WORLD2POLICY: CONTINUAL POLICY IMPROVEMENT VIA SCENARIO CURRICULUM IN INTERACTIVE GAUSSIAN WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive Gaussian worlds enable photorealistic closed-loop simulation for autonomous driving. However, sustained policy improvement requires addressing two coupled challenges: how to resolve heterogeneous safety constraints during policy learning, and which scenarios should support continual policy update. A monolithic policy often entangles avoidance behaviors across dynamic actors, static obstacles, and road boundaries, while fixed scenario distributions yield diminishing returns as the policy improves. We present **World2Policy**, a co-evolutionary framework that unifies policy optimization with failure-conditioned scenario generation. In the *Policy Optimization Engine*, three constraint-specialized experts decouple safety modeling across foreground dynamics, static hazards, and drivable-area constraints through constrained reinforcement learning, while a privileged selector evaluates their complementary proposals to balance safety, progress, and comfort. In the *Environment Engine*, closed-loop rollouts expose residual failures, which guide the synthesis of physically plausible scenarios near the policy's capability boundary and update the training curriculum for policy optimization. Experiments on the closed-loop HUGSIM benchmark demonstrate that integrating WORLD2POLICY with UniAD, VAD, and LTF achieves HD-Score gains of +4.9, +6.1, and +6.9 points, respectively. Crucially, the co-evolutionary loop continually advances the policy's capability and sustains improvements across successive training rounds.

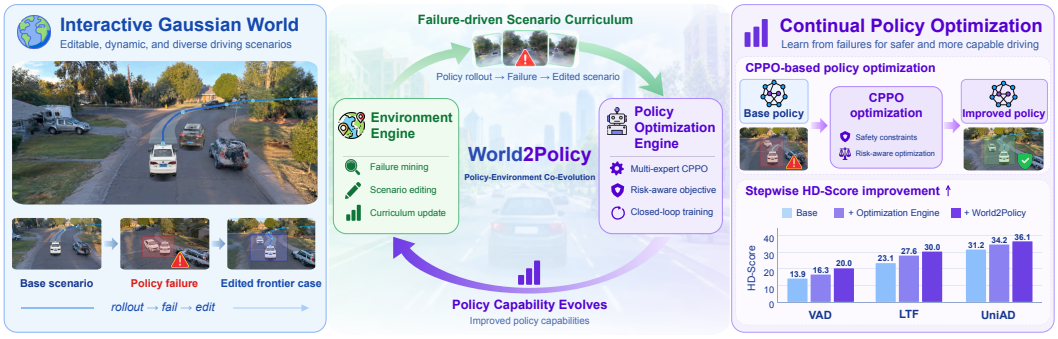

Figure 1: **Overview of World2Policy.** The *Policy Optimization Engine* improves the policy with constrained PPO under risk-aware objectives. The *Environment Engine* mines closed-loop failures and edits physically valid, solvable frontier scenarios that target the policy's weaknesses for curriculum updates. Their feedback loop co-evolves the policy and its training worlds for continual improvement.

1 INTRODUCTION

Interactive learning is crucial for autonomous driving, as a policy's decisions inherently govern its subsequent states and safety-critical risks. Once the policy deviates from demonstrated trajectories, static offline datasets fail to cover these newly induced states, causing minor errors to compound into catastrophic failures such as collisions, off-road excursions, or deadlocks. To bridge this gap,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

interactive Gaussian worlds (Zhou et al., 2025) provide a compelling solution by seamlessly coupling photorealistic reconstruction with closed-loop interaction and controllable scenario editing.

Unlocking the full potential of such worlds requires two tightly coupled components: a Policy Optimization Engine that learns from active scenarios, and an Environment Engine that dynamically adapts the scenario distribution as the policy evolves. Together, they form a co-evolving feedback loop: policy performance is continually refined through closed-loop interaction, while emergent failure modes conversely guide targeted updates to the training curriculum. Realizing this symbiotic loop, however, introduces non-trivial challenges for both engines.

On the policy side, driving involves heterogeneous hazards across dynamic actors, static obstacles, and road boundaries, each demanding distinct perceptual cues and evasion maneuvers. Jointly optimizing these constraints often induces objective conflicts, shifts risks, or yields overly conservative behavior, while a single policy lacks behavioral diversity. The *Policy Optimization Engine* must therefore cultivate complementary constraint-aware behaviors and arbitrate them into safe, effective decisions.

Conversely, on the environment side, training on static benchmarks inevitably yields diminishing returns. Yet, active scenario generation often faces a dilemma between sparse random exploration and physically invalid adversarial crashes. The *Environment Engine* resolves this by synthesizing physically plausible, resolvable scenarios anchored strictly at the policy's shifting capability boundary.

To address these coupled challenges, we present **World2Policy**, an interactive framework synergizing a multi-expert *Policy Engine* with a failure-conditioned *Environment Engine*. At each co-evolution iteration, three specialized experts are trained via closed-loop reward-cost learning, targeting foreground, background, and drivability constraints. Their complementary trajectory proposals are arbitrated by a privileged selector that balances safety, progress, and comfort. Closed-loop rollouts of the unified policy then expose residual failures, which explicitly guide the generation of physically valid, resolvable scenarios along the policy's capability boundary. Incorporating these curated new scenarios into the training pool closes the loop, driving sustained policy improvement.

Integrating WORLD2POLICY with existing state-of-the-art policies consistently improves performance on the challenging closed-loop HUGSIM benchmark: the Policy Optimization Engine raises overall HD-Score by 3.0, 2.4, and 4.5 points for UniAD (Hu et al., 2023b), VAD (Jiang et al., 2023), and LTF (Dauner et al., 2024), respectively, while the complete policy-environment loop yields total gains of +4.9, +6.1, and +6.9 over the corresponding base policies. The results demonstrate that constraint-specialized experts learn complementary risk-aware behaviors, and privileged arbitration combines them into safer, more effective decisions. Moreover, failure-conditioned scenario generation enables an adaptive training curriculum, raising the held-out HD-Score from 29.0 to 31.5.

In summary, our contributions are threefold:

- We propose World2Policy, a closed-loop co-evolutionary framework that unifies policy optimization with adaptive scenario synthesis in interactive Gaussian worlds for continual policy improvement.
- We design a multi-expert *Policy Optimization Engine* that decouples risk-aware behaviors for distinct driving constraints and arbitrates diverse trajectory proposals via privileged selection.
- We develop a failure-conditioned *Environment Engine* that synthesizes physically plausible scenarios anchored at the policy's capability boundary, enabling continual curriculum refinement.

2 RELATED WORK

Interactive World Models and Simulators for Autonomous Driving. Interactive driving environments have progressed from engine-based simulators such as CARLA and NAVSIM (Dosovitskiy et al., 2017; Dauner et al., 2024) to neural-rendered and Gaussian scene representations that reconstruct photorealistic, controllable worlds (Yang et al., 2023; Tonderski et al., 2024; Zhou et al., 2024; 2025). In parallel, generative world models predict action-conditioned future observations or 3D states for planning (Wang et al., 2024a; Gao et al., 2024a;b; Zheng et al., 2024a; Wang et al., 2026b). Despite improved realism and controllability, existing approaches primarily target reconstruction, prediction, or fixed-policy evaluation; WORLD2POLICY instead uses the interactive world to mine policy failures, generate targeted and physically valid scenarios, and adapt the training distribution, continually advancing the policy's capability and sustaining gains.

Closed-Loop Policy Optimization for Autonomous Driving. Closed-loop policy learning updates driving policies through environment rollouts, exposing compounding errors that open-loop imitation (Chitta et al., 2022; Jia et al., 2025) cannot observe. Recent reinforcement-learning, trajectory-optimization, and learned-world-model methods (Li et al., 2024; Zhang et al., 2025; Gao et al., 2025; Yan et al., 2026; Yang et al., 2026; He et al., 2026) improve driving through interaction, while constrained RL explicitly trades off task rewards and safety costs (Achiam et al., 2017; Ray et al., 2019). Yet most methods train on fixed scenario distributions and a single policy, leaving newly exposed failures underrepresented and risk transfer across heterogeneous constraints unresolved; WORLD2POLICY tackles this gap with constraint-specialized optimization and failure-driven curriculum updates. Additional related works are discussed in Appendix A.

3 PRELIMINARIES

Interactive Gaussian Worlds. We represent the driving environment as an interactive Gaussian world $\mathcal{W}$, composed of a static scene and a set of dynamic actors. Following HUGSIM (Zhou et al., 2025), the static scene is represented by 3D Gaussians that capture stationary elements such as roads, buildings, and vegetation. Each dynamic actor is represented at the object level, together with its appearance, pose, and motion state. Formally, an interactive Gaussian world at time t is written as:

$$\mathcal{W}_t = \left(\mathcal{G}^{\text{static}}, \mathcal{A}_t, T_t^{\text{ego}}\right), \qquad \mathcal{A}_t = \left\{\left(\overline{\mathcal{G}}^j, T_t^j, \xi_t^j\right)\right\}_{j=1}^{N} \tag{1}$$

where $\mathcal{G}^{\text{static}}$ denotes the static Gaussian scene, $\mathcal{A}_t$ denotes the set of dynamic actors, and T_t^{ego} is the ego-vehicle pose. For actor j, $\overline{\mathcal{G}}^j$ specifies its Gaussian appearance, while T_t^j and ξ_t^j denote its time-varying pose and motion state, such as velocity and yaw rate. The Gaussian scene and actor appearances remain fixed within an episode, whereas the ego and actor states evolve over time. This formulation allows the same world to support diverse ego rollouts and actor interactions, providing the basis for an interactive Gaussian environment.

Closed-Loop Policy–World Interaction. We formulate the interaction between the driving policy and the Gaussian world as a Markov Decision Process (MDP). At each time step t, the world state $\mathcal{W}_t$ is rendered into an egocentric observation o_t, from which the policy π_θ generates a local waypoint trajectory a_t. Executing a_t updates both the ego vehicle and surrounding actors, transitioning to the next world state $\mathcal{W}_{t+1}$ and yielding an immediate driving reward r_t:

$$\mathcal{W}_t \xrightarrow{\phi} o_t \xrightarrow{a_t \sim \pi_\theta(\cdot|o_t)} a_t \xrightarrow{\mathcal{P}(\cdot|\mathcal{W}_t, a_t)} \mathcal{W}_{t+1}, \quad r_t = R(\mathcal{W}_t, a_t) \tag{2}$$

where ϕ denotes the Gaussian rendering of ego observations, $\mathcal{P}$ is the transition dynamics updating the ego and actor states $(T_t^{\text{ego}}, T_t^j, \xi_t^j) \to (T_{t+1}^{\text{ego}}, T_{t+1}^j, \xi_{t+1}^j)$, and R evaluates driving performance for policy optimization. Crucially, since actions directly reshape subsequent environmental states and observations, this formulation establishes a genuine closed-loop setting for on-policy learning.

4 METHOD

4.1 WORLD2POLICY OVERVIEW

WORLD2POLICY establishes a closed-loop framework for continual driving policy improvement by coupling a ***Policy Optimization Engine*** with an ***Environment Engine*** (Fig. 2). The Policy Optimization Engine trains constraint-specialized experts with constrained PPO, while a privileged selector chooses a safe and progress-preserving trajectory from their complementary proposals. The Environment Engine evaluates the resulting policy in closed loop, mines failure cases, and synthesizes failure-conditioned scenarios targeting the exposed capability bottlenecks. By co-evolving the training scenarios with the policy capabilities, the framework continually improves the driving policy.

