On events where this conditional second moment is zero, the choice of baseline is immaterial. Moreover, for any other baseline b_i,

$$\mathcal{V}(b_i) = \mathcal{V}(b_i^\star) + \mathbb{E}\left[\mathbb{E}[\|\Psi_i\|_2^2 \mid \mathcal{F}_{-i}] \big(b_i(\mathcal{F}_{-i}) - b_i^\star(\mathcal{F}_{-i})\big)^2 \right]. \tag{91}$$

If $\mathbb{E}[\|\Psi_i\|_2^2 \mid \mathcal{F}_{-i}, R_s(A)] = \mathbb{E}[\|\Psi_i\|_2^2 \mid \mathcal{F}_{-i}]$, then

$$b_i^\star(\mathcal{F}_{-i}) = \mathbb{E}[R_s(A) \mid \mathcal{F}_{-i}]. \tag{92}$$

Proof. Condition on $\mathcal{F}_{-i}$. For any scalar value $b_i = b_i(\mathcal{F}_{-i})$, the conditional objective is

$$\begin{aligned}\mathcal{V}(b_i \mid \mathcal{F}_{-i}) = \mathbb{E}[\|\Psi_i\|_2^2 R_s(A)^2 \mid \mathcal{F}_{-i}] - 2 b_i \mathbb{E}[\|\Psi_i\|_2^2 R_s(A) \mid \mathcal{F}_{-i}] \\ + b_i^2 \mathbb{E}[\|\Psi_i\|_2^2 \mid \mathcal{F}_{-i}].\end{aligned} \tag{93}$$

This is a quadratic in b_i. Differentiating and setting the derivative to zero gives Eq. 90. Completing the square yields Eq. 91. If the score magnitude is conditionally independent of the reward given $\mathcal{F}_{-i}$, the numerator in Eq. 90 factorizes, giving Eq. 92. □

Remark C.8 (Pairwise Shapley reconstruction) *When the admitted coalition has size two, $A = \{i, j\}$, define the cooperative game*

$$v(B) := U_s(\mathrm{bg}(A)) - U_s(B), \qquad B \subseteq A, \tag{94}$$

so that $v(\emptyset) = 0$ and $v(A) = \Delta_s^{\mathrm{team}}$. The Shapley values of the two attackers are

$$\begin{aligned}\phi_i &= \frac{1}{2}\big(v(\{i\}) - v(\emptyset)\big) + \frac{1}{2}\big(v(\{i,j\}) - v(\{j\})\big), \\ \phi_j &= \frac{1}{2}\big(v(\{j\}) - v(\emptyset)\big) + \frac{1}{2}\big(v(\{i,j\}) - v(\{i\})\big).\end{aligned} \tag{95}$$

Using

$$\Delta_i^{\mathrm{loo}} = v(\{i,j\}) - v(\{j\}), \qquad \Delta_j^{\mathrm{loo}} = v(\{i,j\}) - v(\{i\}), \tag{96}$$

we obtain

$$\phi_i = \frac{1}{2}\big(\Delta_s^{\mathrm{team}} + \Delta_i^{\mathrm{loo}} - \Delta_j^{\mathrm{loo}}\big), \qquad \phi_j = \frac{1}{2}\big(\Delta_s^{\mathrm{team}} + \Delta_j^{\mathrm{loo}} - \Delta_i^{\mathrm{loo}}\big). \tag{97}$$

Thus, in the $K_{\max} = 2$ regime used by our method, the team and leave-one-out counterfactuals span the exact fair-credit decomposition of the corresponding pairwise cooperative game.

Interpretation. Proposition C.6 and Lemma C.7 show that role-switched counterfactual rewards are not arbitrary heuristics. They are difference-reward control variates for score-function optimization [17, 15] under the action-independence condition stated above. Because the factual and counterfactual rollouts share the same model, token vocabulary, transition operator, and scene context, the baselines remove nuisance scene variation; because the counterfactual utility is detached and independent of the factual attacker sample, the attacker-side score-function direction is preserved. The team term

further preserves non-additive coalition information, which is essential for learning cooperative pair attacks.

C.5 Regret as a Reference-Anchored Control Variate

We now analyze the planner-side robust objective. Stage B freezes the learned AWM and optimizes the planner against the induced top1/pair branch distribution. Throughout this subsection, the branch identity rule and the calibrated probability p_s are treated as frozen with respect to the planner parameters. Let

$$J_s^b(\pi), \qquad b \in \{\text{top1}, \text{pair}\}, \tag{98}$$

denote the planner utility in scene s under a frozen adversarial branch b. We define the reference-relative adversarial regret as

$$\Delta_s^b(\theta) := J_s^b(\pi_{\text{ref}}) - J_s^b(\pi_\theta). \tag{99}$$

A positive value means that the current planner underperforms the reference planner under the same attack.

Proposition C.9 (Regret preserves branchwise gradient direction and reduces scene difficulty)
For any branch b,

$$\nabla_\theta \mathbb{E}_s[\Delta_s^b(\theta)] = -\nabla_\theta \mathbb{E}_s[J_s^b(\pi_\theta)]. \tag{100}$$

Thus, minimizing regret is branchwise equivalent to maximizing utility. Moreover,

$$\begin{aligned}\text{Var}_s[\Delta_s^b(\theta)] = \text{Var}_s[J_s^b(\pi_\theta)] + \text{Var}_s[J_s^b(\pi_{\text{ref}})] \\ - 2\text{Cov}_s\big(J_s^b(\pi_\theta), J_s^b(\pi_{\text{ref}})\big).\end{aligned} \tag{101}$$

Consequently,

$$\text{Var}_s[\Delta_s^b(\theta)] \le \text{Var}_s[J_s^b(\pi_\theta)] \tag{102}$$

whenever

$$2\text{Cov}_s\big(J_s^b(\pi_\theta), J_s^b(\pi_{\text{ref}})\big) \ge \text{Var}_s[J_s^b(\pi_{\text{ref}})]. \tag{103}$$

Finally, if $\pi_\theta \in \Pi_\epsilon$, then

$$|\Delta_s^b(\theta)| \le \Lambda_{H,\gamma}\delta_\epsilon, \qquad \text{Var}_s[\Delta_s^b(\theta)] \le \Lambda_{H,\gamma}^2\delta_\epsilon^2. \tag{104}$$

Proof. Since π_{ref} is frozen,

$$\nabla_\theta \Delta_s^b(\theta) = -\nabla_\theta J_s^b(\pi_\theta), \tag{105}$$

which proves Eq. 100 after taking expectation over scenes. Eq. 101 follows from

$$\text{Var}(Y - X) = \text{Var}(Y) + \text{Var}(X) - 2\text{Cov}(X, Y), \tag{106}$$

with $Y = J_s^b(\pi_{\text{ref}})$ and $X = J_s^b(\pi_\theta)$. The sufficient condition in Eq. 103 follows immediately.

For Eq. 104, apply Lemma C.1 to the fixed branch environment:

$$|\Delta_s^b(\theta)| = |J_s^b(\pi_{\text{ref}}) - J_s^b(\pi_\theta)| \le \Lambda_{H,\gamma}\delta(\pi_\theta, \pi_{\text{ref}}) \le \Lambda_{H,\gamma}\delta_\epsilon. \tag{107}$$

A random variable bounded in absolute value by c has variance at most c^2, yielding the variance bound. □

Remark C.10 (Generalized control-variate coefficient) *More generally, define*

$$\Delta_{s,\beta}^b(\theta) := \beta J_s^b(\pi_{\text{ref}}) - J_s^b(\pi_\theta). \tag{108}$$

For every $\beta \in \mathbb{R}$,

$$\nabla_\theta \mathbb{E}_s[\Delta_{s,\beta}^b(\theta)] = -\nabla_\theta \mathbb{E}_s[J_s^b(\pi_\theta)]. \tag{109}$$

When $\text{Var}_s(J_s^b(\pi_{\text{ref}})) > 0$, the variance-minimizing coefficient is

$$\beta^\star = \frac{\text{Cov}_s(J_s^b(\pi_\theta), J_s^b(\pi_{\text{ref}}))}{\text{Var}_s(J_s^b(\pi_{\text{ref}}))}. \tag{110}$$

Our implementation uses the simple unit coefficient $\beta = 1$, which is natural when π_θ remains close to π_{ref} under the trust region and the two branch utilities are highly correlated.

Interpretation. Regret does not change the branchwise optimization direction. It changes the cross-scene random variable to which the tail-risk operator is applied. In autonomous driving, absolute utility can be dominated by scene-intrinsic difficulty. If

$$J_s^b(\pi) = d_s^b + \widetilde{J}_s^b(\pi), \tag{111}$$

where d_s^b is independent of the planner, then

$$\Delta_s^b(\theta) = \widetilde{J}_s^b(\pi_{\mathrm{ref}}) - \widetilde{J}_s^b(\pi_\theta), \tag{112}$$

so the intrinsic difficulty term cancels exactly. This explains why CVaR should be applied to regret rather than absolute utility: absolute-utility CVaR can overemphasize intrinsically hard scenes, whereas regret-CVaR focuses on planner-induced degradation relative to the same reference policy.

C.6 Closed-Form CVaR on the AWM-Induced Two-Point Risk Distribution

For each scene, the frozen AWM induces a two-point adversarial branch distribution:

$$\mathcal{D}_s^\theta = \left\{ \left(\Delta_s^{\mathrm{top1}}(\theta), 1 - p_s\right), \left(\Delta_s^{\mathrm{pair}}(\theta), p_s\right) \right\}, \tag{113}$$

where p_s is the calibrated pair-admission probability and is frozen during the planner update. We use upper-tail CVaR with tail mass $\tau \in (0, 1]$:

$$\mathrm{CVaR}_\tau(Z) := \min_{\eta \in \mathbb{R}} \left\{ \eta + \frac{1}{\tau} \mathbb{E}[(Z - \eta)_+] \right\}. \tag{114}$$

This is the standard Rockafellar–Uryasev variational form for upper-tail risk [36], written with tail mass τ.

