CRAFT: Counterfactual-to-Interactive Reinforcement Fine-Tuning for Driving Policies

Keyu Chen[1,2] **Nanfei Ye**[2] **Yida Wang**[2] **Wenchao Sun**[1] **Danqi Zhao**[1] **Hao Cheng**[1] **Sifa Zheng**[1]

[1] School of Vehicle and Mobility, Tsinghua University [2] Li Auto Inc

Abstract

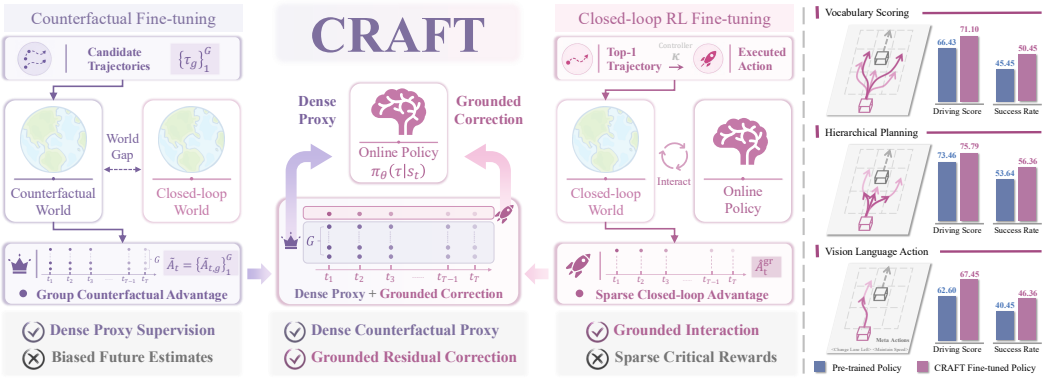
Open-loop imitation learning has advanced modern autonomous driving policy architectures, but closed-loop deployment remains vulnerable to policy-induced distribution shift. Existing post-training paradigms exhibit fundamental trade-offs: closed-loop RL fine-tuning provides grounded feedback from executed actions but is constrained by the sparsity of informative events, whereas counterfactual fine-tuning provides dense supervision over candidate futures but inherits bias from imperfect future estimates. We introduce **Counterfactual-to-Interactive Reinforcement Fine-Tuning (CRAFT)**, an on-policy framework that formulates closed-loop post-training as proxy-residual optimization. CRAFT uses group-normalized counterfactual advantages as a dense proxy for real closed-loop advantages and aligns this proxy with the closed-loop world through grounded residual correction from interaction-critical events. To stabilize adaptation, CRAFT regularizes the online policy toward an EMA teacher via asymmetric KL self-distillation. Theoretically, CRAFT decomposes the real closed-loop policy gradient into proxy and residual terms under the same visited-state distribution, reducing residual variance with an aligned proxy while mitigating proxy bias through grounded residual approximation. Empirically, CRAFT achieves the strongest closed-loop gains on Bench2Drive across hierarchical planning, vision-language-action, and vocabulary-scoring architectures. Ablations, scaling behavior, stability analyses, and transfer results further validate the complementary roles of dense counterfactual proxy and grounded residual correction. Project page: https://currychen77.github.io/CRAFT/.

1 Introduction

Modern autonomous driving policies have rapidly evolved beyond modular perception–planning pipelines. Representative frameworks include unified end-to-end models [1, 2, 3], vision-language-action models [4, 5], and world action models [6, 7]. While these architectures expand the capability of driving policies, their training remains largely governed by an open-loop imitation learning (IL) paradigm. This paradigm often leads to distribution shift due to the inherent discrepancy between open-loop training and closed-loop deployment. Therefore, further improving the closed-loop performance of pre-trained driving policies remains critical for safe and reliable deployment.

To address the challenges of distribution shift, two distinct post-training paradigms have emerged. As illustrated in Figure 1, closed-loop RL fine-tunes pre-trained driving policies within interactive simulators to obtain grounded feedback from executed actions. However, the inherent sparsity of informative events across diverse scenarios often constrains the training efficiency of pure closed-loop RL. Conversely, counterfactual fine-tuning offers an alternative by evaluating multiple candidate futures from the same visited state, transforming these relative performance differences into dense supervisory signals through group-relative objectives, such as GRPO [8]. Nevertheless, counterfactual estimates of the future remain inherently biased, as the counterfactual world cannot faithfully capture the interaction dynamics of the real closed-loop world.

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



(a) CRAFT: *Proxy–Residual* Integration of Fine-Tuning Paradigms (b) Cross-Paradigm Performance Gains

Figure 1: (a) CRAFT integrates dense counterfactual supervision and grounded correction through on-policy proxy–residual optimization. (b) CRAFT improves closed-loop performance across three driving-policy paradigms.

The complementary strengths of these two paradigms motivate a natural question: can counterfactual supervision serve as a dense proxy, while closed-loop interaction corrects its residual mismatch with the real closed-loop world? CRAFT addresses this question by assigning distinct optimization roles to the two signals. Counterfactual supervision serves as a dense proxy over states visited by the current policy, whereas closed-loop interaction provides grounded residual correction to align this proxy with the real closed-loop world.

Building on this proxy-residual decomposition, we introduce Counterfactual-to-Interactive Reinforcement Fine-Tuning (CRAFT), an on-policy post-training framework for pre-trained driving policies. As illustrated in Figure 1, CRAFT uses the counterfactual world to evaluate groups of candidates from states visited by the online policy and converts their efficiency, safety, and rule-compliance returns into group-normalized counterfactual advantages. These advantages serve as a dense proxy for real closed-loop advantages. CRAFT then derives grounded residual correction from interaction-critical traffic events observed in executed closed-loop rollouts, aligning the proxy with the real closed-loop world through a value-free dual-clipped update [9, 10]. To stabilize adaptation, asymmetric-KL self-distillation regularizes the online policy toward an EMA teacher initialized from the pre-trained policy, preserving reliable behaviors while allowing interaction-driven correction.

Theoretically, CRAFT decomposes the real closed-loop policy gradient into proxy and residual terms under the same state distribution, reducing residual variance with an aligned proxy while mitigating proxy bias through grounded residual approximation. Dual clipping further keeps rare critical events informative without dominating the update. Empirically, CRAFT achieves the largest closed-loop gains on Bench2Drive across policy architectures, including hierarchical planning, vision-language-action, and vocabulary-scoring. Ablations verify the complementary roles of each component, with the counterfactual proxy providing broad optimization signals, grounded residual correction delivering substantial closed-loop gains on interaction-sensitive metrics, and self-distillation stabilizing adaptation around reliable pre-trained behavior. Scaling and stability analyses further demonstrate that, with increasing fine-tuning data, CRAFT delivers more consistent gains and better training stability than baselines, while transfer results indicate partial robustness beyond the training distribution.

Our main contributions are summarized as follows:

- We formulate closed-loop post-training for pre-trained driving policies through a proxy-residual gradient decomposition, revealing the trade-off between dense but biased counterfactual supervision and sparse but grounded closed-loop interaction under the same visited-state distribution.
- We propose CRAFT, an on-policy post-training framework that turns counterfactual supervision into a dense proxy for real closed-loop advantages and uses interaction-critical closed-loop events as grounded residual correction, stabilized by self-distillation from the pre-trained policy.
- We substantiate the proxy-residual design theoretically and empirically. CRAFT reduces residual variance with an aligned proxy while mitigating proxy bias through grounded residual approximation, and achieves consistent closed-loop gains on Bench2Drive across diverse policy architectures, including hierarchical planning, vision-language-action, and vocabulary-scoring.