4.2 POLICY OPTIMIZATION ENGINE: MULTI-EXPERT LEARNING

Driving policies face heterogeneous constraints from dynamic actors, static obstacles, and drivable regions. Jointly optimizing these constraints can induce risk transfer and limit action diversity.

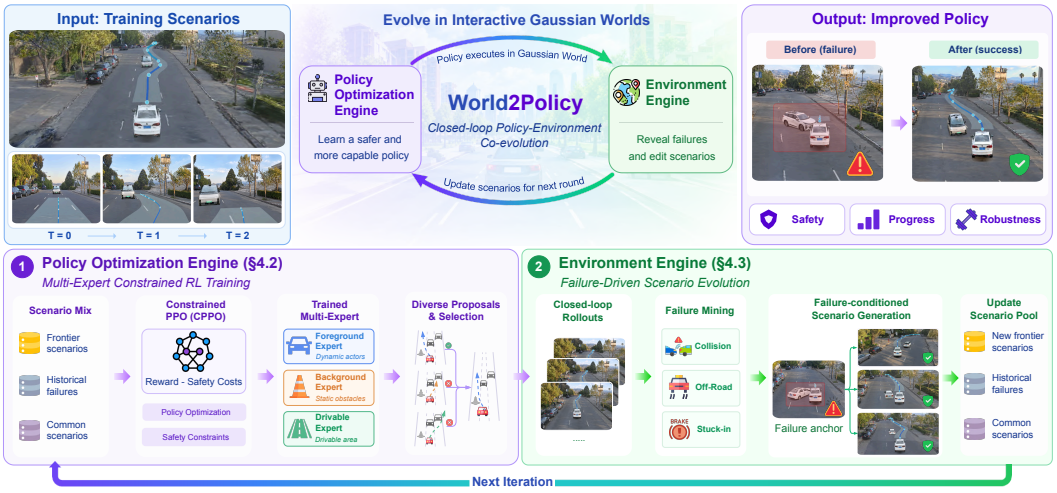

Figure 2: **Pipeline of World2Policy.** The *Policy Optimization Engine* trains constraint-specialized experts with constrained PPO and selects trajectories from their complementary proposals. The *Environment Engine* mines closed-loop failures and generates targeted, failure-conditioned scenarios. Together, the two engines update training distribution in closed-loop for continual policy improvement.

The Policy Optimization Engine learns constraint-specialized experts to generate complementary proposals and uses privileged trajectory selection to choose safe and effective actions.

Constraint-Specialized Experts. We define three constraint-specialized experts $\{\pi_\theta^e\}_{e\in\mathcal{E}}$, indexed by $\mathcal{E} = \{\text{fg}, \text{bg}, \text{drv}\}$. The foreground, background, and drivable experts emphasize dynamic-actor collision, static-obstacle collision, and drivable-area violation risks, respectively. Let C^{fg}, C^{bg}, and C^{drv} denote the corresponding component costs. For expert e, we define the joint safety cost as:

$$C^e(\mathcal{W}_t, a_t) = w^e_{\text{fg}} C^{\text{fg}}(\mathcal{W}_t, a_t) + w^e_{\text{bg}} C^{\text{bg}}(\mathcal{W}_t, a_t) + w^e_{\text{drv}} C^{\text{drv}}(\mathcal{W}_t, a_t), \tag{3}$$

where w^e_{fg}, w^e_{bg}, and w^e_{drv} are expert-specific weights that assign a higher weight to each expert's designated risk. Thus, all experts consider the same three safety risks and share the same driving objective, while their different weightings induce complementary behaviors.

Multi-Expert Closed-Loop Training. Each expert π_θ^e is optimized via on-policy, closed-loop interaction within the Gaussian world. All experts share a driving reward $r_t = R(\mathcal{W}_t, a_t)$ that encourages route progress, task completion, and motion comfort, and use the joint safety cost C^e defined in Eq. 3 to impose their safety constraints. Formally, each expert is trained to maximize the shared driving reward subject to its expert-specific safety budget:

$$\begin{gathered} \max_\theta \; J_R(\pi_\theta^e) \quad \text{s.t.} \quad J_{C^e}(\pi_\theta^e) \le d^e, \qquad e \in \mathcal{E} \\ J_R(\pi_\theta^e) = \mathbb{E}_{\tau\sim\pi_\theta^e}\left[\textstyle\sum_{t=0}^{T-1} \gamma^t R(\mathcal{W}_t, a_t)\right], \quad J_{C^e}(\pi_\theta^e) = \mathbb{E}_{\tau\sim\pi_\theta^e}\left[\textstyle\sum_{t=0}^{T-1} \gamma^t C^e(\mathcal{W}_t, a_t)\right] \end{gathered} \tag{4}$$

Here, τ denotes a closed-loop rollout, γ is the discount factor, $J_R(\pi_\theta^e)$ and $J_{C^e}(\pi_\theta^e)$ are the expected discounted reward and expert-specific safety cost, respectively, and d^e is the safety budget. We optimize Eq. 4 with constrained proximal policy optimization (CPPO) (Ray et al., 2019) using the Lagrangian:

$$\min_{\lambda^e\ge 0} \max_\theta \; \mathcal{L}^e(\theta, \lambda^e) = J_R(\pi_\theta^e) - \lambda^e \left(J_{C^e}(\pi_\theta^e) - d^e\right) \tag{5}$$

In practice, policy parameters θ are updated via a clipped PPO objective penalized by cost advantages, while the Lagrange multiplier λ^e is dynamically adjusted in response to safety-budget violations.

Multi-Expert Proposals and Privileged Selection. At each decision step, a base policy generates a nominal trajectory $\bar{\tau}_t$, which each constraint-specialized expert π_θ^e refines by sampling a discrete driving mode m_t^e (e.g., keep, slow, stop, or lateral avoidance) alongside residual parameters δ_t^e:

$$(m_t^e, \delta_t^e) \sim \pi_\theta^e(\cdot \mid o_t), \quad \tau_t^e = \bar{\tau}_t + \Delta\tau(m_t^e, \delta_t^e) \tag{6}$$

where δ_t^e denotes the mode-conditioned longitudinal and lateral residuals, and $\Delta\tau$ maps the mode and residuals into a trajectory displacement. This design preserves the base policy's driving capability while leveraging constraint-specific refinements to expand the support of feasible trajectories.

The proposals from all experts span distinct driving modes, forming a diverse candidate set of physically valid trajectories. Then, a privileged selector evaluates each candidate using a learned dynamic-actor risk model and geometric safety checks for static obstacles and drivable-area violations, choosing a safe trajectory with favorable progress and comfort. Together, the specialized experts and the privileged selector cooperate to produce the optimal trajectory for execution.

4.3 Environment Engine: Failure-Adaptive Scenario Curriculum

Building on the optimized expert policies, the Environment Engine closes the loop by evaluating policy rollouts, mining failure cases, and generating failure-conditioned scenarios. This failure-adaptive curriculum aligns the scenario distribution with the policy's capability boundary.

Closed-Loop Failure Mining. We evaluate the current policy through closed-loop rollouts in the Gaussian world and record the state transitions. Failure mining first detects critical events, such as collisions, drivable-area violations, and stagnation. For each failure, it extracts the preceding interaction window as an anchor to identify relevant interacting actors and context. Cases without a verified recovery are discarded, yielding solvable failure records for targeted scenario generation.

Failure-Conditioned Scenario Generation. For each retained failure, we synthesize targeted scenarios by perturbing the critical actor's initial pose T_0^j and motion parameters ξ^j (e.g., velocity, heading, or trigger timing), while keeping the reconstructed background $\mathcal{G}^{\text{static}}$ fixed. To target the policy's capability boundary, we use the cross-entropy method (CEM) to search for challenging yet physically valid parameter configurations. The resulting scenarios cover diverse interaction events, such as cut-ins, crossings, and sudden braking, that expose the policy's unresolved failure modes.

4.4 Continual Policy Improvement via Engine Collaboration

The two engines collaborate through an iterative scenario curriculum. The newly generated failure-conditioned scenarios are combined with the existing training set for further multi-expert training by the Policy Optimization Engine. Closed-loop evaluation of the updated policy then reveals remaining failure modes, guiding the Environment Engine to generate the next round of targeted scenarios. Through this closed-loop feedback, World2Policy continually aligns the training distribution with the policy's capability boundary, enabling sustained policy improvement.

5 Experiments

5.1 Experimental Setup

Implementation Details. We use pretrained UniAD (Hu et al., 2023b), VAD (Jiang et al., 2023), and LTF (Dauner et al., 2024) as base policies, and construct 300 HUGSIM training scenarios separate from the benchmark evaluation set. The Policy Optimization Engine freezes each base policy and trains three residual experts for 24 CPPO updates. Each update collects 1,024 closed-loop interaction steps, followed by optimization with Adam using a learning rate of 1×10^{-4} and a minibatch size of 256. During evaluation, the experts sample 16 proposals each, yielding 48 candidates for privileged selection. The Environment Engine mines failures on held-out validation scenarios and uses CEM to search the poses and motion parameters of causal actors. Accepted scenarios are incorporated into the next training curriculum. Appendix B provides additional implementation details of both engines.

Benchmarks and Metrics. We evaluate on HUGSIM (Zhou et al., 2025), a photorealistic closed-loop driving benchmark containing 436 scenarios reconstructed from KITTI-360 (Liao et al., 2022), nuScenes (Caesar et al., 2020), Waymo (Sun et al., 2020), and PandaSet (Xiao et al., 2021). The scenarios span four difficulty levels: Easy, Medium, Hard, and Extreme. We report all official metrics: No Collision (NC), Drivable Area Compliance (DAC), Time to Collision (TTC), Comfort (COM), Route Completion (RC), and HUGSIM Driving Score (HD-Score), as detailed in Appendix C.1.

Table 1: **Closed-loop results on the HUGSIM benchmark.** We integrate WORLD2POLICY with UniAD, VAD, and LTF backbones to evaluate the Policy Optimization and Environment Engines.