Proposition C.11 (Closed-form CVaR weights and rare-risk amplification) *Let the two branch regrets be sorted as*

$$\Delta_s^{b_{(1)}}(\theta) \geq \Delta_s^{b_{(2)}}(\theta), \tag{115}$$

and let $\mu_s^{b_{(1)}} \in \{p_s, 1 - p_s\}$ be the probability mass of the worse branch. Then

$$\mathrm{CVaR}_\tau(\mathcal{D}_s^\theta) = w_s^{b_{(1)}}(\tau) \Delta_s^{b_{(1)}}(\theta) + w_s^{b_{(2)}}(\tau) \Delta_s^{b_{(2)}}(\theta), \tag{116}$$

where

$$w_s^{b_{(1)}}(\tau) = \begin{cases} 1, & \mu_s^{b_{(1)}} \geq \tau, \\ \dfrac{\mu_s^{b_{(1)}}}{\tau}, & \mu_s^{b_{(1)}} < \tau, \end{cases} \qquad w_s^{b_{(2)}}(\tau) = 1 - w_s^{b_{(1)}}(\tau). \tag{117}$$

Away from the tie set $\Delta_s^{\mathrm{top1}} = \Delta_s^{\mathrm{pair}}$, the subgradient is

$$\nabla_\theta \mathrm{CVaR}_\tau(\mathcal{D}_s^\theta) = w_s^{\mathrm{top1}}(\tau) \nabla_\theta \Delta_s^{\mathrm{top1}}(\theta) + w_s^{\mathrm{pair}}(\tau) \nabla_\theta \Delta_s^{\mathrm{pair}}(\theta). \tag{118}$$

If the pair branch is the worse branch and $p_s < \tau$, then its CVaR coefficient is p_s/τ. Compared with ordinary expected regret, where the coefficient is p_s, the pair branch receives the amplification factor

$$\frac{p_s/\tau}{p_s} = \frac{1}{\tau}. \tag{119}$$

Proof. Let $z_{(1)} \geq z_{(2)}$ denote the two sorted regret values and let μ be the probability mass of $z_{(1)}$. For $\eta \in \mathbb{R}$, define

$$f(\eta) := \eta + \frac{1}{\tau} \left[\mu (z_{(1)} - \eta)_+ + (1 - \mu)(z_{(2)} - \eta)_+ \right]. \tag{120}$$

We analyze three regions.

If $\eta \geq z_{(1)}$, both hinge terms vanish and $f(\eta) = \eta$, minimized at $\eta = z_{(1)}$ with value $z_{(1)}$.

If $z_{(2)} \leq \eta \leq z_{(1)}$, only the first hinge is active:

$$f(\eta) = \eta + \frac{\mu}{\tau}(z_{(1)} - \eta) = \frac{\mu}{\tau} z_{(1)} + \left(1 - \frac{\mu}{\tau}\right) \eta. \tag{121}$$

If $\mu \geq \tau$, the slope is non-positive, so the minimum over this interval is attained at $\eta = z_{(1)}$, giving value $z_{(1)}$. If $\mu < \tau$, the slope is positive, so the minimum is attained at $\eta = z_{(2)}$, giving

$$\frac{\mu}{\tau} z_{(1)} + \left(1 - \frac{\mu}{\tau}\right) z_{(2)}. \tag{122}$$

If $\eta \leq z_{(2)}$, both hinge terms are active:

$$\begin{aligned} f(\eta) &= \eta + \frac{1}{\tau}\left[\mu(z_{(1)} - \eta) + (1-\mu)(z_{(2)} - \eta)\right] \\ &= \frac{\mu}{\tau} z_{(1)} + \frac{1-\mu}{\tau} z_{(2)} + \left(1 - \frac{1}{\tau}\right)\eta. \end{aligned} \tag{123}$$

Since $\tau \in (0, 1]$, the slope is non-positive, so the minimum over this region is attained at the largest feasible η, namely $\eta = z_{(2)}$. This gives the same value as the previous case when $\mu < \tau$. Combining the cases proves Eq. 116 and Eq. 117. The subgradient expression follows by differentiating the piecewise-linear closed form away from branch ties. The amplification claim follows by setting $\mu = p_s$ when the pair branch is worse and $p_s < \tau$. □

Interpretation. The calibrated Pair attacks are intentionally sparse: in most scenes, the calibrated probability p_s may be small. Expected regret would multiply the pair-branch gradient by p_s, causing rare but severe cooperative failures to be under-optimized. CVaR prevents this dilution. If the rare pair branch is currently the worst branch, it is amplified by exactly $1/\tau$ until it either no longer dominates the tail or its probability mass saturates the entire tail. This is why the planner-side objective is regret-sensitive and CVaR-oriented rather than a simple expectation over AWM samples.

In the implementation, the CVaR weights are treated with stop-gradient. On every region where the branch ordering is fixed, this coincides with one valid subgradient of the piecewise-linear CVaR objective in Eq. 118. At branch ties, the CVaR subdifferential is set-valued; any convex combination of the tied-branch gradients is valid.

C.7 Nominal Retention as a Local Performance Floor

We finally analyze the nominal-retention and safety-constraint mechanisms. Let

$$J_s^{\mathrm{norm}}(\pi) := U_s(\pi, \pi_{\mathrm{pred}}) \tag{124}$$

be the scene-level nominal utility, and define

$$J^{\mathrm{norm}}(\pi) := \mathbb{E}_{s\sim\mathcal{D}}[J_s^{\mathrm{norm}}(\pi)]. \tag{125}$$

Recall the normal utility margin

$$m_s^{\mathrm{norm}}(\theta) := J_s^{\mathrm{norm}}(\pi_\theta) - J_s^{\mathrm{norm}}(\pi_{\mathrm{ref}}) \tag{126}$$

and the asymmetric penalty

$$\psi_s^{\mathrm{norm}}(\theta) := \left[-\epsilon_{\mathrm{norm}} - m_s^{\mathrm{norm}}(\theta)\right]_+^2. \tag{127}$$

Proposition C.12 (Nominal performance floor) *For every $\pi_\theta \in \Pi_\epsilon$ and every scene s,*

$$\left|J_s^{\mathrm{norm}}(\pi_\theta) - J_s^{\mathrm{norm}}(\pi_{\mathrm{ref}})\right| \leq \Lambda_{H,\gamma}\delta_\epsilon. \tag{128}$$

Moreover, the asymmetric margin penalty implies the pointwise bound

$$m_s^{\mathrm{norm}}(\theta) \geq -\epsilon_{\mathrm{norm}} - \sqrt{\psi_s^{\mathrm{norm}}(\theta)}. \tag{129}$$

Taking expectation over scenes gives

$$J^{\mathrm{norm}}(\pi_\theta) \geq J^{\mathrm{norm}}(\pi_{\mathrm{ref}}) - \epsilon_{\mathrm{norm}} - \sqrt{\mathbb{E}_s[\psi_s^{\mathrm{norm}}(\theta)]}. \tag{130}$$

Therefore,

$$J^{\mathrm{norm}}(\pi_\theta) \geq J^{\mathrm{norm}}(\pi_{\mathrm{ref}}) - \min\left\{\Lambda_{H,\gamma}\delta_\epsilon,\ \epsilon_{\mathrm{norm}} + \sqrt{\mathbb{E}_s[\psi_s^{\mathrm{norm}}(\theta)]}\right\}. \tag{131}$$

In particular, if $\psi_s^{\mathrm{norm}}(\theta) = 0$ almost surely, then

$$J_s^{\mathrm{norm}}(\pi_\theta) \geq J_s^{\mathrm{norm}}(\pi_{\mathrm{ref}}) - \epsilon_{\mathrm{norm}} \qquad \text{for almost every scene } s. \tag{132}$$

Proof. Eq. 128 is Lemma C.1 applied to the nominal environment π_{pred}. For the margin bound, define

$$a_s(\theta) := [-\epsilon_{\text{norm}} - m_s^{\text{norm}}(\theta)]_+ = \sqrt{\psi_s^{\text{norm}}(\theta)}. \tag{133}$$

Since $x \le [x]_+$ for all real x,

$$-\epsilon_{\text{norm}} - m_s^{\text{norm}}(\theta) \le a_s(\theta), \tag{134}$$

which rearranges to Eq. 129. Taking expectations,

$$J^{\text{norm}}(\pi_\theta) - J^{\text{norm}}(\pi_{\text{ref}}) = \mathbb{E}_s[m_s^{\text{norm}}(\theta)] \tag{135}$$

$$\ge -\epsilon_{\text{norm}} - \mathbb{E}_s[a_s(\theta)]. \tag{136}$$

By Jensen's inequality,

$$\mathbb{E}_s[a_s(\theta)] \le \sqrt{\mathbb{E}_s[\psi_s^{\text{norm}}(\theta)]}, \tag{137}$$

which proves Eq. 130. Both Eq. 128 and Eq. 130 are valid lower bounds on expected nominal utility, so taking the tighter one gives Eq. 131. If $\psi_s^{\text{norm}}(\theta) = 0$, Eq. 129 reduces to Eq. 132. □

Interpretation. Proposition C.12 is the finite-horizon autoregressive analogue of trust-region and conservative policy-improvement arguments [23, 38]. The KL anchor limits how far the planner can move from the reference behavior, while the asymmetric margin penalty directly controls degradation in normal traffic. Thus robustness is not obtained by unconstrained overfitting to adversarial samples; it is obtained by a controlled local update with an explicit nominal-performance floor.