2 Related Work

End-to-End Autonomous Driving Policy. Progress in end-to-end driving has been driven largely by advances in model architecture. Diffusion-based models [11, 12, 13, 14], vision-language-action models [15, 16, 17, 18], and trajectory-vocabulary scoring methods [19, 20, 21, 22] all improve policy expressiveness. Yet their training remains largely imitation-based, with policies trained near expert states and then expected to remain reliable after their own actions shift future observations. This assumption is especially fragile in closed-loop deployment, where small planning errors can move the ego vehicle outside the training distribution. CRAFT addresses this gap by fine-tuning representative driving policies under closed-loop interaction, rather than introducing another architecture.

Closed-Loop RL in Autonomous Driving. Closed-loop RL mitigates distribution shift by optimizing policies on executed rollouts, often with policy-gradient methods such as PPO [23] in CARLA [24, 25] or reconstructed simulators [26, 27, 28]. While grounded in real interaction, these methods remain sample-inefficient because informative events are rare, requiring extensive rollouts to obtain useful gradient estimates. Recent approaches improve efficiency through VLM-based reward densification [29] or world-model rollouts [30, 31], but their supervision depends on evaluator or model fidelity. CRAFT instead uses closed-loop interaction only to correct a dense counterfactual proxy, preserving grounded feedback without estimating full advantages from sparse rollouts.

Counterfactual Post-Training in Autonomous Driving. Counterfactual post-training improves learning efficiency through dense local comparisons on real visited states [32], making preference-based and group-relative objectives attractive [8, 33]. Recent methods combine PDM Score [34, 35, 36, 37, 38, 39] with GRPO in NAVSIM [40] for policy fine-tuning, or use synthetic counterfactual data [41, 42] for offline post-training. A key limitation lies in the rollout condition, since counterfactual futures generated offline or in a non-reactive simulator do not capture true closed-loop interaction with the environment. CRAFT uses counterfactual futures as a dense proxy on real visited states, while grounded residual correction refines the proxy with feedback from true closed-loop rollouts.

3 Methodology

CRAFT is a general paradigm for closed-loop post-training of supervised driving policies. The target is to improve closed-loop performance under the on-policy state distribution. Existing post-training paradigms present a fundamental trade-off: closed-loop RL is physically grounded but relies on sparse, high-variance interaction feedback, whereas counterfactual optimization provides dense supervision but suffers from biased future estimates. CRAFT addresses this trade-off with two complementary components: a dense counterfactual proxy that provides trajectory-level supervision via candidate evaluation, and a grounded residual correction that compensates for proxy bias using executed closed-loop feedback. To preserve pre-trained behavior during fine-tuning, CRAFT further applies on-policy self-distillation with an EMA teacher. The framework is illustrated in Figure 2.

3.1 Real Closed-Loop Objective and Proxy–Residual Decomposition

We first formalize the real closed-loop objective that CRAFT aims to improve. We then show that CRAFT optimizes it through two complementary terms defined on the same on-policy state distribution. The key idea is to use dense counterfactual supervision as a proxy for the full closed-loop learning signal, while using grounded interaction feedback to correct the residual mismatch.

Let $\mathcal{M}_{\text{real}} = (\mathcal{S}, \mathcal{A}, P_{\text{real}}, r_{\text{real}}, \gamma)$ denote the real closed-loop MDP, with state space $\mathcal{S}$, low-level action space $\mathcal{A}$, transition function P_{real}, reward function r_{real}, and discount factor $\gamma \in (0, 1)$. The policy scores candidate trajectories $\tau \in \mathcal{T}$ by $\pi_\theta(\tau \mid s)$, and the controller κ tracks the selected trajectory τ_t by executing $a_t = \kappa(s_t, \tau_t)$, with rewards evaluated after the induced controller action. Let $d^{\pi_\theta}_{\text{real}}$ be the discounted state-visitation measure. For closed-loop rollout ξ, the objective is

$$J_{\text{real}}(\pi_\theta) = \mathbb{E}_{\xi \sim (\pi_\theta, \kappa, P_{\text{real}})}\Big[\sum_{t=0}^{\infty} \gamma^t r_{\text{real}}(s_t, a_t, s_{t+1})\Big], \tag{1}$$

whose policy gradient is

$$\nabla_\theta J_{\text{real}}(\pi_\theta) = \mathbb{E}_{s \sim d^{\pi_\theta}_{\text{real}},\, \tau \sim \pi_\theta(\cdot|s)}\Big[\nabla_\theta \log \pi_\theta(\tau \mid s)\, A^{\text{real}}_{\pi_\theta}(s, \tau)\Big]. \tag{2}$$

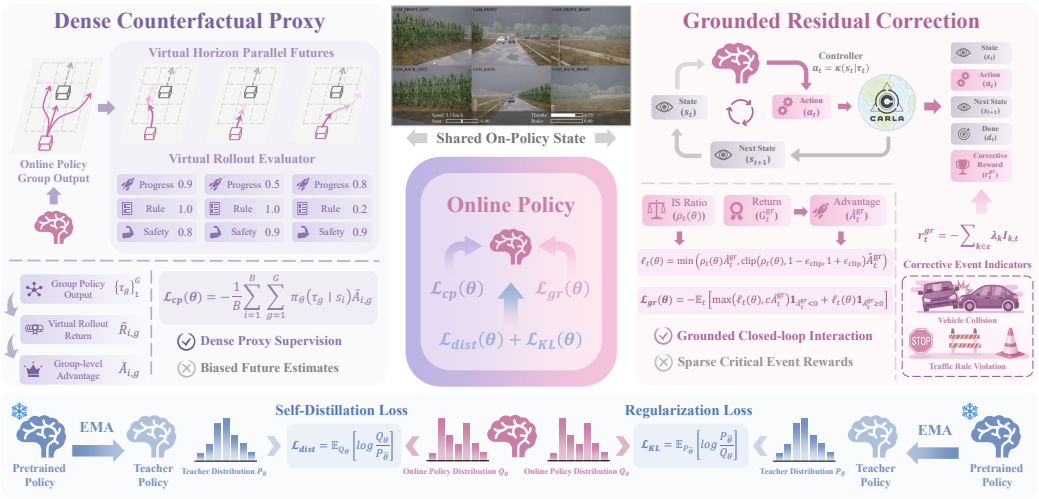


Figure 2: CRAFT combines dense counterfactual proxy supervision, grounded residual correction from real closed-loop interaction, and on-policy self-distillation toward reliable pre-trained behavior.

Here $A^{\text{real}}_{\pi_\theta}(s,\tau)$ denotes the real advantage of selecting τ and executing its induced controller action a. Grounded rollouts alone are sample-inefficient because rare critical events provide weak learning signal for most visited states, while the resulting failures dominate return variance. Consequently, value estimation is brittle and value-free optimization suffers from high-variance updates.