Method	NC ↑	DAC ↑	TTC ↑	COM ↑	RC ↑	HD-Score ↑				
						Easy	Medium	Hard	Extreme	Overall
Reference Methods										
DrivoR (Kirby et al., 2026)	52.2	95.6	46.2	93.9	47.2	78.0	29.1	20.0	14.1	32.5
Gigapixel-DrivoR (Rowe et al., 2026)	-	-	-	-	50.1	67.4	51.9	19.1	21.6	38.5
WA-JEPA (Wang et al., 2026c)	68.6	96.4	61.2	66.2	56.9	79.8	55.6	30.6	13.6	44.6
DriveZero-Scale (He et al., 2026)	-	-	-	-	54.5	81.9	60.5	24.7	27.6	46.6
World2Policy Integration										
UniAD baseline (Hu et al., 2023b)	65.6	93.2	51.6	**66.3**	43.8	64.0	37.2	21.0	6.3	31.2
UniAD + Policy Optimization Engine	72.0	93.8	58.0	62.5	46.5	65.5 (+1.5)	42.5 (+5.3)	23.5 (+2.5)	7.5 (+1.2)	34.2 (+3.0)
UniAD + **World2Policy (Ours)**	**72.8**	**93.9**	**61.3**	61.0	**48.0**	**67.0** (+3.0)	**45.5** (+8.3)	**25.0** (+4.0)	**8.2** (+1.9)	**36.1** (+4.9)
VAD baseline (Jiang et al., 2023)	41.2	90.3	28.0	**95.3**	30.1	42.0	8.5	7.7	6.3	13.9
VAD + Policy Optimization Engine	53.8	90.2	38.8	90.5	32.0	46.1 (+4.1)	10.9 (+2.4)	9.5 (+1.8)	8.1 (+1.8)	16.3 (+2.4)
VAD + **World2Policy (Ours)**	**64.6**	**90.3**	**49.8**	92.1	**34.0**	**51.2** (+9.2)	**15.5** (+7.0)	**12.0** (+4.3)	**10.6** (+4.3)	**20.0** (+6.1)
LTF baseline (Dauner et al., 2024)	44.3	92.8	37.5	**94.8**	38.0	66.1	15.5	12.0	11.7	23.1
LTF + Policy Optimization Engine	51.5	94.1	43.8	92.2	44.2	66.3 (+0.2)	26.7 (+11.2)	14.1 (+2.1)	**12.1** (+0.4)	27.6 (+4.5)
LTF + **World2Policy (Ours)**	**55.4**	**94.9**	**47.4**	89.4	**47.6**	**66.4** (+0.3)	**32.5** (+17.0)	**15.8** (+3.8)	11.6 (-0.1)	**30.0** (+6.9)

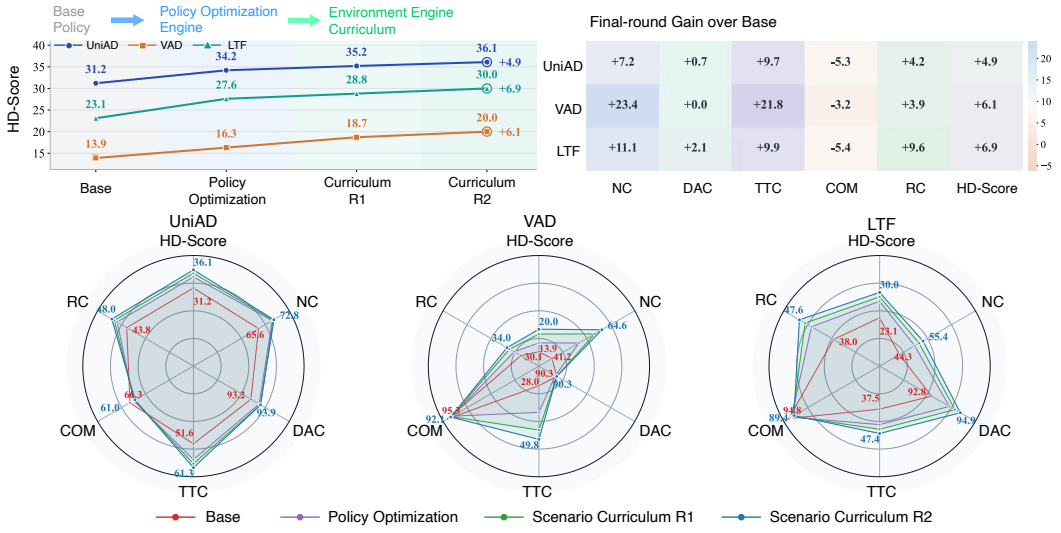

Figure 3: **Continual policy improvement across WORLD2POLICY curriculum.** The upper panels show HD-Score progression and final metric gains over each base policy; the lower panels compare the base policy, Policy Optimization Engine, and successive Environment Engine curriculum rounds.

5.2 MAIN RESULTS

Closed-Loop Policy Improvement on HUGSIM. Table 1 reports closed-loop results on HUGSIM, including reference methods and WORLD2POLICY integration with UniAD, VAD, and LTF. For each backbone, we compare the base policy, the Policy Optimization Engine, and the full framework with both the Policy Optimization and Environment Engines.

The Policy Optimization Engine improves the overall HD-Score by +3.0, +2.4, and +4.5 points on UniAD, VAD, and LTF, respectively. Adding the Environment Engine yields further gains, resulting in total improvements of +4.9, +6.1, and +6.9 points over the corresponding base policies. These consistent gains demonstrate the complementary benefits of policy optimization and failure-conditioned scenario curriculum across diverse policy architectures.

Continual Improvement across Curriculum Rounds. Figure 3 tracks HD-Score from each base policy through Policy Optimization Engine training and two successive Environment Engine curriculum rounds. The curriculum is generated on the validation split, and performance is measured on a held-out HUGSIM test split. For instance, LTF gains +4.5, +1.2, and +1.2 HD-Score points after policy optimization, Curriculum R1, and Curriculum R2, respectively. UniAD and VAD exhibit the same stagewise trend, indicating that failure-conditioned curriculum learning complements policy

Table 2: **Ablation of CPPO training and expert composition.** Joint Safety combines all three safety costs, while Expert-Specific Weights emphasize each expert's designated risk. The two blocks compare PPO vs. CPPO training and single-expert vs. multi-expert designs under a fixed selector.

Training Variant	# Experts	Training Objective			Safety Diagnostics			HUGSIM Metrics					
		Driving Reward	Joint Safety	Expert-Specific Weights	FG Coll. (%) ↓	BG Coll. (%) ↓	DRV Viol. (%) ↓	NC↑	DAC↑	TTC↑	COM↑	RC↑	HD-Score↑
Base Policy	-	-	-	-	40.63	32.81	32.81	47.4	89.3	42.5	**93.3**	40.3	24.5
CPPO Training													
Single-Expert PPO	1	✓	-	-	37.50	10.42	17.71	53.5	93.5	45.7	83.9	43.7	27.0 (+2.5)
Single-Expert CPPO	1	✓	✓	-	39.58	10.42	16.67	55.2	93.6	48.7	85.7	42.1	26.5 (+2.0)
Multi-Expert CPPO	3	✓	✓	-	35.42	8.33	18.13	54.9	94.8	48.2	86.0	41.3	25.7 (+1.2)
Expert Composition													
Foreground Expert	1	✓	✓	✓	36.98	7.29	21.35	55.3	94.5	48.1	85.0	40.7	24.5 (+0.0)
Background Expert	1	✓	✓	✓	40.63	6.77	18.23	54.5	94.1	47.5	84.4	41.9	25.6 (+1.1)
Drivable Expert	1	✓	✓	✓	39.06	8.33	17.19	**59.6**	94.3	**52.7**	85.0	43.3	28.1 (+3.6)
Multi-Expert CPPO (Ours)	3	✓	✓	✓	**33.85**	**3.65**	**16.67**	57.9	**95.0**	51.3	84.8	**44.7**	**29.3** (+4.8)

Table 3: **Ablation of privileged trajectory selection.** We compare random and privileged selection using 16 candidates from single-expert policy and 48 candidates from three-expert policies, selecting one trajectory at each step and evaluating safety, progress, comfort, and closed-loop performance.

Selection Variant	Candidates → Selected	Selection Signal			Safety Diagnostics			HUGSIM Metrics					
		Safety	Progress	Comfort	FG Coll. (%) ↓	BG Coll. (%) ↓	DRV Viol. (%) ↓	NC↑	DAC↑	TTC↑	COM↑	RC↑	HD-Score↑
Base Trajectory	1 → 1	-	-	-	40.63	32.81	32.81	47.4	89.3	42.5	**93.3**	40.3	24.5
Random Selection													
Joint-Safety Random	16 → 1	-	-	-	45.31	26.56	23.44	59.2	90.6	51.5	76.8	31.1	18.4 (-6.1)
Single-Expert Random (Avg.)	16 → 1	-	-	-	43.23	31.77	24.48	56.9	90.9	48.9	79.3	30.5	17.8 (-6.7)
Multi-Expert Random	48 → 1	-	-	-	43.75	35.94	28.13	52.2	90.5	44.6	77.1	29.1	16.1 (-8.4)
Privileged Selection													
Safety-Only Selector	48 → 1	✓	-	-	34.38	0.00	**15.63**	**71.8**	94.5	**64.7**	32.6	6.4	3.4 (-21.1)
Safety + Progress Selector	48 → 1	✓	✓	-	54.69	0.00	18.75	50.2	93.8	41.2	73.8	**48.6**	27.0 (+2.5)
Full Privileged Selector (Ours)	48 → 1	✓	✓	✓	**33.85**	3.65	16.67	57.9	**95.0**	51.3	84.8	44.7	**29.3** (+4.8)

optimization and sustains improvement across rounds. The radar charts show that Curriculum R2 consistently improves NC, TTC, RC, and HD-Score over the base policies while trading off COM, revealing a shift toward safer and more effective driving with a moderate comfort cost.

5.3 Policy Optimization Engine: Ablations and Analysis

We ablate the Policy Optimization Engine's multi-expert CPPO training and privileged selector, using LTF as the base policy and evaluating all variants on a 64-scenario subset of the HUGSIM test set.

Multi-Expert CPPO Training. Table 2 ablates CPPO training and expert composition. In the CPPO training block, reward-only PPO raises the HD-Score from 24.5 to 27.0, showing the benefit of direct reward optimization, but leaves foreground and background collision rates at 37.50% and 10.42%. Adding joint safety constraints improves NC, DAC, and TTC, confirming the value of constraint-aware optimization. Multi-expert CPPO further lowers foreground and background collision rates to 35.42% and 8.33%, showing that complementary experts broaden risk-aware behavior. Expert-specific weighting in the full World2Policy reduces these rates to 33.85% and 3.65% and yields the highest HD-Score (29.3), demonstrating the value of differentiated risk emphasis.

In the expert composition block, each expert has the lowest failure rate for its designated risk among individual experts (36.98% foreground, 6.77% background, and 17.19% drivable), but leaves other risks less controlled. Combining all three experts lowers these rates to 33.85%, 3.65%, and 16.67%, respectively, and achieves the best overall HD-Score, demonstrating their complementary strengths.

Privileged Trajectory Selection. Table 3 isolates the effect of trajectory selection by keeping the trained experts fixed and varying only the selection rule. Random selection fails to effectively exploit the candidate pool: Multi-Expert Random achieves an HD-Score of just 16.1, compared with 24.5 for the base trajectory. Safety-Only selection eliminates collisions and reduces drivable-area violations to 15.6%, but lowers route completion to 6.4, suggesting that prioritizing safety alone shifts failures toward stuck-in behaviors. Incorporating progress improves route completion to 48.6, albeit at the cost of more foreground collisions. Our full privileged trajectory selector achieves the highest HD-Score of 29.3, with foreground, background, and drivable-area violation rates of 33.9%, 3.7%, and 16.7%, respectively. These results highlight the importance of balancing safety, progress, and comfort to translate candidate diversity into improved driving performance.

Table 4: **Ablation of failure-conditioned scenario generation.** Valid Scenarios and Challenge Rate denote the valid-candidate and recoverable-failure proportions, respectively. Validation safety metrics are computed on the validation scenarios, while driving performance is measured on the test set.