C.8 Safety Dual Variables as a Projected Max-Violation Surrogate

Finally, we analyze the safety dual update in Section 3.3. For each safety channel m, define the branch-mixed violation gap

$$h_m(\theta) := \mathbb{E}_{s\sim\mathcal{D}}[\bar{v}_{s,m}] - \kappa_m, \qquad \bar{v}_{s,m} := (1-p_s)v_{s,m}^{\text{top1}} + p_s v_{s,m}^{\text{pair}}. \tag{138}$$

In a classical Lagrangian relaxation, one would use nonnegative multipliers $\lambda_m \ge 0$ and maximize $\sum_m \lambda_m h_m(\theta)$. Our implementation instead uses a bounded softmax parameterization

$$[\lambda_R, \lambda_1, \dots, \lambda_M] = \text{softmax}([\alpha_R, \zeta_1, \dots, \zeta_M]), \tag{139}$$

and maximizes

$$\mathcal{L}_{\text{dual}}(\zeta) = \sum_{m=1}^{M} \lambda_m(\zeta) h_m(\theta). \tag{140}$$

For analysis, define $h_R := 0$ for the progress channel.

Proposition C.13 (Softmax safety multipliers as a bounded max-violation surrogate) *For fixed planner parameters θ, the gradient of Eq. 140 with respect to the safety logit ζ_m is*

$$\frac{\partial \mathcal{L}_{\text{dual}}}{\partial \zeta_m} = \lambda_m \left(h_m - \sum_{\ell=1}^{M} \lambda_\ell h_\ell \right). \tag{141}$$

Thus dual ascent increases the relative weight of safety channels whose violation gap exceeds the current softmax-weighted average, and decreases the relative weight of channels below that average. Moreover,

$$\sup_{\zeta\in\mathbb{R}^M} \mathcal{L}_{\text{dual}}(\zeta) = \max\left\{0, \max_{m\in\{1,\dots,M\}} h_m(\theta)\right\}. \tag{142}$$

Proof. The softmax derivative gives

$$\frac{\partial \lambda_\ell}{\partial \zeta_m} = \lambda_\ell(\mathbb{I}\{\ell = m\} - \lambda_m), \qquad \ell \in \{1, \dots, M\}. \tag{143}$$

Therefore,

$$\begin{aligned} \frac{\partial \mathcal{L}_{\text{dual}}}{\partial \zeta_m} &= \sum_{\ell=1}^{M} h_\ell \frac{\partial \lambda_\ell}{\partial \zeta_m} = \lambda_m h_m - \lambda_m \sum_{\ell=1}^{M} \lambda_\ell h_\ell \\ &= \lambda_m \left(h_m - \sum_{\ell=1}^{M} \lambda_\ell h_\ell \right). \end{aligned} \tag{144}$$

This proves Eq. 141.

For Eq. 142, note that the softmax weights range over the relative interior of the simplex over the M safety channels plus the progress channel with value $h_R = 0$. Thus $\mathcal{L}_{\mathrm{dual}}$ is a convex combination of $\{0, h_1, \ldots, h_M\}$ and can approach, but not necessarily attain at finite logits, the maximum of these values. Therefore the supremum is $\max\{0, \max_m h_m\}$. □

Interpretation. The softmax dual is a normalized, bounded surrogate for the classical Lagrangian dual. It does not provide a global KKT certificate for the original non-convex constrained problem, because the multipliers are bounded and normalized. Instead, Proposition C.13 shows the precise optimization effect of the implementation: it behaves like a bounded projected max-violation surrogate, automatically concentrating penalty mass on whichever safety constraint is most violated while keeping the reward scale numerically stable. If one replaces the softmax parameterization with unconstrained nonnegative Lagrange multipliers and the non-convex optimization converges to a local primal-dual saddle point under standard constraint qualifications, then the usual local KKT complementary-slackness conditions apply. Our implementation should be understood as a stable projected approximation to that ideal dual update.

C.9 Summary and Theory-to-Practice Gap

The analysis above provides a formal justification for the proposed self-play fine-tuning framework. Rather than presenting the global solution to the non-convex constrained min-max game in Eq. 2, we aim to demonstrate that the implemented AWM-side and planner-side surrogates are locally aligned with the ideal robust game under explicit approximations.

Theoretical implications. The central game-level conclusion follows from Lemma C.1, Proposition C.2, and Corollaries C.3–C.4. Lemma C.1 establishes a finite-horizon sensitivity bound for the shared discrete autoregressive rollout interface: *when the planner remains inside a token-level trust region around the reference planner, the induced utility changes smoothly under any fixed adversarial environment distribution*. This local stability is the key condition that makes it meaningful to freeze the adversary in the first stage before updating the planner in the second stage. In particular, Corollary C.3 implies the local robust-value guarantee:

$$\mathbb{E}_{s\sim\mathcal{D}}[g_s(\widehat{\pi})] \geq V_\epsilon^\star - \varepsilon_B - \bar{\varepsilon}_A - \bar{\delta}_{\mathrm{cov}} - 2\Lambda_{H,\gamma}\delta_\epsilon, \qquad \delta_\epsilon = \sqrt{\epsilon/2}.$$

Thus, the gap between the planner returned by the decoupled solver and the ideal local robust best response decomposes into four interpretable terms: (1) the adversary's optimization error $\bar{\varepsilon}_A$, (2) the planner's best-response error ε_B, (3) the adversary coverage gap $\bar{\delta}_{\mathrm{cov}}$, and (4) the trust-region drift term $2\Lambda_{H,\gamma}\delta_\epsilon$. Corollary C.4 further interprets $\bar{\delta}_{\mathrm{cov}}$ through omitted support mass, explaining why the proposal-primary-pair-calibrator pipeline should cover plausible low-utility executions while avoiding an intractable exhaustive search over all attacker subsets and token trajectories.

The remaining results justify the main algorithmic components. Proposition C.5 interprets the scene-adaptive pair calibrator as a Bayes admission gate: *the cooperative branch should be activated only when its conditional marginal gain over the primary attacker is positive*. Proposition C.6 and Lemma C.7 show that role-switched counterfactual rewards are *gradient-aligned difference reward control variates* for attacker-side score function optimization; Remark C.8 further shows that, for a two-agent coalition, the role-switched utilities contain sufficient information to *reconstruct the exact pairwise Shapley credit*. On the planner side, Proposition C.9 shows that reference-relative regret *preserves the branchwise optimization direction while removing scene-intrinsic difficulty* from the risk statistic. Proposition C.11 derives the exact CVaR weights for the frozen two-point top1/pair regret distribution and explains the rare-risk amplification effect: *a sparse cooperative pair attack can receive large optimization weight when it exposes the worst current vulnerability*. Finally, Proposition C.12 gives an explicit lower bound on nominal performance induced by the trust-region anchor and the asymmetric normal margin penalty, while Proposition C.13 characterizes the softmax safety multipliers as a bounded projected max-violation surrogate.

Theory-to-practice gap. The theory intentionally analyzes an idealized version of the implemented solver. We summarize the main gaps below to clarify the precise scope of the guarantees.

- *Local Stackelberg approximation.* Proposition C.2 is a local result around π_{ref}. It justifies freezing the learned AWM and then computing a constrained best response only inside the trust region Π_ϵ.

It does not establish convergence of simultaneous non-convex min-max updates, nor does it imply that the learned AWM is a global worst-case adversary for every possible planner.

- *Structured adversary coverage.* The full adversary class $\mathcal{Q}_s^\star$ is replaced by the structured AWM class $\widehat{\mathcal{Q}}_s$, induced by proposal pruning, primary selection, conditional pair ranking, and scene-level calibration. The coverage gap $\delta_{\mathrm{cov},s}$ captures adversarial executions that are feasible in the ideal game but not represented by the top1 (pair) support of AWM. In practice, this term is not directly observable. It is controlled only indirectly through the expressiveness and proposal coverage of the AWM pipeline.
- *Approximate optimization.* The error terms ε_A and ε_B aggregate neural approximation error, finite rollout noise, stochastic policy-gradient error, proposal pruning error, and selector-ranker-calibrator error. They should therefore be interpreted as analytical proxies for practical approximation quality.
- *Counterfactual credit with Monte Carlo role switching.* Proposition C.6 is exact for score-function gradients only when the counterfactual utilities serve as action-independent stop-gradient baselines. Accordingly, our implementation estimates these baselines using independent role-switched Monte Carlo rollouts under the same AWM, token vocabulary, transition operator, and scene context. This preserves the intended gradient-alignment interpretation under the independence condition, at the cost of Monte Carlo variance and dependence on rollout quality.
- *Surrogate planner losses.* Proposition C.11 analyzes the exact CVaR of the frozen two-point regret distribution $\{\Delta_s^{\mathrm{top1}}, \Delta_s^{\mathrm{pair}}\}$. The implemented optimizer detaches the CVaR weights for stability and optimizes GRPO-style policy-gradient surrogate losses rather than differentiating through every rollout dependency. When the branch ordering is fixed, the detached weight corresponds to a valid subgradient of the piecewise-linear CVaR objective; at branch ties, the subgradient is set-valued, and the implementation selects a stable representative.
- *Ideal trust region and empirical regularization.* Proposition C.12 uses a stronger statewise KL trust region than the empirical trajectory-level KL penalty used in training. Therefore, the nominal performance bound should be interpreted as a sufficient condition explanation of why reference anchoring and the asymmetric normal margin reduce the drift of the planner.
- *Projected safety dual.* Proposition C.13 characterizes the softmax safety multipliers as a normalized and bounded projected dual surrogate. This update concentrates penalty mass on the most violated safety channel while keeping the reward scale numerically stable. However, because the multipliers are bounded and the planner optimization is non-convex, the result does not imply global KKT optimality or guaranteed satisfaction of every safety constraint at every training iterate.