CRAFT introduces a dense counterfactual proxy on real visited states. For each $s \sim d^{\pi_\theta}_{\text{real}}$, the counterfactual world assigns each candidate trajectory τ a proxy value $\Phi_{\pi_\theta}(s,\tau)$, such as a short-horizon rollout score or a group-relative preference. As Φ_{π_θ} only approximates the real advantage, its residual mismatch is $\Delta_{\pi_\theta}(s,\tau) \triangleq A^{\text{real}}_{\pi_\theta}(s,\tau) - \Phi_{\pi_\theta}(s,\tau)$.

Proposition 3.1 (Exact Counterfactual-Residual Decomposition)**.** *For any integrable dense counterfactual proxy $\Phi_{\pi_\theta}(s,\tau)$ defined on real visited states, the closed-loop gradient decomposes as*

$$\nabla_\theta J_{\text{real}}(\pi_\theta) = \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(\tau \mid s)\, \Phi_{\pi_\theta}(s,\tau)\Big] + \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(\tau \mid s)\, \Delta_{\pi_\theta}(s,\tau)\Big], \tag{3}$$

where both expectations are over $s \sim d^{\pi_\theta}_{\text{real}}$ and $\tau \sim \pi_\theta(\cdot \mid s)$. The proof is provided in Appendix A.1.

The proxy term captures the dense component of the real objective, while the residual term accounts for the mismatch between the proxy and real advantage. Since both terms are defined on real visited states, the decomposition introduces no state-distribution mismatch.

The decomposition can reduce variance when the proxy is aligned with the real advantage, since the residual term only needs to estimate the remaining mismatch through grounded rollouts.

Proposition 3.2 (Conditional Variance Reduction)**.** *Fix a state s and let $Y(s,\tau) = A^{\text{real}}_{\pi_\theta}(s,\tau)$ and $C(s,\tau) = \Phi_{\pi_\theta}(s,\tau)$. For $R_\alpha(s,\tau) = Y(s,\tau) - \alpha C(s,\tau)$ with $\tau \sim \pi_\theta(\cdot \mid s)$ and $\mathrm{Var}_\tau[C(s,\tau)] > 0$,*

$$\mathrm{Var}_\tau[R_\alpha(s,\tau)] = \mathrm{Var}_\tau[Y(s,\tau)] + \alpha^2 \mathrm{Var}_\tau[C(s,\tau)] - 2\alpha \mathrm{Cov}_\tau[Y(s,\tau), C(s,\tau)], \tag{4}$$

and the minimizing coefficient is $\alpha^\star(s) = \mathrm{Cov}_\tau[Y(s,\tau), C(s,\tau)]/\mathrm{Var}_\tau[C(s,\tau)]$. In particular, if $\mathrm{Cov}_\tau[Y(s,\tau), C(s,\tau)] > \frac{1}{2}\mathrm{Var}_\tau[C(s,\tau)]$, then $\mathrm{Var}_\tau[Y(s,\tau) - C(s,\tau)] < \mathrm{Var}_\tau[Y(s,\tau)]$. If this condition holds for $d^{\pi_\theta}_{\text{real}}$-almost every s, the same strict inequality holds after averaging over s.

The proof is in Appendix A.3. Importantly, the proxy need not be unbiased, since residual variance reduction only requires the proxy to be sufficiently aligned with the real advantage.

CRAFT instantiates this decomposition with the surrogate objective:

$$\max_{\pi_\theta} \underbrace{\mathbb{E}_{s\sim d^{\pi_\theta}_{\text{real}}}\Big[\mathcal{R}_{\text{cp}}(s;\pi_\theta)\Big]}_{\text{Dense Counterfactual Proxy}} + \underbrace{\mathbb{E}_{\xi\sim(\pi_\theta,\kappa,P_{\text{real}})}\Big[\sum_t \gamma^t r^{\text{gr}}_t\Big]}_{\text{Grounded Residual Correction}}, \tag{5}$$

where the two objective components are defined in Sections 3.2 and 3.3.

3.2 Dense Counterfactual Proxy

For each on-policy state s_i collected from closed-loop interaction, the counterfactual world evaluates a group of G candidate trajectories $\{\tau_g\}_{g=1}^{G}$ over a finite virtual horizon. Evaluation criteria covering efficiency, safety, and rule compliance yield a scalar return $\tilde{R}_{i,g}$ for each candidate. Implementation details of the counterfactual world and reward design are provided in Appendices D.3 and D.4. We convert these returns into local relative scores by normalizing within each candidate group:

$$\tilde{A}_{i,g} = \frac{\tilde{R}_{i,g} - \mu_i}{\sigma_i + \varepsilon_{\text{norm}}}, \qquad \mu_i = \frac{1}{G}\sum_{g=1}^{G} \tilde{R}_{i,g}, \tag{6}$$

where σ_i is the empirical standard deviation of $\{\tilde{R}_{i,g}\}_{g=1}^{G}$. In CRAFT, this normalized score is the proxy value $\Phi_{\pi_\theta}(s_i, \tau_g) = \tilde{A}_{i,g}$. Since fine-tuning is performed in a discrete action space, the counterfactual objective takes the exact expectation over candidate trajectories:

$$\mathcal{L}_{\text{cp}} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{g=1}^{G} \pi_\theta(\tau_g \mid s_i)\, \tilde{A}_{i,g}. \tag{7}$$

This objective instantiates the dense counterfactual proxy term in Equation (5), where $\mathcal{R}_{\text{cp}}(s_i; \pi_\theta) = \sum_{g=1}^{G} \pi_\theta(\tau_g \mid s_i)\tilde{A}_{i,g}$. It provides dense group-wise comparative supervision at each on-policy state by evaluating candidate trajectories in a counterfactual world designed to approximate the real closed-loop world. However, unrealistic traffic interactions and dynamics biases still introduce a discrepancy between the proxy advantage $\tilde{A}_{i,g}$ and the real advantage $A^{\text{real}}_{\pi_\theta}(s_i, \tau_g)$, which motivates the grounded residual correction.

3.3 Grounded Residual Correction

Rather than estimating the exact residual in Equation (3), CRAFT approximates the interaction-critical residual component using grounded feedback from traffic events that are not fully captured by the counterfactual proxy, such as collisions and traffic-rule violations. For a selected trajectory τ_t, the controller executes $a_t = \kappa(s_t, \tau_t)$, after which the event-driven corrective reward defined in Appendix D.4 is computed from the closed-loop transition and converted into bounded corrective advantages $\hat{A}^{\text{gr}}_t$, as detailed in Appendix D.5.

Let $\rho_t(\theta) = \pi_\theta(\tau_t \mid s_t)/\pi_{\theta_{\text{old}}}(\tau_t \mid s_t)$ denote the importance sampling ratio. We first form

$$\ell_t(\theta) = \min\Big(\rho_t(\theta)\hat{A}^{\text{gr}}_t,\ \text{clip}(\rho_t(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})\hat{A}^{\text{gr}}_t\Big), \tag{8}$$

and then apply the dual-clipping [9, 10] with constant $c > 1$ to form the grounded residual loss

$$\mathcal{L}_{\text{gr}} = -\mathbb{E}_{t\in\mathcal{B}}\Big[\max\big(\ell_t(\theta), c\hat{A}^{\text{gr}}_t\big)\mathbf{1}_{\hat{A}^{\text{gr}}_t<0} + \ell_t(\theta)\mathbf{1}_{\hat{A}^{\text{gr}}_t\geq 0}\Big]. \tag{9}$$

Proposition A.2 formalizes the stabilization of dual clipping, which preserves informative gradients from rare catastrophic rollouts while preventing extreme negative events from dominating the update.