Scenario Generation	Validation Set					Test Set (Held Out)					
	Scenario Generation		Policy Evaluation			Driving Performance					
	Valid Scenarios (%) ↑	Challenge Rate (%) ↑	FG Coll. (%) ↓	BG Coll. (%) ↓	DRV Viol. (%) ↓	NC↑	DAC↑	TTC↑	COM↑	RC↑	HD-Score↑
Baseline (No Failure-Aware Updates)	-	-	34.92	3.87	17.71	57.9	95.0	51.3	84.8	44.7	29.3
Random Scenario Generation	87.8	6.5	34.70	4.15	17.46	57.5	94.8	51.0	84.6	44.5	29.0 (-0.3)
Failure-Conditioned, No CEM	94.0	30.3	32.63	3.32	16.16	58.1	95.1	51.9	85.0	45.4	30.3 (+1.0)
Failure-Conditioned + CEM (Ours)	**97.5**	**54.4**	**31.05**	**2.81**	**15.24**	**58.8**	**95.3**	**52.7**	**85.2**	**46.2**	**31.5** (+2.2)

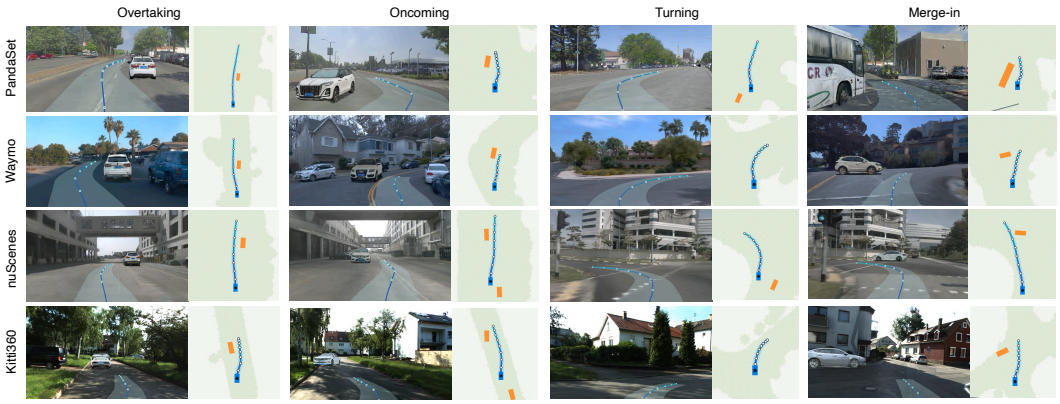

Figure 4: **Qualitative results of the optimized policy.** Rendered and BEV views illustrate smooth, drivable responses to overtaking, oncoming, turning, and merging interactions across PandaSet, Waymo, nuScenes, and KITTI-360 datasets, demonstrating the effectiveness of WORLD2POLICY.

5.4 ENVIRONMENT ENGINE: ABLATIONS AND ANALYSIS

We ablate the Environment Engine's scenario generation and dual engine collaboration, using LTF as the base policy and evaluating all variants on 64 scenarios from the HUGSIM test set.

Failure-Conditioned Scenario Generation. Table 4 evaluates the scenario generation mechanisms of the Environment Engine. Random generation achieves 87.8% validity but only a 6.5% challenge rate, indicating that valid perturbations are rarely targeted to the policy's weaknesses. Failure conditioning without CEM raises the challenge rate to 30.3% and the test HD-Score to 30.3, while the full process reaches a 97.5% validity, 54.4% challenge rate, and a 31.5 HD-Score. These results show that failure conditioning supplies policy-specific failure signals and that CEM concentrates them near policy's capability boundaries, producing more effective training scenarios than random exploration.

Policy and Environment Engine Collaboration. Table 5 presents a 2×2 ablation that combines single- vs. multi-expert CPPO, and fixed vs. failure-conditioned scenarios. With fixed scenarios, multi-expert CPPO raises the HD-Score from 26.5 to 29.3 (+2.8), showing that the Policy Optimization Engine improves the policy by supplying complementary risk-aware behaviors. With single-expert CPPO, replacing fixed scenarios with failure-conditioned scenarios raises the score from 26.5 to 28.0 (+1.5), demonstrating that the Environment Engine provides targeted, policy-specific training signals. Combining both components achieves an HD-Score of 31.5 (+5.0 over the single-expert fixed baseline), exceeding either component alone and confirming that adaptive scenario curricula amplify the benefits of complementary policy optimization.

5.5 QUALITATIVE ANALYSIS

Optimized Policy Behaviors. Figure 4 shows that the optimized policy handles diverse interactions with smooth, drivable trajectories: it overtakes blocking vehicles, yields to oncoming traffic, follows road geometry through turns, and merges around crossing traffic. These examples show that constraint-specialized experts and privileged selection translate complementary risk-aware proposals into

Table 5: **Ablation of Policy–Environment Engine collaboration.** The 2×2 design combines single- and multi-expert policy optimization with fixed and failure-conditioned scenario curricula.

Variant	Policy Engine		Environment Engine		HUGSIM Metrics					
	Single-Expert CPPO	Multi-Expert CPPO	Fixed Scenarios	Failure-Conditioned Scenarios	NC↑	DAC↑	TTC↑	COM↑	RC↑	HD-Score↑
Single-Expert + Fixed	✓	-	✓	-	55.2	93.6	48.7	85.7	42.1	26.5
Single-Expert + Failure-Conditioned	✓	-	-	✓	56.5	94.2	49.8	85.5	43.5	28.0 (+1.5)
Multi-Expert + Fixed	-	✓	✓	-	57.9	95.0	51.3	84.8	44.7	29.3 (+2.8)
Multi-Expert + Failure-Conditioned (Ours)	-	✓	-	✓	**58.8**	**95.3**	**52.7**	**85.2**	**46.2**	**31.5** (+5.0)



Figure 5: **Scenario curriculum across difficulty levels.** Examples from clear-road scenes to increasingly challenging scenarios, including lane blocking, oncoming vehicles, and head-on vehicles.

executable behavior while preserving route progress under heterogeneous hazards. Together, these components enable the policy to maintain a safety-progress trade-off across diverse scenes.

Adaptive Scenario Curriculum. Figure 5 illustrates how the Environment Engine varies the initial positions and motions of causal actors to produce a graded range of challenges, from clear roads to lane blocking, cut-in, oncoming, and collision-course interactions. By conditioning the search on observed failures and exploring physically valid, recoverable configurations near the current policy's capability boundary, it generates targeted training scenarios rather than arbitrary perturbations. The resulting curriculum supplies progressively more informative signals, exposes newly unresolved failure modes, and supports continual policy improvement across successive rounds.

6 Conclusion

We presented World2Policy, a co-evolutionary framework for continual closed-loop driving policy improvement in interactive Gaussian worlds. Its Policy Optimization Engine learns complementary risk-aware behaviors with three constraint-specialized experts and privileged trajectory selection, while its Environment Engine mines recoverable failures to generate physically plausible scenarios near the policy's evolving capability boundary. Across UniAD, VAD, and LTF, policy optimization alone improves overall HD-Score by 3.0, 2.4, and 4.5 points, while the complete policy-environment loop delivers total gains of 4.9, 6.1, and 6.9 points, respectively. Ablations further show that failure-conditioned generation raises held-out HD-Score from 29.0 to 31.5, confirming that targeted recoverable failures provide more effective training signals. Together, these results show that coupling complementary policy behaviors with an adaptive, failure-driven curriculum can continually advance the capability boundary and sustain safer, more effective driving across successive training rounds.

AI USE STATEMENT

We used generative AI tools solely for language polishing to improve grammar, clarity, and readability. They were not used to develop the research ideas or methods, conduct experiments, or analyze or interpret results. The authors reviewed all AI-assisted edits to ensure that the scientific meaning was preserved and take full responsibility for the final manuscript.

ETHICS STATEMENT

This work is conducted entirely in simulation using existing datasets and pretrained models. It does not involve human subjects, real-world deployment, or new data collection, and we do not identify additional ethical concerns.

REPRODUCIBILITY STATEMENT

We provide implementation and experimental details in Secs. 4–5, with additional training, algorithmic, and evaluation details in Apps. B and C. Our experiments are based on publicly available datasets and open-source baselines, including HUGSIM, UniAD, VAD, and LTF. We will release the code and framework for both the Policy Optimization Engine and the Environment Engine, together with configuration and evaluation instructions.

REFERENCES

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

Mohammad Al-Sharman, Rowan Dempster, Mohamed A Daoud, Mahmoud Nasr, Derek Rayside, and William Melek. Self-learned autonomous driving at unsignalized intersections: A hierarchical reinforced learning approach for feasible decision-making. *IEEE Transactions on Intelligent Transportation Systems*, 24(11): 12345–12356, 2023.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11618–11628. IEEE, 2020.

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.

Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo De Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. In *International Conference on Learning Representations*, volume 2025, pp. 85508–85527, 2025.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

Daphne Cornelisse, Aarav Pandya, Kevin Joseph, Joseph Suárez, and Eugene Vinitsky. Building reliable sim driving agents by scaling self-play. *arXiv preprint arXiv:2502.14706*, 2025.

Chenxu Dang, Haiyan Liu, Jason Bao, Pei An, Xinyue Tang, An Pan, Jie Ma, Bingchuan Sun, and Yan Wang. Sparseworld: A flexible, adaptive, and efficient 4d occupancy world model powered by sparse and dynamic queries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3497–3505, 2026.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Xin Fei, Wenzhao Zheng, Yueqi Duan, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Jiwen Lu. Driv3r: Learning dense 4d reconstruction for autonomous driving. *arXiv preprint arXiv:2412.06777*, 2024.

Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, Xiangyu Li, Wenyu Liu, Qian Zhang, et al. Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. *Advances in Neural Information Processing Systems*, 38:32551–32576, 2025.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *International Conference on Learning Representations*, volume 2024, pp. 22841–22860, 2024a.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2024b.

Hao He, Chengcheng Hu, Zirun Su, Heng Zhang, Haisong Liu, Jinke Li, Haochen Tian, Zhenwei Shen, Hongyang Li, Zhichao Li, et al. Drivezero: End-to-end driving beyond human demonstrations. *arXiv preprint arXiv:2609.06055*, 2026.

Georg Hess, Carl Lindström, Maryam Fatemi, Christoffer Petersson, and Lennart Svensson. Splatad: Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11982–11992. IEEE, 2025.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023a.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023b.

Bernhard Jaeger, Daniel Dauner, Jens Beißwenger, Simon Gerstenecker, Kashyap Chitta, and Andreas Geiger. Carl: Learning scalable planning policies with simple rewards. *arXiv preprint arXiv:2504.17838*, 2025.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2025, pp. 67227–67243, 2025.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8306–8316. IEEE, 2023.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, et al. Driving on registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32058–32069, 2026.

Lingdong Kong, Yu Yang, Jianbiao Mei, Youquan Liu, Ao Liang, Dekai Zhu, Dongyue Lu, Wei Yin, Xiaotao Hu, Mingkai Jia, et al. 3d and 4d world modeling: A survey. *arXiv preprint arXiv:2509.07996*, 2025.

Lingdong Kong, Ao Liang, Tianyi Yan, Hongsi Liu, Wesley Yang, Ziqi Huang, Xian Sun, Wei Yin, Jialong Zuo, Yixuan Hu, et al. Is your driving world model an all-around player? *arXiv preprint arXiv:2605.10858*, 2026.