D Detailed Experimental Setups

D.1 Training Data, Computation, and Hyperparameters

Datasets. All models are trained on the nuPlan training split, and the benchmark splits used for Table 1 are held out from training. The base autoregressive model follows the Plan-R1 training recipe [44]. We first pretrain the traffic prediction branch with next-token maximum likelihood on 1,000,000 nuPlan training scenarios, split into 900,000 training scenarios and 100,000 validation scenarios. This stage learns the shared motion-token vocabulary interface and the naturalistic predictive prior used by the later planner and world-model branches. The nominal planner reference is then initialized from this pretrained model and fine-tuned via GRPO [40, 44] on 100,000 planning scenarios, split into 90,000 training scenarios and 10,000 validation scenarios. The proposed AWM fine-tuning uses the same 90,000/10,000 planning split for both stages: Stage A freezes the nominal planner and trains the role-conditioned adversarial world model, while Stage B freezes the learned AWM opponent distribution and fine-tunes the ego planner as the constrained robust best response.

Existing assets and licenses. We use the nuPlan dataset and devkit [3], InterPlan [18], and cited baseline planners only for research benchmarking. The nuPlan devkit is released under the Apache-2.0 license, while the nuPlan dataset is listed under a commercial data license and is accessed under the official Motional/nuPlan distribution terms; raw nuPlan logs are not redistributed. The InterPlan repository states that its code and assets are under Apache-2.0 unless otherwise specified, and that inherited nuPlan data remain governed by their own distribution licenses.

Hyperparameters. The shared autoregressive architecture uses a 1024-token motion vocabulary, a 2s history window, and an 8s rollout horizon at 10Hz. Unless otherwise stated, the encoder has hidden dimension 128, 6 attention layers, 8 attention heads, 4 message-passing hops, a 60m agent radius, a 30m map-polygon radius, dropout 0.1, and weight decay 10^{-4}. Prediction pretraining uses learning rate 3×10^{-4}, warmup for 4 epochs, cosine scheduling, and 32 total epochs. Nominal planner initialization uses learning rate 4×10^{-6} and 5 epochs, following [44]. The AWM and planner fine-tuning stages use GRPO-style rollouts [40] with 4 samples per scene, clipping parameter 0.25, and top-50 token sampling for rollouts. The scene utility uses the reward weights described in Section 3: comfort 2, time-to-collision 5, speed-limit compliance 2, and progress 1. For Stage A, the planner is kept fixed and only the AWM-side parameters are updated. We use adversarial budget $K_{\max} = 2$, a proposal pool of 5 candidate attackers, and up to 5 candidate partners for pair ranking. The AWM policy learning rate is 4×10^{-6}, and the role-embedding learning rate is 2×10^{-5}. Stage A is trained for up to 8 epochs on the 90,000-scenario training split. For Stage B, the AWM is frozen and the planner is optimized with learning rate 10^{-5}. The regret-CVaR tail mass is $\tau = 0.25$, the regret-CVaR weight is 1.0, the normal-branch policy weight is 1.0, the reference KL weight is 0.1, and the progress-retention weight is 10.0. The planner-side safety thresholds are 0.05 for agent collision, 0.03 for static-object collision, and 0.02 for off-road violation, with dual learning rate 10^{-3}. Stage B is trained for up to 6 epochs on the same 90,000-scenario training split.

Computing Platform. The reported AWM fine-tuning runs are executed with distributed data parallel training on a single 8-GPU node with NVIDIA H200 GPUs. Stage A and Stage B use per-GPU batch size 32, giving an effective batch size of 256 scenes. The nominal prediction and planner pretraining stages are lighter and can be reproduced with the same architecture using 2 GPUs, with per-GPU batch sizes 32 for prediction pretraining and 8 for nominal planner initialization. Table 3 reports relative GPU-hours, plateau update counts, seed stability, and seed-wise closed-loop variability for the compared training paradigms; runtime is reported relatively because simulator backend and cluster scheduling affect absolute wall-clock time.

D.2 Evaluation Setup and Protocol

This appendix is organized by experimental protocol: closed-loop benchmarks and sim-agent evaluation, dual-rollout diagnostics, AWM-side mechanism ablations, planner-side adaptation ablations, and adapted prior-generator comparisons. Our experiments use three different evaluation protocols. Ordinary closed-loop planner evaluation, used in Table 1 and in the closed-loop score columns of Tables 3-15, measures the ego planner in the benchmark simulator and reports the official weighted score on a percentage scale. Dual rollout evaluation, used in Tables 8, 9, and 16, evaluates paired

open-loop rollouts under controlled world model branches and reports the planner's utility. Sim-agent evaluation, used in Tables 10-13, 2, and 7, replaces the non-ego traffic host with a learned generator and is treated as an adversarial stress test in the closed-loop simulator. In particular, an AWM-sim score (attack strength) is not directly comparable to an ordinary benchmark score (planning performance).

All open-loop rewards are reported on the same normalized scene-utility scale, where higher is better for the ego planner. A drop is always computed against the paired nominal branch on the same scene bank, so larger drops indicate stronger adversarial degradation only within the same table. Tail-CVaR is the mean utility over the 20% of scenes with the lowest rewards unless the table states otherwise; Table 10 defines Tail Drop using the fixed paired nominal reference, and LCB denotes the scene mean minus 1.64 standard errors. For realism and validity diagnostics, ADE/FDE, kinematic violation, off-road/static-object collision, and background collision are lower-is-better; benchmark submetrics such as progress, collision, time-to-collision, drivable-area compliance, speed-limit compliance, and driving-direction compliance are higher-is-better. All table-level comparisons are paired by scene whenever the table compares branches, hosts, or ablations.

D.2.1 Closed-Loop and Sim-Agent Evaluation

For `nuPlan` [3], Table 1 evaluates each planner on Val14, Test14-hard, and Test14-random in both non-reactive (NR) and reactive (R) modes. For `InterPlan` [18], Table 1 reports the official benchmark and InterPlan10 when available, and reports `InterPlan-LongTail` separately because it is a custom stress suite constructed in Appendix D.3. The transferable sim-agent evaluation in Table 2 uses Test14-hard and changes only the non-ego traffic host. The NORMAL host samples from the learned non-adversarial predictive policy. The AWM host activates the scene-adaptive adversarial coalition described in Appendix B.4, with at most two adversarial non-ego agents at a time. If an evaluated planner fails to complete a scenario, the scenario remains in the benchmark denominator and contributes zero to the closed-loop score. Simulation diagnostics such as fallback rate and active-adversary count are then averaged over the scenarios whose rollouts completed.

D.2.2 Dual-Rollout Open-Loop Evaluation

Dual rollouts are used to isolate the world model from full simulation. For a fixed scene and planner, we first evaluate a nominal branch under the benign predictive host. We then evaluate adversarial branches under a frozen AWM host. The TOP-1 branch activates only the primary selected attacker, the PAIR branch activates the primary attacker and the ranked secondary attacker, and the ADAPTIVE branch applies the scene-level admission probability to decide whether the pair should be used. Table 8 uses this paired design as a pre/post cross-validation matrix. The pre-planner is the nominal reference planner before AWM-based robust adaptation, and the post-planner is the selected AWM-trained planner. The pre-world column evaluates the nominal branch under the benign world model. The post-world column evaluates the ADAPTIVE adversarial branch under a frozen held-out AWM. Table 9 uses the same scene bank and branch definitions. We report both utility and diagnostic proxies for these dual-rollout experiments. Utility summarizes the ego planner's objective, while ADE/FDE, kinematic validity, diversity, off-road/static-object violations, and background collisions audit whether the non-ego generator remains a plausible world model.

D.2.3 AWM Mechanism Ablations

Tables 10, 11, 12, and 13 are AWM-side mechanism tests. They keep the ego planner (Plan-R1 or AWM-Planner), scene bank, reward definition, and rollout interface fixed, and change only the mechanism of AWM. Each row is therefore a controlled variant of the same AWM pipeline.

Table 10 tests the credit assignment mechanism. The GLOBAL-REWARD BROADCAST baseline assigns the same scene-level adversarial reward to all candidate attackers. The HEURISTIC-ONLY selector uses fixed threat statistics without learned counterfactual credit. The ASSIGNMENT-ONLY selector learns a primary attacker assignment without role-switched marginal utilities. The NO-PAIR-RANKER variant keeps the primary attacker mechanism but removes the learned secondary-attacker ranking stage. Tables 11 and 12 test coalition formation. FIXED $K{=}1$ disables cooperative attacks; FIXED $K{=}2$ forces the two-agent coalition whenever a partner is available; and the ADAPTIVE CALIBRATED setting admits the pair only when the scene-level gate predicts positive marginal value. The NO-PAIR-RANKER row removes learned partner ordering, the NO-SCENE-CALIBRATOR row removes

adaptive pair admission, and the REDUCED/FULL PROPOSAL-BUDGET variants test whether candidate pool coverage changes the result. Table 13 tests whether adversarial stress comes from coherent role-conditioned rollout or from post-hoc trajectory insertion. EXPLICIT REPLACEMENT swaps in adversarial predictions more directly, INACTIVE-REVERT overwrites inactive agents with nominal behavior after generation, and SAFE-BACKGROUND injection keeps a conservative background while inserting adversarial proposals.