To characterize the grounded residual correction under the decomposition in Equation (3), let $\widehat{\Delta}_{\pi_\theta}(s, \tau)$ be the residual approximation induced by grounded feedback, and define

$$\widehat{g}(\theta) = \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(\tau \mid s)\,\big(\Phi_{\pi_\theta}(s,\tau) + \widehat{\Delta}_{\pi_\theta}(s,\tau)\big)\Big]. \tag{10}$$

Theorem 3.3 (Gradient Bias under Residual Approximation)**.** *Let $g(\theta)$ denote the real gradient in Equation (3). If $\mathbb{E}\|\nabla_\theta \log \pi_\theta(\tau \mid s)\|_2^2 < \infty$ and $\mathbb{E}|\widehat{\Delta}_{\pi_\theta}(s,\tau) - \Delta_{\pi_\theta}(s,\tau)|^2 < \infty$, then*

$$\|\widehat{g}(\theta) - g(\theta)\|_2 \leq \Big(\mathbb{E}\|\nabla_\theta \log \pi_\theta(\tau \mid s)\|_2^2\Big)^{1/2}\Big(\mathbb{E}|\widehat{\Delta}_{\pi_\theta}(s,\tau) - \Delta_{\pi_\theta}(s,\tau)|^2\Big)^{1/2}. \tag{11}$$

If $\|\nabla_\theta \log \pi_\theta(\tau \mid s)\|_2 \leq G_\nabla$ almost surely, the first factor reduces to G_∇. Proof in Appendix A.5.

Theorem 3.3 establishes the role of grounded residual correction in Equation (5). Grounded feedback captures the dominant mismatch left by the dense counterfactual proxy, focusing correction on the interaction-sensitive part of the real closed-loop objective.

3.4 On-Policy Self-Distillation

The proxy-residual objective provides dense and grounded learning signals, but unconstrained policy updates may deviate from reliable pre-trained behavior. CRAFT limits this drift through on-policy self-distillation from an EMA teacher π_T,

$$\theta_T^{(k+1)} \leftarrow m\theta_T^{(k)} + (1-m)\theta^{(k)}, \qquad \theta_T^{(0)} = \theta_{\text{pre}}, \quad 0 \le m < 1. \tag{12}$$

The teacher is initialized from the pre-trained checkpoint, as a conservative anchor early in training, and subsequently tracks the smoothed online policy through EMA. For a visited state s, let $U(s,\tau) = \Phi_{\pi_\theta}(s,\tau) + \widehat{\Delta}_{\pi_\theta}(s,\tau)$ denote the proxy-residual advantage estimate. Proposition A.1 indicates that policy improvement constrained by reverse KL induces an exponential reweighting of the teacher distribution according to this advantage estimate. This provides a principled update rule that improves the policy under the proxy-residual objective while remaining close to teacher-supported behavior. CRAFT implements this rule with asymmetric KL penalties,

$$\mathcal{L}_{\text{dist}} = \mathbb{E}_{s\sim d^{\pi_\theta}_{\text{real}},\, \tau\sim\pi_\theta(\cdot|s)}\left[\log \frac{\pi_\theta(\tau \mid s)}{\pi_T(\tau \mid s)}\right], \quad \mathcal{L}_{\text{KL}} = \mathbb{E}_{s\sim d^{\pi_\theta}_{\text{real}},\, \tau\sim\pi_T(\cdot|s)}\left[\log \frac{\pi_T(\tau \mid s)}{\pi_\theta(\tau \mid s)}\right]. \tag{13}$$

The reverse KL term is the primary constraint, with $\beta_r > \beta_f$, and suppresses deviations from reliable pre-trained behavior. The weaker forward KL term preserves teacher-supported alternatives and mitigates collapse to a narrow subset of trajectories favored by the proxy-residual objective.

Overall, CRAFT combines the dense counterfactual proxy, grounded residual correction, and asymmetric self-distillation into the final objective, with Algorithm 1 detailing the full training procedure:

$$\mathcal{L}_{\text{all}} = \lambda_{\text{cp}}\mathcal{L}_{\text{cp}} + \lambda_{\text{gr}}\mathcal{L}_{\text{gr}} + \beta_r\mathcal{L}_{\text{dist}} + \beta_f\mathcal{L}_{\text{KL}}. \tag{14}$$

4 Experiments

4.1 Experimental Setup

Benchmark We conduct experiments on Bench2Drive [43], a CARLA-based benchmark [44] for closed-loop evaluation in autonomous driving. Our setup follows Bench2Drive as the primary protocol, with additional scenario configurations from LEAD [45] and Longest6 V2 [25]. We also apply several protocol-level refinements to the original scenarios and evaluation protocol to improve efficiency and reduce repeated-run variance. Details are provided in Appendix B.

Metrics We adopt the standard Bench2Drive metrics, including Driving Score (DS), Success Rate (SR), and Multi-Ability. DS combines Route Completion (RC) and Infraction Score (IS), where RC measures route progress and IS reflects safety and traffic-rule compliance. SR measures successful scenario completion, while Multi-Ability separately evaluates five advanced urban driving skills.

Driving Policies and Fine-tuning Baselines To demonstrate the generality of CRAFT, we select three representative driving policies from distinct paradigms.

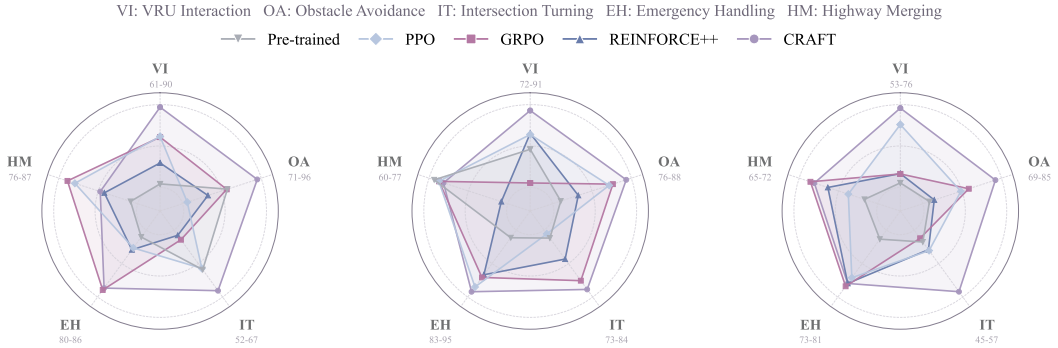
- **Hierarchical planning.** HiP-AD [46] uses multi-granularity planning queries together with deformable attention to couple perception and planning within a unified decoder.
- **Vision-language-action.** MindDrive [24] uses a VLA architecture with separate decision and action experts, translating high-level semantic decisions into low-level trajectory generation.
- **Vocabulary scoring.** SparseDriveV2 [22] uses sparse scene representation for end-to-end driving, and here we use its vocabulary variant to provide a dense candidate-trajectory space for RL. This property also makes it the default driving policy in our subsequent analysis experiments.