Hongchen Li, Tianyu Li, Jiazhi Yang, Mingyang Shang, Gaoqiang Wu, Caojun Wang, Haochen Tian, Zengrong Lin, Zhihui Hao, XianPeng Lang, et al. Plannerrft: Reinforcing diffusion planners through closed-loop and sample-efficient fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24929–24938, 2026a.

Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in carla-v2). In *European conference on computer vision*, pp. 142–158. Springer, 2024.

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

Tianyu Li, Li Chen, Caojun Wang, Haochen Liu, Kashyap Chitta, Zhenjie Yang, Yuhang Lu, Naisheng Ye, Yihang Qiu, Yufei Wang, et al. World engine: Towards the era of post-training for autonomous driving. *arXiv preprint arXiv:2606.19836*, 2026b.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 7890–7911, 2026c.

Alan Liang, Youquan Liu, Yu Yang, Dongyue Lu, Linfeng Li, Lingdong Kong, Huaici Zhao, and Wei Tsang Ooi. Lidarcrafter: Dynamic 4d world modeling from lidar sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18406–18414, 2026a.

Ao Liang, Lingdong Kong, Tianyi Yan, Hongsi Liu, Yu Yang, Ziqi Huang, Wei Yin, Jialong Zuo, Yixuan Hu, Dekai Zhu, et al. Worldlens: Full-spectrum evaluations of driving world models in real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36385–36399, 2026b.

Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022.

Jiangfan Liu, Yongkang Guo, Fangzhi Zhong, Tianyuan Zhang, Zonglei Jing, Siyuan Liang, Jiakai Wang, Mingchuan Zhang, Aishan Liu, and Xianglong Liu. Adversarial generation and collaborative evolution of safety-critical scenarios for autonomous vehicles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 38926–38934, 2026a.

Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, Long Chen, et al. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving. *arXiv preprint arXiv:2602.06521*, 2026b.

William Ljungbergh, Adam Tonderski, Joakim Johnander, Holger Caesar, Kalle Åström, Michael Felsberg, and Christoffer Petersson. Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. In *European Conference on Computer Vision*, pp. 161–177. Springer, 2024.

Jianbiao Mei, Tao Hu, Xuemeng Yang, Licheng Wen, Yu Yang, Tiantian Wei, Yukai Ma, Min Dou, Botian Shi, and Yong Liu. Dreamforge: Motion-aware autoregressive video generation for multi-view driving scenes. *arXiv preprint arXiv:2409.04003*, 2024.

Jianbiao Mei, Yu Yang, Xuemeng Yang, Licheng Wen, Jiajun Lv, Botian Shi, and Yong Liu. Vision-centric 4d occupancy forecasting and planning via implicit residual world models. *arXiv preprint arXiv:2510.16729*, 2025.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

Davis Rempe, Jonah Philion, Leonidas J Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17284–17294. IEEE, 2022.

Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, et al. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models. *arXiv preprint arXiv:2506.09042*, 2025.

Luke Rowe, Roger Girgis, Rodrigue de Schaetzen, Daphne Cornelisse, Alaap Grandhi, Felix Heide, Eugene Vinitsky, Christopher Pal, and Liam Paull. Scaling self-play for end-to-end driving. *arXiv preprint arXiv:2606.19641*, 2026.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451. IEEE, 2020.

Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, et al. Simscale: Learning to drive via real-world simulation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36365–36374, 2026.

Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14895–14904. IEEE, 2024.

Hang Wang, Xin Ye, Feng Tao, Chenbin Pan, Abhirup Mallik, Burhan Yaman, Liu Ren, and Junshan Zhang. Adawm: Adaptive world model based planning for autonomous driving. In *International Conference on Learning Representations*, volume 2025, pp. 85591–85615, 2025.

Jiawei Wang, Hao Yu, Yongzhen Hu, Xinyi Yang, Tao Ni, Xin Zhan, Junbo Chen, Xiaowei Zhou, Ruizhen Hu, and Sida Peng. Infinisplat: Implicit gaussian decoding for large-baseline monocular view synthesis. *arXiv preprint arXiv:2608.02437*, 2026a.

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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9904–9913. IEEE, 2021.

Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *IEEE Transactions on Image Processing*, 35:4947–4960, 2026b.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024a.

Xinlin Wang, Yujiao Xiang, Yuheng Zhou, Jingqi Wang, Minqing Huang, Jiajie Huang, Dongxu Wei, Tingguang Zhou, Xiyang Wang, Gong Chen, et al. Wa-jepa: Rethinking the video jepa paradigm for world-action modeling in autonomous driving. *arXiv preprint arXiv:2608.20974*, 2026c.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14749–14759. IEEE, 2024b.

Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pp. 3–15. Springer, 2023.

Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE international intelligent transportation systems conference (ITSC)*, pp. 3095–3101. IEEE, 2021.

Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, et al. Scal3r: Scalable test-time training for large-scale 3d reconstruction. *arXiv preprint arXiv:2604.08542*, 2026.

Zhiyuan Xu, Bohan Li, Huan-ang Gao, Mingju Gao, Yong Chen, Ming Liu, Chenxu Yan, Hang Zhao, Shuo Feng, and Hao Zhao. Challenger: Affordable adversarial driving video generation. *arXiv preprint arXiv:2505.15880*, 2025.

Tianyi Yan, Dongming Wu, Wencheng Han, Junpeng Jiang, Xia Zhou, Kun Zhan, Cheng-zhong Xu, and Jianbing Shen. Drivingsphere: Building a high-fidelity 4d world for closed-loop simulation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27531–27541. IEEE, 2025a.

Tianyi Yan, Tao Tang, Xingtai Gui, Yongkang Li, Jiasen Zheng, Weiyao Huang, Lingdong Kong, Wencheng Han, Xia Zhou, Xueyang Zhang, et al. Ad-r1: Closed-loop reinforcement learning for end-to-end autonomous driving with impartial world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1085–1095, 2026.

Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pp. 156–173. Springer, 2024.

Ziyang Yan, Wenzhen Dong, Yihua Shao, Yuhang Lu, Haiyang Liu, Jingwen Liu, Haozhe Wang, Zhe Wang, Yan Wang, Fabio Remondino, et al. Renderworld: World model with self-supervised 3d label. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6063–6070. IEEE, 2025b.

Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. In *International Conference on Learning Representations*, volume 2024, pp. 16739–16766, 2024a.

Pengxuan Yang, Yupeng Zheng, Deheng Qian, Zebin Xing, Qichao Zhang, Linbo Wang, Yichen Zhang, Shaoyu Guo, Zhongpu Xia, Qiang Chen, et al. Dreamerad: Efficient reinforcement learning via latent world model for autonomous driving. *arXiv preprint arXiv:2603.24587*, 2026.

Xuemeng Yang, Licheng Wen, Tiantian Wei, Yukai Ma, Jianbiao Mei, Xin Li, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, et al. Drivearena: A closed-loop generative simulation platform for autonomous driving. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26933–26943. IEEE, 2025a.

Yu Yang, Alan Liang, Jianbiao Mei, Yukai Ma, Yong Liu, and Gim Hee Lee. X-scene: Large-scale driving scene generation with high fidelity and flexible controllability. *Advances in Neural Information Processing Systems*, 38:104415–104451, 2025b.

Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9327–9335, 2025c.

Yuanbo Yang, Yujun Shen, Yue Wang, Andreas Geiger, and Yiyi Liao. Urbangen: Urban generation with compositional and controllable neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025d.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1389–1399. IEEE, 2023.

Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14673–14684. IEEE, 2024b.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17239–17248. IEEE, 2025.

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

Zongchuang Zhao, Xin Zhou, Tianyang Xu, Zhengyang Sun, Kaixuan Zhou, Honglin Li, Dingkang Liang, and Xiang Bai. Simwam: A simple world action model for end-to-end autonomous driving. *arXiv preprint arXiv:2608.07468*, 2026.

Chaoda Zheng, Sean Li, Jinhao Deng, Zhennan Wang, Shijia Chen, Liqiang Xiao, Ziheng Chi, Hongbin Lin, Kangjie Chen, Boyang Wang, et al. X-world: Controllable ego-centric multi-camera world models for scalable end-to-end driving. *arXiv preprint arXiv:2603.19979*, 2026.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024a.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 87–104. Springer, 2024b.

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48:4673–4691, 2025.

Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21634–21643. IEEE, 2024.

Ziyue Zhu, Zhanqian Wu, Zhenxin Zhu, Lijun Zhou, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Jin Xie, et al. Worldsplat: Gaussian-centric feed-forward 4d scene generation for autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 68258–68277, 2026.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

World2Policy: Continual Policy Improvement via Scenario Curriculum in Interactive Gaussian Worlds

Appendix

Contents

A Additional Related Work **16**

A.1 Interactive World Models and Simulators for Autonomous Driving 16

A.2 Closed-Loop Policy Optimization for Autonomous Driving 16

A.3 Generative World Models for Autonomous Driving 16

B Additional Implementation Details **17**

B.1 General Setup 17

B.2 Policy Optimization Engine 17

B.3 Environment Engine 18

B.4 Dual-Engine Policy Improvement 18

C Additional Quantitative Results **20**

C.1 HUGSIM Metrics and Aggregation 20

C.2 Reward Component Ablation 21

C.3 Safety-Cost Design Ablation 21

C.4 Results by Difficulty Level 22

D Additional Qualitative Results **22**

D.1 Optimized Policy Behaviors 22

D.2 Comparison with the Base Policy 22

D.3 Policy Capability Boundaries across Scenario Curricula 22

D.4 Environment Engine across Curriculum Levels 27

A Additional Related Work

A.1 Interactive World Models and Simulators for Autonomous Driving

Autonomous driving simulation has evolved from engine-based simulators to data-driven interactive worlds. Engine- and logic-based simulators, such as CARLA and NAVSIM (Dosovitskiy et al., 2017; Dauner et al., 2024; Caesar et al., 2021; Li et al., 2022), provide scalable closed-loop evaluation and controllable interactions, but rely heavily on manually designed assets and rendering pipelines, which may introduce discrepancies from real-world sensor observations. Neural rendering approaches (Yang et al., 2023; Wu et al., 2023; Tonderski et al., 2024; Ljungbergh et al., 2024; Yang et al., 2024a; 2025d) address this limitation by learning photorealistic scene representations from real driving data, enabling realistic view synthesis and sensor simulation. However, their implicit representations often limit interactive editing and require costly scene-specific optimization. Recent Gaussian-based approaches (Yan et al., 2024; Zhou et al., 2024; Fei et al., 2024; Chen et al., 2025; Hess et al., 2025; Wang et al., 2026a; Xie et al., 2026) introduce explicit and editable scene representations, enabling efficient reconstruction, real-time rendering, and interactive driving simulation. In particular, HUGSIM (Zhou et al., 2025) reconstructs real-world scenes into photorealistic interactive Gaussian worlds with privileged geometry for closed-loop evaluation. Nevertheless, existing neural simulators mainly focus on constructing realistic environments and evaluating policies, while adapting environments according to policy capabilities remains an open challenge.

Complementary to environment reconstruction, safety-critical scenario generation methods (Wang et al., 2021; Rempe et al., 2022; Zhang et al., 2024; Liu et al., 2026a; Xu et al., 2025) explore rare-event discovery and controllable scenario synthesis for autonomous driving evaluation. These methods improve robustness analysis by generating challenging interactions, but typically target testing under fixed policies rather than continual policy improvement. Different from these approaches that primarily construct or evaluate interactive environments, World2Policy further leverages the interactive Gaussian world as an adaptive training environment for discovering policy failures and evolving scenario distributions.