D.2.4 Planner Mechanism Ablations

Planner-side ablations evaluate the robust adaptation stage after the adversarial host has been learned. Table 3 compares four training recipes under the same evaluation suite. PLANNER-ONLY refinement updates the ego planner without an adversarial world-model opponent. SIMULTANEOUS ALTERNATING SELF-PLAY updates the adversary and planner in one moving loop. ALTERNATING WARMUP WITH FIXED-AWM REFINEMENT first performs a short alternating phase and then freezes the adversary for planner refinement. The DECOUPLED TWO-STAGE PIPELINE learns the AWM around a frozen reference planner, freezes the learned opponent distribution, and then trains the constrained robust best response. Table 14 changes only the frozen world model used inside inference-time rollout. The NOMINAL HOST removes internal adversarial predictions, the MATCHED AWM is the host used by the main method, and the HELD-OUT AWM tests mismatch robustness. The deployed ego planner and the benchmark scenario lists are otherwise fixed, so the closed-loop columns measure sensitivity to the internal opponent model. Table 4 varies the planner's training objective while keeping the AWM fixed. The EXPECTED-ONLY variant replaces tail-risk optimization with the expected adversarial branch objective; the NO-CVAR variant removes the tail allocation; and the NO-NORMAL-RETENTION variant removes the nominal preservation constraint.

D.2.5 Baseline Adversarial Generator Adaptation

Table 16 adapts CAT- [52], STRIVE- [35], and KING [19]-style adversarial generators to the same dual-rollout interface used by AWM. Note that this is an adapted comparison rather than a strict reproduction of the original systems. The adapted generators share the same initial scene state, tokenized transition operator, ego planner, reward, and diagnostic metrics, so the comparison focuses on adversarial generation strategy and search budget. CAND.=1 samples a single adversarial rollout without reward-guided reranking, measuring the generation quality without a search advantage. BEST-4 samples four candidates and selects the rollout with the lowest planner reward, matching a stronger reward-guided search budget. The STRICT-CONSTRAINT protocol increases behavioral prior and realism penalties for adapted prior generators, measuring how much attack strength remains when generated traffic is forced closer to plausible behavior. KING-STYLE kinematic search is included as an aggressive raw-attack upper bound as its realism proxies degrade.

D.3 Construction of the `InterPlan-LongTail` Benchmark

Base benchmark. `InterPlan` is a closed-loop motion-planning benchmark built on top of the `nuPlan` dataset and simulator. It starts from real-world `nuPlan` [3] scenarios and constructs highly interactive driving cases by injecting additional traffic participants or static objects, modifying navigation goals, and varying behaviors of surrounding agents [18]. The resulting benchmark is designed to evaluate whether planners that perform well on common closed-loop `nuPlan` scenarios can generalize to safety-critical interactions, such as jaywalking pedestrians, accident sites, construction zones, obstacle overtaking, and lane changes under different traffic densities. `InterPlan` also exposes a configuration-level scenario-modification interface, which allows users to instantiate customized scenarios by specifying base `nuPlan` tokens, traffic-density suffixes, navigation-goal suffixes, behavior-policy suffixes, and special-scenario edits [18]. We use this interface to construct `InterPlan-LongTail`, a frozen auxiliary stress-test suite for evaluating robustness under dense interactions, such as pedestrian yielding and near-conflict maneuvers.

Benchmark objective. `InterPlan-LongTail` serves as a controlled diagnostic benchmark for testing whether a planner remains reliable when the interaction structure is shifted toward recoverable but safety-critical rare scenarios. This design is aligned with the goal of evaluating adversarial robustness: the benchmark emphasizes close-range conflicts, dense-yield decisions, and pedestrian-triggered reaction requirements, while avoiding simulator artifacts that would make the task trivially

unsolvable for most methods. All results on this benchmark are therefore reported separately from the official InterPlan and InterPlan10 scores.

Configuration and construction. The benchmark is constructed entirely through InterPlan's official configuration interface. We do not modify the nuPlan simulator, the InterPlan simulation logic, or the metric engine. Each scenario is defined by a frozen scenario-token list and a paired modification specification, which together determine the navigation goal, traffic density, surrounding-agent behavior, and special-scenario edits. Following the token convention of InterPlan, each modified scenario token is parsed into a base nuPlan scenario token and a suffix describing the applied modification. The density codes dl, dm, and dh denote low-, medium-, and high-density traffic, respectively; gl, gr, and gs denote left-turn, right-turn, and straight-route goals; and oa, om, and oc denote assertive, mixed, and cautious surrounding agent behavior. Customized special scenarios are indexed by zero-padded suffixes s00–s07. For example, the configuration token dhgsoa corresponds to a high-density, straight-goal, assertive-agent setting. The final benchmark contains ten base scenario anchors, each instantiated under eight interaction templates, yielding $10 \times 8 = 80$ frozen scenarios.

Scenario templates. Table 5 summarizes the composition of the benchmark. All 80 entries use the jaywalking_pedestrian special scenario type and contain at least one injected pedestrian. The templates cover three close-range straight-driving jaywalker variants with different surrounding-agent behavior, two high-density turning-yield cases, two medium-density yielding cases, and one sequential dual pedestrian yielding case. This design concentrates the benchmark on interaction patterns where robustness failures are safety-relevant but not purely caused by blocked roads or static obstacles. To evaluate robustness in recoverable interactive scenarios rather than stress-test limitations of the simulator, we exclude cone- and stopped-object-dominated categories from the evaluation suite and focus on interactions that require yielding. These scenarios remain difficult while preserving meaningful planner-dependent variation. To reduce construction artifacts, the modification specification deliberately excludes stopped_vehicle, stopped_bus, and roadblock_locations_to_spawn_agents in the custom entries. We also require every injected participant to have an explicit activation iteration of at least one, which avoids an edge case in which participant activation may be evaluated before the ego lane is fully available during simulation propagation.

Table 5: Composition of InterPlan-LongTail. The suite contains 80 frozen scenarios, organized into eight interaction templates with ten scenarios per template.

Template	Main stress factor	Count
close_jaywalker_straight_assertive	High-density straight driving with assertive pedestrian interaction	10
close_jaywalker_straight_mixed	High-density straight driving with mixed surrounding-agent behavior	10
close_jaywalker_straight_cautious	High-density straight driving with cautious surrounding-agent behavior	10
left_turn_pedestrian_yield_assertive	High-density left-turn yielding to a pedestrian	10
right_turn_pedestrian_yield_assertive	High-density right-turn yielding to a pedestrian	10
medium_left_yield_assertive	Medium-density left-turn yielding	10
medium_straight_yield_assertive	Medium-density straight yielding	10
dual_pedestrian_chain_yield	Sequential pedestrian-yielding pressure	10

E Supplementary Results

This section provides the complete empirical evidence underlying Section 4. We first report the `InterPlan-LongTail` breakdown and simulator-side transfer diagnostics, and then give detailed analyses of planner-world cross-validation, AWM mechanisms, planner adaptation, adapted prior generators, and qualitative visualizations. All experimental protocols and metric definitions are specified in Appendix D. The discussion below focuses on the additional conclusions supported by each table or figure.

E.1 Closed-Loop Long-Tail Breakdown

Table 6 decomposes the `InterPlan-LongTail` score by scenario template. Scores use the same percentage scale as Table 1. The suite composition is described in Appendix D.3 and summarized in Table 5. The per-template breakdown shows that AWM-Planner improves over Plan-R1 on seven of eight templates. The largest gains appear in close straight-driving jaywalker and medium-density yielding cases, which are precisely the settings where earlier recovery from interactive conflicts is valuable. The only negative row is `left_turn_yield`, where the difference is small. Thus, the long-tail result shows that AWM improves a broad set of recoverable interaction failures.

Table 6: Per-template scores on `InterPlan-LongTail`. Each template contains ten scenarios. Δ is AWM-Planner minus Plan-R1. Higher is better.

Template	Plan-R1	AWM-Planner	Δ
`close_straight_assertive`	33.30	42.02	+8.72
`close_straight_cautious`	33.30	42.08	+8.78
`close_straight_mixed`	33.30	41.19	+7.90
`dual_pedestrian_chain`	25.18	26.12	+0.94
`left_turn_yield`	52.56	51.87	-0.69
`medium_left_yield`	31.17	37.95	+6.78
`medium_straight_yield`	26.83	34.05	+7.22
`right_turn_yield`	43.50	50.34	+6.84
Overall	34.89	**40.70**	**+5.81**

E.2 Adversarial Sim-Agent Diagnostics

Table 7 reports the simulator-side diagnostics used to examine the AWM sim-agent interface described in Appendix B.4. The fallback rate is the fraction of proposed sim-agent updates that revert to the normal policy after feasibility checks. The ego-gate rate measures how often the stricter ego-local guard suppresses an adversarial proposal. The active-adversary count measures the coalition size under the attack budget $K_{\max} = 2$, and budget utilization is the active-adversary count divided by $K_{\max}$. These diagnostics complement the score table in the main text: AWM consistently uses nearly the full two-agent budget, but its high fallback rate shows that the adversarial host is substantially constrained by the feasibility gate. The ego-local gate remains low across planners, indicating that the observed score drop is mainly caused by feasible interaction pressure rather than by frequent emergency suppression around the ego.

E.3 Planner-World Cross-Validation Details

Table 8 reports the full planner and world matrix summarized in Figure 2(a). The post-trained planner improves under both the normal and adversarial worlds, but the increase is larger under AWM, especially for Tail-CVaR. This confirms that Stage B does not merely shift the planner toward a different nominal policy. It specifically improves the lower tail exposed by the adversarial model.