We compare CRAFT against the pre-trained policy and three RL fine-tuning baselines under the same setting and evaluation protocol. The baselines differ only in their training objective.

- PPO [23] employs a clipped surrogate objective together with a learned value baseline, making it the standard actor–critic baseline for stable on-policy fine-tuning.
- REINFORCE++ [47] removes the learned critic and uses return-based updates normalized at the batch level, providing a simple critic-free baseline for on-policy fine-tuning.
- GRPO [8] replaces a separate critic with relative rewards computed within output groups, making it the closest baseline to the group-based counterfactual update used in CRAFT.

Table 1: **Closed-loop performance on Bench2Drive.** † denotes the vocabulary-based variant of SparseDriveV2. Best results are highlighted in **bold**, and second-best results are underlined.

Driving Policy	Method	Closed-loop Metrics		Multi-Ability (%)					
		DS ↑	SR (%) ↑	Merging ↑	Overtaking ↑	Emergency Brake ↑	Give Way ↑	Traffic Sign ↑	Mean ↑
HiP-AD [46] *(Hierarchical Planning)*	Pre-trained	73.46	53.64	**46.30**	53.30	63.30	**50.00**	46.30	51.80
	REINFORCE++	74.82	55.00	36.20	**64.40**	71.70	**50.00**	44.20	53.30
	PPO	73.58	53.18	41.20	55.60	68.30	30.00	43.20	47.70
	GRPO	75.27	55.91	42.50	60.00	70.00	40.00	**48.40**	52.20
	CRAFT (Ours)	**75.79**	**56.36**	40.00	**64.40**	**73.30**	**50.00**	**48.40**	**55.20**
MindDrive [24] *(Vision Language Action)*	Pre-trained	62.60	40.45	26.30	51.10	50.00	40.00	32.60	40.00
	REINFORCE++	63.68	41.36	32.50	46.70	50.00	**50.00**	33.70	42.60
	PPO	64.61	42.73	30.00	48.90	55.00	**50.00**	35.80	43.90
	GRPO	65.12	43.18	27.50	**57.80**	53.30	**50.00**	34.70	44.70
	CRAFT (Ours)	**67.45**	**46.36**	**37.50**	51.10	**56.70**	**50.00**	**37.90**	**46.60**
SparseDriveV2† [22] *(Vocabulary Scoring)*	Pre-trained	66.43	45.45	**43.80**	40.00	56.70	**50.00**	32.60	44.60
	REINFORCE++	66.60	46.36	41.20	**53.30**	55.00	**50.00**	26.30	45.20
	PPO	67.44	49.09	**43.80**	**53.30**	61.70	**50.00**	33.70	48.50
	GRPO	68.40	48.18	41.20	48.90	61.70	**50.00**	36.80	47.70
	CRAFT (Ours)	**71.10**	**50.45**	40.00	**53.30**	**68.30**	**50.00**	**37.90**	**49.90**



(a) SparseDriveV2 (b) HiP-AD (c) MindDrive

Figure 3: **Fine-grained ability profiles.** Radar plots compare fine-tuning methods across driving policies. Each axis represents a scenario-specific capability, and higher values indicate better performance.

Implementation Details For all policies, we fine-tune only the planning-related modules while freezing pre-trained visual processing. All fine-tuning methods share a CaRL-style reward formulation based on progress and infractions [25]. Our RL infrastructure is built on RIFT [48] and supports asynchronous rollout collection and policy updates. Details appear in Appendices D.2 to D.5.

4.2 Main Closed-Loop Results

The main comparison reveals a consistent pattern across policy architectures. CRAFT improves closed-loop metrics without relying on any specific model architecture. This improvement remains evident even for HiP-AD, where the pre-trained policy is already strong, and is larger for MindDrive and SparseDriveV2, where the initial closed-loop performance indicates greater potential for fine-tuning. In Table 1, the same ordering holds across hierarchical planning, vision-language-action, and vocabulary-scoring, supporting the claim that CRAFT generalizes across diverse policy architectures.

The comparison also highlights the value of the full CRAFT objective. Even GRPO, the closest proxy-based baseline, underperforms CRAFT across policy architectures. PPO and REINFORCE++ improve specific driving abilities, but their gains are less consistent. This pattern further supports the proxy-residual design of CRAFT, suggesting that dense counterfactual supervision is most effective when refined by grounded residual feedback.

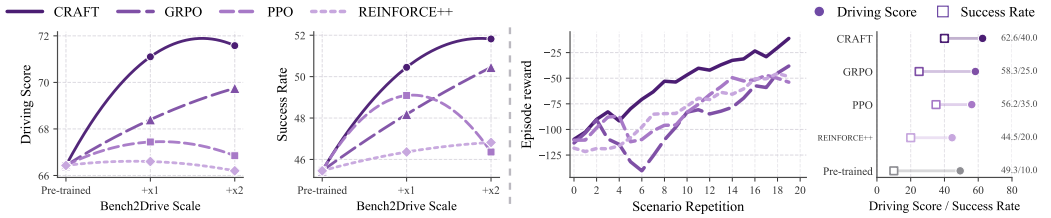
The ability profiles in Figure 3 provide a finer-grained view of the aggregate results. CRAFT leads on most abilities, with the largest gains in safety-critical behaviors such as VRU Interaction, Obstacle Avoidance, and Intersection Turning. These abilities depend more strongly on closed-loop interaction and are therefore less likely to be captured by a dense counterfactual proxy alone. This pattern further supports the proxy-residual design of CRAFT, suggesting that dense counterfactual supervision alone is insufficient for safety-critical events and that grounded residual correction is essential.

Table 2: **Ablation study of CRAFT.** SD: self-distillation; CP: counterfactual proxy; GR: grounded residual.

ID	SD	CP	GR	Closed-loop Metrics			
				DS ↑	RC ↑	IS ↑	SR ↑
1	✗	✗	✗	66.43	77.53	0.80	45.45
2	✓	✗	✗	66.49	76.65	0.81	46.36
3	✓	✓	✗	68.40	78.28	0.82	48.18
4	✓	✓	✓	**71.10**	**80.01**	**0.84**	**50.45**

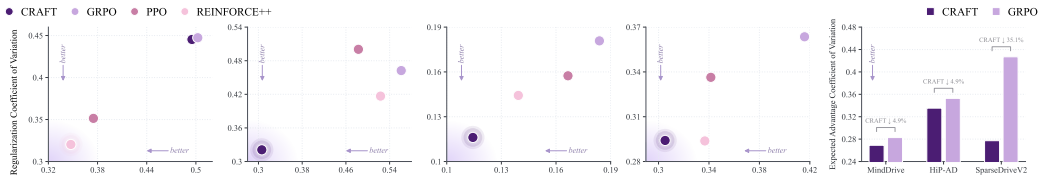
Table 3: **Generalization across datasets.** Gains over the pre-trained policy under each train-test split. ‡: Town12 subset of LEAD scenarios.