A.2 Closed-Loop Policy Optimization for Autonomous Driving

Open-loop imitation learning methods, including recent end-to-end driving policies (Chitta et al., 2022; Al-Sharman et al., 2023; Jia et al., 2025; Tian et al., 2026), have achieved significant progress by learning from expert demonstrations. However, they rely on expert state-action distributions and cannot directly capture the consequences of policy-induced deviations during deployment. Closed-loop policy learning addresses this limitation by optimizing policies through environment interaction. Recent approaches, such as CarPlanner, WorldEngine, and DriveZero (Li et al., 2024; Zhang et al., 2025; Zou et al., 2025; Jaeger et al., 2025; Cornelisse et al., 2025; Gao et al., 2025; Yang et al., 2026; Yan et al., 2026; Li et al., 2026a;b; He et al., 2026), improve closed-loop driving through reinforcement learning, trajectory optimization, or learned world models. However, most existing methods optimize policies under fixed scenario distributions and lack mechanisms for adapting training scenarios to emerging failure modes.

Safe reinforcement learning formulates driving as a constrained Markov decision process and explicitly optimizes safety costs alongside task rewards. Methods such as CPO and CPPO (Achiam et al., 2017; Ray et al., 2019) provide principled solutions for balancing safety and performance. However, autonomous driving involves diverse safety constraints from dynamic actors, static obstacles, and drivable-area boundaries, which may require different avoidance strategies. A single policy may therefore suffer from risk transfer across failure modes or excessive conservatism. World2Policy combines constraint-specialized policy optimization with adaptive scenario generation to improve driving policies through continual environment-policy interaction.

A.3 Generative World Models for Autonomous Driving

Recent world models for autonomous driving learn to predict or generate future driving dynamics from large-scale visual and multimodal data. Video and multi-view generative models (Hu et al., 2023a; Wang et al., 2024a; Gao et al., 2024a;b; Ren et al., 2025; Yang et al., 2025a; Mei et al., 2024; Yan et al., 2025a; Zheng et al., 2026; Liang et al., 2026b; Kong et al., 2026) synthesize future observations and provide planning priors for end-to-end and vision-language-action driving policies.

Meanwhile, geometry-oriented world models (Zheng et al., 2024a; Yang et al., 2024b; 2025c; Wang et al., 2026b; Yan et al., 2025b; Mei et al., 2025; Zhu et al., 2026; Yang et al., 2025b; Dang et al., 2026; Liang et al., 2026a) predict future occupancy, BEV, point clouds, or Gaussian states to improve action-conditioned 3D consistency and planning. However, these models still face challenges in balancing generation quality, physical consistency, and long-horizon prediction accuracy.

Another line of work integrates world modeling objectives into driving policies, leveraging future prediction and action-conditioned latent representations to improve planning and end-to-end driving (Wang et al., 2024b; Zheng et al., 2024b; Kong et al., 2025; Li et al., 2026c; Zhao et al., 2026; Liu et al., 2026b; Wang et al., 2025). These approaches demonstrate the value of future-aware representations, but typically use world models as internal predictors or action evaluators rather than editable environments for policy improvement. Complementary to generative world models, WORLD2POLICY does not aim to predict future observations, but uses reconstructed interactive worlds to diagnose policy failures and generate targeted training scenarios.

B ADDITIONAL IMPLEMENTATION DETAILS

B.1 GENERAL SETUP

We use pretrained UniAD (Hu et al., 2023b), VAD (Jiang et al., 2023), and LTF (Dauner et al., 2024) as frozen base planners and train the residual experts on 300 HUGSIM scenarios disjoint from the 436 benchmark evaluation scenarios. HUGSIM renders the observations at each decision step and executes the selected waypoint trajectory through its low-level controller.

B.2 POLICY OPTIMIZATION ENGINE

The Policy Optimization Engine trains the residual experts while keeping the base planner fixed, then samples and selects trajectories as described in Sec. 4.2.

Multi-Expert Closed-Loop CPPO. We optimize each expert π_θ^e with CPPO using the objective in Eq. 4 and the Lagrangian in Eq. 5:

- **Rollouts and advantages.** Each expert collects on-policy transitions in the Gaussian world. Reward and expert-specific cost advantages are estimated separately with GAE.
- **Value estimation.** The residual policy has a reward-value head and separate cost-value heads for the safety channels. These value functions are initialized during value warm-up and updated alongside the residual policy.
- **Actor update.** Each expert uses the clipped CPPO actor objective defined in Eq. 9. The reward and cost terms share the same likelihood ratio, while the cost term uses pessimistic clipping. The full loss also includes value fitting, entropy regularization, and a KL penalty to a frozen reference policy.
- **Dual update.** Before each PPO update, the safety multiplier is updated according to Eq. 10, using the expert's discounted episode cost and safety budget. The effective violation includes a relative deadband and a per-update step cap.

Table 6 summarizes the expert fine-tuning settings.

Table 6: Hyperparameter configuration for constrained PPO training of residual experts.

Setting	Value	Setting	Value
Optimizer	Adam	Learning rate	1×10^{-4}
Updates per expert	24	Transitions per update	1,024
PPO epochs per update	4	Minibatch size	256
Discount factor	0.99	GAE parameter	0.95
PPO clipping parameter	0.2	Target KL divergence	0.02
Entropy coefficient	0.01	Lagrange learning rate	0.05
Initial Lagrange multiplier	0	Specialist cost multiplier	2

Expert Proposals and Selection. At each decision step, the base planner provides a nominal trajectory, and the experts produce complementary refinements that are pooled for privileged selection.

- **Base proposal.** The planner produces a nominal waypoint trajectory $\bar{\tau}_t$ and shared policy features.
- **Expert sampling.** Each expert samples one of seven driving modes {keep, slow, stop, left, right, left-slow, right-slow} from a categorical head and two mode-conditioned residual parameters from a diagonal Gaussian:

$$m_t^e \sim p_\theta^e(m \mid o_t), \quad z_t^e \sim \mathcal{N}\big(\mu_\theta^e(o_t, m_t^e), \mathrm{diag}\big((\sigma_\theta^e(m_t^e))^2\big)\big), \quad \delta_t^e = \frac{1 + \tanh z_t^e}{2} \in [0,1]^2 \tag{7}$$

  The entries of δ_t^e control longitudinal progress and lateral displacement.
- **Decoding and selection.** The decoder applies the sampled residual to $\bar{\tau}_t$ as in Eq. 6, anchoring the result to the current ego state and limiting progress to the reachable interval. The selector pools all proposals and assesses dynamic-actor risk, static-obstacle geometry, and drivable-area compliance before choosing a trajectory based on safety, progress, and comfort.

B.3 Environment Engine

The Environment Engine uses failures observed in closed-loop evaluation to construct new scenarios. Each generated case records its source failure and parent world, allowing us to distinguish the edited interaction from the reconstructed scene.

Failure Mining. Failure mining reduces each rollout to a compact record that identifies the failure event and its preceding interaction.

- **Event detection.** We record the first onset of a foreground collision, background collision, or drivable-area violation. If no such event occurs, an episode with route completion below 0.2 is recorded as a progress failure.
- **Failure anchoring.** For each event, we retain the preceding eight steps and use privileged actor states, when available, to identify the interacting actor and characterize the behavior leading to failure.
- **Recoverability filtering.** Counterfactual replay evaluates policy and recovery candidates around the anchor. Cases without a valid context or a verified safe alternative are excluded; each retained record stores the failure time, type, actor identity when available, and behavior pattern.

Scenario Parameterization and Search. For each retained failure, generation edits the associated interaction while preserving the parent Gaussian world. The process has three parts:

- **Recipe construction.** We change one bounded interaction parameter, typically the relevant actor's initial position, heading, or speed, while reusing the static Gaussian scene in Eq. 1. The edit is stored with its parent scenario and failure record.
- **Boundary search.** In the full-parent search, we evaluate eight nearby parameter settings around each stable source case to probe the policy's local failure boundary.
- **Validity and acceptance.** Before simulation, candidates must pass parameter, asset, route-coverage, initial-overlap, and short-rollout checks. We then evaluate valid candidates with the current policy and retain those that expose a recoverable failure.

Curriculum Update. Accepted scenarios are recorded in an overlay manifest alongside their parent and failure identifiers. The next training set combines these cases with the existing scenarios, providing the Policy Optimization Engine with new interactions while preserving coverage of earlier driving conditions.

B.4 Dual-Engine Policy Improvement

The Policy Optimization Engine and the Environment Engine form a coupled policy-improvement procedure. Starting from a frozen base planner and an initial scenario distribution, each curriculum iteration follows four stages: rollout collection, offline risk modeling, online constrained optimization, and expert specialization with scenario-curriculum update.

- **Base-policy initialization and rollout collection.** We freeze the pretrained base planner and initialize a shared residual policy conditioned on its nominal trajectory, shared features, and ego

state. Closed-loop rollouts in the Gaussian world provide the observations, rewards, safety costs, and context needed by the subsequent stages.

- **Offline risk modeling.** We fit separate cost critics for the three component safety costs that define Eq. 3 using the collected rollouts, and validate their estimates on held-out scene groups. We calibrate the expert-specific safety budgets from discounted episode costs under reward-only rollouts and freeze them before constrained optimization.
- **Online constrained optimization.** We first warm up the reward and cost value functions with online rollouts while keeping the actor and Lagrange multipliers fixed. After the value estimates pass their validation checks, we jointly optimize the residual policy, value functions, and multipliers with CPPO. This stage establishes a common initialization that balances the driving reward against the safety constraints.
- **Expert specialization and curriculum update.** We initialize constraint-specialized experts from the common policy and fine-tune them independently with the expert-specific cost weights in Eq. 3. All experts retain the shared reward and all safety constraints. Their stochastic trajectory proposals are pooled for privileged selection as described in Sec. 4.2; the Environment Engine then mines recoverable failures and augments the scenario distribution for the next iteration.

The first two stages establish the rollout data, cost estimates, and safety budgets. The following two components formalize how the Policy Optimization Engine updates the experts and how the Environment Engine uses the resulting closed-loop behavior to construct the next training distribution. We next describe the corresponding optimization and scenario-update rules, followed by their complete execution procedure.

Policy Optimization Engine: Constrained Expert Learning. For expert e, let $\rho_t^e(\theta) = \pi_\theta^e(a_t \mid o_t)/\pi_{\text{old}}^e(a_t \mid o_t)$ denote the policy ratio, and let $\hat{A}_t^{R,e}$ and $\hat{A}_t^{C,e}$ be the reward and safety-cost advantages estimated with GAE:

$$\hat{A}_t^{R,e} = \text{GAE}(r_t, \hat{V}_R^e), \qquad \hat{A}_t^{C,e} = \text{GAE}(c_t^e, \hat{V}_C^e), \qquad c_t^e = C^e(\mathcal{W}_t, a_t). \tag{8}$$

$$\mathcal{L}_{\text{actor}}^e = -\mathbb{E}_t\Big[\min\Big(\rho_t^e \hat{A}_t^{R,e}, \text{clip}(\rho_t^e, 1-\epsilon, 1+\epsilon)\hat{A}_t^{R,e}\Big)\Big] + \lambda^e \mathcal{L}_{\text{cost}}^e, \tag{9}$$

where $\mathcal{L}_{\text{cost}}^e$ uses the corresponding safety-cost advantage with pessimistic clipping. The complete CPPO objective additionally includes value fitting, entropy regularization, and a KL penalty to a frozen reference policy. Before each policy update, the multiplier is updated from the discounted episode cost $\widehat{J}_{C^e}$:

$$\lambda^e \leftarrow \text{clip}(\lambda^e + \eta_\lambda \widetilde{g}^e, 0, \lambda_{\max}), \qquad g^e = \frac{\widehat{J}_{C^e} - d^e}{d^e}, \tag{10}$$

where $\widetilde{g}^e$ applies the relative deadband and per-update step cap to g^e, and d^e is the expert-specific budget.