Table 9 decomposes the adversarial branch used in the planner-world cross-validation of Section 4. The training host produces moderate drops of 0.012–0.020, while the held-out AWM increases the drops to 0.025–0.040. Pair and Adaptive branches expose larger lower-tail risk than the single-agent Top-1 branch, with Adaptive remaining less aggressive than forced Pair because it controls

Table 7: Sim-agent behavior diagnostics for the cross-planner evaluation. FALLBACK is the rate of reverting to the normal policy after feasibility checks, EGO GATE is the ego-local safety-suppression rate, ACTIVE ADV. is the actual number of adversarial non-ego agents, and BUDGET UTIL. is active adversaries divided by the two-agent budget. FALLBACK, EGO GATE, and BUDGET UTIL. are reported as percentages, while ACTIVE ADV. and MAX ADV. are counts. Gray cells mark Normal reference diagnostics, and blue cells mark AWM-specific adversarial-budget diagnostics.

Planner	World	Fallback ↓	Ego gate ↓	Active adv.	Max adv.	Budget util.
Diffusion Planner [54]	Normal	6.35	0.34	–	–	–
	AWM	50.07	0.33	1.93	2.00	96.45
IDM+Mobil	Normal	6.32	0.37	–	–	–
	AWM	50.35	0.61	1.92	2.00	96.19
PDM-Closed [10]	Normal	6.25	0.43	–	–	–
	AWM	49.98	0.29	1.92	2.00	96.22
PLUTO [7]	Normal	6.40	0.36	–	–	–
	AWM	50.47	0.36	1.94	2.00	96.91
PlanTF [8]	Normal	6.48	0.27	–	–	–
	AWM	50.20	0.26	1.92	2.00	96.16
Plan-R1 [44]	Normal	6.08	0.18	–	–	–
	AWM	50.24	0.25	1.92	2.00	96.22
AWM-Planner	Normal	6.19	0.23	–	–	–
	AWM	49.80	0.20	1.93	2.00	96.28

Table 8: Pre/post planner × pre/post world-model offline matrix. Avg. reward and progress are scene means. Tail-CVaR is the average reward over the worst 20% scenes under the adversarial rollout.

	Pre-world: normal WM		Post-world: AWM	
Planner	Nominal R ↑	Progress ↑	Adversarial R ↑	Tail-CVaR ↑
Pre-planner	0.980221	0.942985	0.967874	0.865963
Post-planner	0.984197	0.958129	0.979767	0.909490
Δ Post–Pre	+0.003976	+0.015144	+0.011893	+0.043527

the behavior through scene-level admission. This pattern supports the use of calibrated coalition admission in the training risk distribution.

Table 9: Branch-level diagnostic via dual-rollout. LCB denotes mean minus 1.64 standard errors.

Host	Branch	Avg. reward ↑	Δ vs. nominal	Reward LCB ↑	Tail-CVaR ↑
Training host	Nominal	0.9842	–	0.9830	–
	Top-1	0.9718	-0.0124	0.9703	0.8520
	Pair	0.9645	-0.0197	0.9628	0.8305
	Adaptive	0.9668	-0.0174	0.9652	0.8421
Held-out AWM	Nominal	0.9836	–	0.9823	–
	Top-1	0.9584	-0.0252	0.9566	0.8015
	Pair	0.9438	-0.0398	0.9417	0.7610
	Adaptive	0.9489	-0.0347	0.9469	0.7802

E.4 AWM Mechanism Ablations

Tables 10–13 provide the detailed mechanism evidence for the AWM design summarized in Section 4. The credit-assignment ablation in Table 10 shows that role-conditioned counterfactual credit gives stronger tail stress than coarse global reward or assignment-only variants while avoiding the severe trajectory-fidelity collapse of coarse baselines.

The coalition ablations in Tables 11 and 12 separate raw attack strength from pair-admission control. Forced $K=2$ and no-calibrator variants can create stronger raw stress, but they admit pair attacks in most scenes and degrade behavior realism. The adaptive calibrated variant keeps pair usage sparse

Table 10: Credit-assignment analysis. Tail Drop is $R_{\text{nominal-ref}} - \text{TailCVaR}_{20\%}(R_{\text{adversarial}})$, where $R_{\text{nominal-ref}}$ is the fixed paired nominal reference and $\text{TailCVaR}_{20\%}$ is the mean adversarial reward over the 20% of scenes with the lowest rewards (TAIL R). The reference here is 0.9842, giving $0.9842 - 0.7764 = 0.2078$ for full AWM. MEAN R is the mean rollout reward for each row; larger Tail Drop indicates stronger degradation of the planner's lower tail. PRED. PAIR GAIN is the learned proxy for pair-coalition utility, and DIV. denotes trajectory diversity. AWM-SIM is the Test14-hard closed-loop score of the original Plan-R1 planner when AWM is used in the simulator as the non-ego policy; lower values indicate stronger closed-loop attacks. This sim-agent score is distinct from ordinary planner closed-loop scores.

Mode	Mean R	Worst R	Tail R	Drop ↑	Worst Drop ↑	Tail Drop ↑	Pred. pair gain ↑	AWM-sim ↓	ADE ↓	FDE ↓	Div.
Nominal reference	0.9842	0.9842	0.9262	0.0000	0.0000	0.0580	–	75.86	1.035	3.285	0.000
Role-conditioned AWM	0.9778	0.9342	0.7764	0.0064	0.0500	0.2078	0.132	71.71	1.55	4.95	1.75
Global-reward broadcast	0.9814	0.9740	0.8970	0.0028	0.0102	0.0872	0.014	75.10	11.80	18.60	15.20
Heuristic-only selector	0.9810	0.9635	0.8500	0.0032	0.0207	0.1342	0.058	73.20	1.77	5.58	2.10
No pair ranker	0.9804	0.9648	0.8425	0.0038	0.0194	0.1417	0.052	73.70	1.44	4.63	1.58
Assignment-only selector	0.9805	0.9590	0.8612	0.0037	0.0252	0.1230	0.021	74.20	19.50	29.00	26.70

while retaining most of the sim-agent attack effect. These results show that pair attacks are useful but should be selected conditionally.

Table 11: Coalition structure ablations. MEAN R and Tail Drop follow the definitions in Table 10. PAIR FRAC. is the fraction of scenes with an admitted two-agent coalition, and SEL./SCENE is the average number of selected attackers.

Mode	Mean R	Worst R	Tail R	Sel./scene	Pair frac.	Pred. pair gain	Tail Drop ↑	AWM-sim ↓	ADE ↓	FDE ↓
Adaptive calibrated	0.9778	0.9342	0.7764	1.34	0.34	0.132	0.2078	71.71	1.55	4.95
Fixed K=1	0.9812	0.9630	0.8720	1.00	0.00	0.000	0.1122	74.40	1.38	4.42
Fixed K=2	0.9776	0.9380	0.7560	2.00	1.00	0.118	0.2282	69.60	2.35	7.40
No pair ranker	0.9807	0.9580	0.8340	1.76	0.76	0.047	0.1502	73.60	1.45	4.70
No scene calibrator	0.9785	0.9410	0.7680	1.82	0.82	0.116	0.2162	71.40	2.10	6.75
Reduced proposal budget	0.9810	0.9620	0.8520	1.18	0.18	0.034	0.1322	74.20	1.42	4.60
Full proposal budget	0.9798	0.9485	0.7920	1.40	0.40	0.135	0.1922	71.80	1.58	5.05

Table 12: Selector/ranker/calibrator diagnostics. $|P_s|$ is the proposal-pool size and $|R_s|$ is the runner-up pool used for pair ranking. PAIR CAND. and PAIR ADMIT denote pre-calibration candidate and final admitted pair-scene fractions. ADM. PRED/LABEL compares the predicted admission rate with the positive-pair label rate, GAIN PRED/ORACLE compares learned and counterfactual pair gain, and CALIB. GAP is the absolute admission-rate mismatch.

Mode	$\lvert P_s\rvert$	$\lvert R_s\rvert$	Sel./scene	Pair cand.	Pair admit	Adm. pred/label	Gain pred/oracle	Calib. gap
Adaptive calibrated	3.09	2.09	1.34	0.72	0.34	0.36/0.34	0.132/0.120	0.02
Fixed K=2	3.11	2.10	2.00	1.00	1.00	1.00/0.34	0.118/0.074	0.66
No scene calibrator	3.08	2.09	1.82	0.82	0.82	0.82/0.34	0.116/0.071	0.48

Table 13 further tests whether adversarial stress arises from coherent role-conditioned rollouts. Post-hoc replacement or injection preserves more nominal behavior and can appear favorable under simple ADE/FDE metrics, but it produces much weaker tail stress and attack strength. The full role-conditioned AWM therefore provides the strongest coherent attack while remaining realistic.

Table 13: Role-conditioning mechanism analysis. MEAN R and Tail Drop follow the definitions in Table 10. Role-conditioned joint rollout produces the strongest coherent attack, while post-hoc variants can look better under per-agent ADE/FDE/kinematic metrics because they preserve more nominal behavior. KIN. is the kinematic-violation rate.

Mode	Mean R	Worst R	Tail R	Drop ↑	Worst Drop ↑	Tail Drop ↑	AWM-sim ↓	ADE ↓	FDE ↓	Kin. ↓
Full role-conditioned	0.9778	0.9342	0.7764	0.0064	0.0500	0.2078	71.71	1.55	4.95	0.0108
Explicit replacement	0.9828	0.9761	0.8913	0.0014	0.0081	0.0929	75.50	1.42	4.80	0.0098
Inactive-revert intervention	0.9762	0.9560	0.8180	0.0080	0.0282	0.1662	73.50	1.12	3.60	0.0039
Safe-bg post-hoc injection	0.9821	0.9753	0.8863	0.0021	0.0089	0.0979	75.80	1.06	3.45	0.0037

E.5 Additional Planner Ablations

Table 14 evaluates the sensitivity of Stage B to the frozen opponent used inside the inference-time rollout. The matched self-play AWM gives the best combined results. Training with a nominal world model or a held-out external AWM remains functional, but the resulting planner is weaker. This indicates that the learned opponent distribution is most effective when the planner is optimized against the same calibrated AWM used to define the adversarial risk distribution.