Train	Test	Closed-loop Metrics			
		Δ DS ↑	Δ RC ↑	Δ IS ↑	Δ SR ↑
Bench2Drive	Bench2Drive	**+4.67**	**+2.48**	**+0.04**	**+5.00**
Bench2Drive	Longest6 v2 [25]	+2.18	+1.12	+0.06	+0.00
LEAD‡	LEAD‡ [45]	**+2.14**	**+2.60**	**+0.01**	**+1.89**
LEAD‡	Bench2Drive	+0.58	-0.34	+0.01	+0.91



(a) Scaling Behavior Across Fine-tuning Methods (b) Training Dynamics and Final Performance

Figure 4: **Scaling behavior and reward dynamics.** (a) Closed-loop performance as the Bench2Drive scale increases. (b) Reward evolution during RL training on a challenging scenario, with final evaluation performance.



(a) MindDrive KL Stability (b) HiP-AD KL Stability (c) SparseDriveV2 KL Stability (d) Mean KL Stability (e) Expected Advantage Stability

Figure 5: **Training stability.** Panels (a)–(d) report the coefficients of variation for KL terms across fine-tuning methods, and panel (e) reports the coefficient of variation for the expected advantage across driving policies.

4.3 Ablation and Generalization Analysis

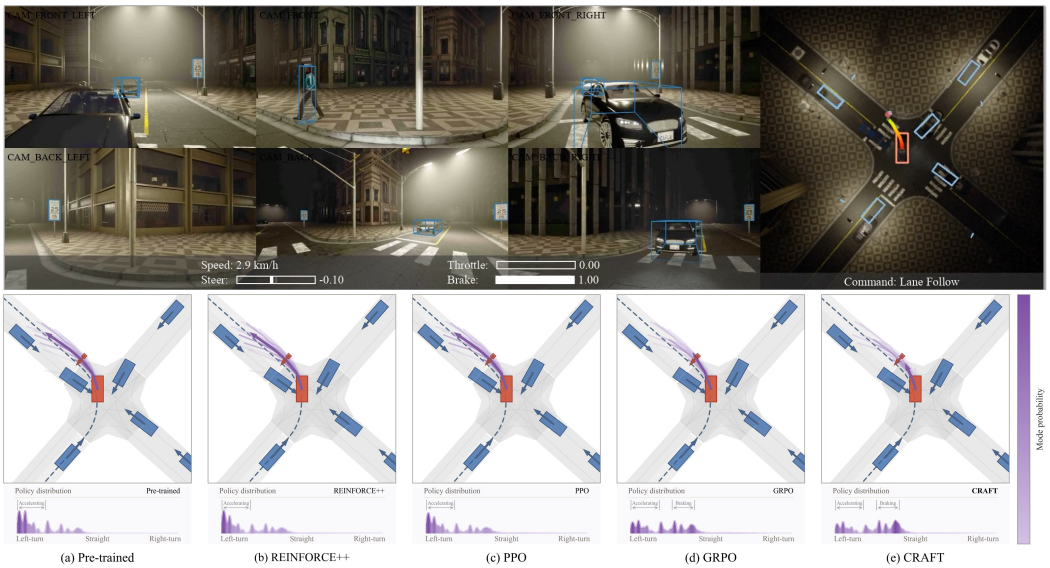
The ablation study in Table 2 clarifies each component of CRAFT. Self-distillation reinforces the dominant modes of the pre-trained policy, while the dense counterfactual proxy drives broad closed-loop gains. With grounded residual correction, the full CRAFT objective achieves the best overall performance, especially on the interaction-related metrics IS and SR. These results support the proxy-residual view, where the dense counterfactual proxy provides the primary optimization signal and grounded residual correction compensates for counterfactual bias through closed-loop interaction.

As shown in Table 3, CRAFT maintains positive generalization under scenario shift. In-domain fine-tuning on Bench2Drive delivers the largest improvement over the pre-trained policy, while cross-scenario transfer still yields consistent gains. The main limitation appears in Route Completion, where the limited improvement under transfer suggests a more conservative policy in unseen scenarios. Overall, CRAFT generalizes beyond the training split, although the magnitude of improvement remains partly dependent on the interaction distribution used for fine-tuning.

4.4 Scaling and Training Dynamics

Figure 4 reveals a clear scaling pattern. CRAFT is the only method whose performance improves consistently with more data. PPO and REINFORCE++, which rely only on real interaction, become less stable at larger scales and show saturating gains, suggesting inefficient learning from sparse closed-loop feedback. GRPO, which depends only on proxy supervision, delivers limited improvement and even deteriorates at larger scales, indicating that proxy bias may become the main bottleneck in later stages of training. These trends demonstrate that efficient scaling requires dense proxy supervision, while sustained gains depend on grounded residual correction.

The results in Figure 5 further confirm the stability of CRAFT. In terms of KL stability, CRAFT shows larger gains for policies with broader action spaces, such as HiP-AD and SparseDriveV2, while



(a) Pre-trained (b) REINFORCE++ (c) PPO (d) GRPO (e) CRAFT

Figure 6: **Policy distribution shift in a pedestrian-crossing scenario.** Top: a closed-loop scene snapshot with SparseDriveV2 as the ego vehicle. Bottom: the policy distributions over the trajectory vocabulary for the pre-trained and fine-tuned policies. CRAFT shifts probability mass toward braking-compatible modes.

the advantage is smaller for MindDrive, which has a more constrained action space. This pattern suggests that a broader action space improves counterfactual proxy stability. The expected-advantage results support the same conclusion from the grounded side. Compared with GRPO, CRAFT reduces expected-advantage variation across policy architectures. Although grounded residual correction is driven by sparse feedback, dual clipping stabilizes training. Together, these results suggest that the stability of CRAFT arises from both components, with broader action spaces improving proxy stability and dual-clipped grounded residual correction further reducing update variance.

4.5 Qualitative Scenario Analysis

Figure 6 illustrates how fine-tuning reshapes the policy distribution in a safety-critical state. The pre-trained SparseDriveV2 assigns most probability mass to a left-turn accelerating mode, which remains largely unchanged after PPO and REINFORCE++ fine-tuning. This similarity suggests that sparse closed-loop updates do not efficiently shift the dominant mode away from unsafe behavior. GRPO shifts part of the distribution toward a braking-compatible mode, but this mode remains secondary, indicating that the counterfactual proxy alone gives a biased estimate of the safer closed-loop response. CRAFT induces the clearest shift, making the braking-compatible mode dominant while retaining the original peak of the pre-trained policy at lower probability. These results highlight the advantage of proxy-residual fine-tuning for safety-critical events by shifting probability toward interaction-grounded corrective behavior while preserving the pre-trained policy manifold.

4.6 Further Analysis of Stability and Interaction

Role of self-distillation Table 4 isolates self-distillation in the full proxy-residual objective under identical post-training settings. Without self-distillation, DS drops from 66.43 to 51.42 and SR from 45.45% to 20.91%, falling below pre-trained performance. A frozen teacher restores performance above the initialization, while the EMA teacher achieves the strongest results. Together with Table 2, this comparison supports self-distillation as a stabilizer of the joint proxy-residual update. The EMA teacher further balances closed-loop adaptation with retention of reliable pre-trained behavior.