After the CPPO updates, the experts form the policy evaluated by the Environment Engine. At each decision step, expert e samples K_e trajectory proposals, which are pooled and ranked by the privileged selector:

$$\mathcal{C}_t = \bigcup_{e \in \mathcal{E}} \{\tau_{t,j}^e\}_{j=1}^{K_e}, \qquad \tau_t^* = S(\mathcal{C}_t), \tag{11}$$

where S is the privileged selector and τ_t^* is the trajectory executed by the controller. This selected policy is the input to the Environment Engine below.

Environment Engine: Failure-Adaptive Scenario Generation. Let Π_k denote the current multi-expert policy with privileged selection and let $\mathcal{D}_k$ denote its training scenario distribution. After the Policy Optimization Engine updates the policy on $\mathcal{D}_k$, closed-loop evaluation of the resulting policy Π_{k+1} produces a failure set:

$$\begin{aligned} \mathcal{F}_k &= \text{MineFailures}(\text{Rollout}(\Pi_{k+1}, \mathcal{D}_k)), \\ \mathcal{G}_k &= \text{Accept}_{\text{valid,recoverable}}(\text{CEM}(\mathcal{F}_k; \Pi_{k+1})), \\ \mathcal{D}_{k+1} &= \mathcal{D}_k \cup \mathcal{G}_k. \end{aligned} \tag{12}$$

where CEM searches bounded actor and motion parameters while preserving the reconstructed world, and the acceptance operator retains only physically valid and recoverable cases.

The interaction between the two engines is summarized in Algorithm 1.

Algorithm 1 Continual Policy Improvement with Policy Optimization and Environment Engines

Require: Frozen base planner π_0, initial scenario distribution $\mathcal{D}_0$, expert set $\mathcal{E}$, curriculum iterations K, CPPO updates U
Ensure: Trained experts $\{\pi_K^e\}_{e\in\mathcal{E}}$ and selected policy Π_K

```
Initialize the shared residual policy from π_0, copy it to the experts, and set D ← D_0.
for k = 0, ..., K − 1 do
    Collect frozen-base rollouts on D; fit the three cost critics and freeze budgets {d^e}.
    Warm up the reward and cost value functions with online rollouts while fixing the actor and multipliers.
    for u = 1, ..., U do
        Collect on-policy rollouts for every expert on D and estimate Â^{R,e} and Â^{C,e} using Eq. 8.
        Update λ^e using Eq. 10; update θ^e ← θ^e − α_θ ∇_θ L^e_actor using Eq. 9, and update the value functions.
    end for
    Form Π_{k+1} by pooling stochastic expert proposals and applying S.
    F_k ← MineFailures(Rollout(Π_{k+1}, D)); G_k ← Accept_{valid,recoverable}(CEM(F_k; Π_{k+1})); set D ← D ∪ G_k.
end for
return {π_K^e}_{e∈E} and the selected policy Π_K.
```

C Additional Quantitative Results

This section describes the HUGSIM metrics and aggregation protocol (Section C.1), ablates the reward components (Section C.2) and safety-cost design (Section C.3), and reports results across difficulty levels (Section C.4).

C.1 HUGSIM Metrics and Aggregation

We use the six HUGSIM metrics below; higher values indicate better performance. The component checks follow the HUGSIM evaluator, including its geometric tests and kinematic thresholds (Zhou et al., 2025).

- **No Collision (NC):** collision avoidance with traffic actors and static scene geometry.
- **Drivable Area Compliance (DAC):** compliance of the ego footprint along the evaluated trajectory with the reconstructed drivable region.
- **Time to Collision (TTC):** satisfaction of the evaluator's predictive collision checks under extrapolated motion.
- **Comfort (COM):** satisfaction of acceleration, jerk, yaw-rate, and yaw-acceleration limits.
- **Route Completion (RC):** progress along the reference route, capped at one.
- **HUGSIM Driving Score (HD-Score):** route completion weighted by the time-averaged joint safety and comfort score.

The evaluator assesses planned trajectories at scored keyframes of a closed-loop rollout. Let $\mathcal{I}$ denote these keyframes and let NC_t, DAC_t, TTC_t, and COM_t be their component scores. The scenario-level HD-Score is

$$\text{HD-Score} = \mathrm{RC}\,\frac{1}{|\mathcal{I}|}\sum_{t\in\mathcal{I}}\mathrm{NC}_t\,\mathrm{DAC}_t\,\frac{5\,\mathrm{TTC}_t + 2\,\mathrm{COM}_t}{7} \tag{13}$$

NC, DAC, TTC, and COM are averaged over the same scored keyframes. HD-Score is computed for each scenario before aggregation. These evaluation metrics are distinct from the reward and safety-cost functions used for policy optimization.

Within each dataset-difficulty group, we average scenario scores uniformly. For each difficulty level, we then average the four dataset scores with equal weight. Overall scores weight the difficulty-level scores by their respective scenario counts. We apply the same aggregation to all six metrics and report them in the native 0–100 range.

Table 7: **Ablation of reward components.** Each variant removes one reward term while keeping the safety costs, selector, and training protocol fixed.

Reward Variant	Reward Terms			Driving Diagnostics			HUGSIM Metrics					
	Progress	Deviation	Comfort	Progress (m)↑	Deviation (m)↓	Comfort Viol. (%)↓	NC↑	DAC↑	TTC↑	COM↑	RC↑	HD-Score↑
Full Reward (Ours)	✓	✓	✓	46.06	0.94	28.5	57.9	95.0	51.3	84.8	44.7	**29.3**
w/o Progress	✗	✓	✓	38.50	0.90	23.5	60.5	95.5	53.8	86.2	36.2	25.0
w/o Deviation	✓	✗	✓	49.52	1.35	31.0	56.8	93.6	49.8	83.8	43.8	27.3
w/o Comfort	✓	✓	✗	47.10	1.02	44.0	56.8	94.7	49.5	76.0	44.9	26.4

Table 8: **Ablation of safety-cost design.** Each variant removes one safety cost from CPPO training, while all foreground, background, and drivable violations remain evaluated.

Cost Variant	Active Safety Costs			Safety Diagnostics			HUGSIM Metrics					
	FG	BG	DRV	FG Coll.↓	BG Coll.↓	DRV Viol.↓	NC↑	DAC↑	TTC↑	COM↑	RC↑	HD-Score↑
All Three Costs (Ours)	✓	✓	✓	**33.9**	**3.7**	**16.7**	**57.9**	**95.0**	**51.3**	**84.8**	**44.7**	**29.3**
w/o FG Cost	✗	✓	✓	42.0	5.2	20.5	54.8	94.2	48.7	83.5	43.5	26.8
w/o BG Cost	✓	✗	✓	36.8	10.5	20.0	55.5	94.0	49.3	83.0	42.9	26.4
w/o DRV Cost	✓	✓	✗	35.5	4.6	26.0	56.8	91.8	50.1	82.5	42.1	25.9
No Safety Cost	✗	✗	✗	43.5	12.5	29.0	51.0	90.5	44.0	80.0	39.5	23.5

C.2 Reward Component Ablation

Table 7 ablates the three driving-reward components: progress, route deviation, and comfort. Progress is the mean cumulative advance along the reference-route arc length after projecting the ego trajectory onto the reference route. Mean route deviation averages the lateral deviation within each episode and then across episodes. Comfort Violation Rate is the percentage of episodes with at least one acceleration, jerk, yaw-rate, or yaw-acceleration threshold violation.

The full reward achieves a mean progress of 46.06 m, a mean deviation of 0.94 m, a 28.5% Comfort Violation Rate, and a 29.3 HD-Score. Without Progress, safety improves (NC/DAC/TTC: $60.5/95.5/53.8$), but RC drops to 36.2 because the policy avoids risk by braking or stopping, causing frequent stagnation. Without Deviation, progress increases to 49.52 m while deviation grows to 1.35 m and RC falls to 43.8, showing that extra distance can come from wider evasions rather than better route completion. Without Comfort, progress and RC remain comparable to the full reward, but the violation rate rises to 44.0% and COM falls to 76.0 due to more aggressive steering. These results show that the three terms address complementary failure modes, with the full reward providing the best overall trade-off.

C.3 Safety-Cost Design Ablation

Table 8 ablates the three safety-cost components. The foreground (FG) cost penalizes collisions with dynamic traffic actors, the background (BG) cost penalizes collisions with static scene geometry, and the drivable (DRV) cost penalizes violations of the drivable area. All three violation types remain evaluated for every variant.

The full cost design achieves FG/BG/DRV violation rates of $33.85/3.65/16.67$ and an HD-Score of 29.3. Removing FG increases FG collisions to 42.0 and lowers NC/TTC to $54.8/48.7$. Removing BG produces the clearest background-risk increase, with BG collisions rising to 10.5 and RC falling to 42.9. Removing DRV increases drivable violations to 26.0 and reduces DAC to 91.8, indicating that the policy more often leaves the drivable region when this constraint is absent. In all three cases, the non-target costs also increase to some extent, showing risk transfer between constraint channels. Removing all safety costs yields the highest combined violation rates, the lowest NC/DAC/TTC/COM values, and the lowest RC and HD-Score (39.5 and 23.5). These results demonstrate that the three costs provide complementary protection and are most effective when enforced jointly.

Table 9: **Difficulty-wise results on HUGSIM.** We report the six official metrics for UniAD, VAD, and LTF integrated with WORLD2POLICY across four difficulty levels.

Method	Difficulty	NC↑	DAC↑	TTC↑	COM↑	RC↑	HD-Score↑
UniAD + World2Policy	Easy	84.73	96.83	74.62	70.41	69.42	66.87
	Medium	75.18	94.71	64.83	61.78	52.37	45.62
	Hard	68.42	92.36	57.28	57.36	37.84	25.14
	Extreme	64.08	91.85	49.42	55.98	34.34	8.11
	Overall	72.81	93.90	61.28	61.04	47.97	36.12
VAD + World2Policy	Easy	78.10	95.02	69.61	95.91	67.85	51.21
	Medium	63.94	90.47	46.29	92.33	31.21	15.46
	Hard	62.10	88.56	45.72	90.85	23.81	11.96
	Extreme	57.67	87.98	43.87	89.99	21.64	10.59
	Overall	64.62	90.33	49.82	92.10	33.96	20.02
LTF + World2Policy	Easy	81.15	96.11	74.63	98.34	86.81	66.37
	Medium	63.19	96.55	52.53	90.52	51.48	32.52
	Hard	41.48	92.88	35.31	86.03	31.47	15.77
	Extreme	36.56	93.32	30.05	83.98	26.52	11.33
	Overall	55.37	94.89	47.43	89.40	47.57	29.96

C.4 RESULTS BY DIFFICULTY LEVEL

Table 9 reports the six HUGSIM metrics across four difficulty levels for UniAD, VAD, and LTF integrated with WORLD2POLICY. Performance degrades consistently as difficulty increases, with the largest reductions in TTC, RC, and HD-Score. UniAD achieves the strongest overall HD-Score and RC among the three integrations, while LTF obtains the highest DAC and VAD the highest COM. At the extreme level, LTF and VAD slightly exceed UniAD in HD-Score, indicating that the relative advantage of the backbone can vary in the most challenging scenarios.