Table 14: Inference-time opponent model sensitivity. Open-loop rewards are on the planner's rollout utility scale; closed-loop (CL) score and submetrics are ordinary nuPlan simulation. ADV. R is the adaptive-calibrated adversarial reward, TAIL is its Tail-CVaR, and Prog./Coll./TTC denote progress, collision, and time-to-collision submetrics.

Internal opponent host	Open-loop rollout			CL: Test14-hard NR			CL: Test14-random R		
	Nom. R ↑	Adv. R ↑	Tail ↑	Score ↑	Prog. ↑	TTC ↑	Score ↑	Coll. ↑	TTC ↑
Nominal WM	0.9815	–	–	79.41	88.83	**82.10**	90.36	97.75	94.60
Matched self-play AWM	0.9818	0.9756	0.8919	**80.05**	**90.16**	81.99	**90.71**	**98.08**	**95.40**
Held-out external AWM	0.9809	0.9738	0.8825	79.29	89.11	80.94	90.25	97.64	94.23

Table 15 and Figure 5 report hyperparameter sensitivity. The main trend is a trade-off between robustness and nominal performance: stronger tail-risk pressure or lower admission thresholds can improve adversarial tail metrics, but overly aggressive settings reduce closed-loop performance. Stronger retention preserves nominal behavior but can under-emphasize adversarial recovery.

Table 15: Hyperparameter sensitivity around the selected planner operating point. The base row uses the default Stage-B setting: CVaR tail mass τ=0.25, regret-CVaR weight 1.0, progress-retention weight 10.0, and training-time AWM threshold 0.45. HARD NR and RANDOM R are ordinary nuPlan closed-loop planner scores reported as percentages, not AWM sim-agent scores. SEL./SCENE and PAIR FRAC. describe the AWM exposure density used by the training rollout.

Variant	Open-loop rollout				Planner CL		AWM exposure	
	Nom. R ↑	Nom. prog. ↑	Adv. R ↑	Tail ↑	Hard NR ↑	Random R ↑	Sel./scene	Pair frac.
Base	0.9818	0.9545	0.9756	0.8919	80.05	90.71	1.34	0.34
CVaR tail τ=0.10	0.9808	0.9487	0.9758	0.8885	79.55	90.20	1.34	0.34
CVaR tail τ=0.50	0.9814	0.9538	0.9734	0.8790	78.60	90.45	1.34	0.34
Expected risk τ=1.00	0.9816	0.9552	0.9712	0.8650	77.85	90.55	1.34	0.34
CVaR weight 0.0	0.9820	0.9560	0.9690	0.8580	77.55	90.40	1.34	0.34
CVaR weight 0.5	0.9821	0.9550	0.9735	0.8835	79.20	90.65	1.34	0.34
CVaR weight 2.0	0.9806	0.9490	0.9760	0.8940	79.70	89.90	1.34	0.34
Progress weight 5.0	0.9807	0.9475	0.9760	0.8940	79.75	89.85	1.34	0.34
Progress weight 20.0	0.9819	0.9560	0.9720	0.8750	78.90	90.82	1.34	0.34
AWM threshold 0.10	0.9809	0.9502	0.9748	0.8868	79.20	90.15	1.52	0.52
AWM threshold 0.20	0.9812	0.9524	0.9752	0.8895	79.55	90.38	1.45	0.45

E.6 Adapted Prior Adversarial Generators

Table 16 gives the full adapted generator comparison summarized in Figure 2(b). The comparison separates adversarial strength from realism and search budget by adapting prior generators to the same autoregressive rollout interface. Single-candidate CAT* and STRIVE* produce relatively weak tail stress, while best-4 search strengthens all generators by selecting more damaging samples. The KING-style aggressive bound obtains the largest raw drop, but at a substantial cost in ADE/FDE and background-collision proxies.

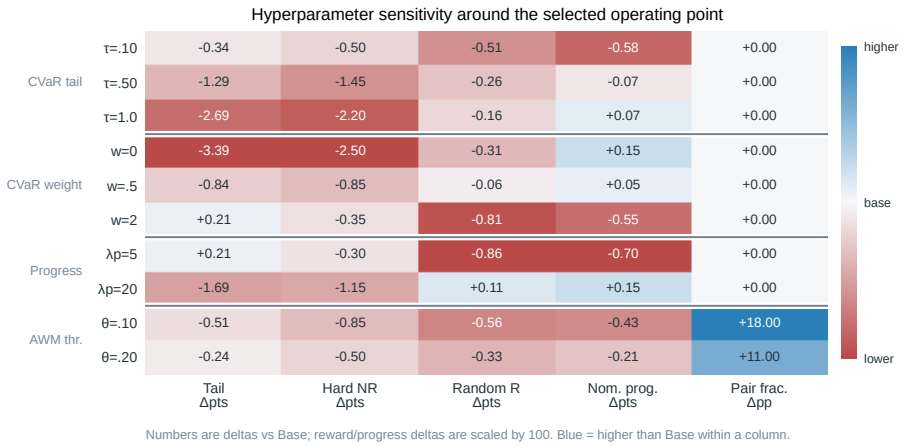


Figure 5: Sensitivity around the selected planner operating point. The default configuration balances adversarial tail improvement with ordinary closed-loop performance, while more aggressive risk or weaker retention settings move along the robustness and nominal performance trade-off.

Table 16: Adapted prior generator comparison. * denotes a prior baseline adapted to the autoregressive rollout interface, so all methods share the same scene state, tokenized transition, planner reward, and metrics. PROTOCOL separates search and constraints: CAND.=1 uses one generated candidate without reward-guided reranking; BEST-4 SEARCH generates four candidates and selects the lowest-reward rollout; STRICT CONSTRAINT strengthens behavioral and realism penalties; and KINEMATIC-SEARCH is an intentionally aggressive baseline used as a raw-attack upper bound. OFF./STAT. reports off-road and static-object collision violation; BG-COLL. reports the background agent collision.

Setting	Method	Protocol	Drop ↑	Worst ↑	Tail ↑	ADE ↓	FDE ↓	Kin. ↓	Off./Stat. ↓	BG-coll. ↓
Full AWM	**Full AWM**	adaptive coalition	0.0064	0.0500	0.2078	1.55	4.95	0.0108	0.280/0.009	0.081
Single candidate	CAT* [52]	cand.=1 sample	0.0049	0.0230	0.0917	2.46	7.71	0.0309	0.380/0.016	0.147
Single candidate	STRIVE* [35]	cand.=1 sample	0.0054	0.0245	0.1007	3.47	7.89	0.0181	0.381/0.016	0.148
Matched search	AWM	best-4 search	0.0242	0.0743	0.3853	1.47	4.73	0.0082	0.300/0.013	0.090
Matched search	CAT* [52]	best-4 search	0.0258	0.0774	0.2530	2.65	8.02	0.0140	0.424/0.020	0.165
Matched search	STRIVE* [35]	best-4 search	0.0204	0.0649	0.2473	3.88	8.17	0.0123	0.406/0.018	0.155
Aggressive bound	KING* [19]	kinematic-search (upp. bound)	0.1249	0.1542	0.3901	5.06	10.57	0.0211	0.579/0.046	0.258
Realism control	CAT* [52]	strict constraint	0.0141	0.0503	0.1174	1.45	4.79	0.0099	0.359/0.011	0.145
Realism control	STRIVE* [35]	strict constraint	0.0113	0.0426	0.1606	1.32	4.57	0.0077	0.372/0.013	0.144

E.7 Supplementary Visualizations

E.7.1 `InterPlan-LongTail` Examples

Figure 6 shows representative `InterPlan-LongTail` cases. These examples complement Table 6 by illustrating the dense interaction patterns introduced by the long-tail scenario modifications.

E.7.2 Time-Unrolled Robust-Recovery Cases

Figures 7 and 8 provide time-unrolled visualizations of the recovery cases in Figure 3. Each row compares Plan-R1 and AWM-Planner in the same adversarial scene. The temporal sequences show that AWM-Planner typically responds earlier, maintains a safer interaction envelope, and avoids the collision or TTC failure reached by the original planner.

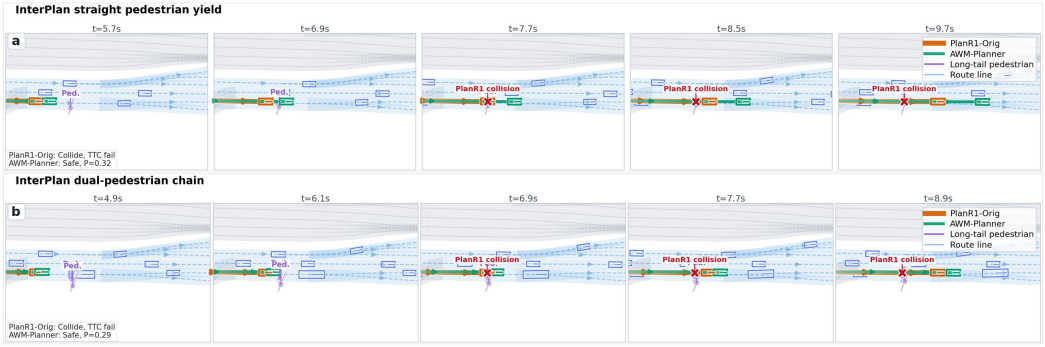


Figure 6: `InterPlan-LongTail` pedestrian-yield visualizations. (a) Straight pedestrian-yield case. (b) Dual-pedestrian chain case.