Scope of grounded feedback Both signals originate from the same on-policy state. The counterfactual proxy evaluates all G candidates, while grounded feedback is available for the executed trajectory. Grounded supervision is therefore sparse across candidate trajectories. All collected transitions enter the grounded residual loss, and discounted corrective returns propagate event penalties to preceding transitions, allowing earlier decisions to receive feedback from subsequent closed-loop outcomes.

Table 4: **Teacher ablation.** All runs use SparseDriveV2 with 512 modes. SD denotes self-distillation.

Configuration	Closed-loop metrics	
	DS ↑	SR (%) ↑
Pre-trained	66.43	45.45
CRAFT w/o SD	51.42	20.91
CRAFT (Frozen)	68.04	47.27
CRAFT (EMA)	**71.10**	**50.45**

Table 5: **Action-support sensitivity.** Frozen and EMA denote CRAFT teacher variants. Gains are relative to pre-training, with SR gains in percentage points.

Configuration	256 modes		512 modes	
	DS ↑	SR (%) ↑	DS ↑	SR (%) ↑
Pre-trained	64.62	43.18	66.43	45.45
CRAFT (Frozen)	65.63	43.18	68.04	47.27
CRAFT (EMA)	**68.16**	**49.09**	**71.10**	**50.45**
Gain (EMA)	+3.54	+5.91	+4.67	+5.00

Table 6: **Merging success by interaction initiator.** Success rates (%) follow the Perfect criterion; Change is measured in percentage points before rounding. Each policy has 15 ego-initiated and 10 other-initiated routes (five per scenario), excluding mixed interactions. The *All policies* row combines results from the three policies (45 and 30 evaluations, respectively).

Driving policy	Ego-initiated			Other-initiated		
	Pre-trained (%) ↑	CRAFT (%) ↑	Change ↑	Pre-trained (%) ↑	CRAFT (%) ↑	Change ↑
HiP-AD	33.3	40.0	**+6.7**	70.0	60.0	**−10.0**
MindDrive	13.3	40.0	**+26.7**	70.0	60.0	**−10.0**
SparseDriveV2	33.3	53.3	**+20.0**	80.0	50.0	**−30.0**
All policies	26.7	44.4	**+17.8**	73.3	56.7	**−16.7**

Sensitivity to pre-trained action support Table 5 compares 256- and 512-mode SparseDriveV2 policies under identical post-training settings. With the EMA teacher, CRAFT improves DS by 3.54 and 4.67, and SR by 5.91 and 5.00 percentage points, respectively. The EMA teacher yields larger gains than the frozen teacher for both initializations, with the 512-mode policy achieving the highest final DS and SR. CRAFT adapts the decision module over existing candidate trajectories, whose coverage determines the behavioral alternatives available for closed-loop adaptation.

Interaction-dependent merging behavior Table 6 shows that all three policies improve in ego-initiated merging but degrade when other agents enter the ego corridor. Across the three policies, SR increases by 17.8 percentage points for ego-initiated merges and decreases by 16.7 points for externally initiated merges. This asymmetry suggests that counterfactual interaction bias remains a limitation. Ego-initiated merging depends mainly on selecting an entry gap, while externally initiated merging requires accurate estimates of another vehicle's motion and response. Grounded residual correction uses the executed outcome, while responses to unexecuted alternatives remain estimated. More accurate modeling of interactive agents, including their responses to candidate ego trajectories, is therefore a natural direction for improving performance in externally initiated merges.

5 Conclusions

This paper presents CRAFT, a proxy-residual framework for closed-loop post-training of supervised driving policies. CRAFT combines a dense counterfactual proxy on on-policy visited states, grounded residual correction from executed rollouts, and self-distillation to preserve reliable pre-trained behavior during fine-tuning. We support this design with an exact gradient decomposition on the real closed-loop state distribution, together with variance, bias, and stability analyses. On Bench2Drive, CRAFT achieves the strongest closed-loop gains across hierarchical-planning, vision-language-action, and vocabulary-scoring architectures, while ablation and scaling results confirm the complementary roles of counterfactual proxy learning, grounded residual correction, and self-distillation.

Limitation. CRAFT currently fine-tunes planning modules in representative driving policies, without exploring more specialized designs for different policy architectures. While this preserves broad applicability, it may not fully exploit the structure of each architecture. Exploring architecture-aware adaptation strategies is a natural direction for future work on proxy-residual fine-tuning.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 52572497 and 52221005.

References

[1] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 17, 18

[2] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023. 1, 18

[3] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8795–8801. IEEE, 2025. 1, 18

[4] Zhenghao Peng, Wenhao Ding, Yurong You, Yuxiao Chen, Wenjie Luo, Thomas Tian, Yulong Cao, Apoorva Sharma, Danfei Xu, Boris Ivanovic, et al. Counterfactual vla: Self-reflective vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2512.24426*, 2025. 1

[5] Chenxu Dang, Sining Ang, Yongkang Li, Haochen Tian, Jie Wang, Guang Li, Hangjun Ye, Jie Ma, Long Chen, and Yan Wang. Drivefine: Refining-augmented masked diffusion vla for precise and robust driving. *arXiv preprint arXiv:2602.14577*, 2026. 1

[6] Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, et al. Drivelaw: Unifying planning and video generation in a latent driving world. *arXiv preprint arXiv:2512.23421*, 2025. 1

[7] 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. *arXiv preprint arXiv:2510.12796*, 2025. 1

[8] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 1, 3, 6

[9] Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, et al. Mastering complex control in moba games with deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6672–6679, 2020. 2, 5

[10] Yiming Gao, Bei Shi, Xueying Du, Liang Wang, Guangwei Chen, Zhenjie Lian, Fuhao Qiu, Guoan Han, Weixuan Wang, Deheng Ye, et al. Learning diverse policies in moba games via macro-goals. *Advances in Neural Information Processing Systems*, 34:16171–16182, 2021. 2, 5

[11] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12037–12047, 2025. 3

[12] Lin Liu, Caiyan Jia, Guanyi Yu, Ziying Song, JunQiao Li, Feiyang Jia, Peiliang Wu, Xiaoshuai Hao, and Yadan Luo. Guideflow: Constraint-guided flow matching for planning in end-to-end autonomous driving. *arXiv preprint arXiv:2511.18729*, 2025. 3

[13] Zhiyu Zheng, Shaoyu Chen, Haoran Yin, Xinbang Zhang, Jialv Zou, Xinggang Wang, Qian Zhang, and Lefei Zhang. Resad: Normalized residual trajectory modeling for end-to-end autonomous driving. *arXiv preprint arXiv:2510.08562*, 2025. 3

[14] Yinan Zheng, Tianyi Tan, Bin Huang, Enguang Liu, Ruiming Liang, Jianlin Zhang, Jianwei Cui, Guang Chen, Kun Ma, Hangjun Ye, et al. Unleashing the potential of diffusion models for end-to-end autonomous driving. *arXiv preprint arXiv:2602.22801*, 2026. 3

[15] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24823–24834, 2025. 3

[16] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11993–12003, 2025. 3

[17] Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025. 3

[18] Xinyang Wang, Qian Liu, Wenjie Ding, Zhao Yang, Wei Li, Chang Liu, Bailin Li, Kun Zhan, Xianpeng Lang, and Wei Chen. Unifying language-action understanding and generation for autonomous driving. *arXiv preprint arXiv:2603.01441*, 2026. 3