D ADDITIONAL QUALITATIVE RESULTS

D.1 OPTIMIZED POLICY BEHAVIORS

Figures 6 and 7 show temporal rollouts of the optimized policy in four interactive driving scenarios. With paired camera and bird's-eye-view (BEV) views, the policy stops or yields when another vehicle blocks its path, performs overtaking maneuvers when appropriate, and resumes route progress after the interaction. These examples indicate that the optimized policy can coordinate longitudinal and lateral decisions across cut-in, oncoming-traffic, lane-changing, and crossing-vehicle interactions.

D.2 COMPARISON WITH THE BASE POLICY

Figure 8 compares the base policy with its WORLD2POLICY-enhanced counterpart in four driving scenes. The base policy incurs collisions, off-road failures, or both, whereas WORLD2POLICY succeeds in all four illustrated cases. These paired examples illustrate improved collision avoidance and drivable-area compliance in challenging interactions.

D.3 POLICY CAPABILITY BOUNDARIES ACROSS SCENARIO CURRICULA

Figure 9 evaluates the policy on two scenes spanning four curriculum levels, from clear road to lane blocking, oncoming traffic, and a head-on vehicle. The first scene remains successful at all levels, whereas the second scene succeeds through the hard level but fails at the extreme level. This contrast shows that the policy has a scenario-dependent capability boundary: increasing interaction difficulty can eventually expose an unresolved failure mode, while the boundary itself depends on the surrounding scene and actor configuration.

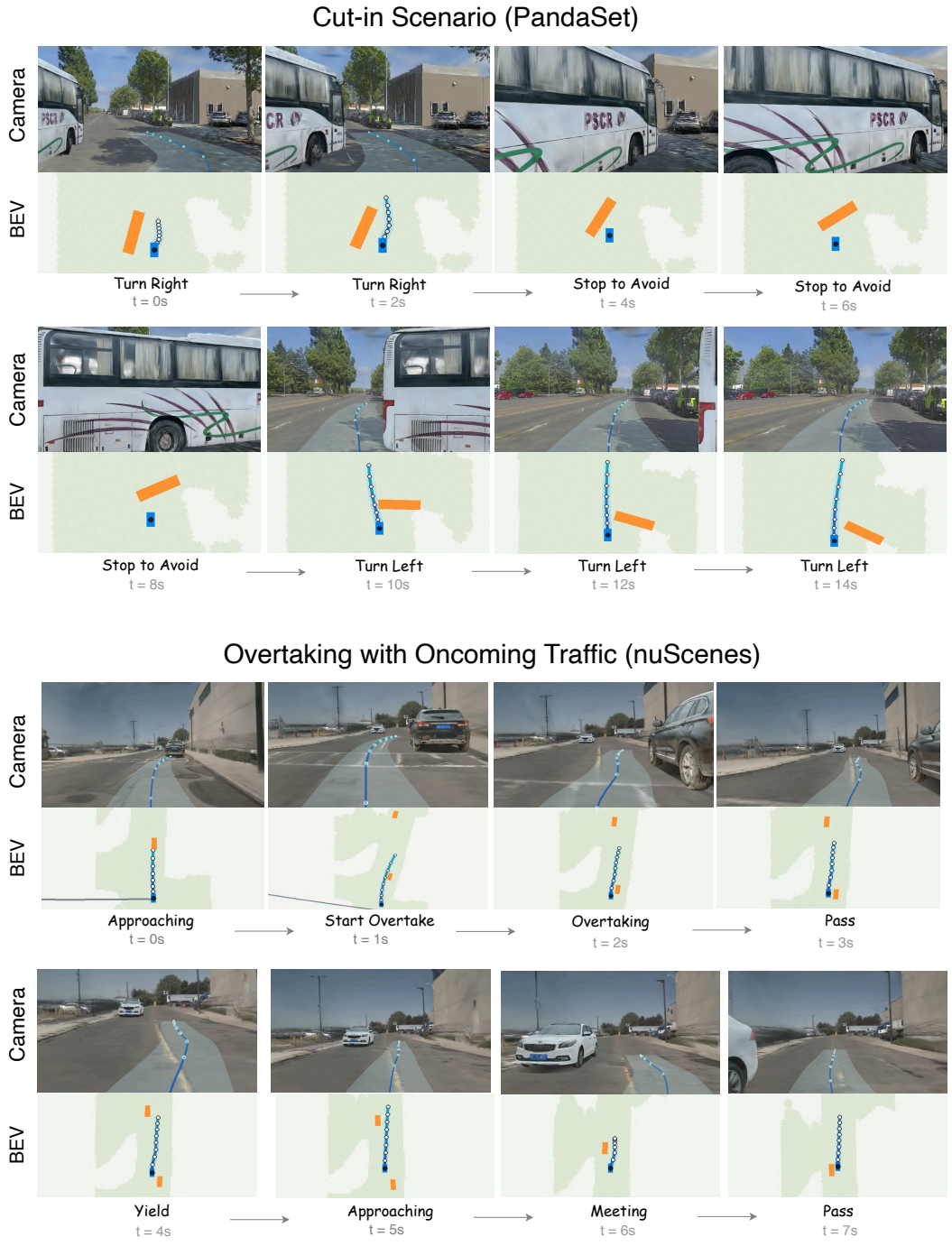

Figure 6: **Additional qualitative results of the optimized policy.** Top: a cut-in scenario in PandaSet, where the ego vehicle stops for a crossing bus before resuming motion. Bottom: an overtaking scenario in nuScenes, where the ego vehicle passes a slower vehicle and yields to oncoming traffic.

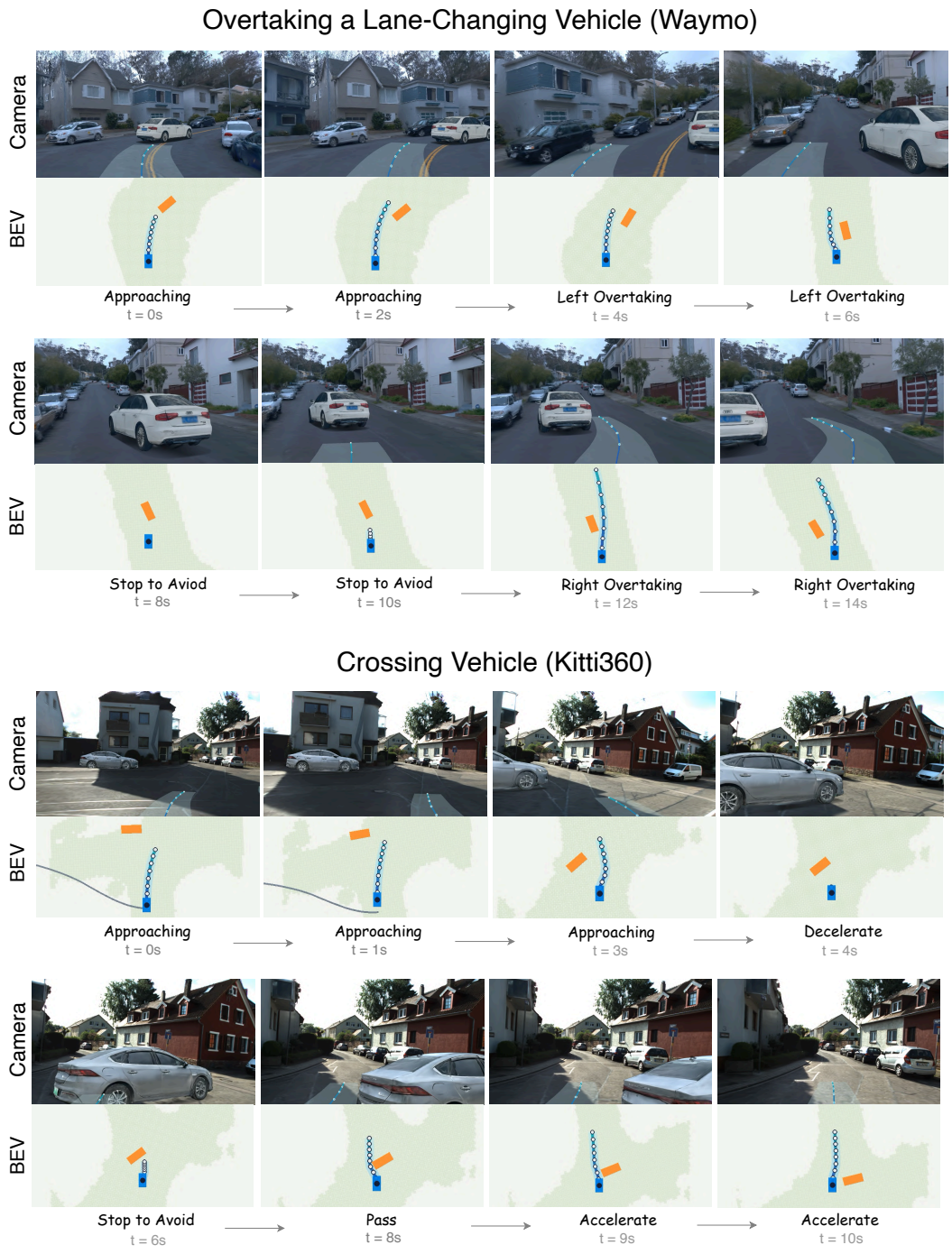

Figure 7: **Additional qualitative results of the optimized policy.** Top: an overtaking scenario in Waymo, where the ego adapts its motion to a lane-changing vehicle. Bottom: a crossing-vehicle scenario in KITTI-360, where the ego decelerates, stops and resumes after the crossing vehicle passes.



Figure 8: **Qualitative comparison between the base policy and World2Policy-enhanced variant.** In four challenging scenes, the base policy exhibits collision or off-road failures, while the World2Policy-enhanced policy completes the illustrated interactions and achieves goals successfully.



Figure 9: **Additional scenario curriculum examples.** Two driving scenes are shown across four difficulty levels: clear road (easy), lane blocking (medium), oncoming vehicle (hard), and head-on vehicle (extreme). The first scene succeeds at all levels, while the second fails at the extreme level.

D.4 Environment Engine across Curriculum Levels

Figure 10 illustrates how the Environment Engine generates a controlled progression from a base scenario (L0) to same-direction (L1), opposite-direction (L2), and adversarial (L3) interactions. By varying the positions and motions of causal actors while preserving the reconstructed scene, the engine produces increasingly targeted and physically plausible scenarios for subsequent policy training. Combined with failure mining, these scenario updates help keep the training distribution aligned with the policy's evolving capability boundary, thereby supporting continual policy improvement.



Figure 10: **Additional scenario curriculum progression.** Two driving scenes progress from the base scenario (L0) through same-direction (L1), opposite-direction (L2), and adversarial (L3) interactions.