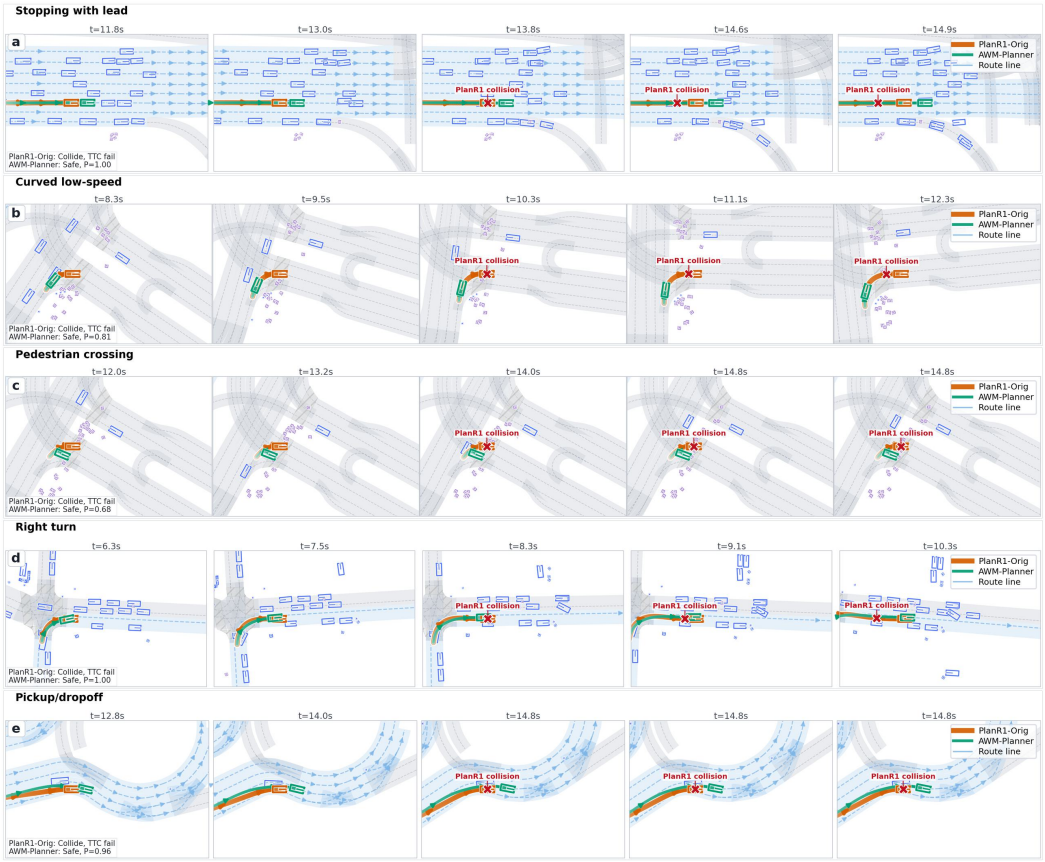


Figure 7: Time-unrolled robust-recovery visualizations on `NuPlan`, group I. (a) Stopping with lead. (b) Curved low-speed interaction. (c) Pedestrian crossing. (d) Right turn. (e) Pickup/dropoff interaction.

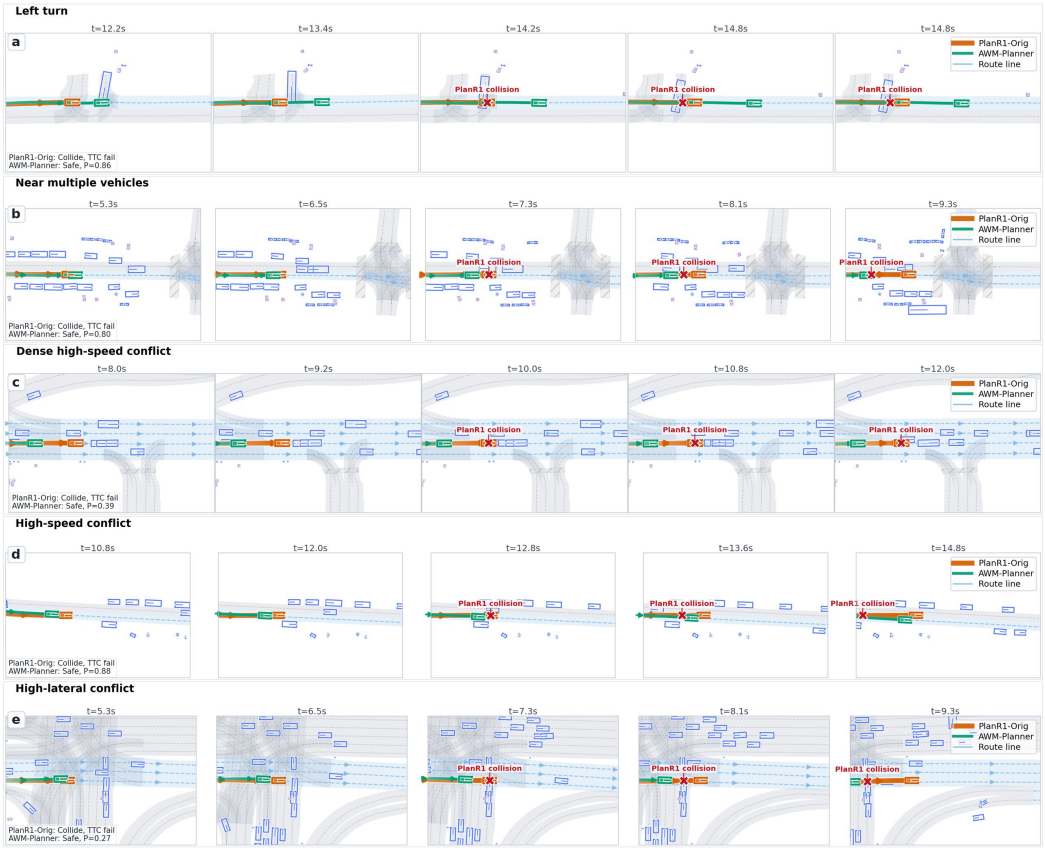


Figure 8: Time-unrolled robust-recovery visualizations on `NuPlan`, group II. (a) Left turn. (b) Dense traffic near multiple vehicles. (c) Dense high-speed conflict. (d) High-speed conflict. (e) High-lateral conflict.

E.7.3 K=2 Cooperative Attack Cases

Figure 9 provides additional $K=2$ cooperative attack visualizations. These cases complement the transfer evaluation in Table 2 by showing how two adversarial vehicles can jointly restrict the ego planner while preserving a coherent traffic scene.



Figure 9: K=2 cooperative attack visualizations. (a) Multi-lane following. (b) Low-speed blocking. (c) Right-turn merge. (d) Stopping with lead.

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 1 state the proposed AWM framework, its two-stage scope, and the claimed empirical gains. The supporting method, theoretical analysis, and experiments are provided in Sections 3–4 and Appendices C–D.

   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: Appendix C discusses the local and surrogate nature of the theoretical guarantees, approximation errors, Monte Carlo counterfactual variance, and trust-region assumptions. Section 4 and Appendix D further discuss attack–realism and robustness–nominal-performance trade-offs.

   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: Appendix C states the assumptions and notation used for the theoretical analysis and provides proofs for the lemma and propositions supporting the decoupled solver, credit assignment, CVaR weighting, nominal retention, and safety-dual surrogate.

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: The architecture, two-stage optimization procedure, training splits, hyperparameters, evaluation protocols, metric definitions, and benchmark construction are described in Section 3, Section 4, and Appendices D–D.3.

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. Appendices B and D describe the implementation, datasets, 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: Appendix D.1 specifies data splits, architecture settings, learning rates, epochs, optimizer-style rollout settings, reward weights, adversarial budgets, and Stage-A/Stage-B fine-tuning details. Appendix D.2 describes the evaluation protocols and metric conventions.

   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 experiments use fixed held-out benchmark scenario lists and paired scene-level comparisons where applicable, as described in Appendix D.2. The paper also reports seed-wise stability statistics in Table 3 and lower-confidence-bound diagnostics based on standard errors in Table 9.

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.1 reports the GPU type, number of GPUs, distributed training setup, and effective batch sizes. Table 3 additionally reports relative GPU-hours, plateau update counts, seed stability, and seed-wise closed-loop variability for the compared training paradigms.

   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 conducted on public autonomous-driving benchmarks and simulation-based stress tests, without collecting private data, involving human subjects, or deploying a real-world driving system. The adversarial component is constrained by feasibility and realism gates as described in Section 4 and Appendix D.

   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 Impact paragraph discusses benefits for robustness evaluation and training in simulation, risks of misuse and false confidence, and mitigation through feasibility and realism checks and separate reporting of attack strength and planner performance.

    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 current paper does not release high-risk data or a broadly deployable model, and the experiments are limited to benchmark simulation. The adversarial traffic model is evaluated under feasibility and realism constraints rather than released as an unrestricted deployment artifact.

    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: [Yes]

    Justification: Appendix D.1 credits the existing datasets, benchmarks, and baseline methods, and explicitly states the license and access constraints for `nuPlan`, `InterPlan`, and third-party assets. Raw restricted datasets are not redistributed and must be obtained from their original sources.

    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: This paper does not release new assets. Appendix D.3 describes the construction of `InterPlan-LongTail`; its scenario lists and modification configurations are not publicly released.

    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 experiments or human-subject studies; it uses existing autonomous-driving benchmarks and simulator-based evaluations.

    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 study participants or human-subject data collection, so IRB or equivalent review for human-subject experiments 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: The core method does not use LLMs as an important, original, or non-standard research component. Mentions of LLMs are used only as motivation for discrete autoregressive modeling, not as part of the proposed algorithm.

    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.