[19] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024. 3

[20] Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M Alvarez, and Zuxuan Wu. Drivesuprim: Towards precise trajectory selection for end-to-end planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11910–11918, 2026. 3

[21] Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025. 3

[22] Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. Sparsedrivev2: Scoring is all you need for end-to-end autonomous driving. *arXiv preprint arXiv:2603.29163*, 2026. 3, 6, 7, 20

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3, 6

[24] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Hongwei Xie, Bing Wang, Guang Chen, Dingkang Liang, and Xiang Bai. Minddrive: A vision-language-action model for autonomous driving via online reinforcement learning. *arXiv preprint arXiv:2512.13636*, 2025. 3, 6, 7, 19

[25] Bernhard Jaeger, Daniel Dauner, Jens Beißwenger, Simon Gerstenecker, Kashyap Chitta, and Andreas Geiger. Carl: Learning scalable planning policies with simple rewards. In *9th Annual Conference on Robot Learning*, 2025. 3, 6, 7, 8, 21

[26] 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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 3

[27] Hao Gao, Shaoyu Chen, Yifan Zhu, Yuehao Song, Wenyu Liu, Qian Zhang, and Xinggang Wang. Rad-2: Scaling reinforcement learning in a generator-discriminator framework. *arXiv preprint arXiv:2604.15308*, 2026. 3

[28] Chaojun Ni, Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Wenkang Qin, Xinze Chen, Guanghong Jia, Guan Huang, and Wenjun Mei. Recondreamer-rl: Enhancing reinforcement learning via diffusion-based scene reconstruction. *arXiv preprint arXiv:2508.08170*, 2025. 3

[29] Zilin Huang, Zihao Sheng, Zhengyang Wan, Yansong Qu, Junwei You, Sicong Jiang, and Sikai Chen. Drivevlm-rl: Neuroscience-inspired reinforcement learning with vision-language models for safe and deployable autonomous driving. *arXiv preprint arXiv:2603.18315*, 2026. 3

[30] 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*, pages 142–158. Springer, 2024. 3

[31] Zhenjie Yang, Xiaosong Jia, Qifeng Li, Xue Yang, Maoqing Yao, and Junchi Yan. Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 3

[32] Haochen Liu, Tianyu Li, Haohan Yang, Li Chen, Caojun Wang, Ke Guo, Haochen Tian, Hongchen Li, Hongyang Li, and Chen Lv. Reinforced refinement with self-aware expansion for end-to-end autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. 3

[33] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023. 3

[34] Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025. 3

[35] 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. 3

[36] Rajeev Yasarla, Deepti Hegde, Shizhong Han, Hsin-Pai Cheng, Yunxiao Shi, Meysam Sadeghigooghari, Shweta Mahajan, Apratim Bhattacharyya, Litian Liu, Risheek Garrepalli, et al. Generative scenario rollouts for end-to-end autonomous driving. *arXiv preprint arXiv:2601.11475*, 2026. 3

[37] Shuyao Shang, Bing Zhan, Yunfei Yan, Yuqi Wang, Yingyan Li, Yasong An, Xiaoman Wang, Jierui Liu, Lu Hou, Lue Fan, et al. Dynvla: Learning world dynamics for action reasoning in autonomous driving. *arXiv preprint arXiv:2603.11041*, 2026. 3

[38] Ruiming Liang, Yinan Zheng, Kexin Zheng, Tianyi Tan, Jianxiong Li, Liyuan Mao, Zhihao Wang, Guang Chen, Hangjun Ye, Jingjing Liu, et al. Dichotomous diffusion policy optimization. *arXiv preprint arXiv:2601.00898*, 2025. 3

[39] Canyu Chen, Yuguang Yang, Zhewen Tan, Yizhi Wang, Ruiyi Zhan, Haiyan Liu, Xuanyao Mao, Jason Bao, Xinyue Tang, Linlin Yang, et al. Devil is in narrow policy: Unleashing exploration in driving vla models. *arXiv preprint arXiv:2603.06049*, 2026. 3

[40] 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. 3

[41] Haohong Lin, Yunzhi Zhang, Wenhao Ding, Jiajun Wu, and Ding Zhao. Model-based policy adaptation for closed-loop end-to-end autonomous driving. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 3

[42] Tianyi Yan, Tao Tang, Xingtai Gui, Yongkang Li, Jiasen Zhesng, 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. *arXiv preprint arXiv:2511.20325*, 2025. 3

[43] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *arXiv preprint arXiv:2406.03877*, 2024. 6

[44] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 6

[45] Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. Lead: Minimizing learner-expert asymmetry in end-to-end driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. 6, 8

[46] Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. Hip-ad: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25605–25615, 2025. 6, 7, 19

[47] Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*, 2025. 6

[48] 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

[49] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pages 256–274. Springer, 2024. 17, 18

[50] Simon Gerstenecker, Andreas Geiger, and Katrin Renz. Plant 2.0: Exposing biases and structural flaws in closed-loop driving. *arXiv preprint arXiv:2511.07292*, 2025. 17, 18

Appendix

A Proofs

This appendix provides the proofs for the formal statements in Section 3. Unless noted otherwise, expectations are taken over $s \sim d_{\text{real}}^{\pi_\theta}$ and candidate trajectory $\tau \sim \pi_\theta(\cdot \mid s)$.

A.1 Exact Decomposition

Proof. By definition of the residual mismatch,

$$A_{\pi_\theta}^{\text{real}}(s,\tau) = \Phi_{\pi_\theta}(s,\tau) + \Delta_{\pi_\theta}(s,\tau). \tag{15}$$

Substituting this identity into the real policy gradient in Equation (2) gives

$$\nabla_\theta J_{\text{real}}(\pi_\theta) = \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(\tau \mid s)\,\big(\Phi_{\pi_\theta}(s,\tau) + \Delta_{\pi_\theta}(s,\tau)\big)\Big] \tag{16}$$

$$= \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(\tau \mid s)\,\Phi_{\pi_\theta}(s,\tau)\Big] + \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(\tau \mid s)\,\Delta_{\pi_\theta}(s,\tau)\Big], \tag{17}$$

which proves the claim. □

A.2 KL-Proximal Policy Improvement

Proposition A.1 (KL-Proximal Policy Improvement)**.** *For a finite or countable discrete trajectory space, fix state s and a strictly positive teacher distribution $q(\tau) = \pi_T(\tau \mid s)$. For any fixed bounded score $U(s,\tau)$ and temperature $\eta > 0$, the non-parametric solution to*

$$\max_{p \in \Delta(\mathcal{T})} \; \mathbb{E}_{\tau\sim p}[U(s,\tau)] - \eta D_{\text{KL}}(p\|q) \tag{18}$$

is

$$p_\eta^\star(\tau \mid s) = \frac{q(\tau)\exp(U(s,\tau)/\eta)}{\sum_{\tau' \in \mathcal{T}} q(\tau')\exp(U(s,\tau')/\eta)}. \tag{19}$$

Proof. Fix s and write $U(\tau) = U(s,\tau)$. The Lagrangian for the optimization problem in Proposition A.1, with multiplier λ for the normalization constraint $\sum_\tau p(\tau) = 1$, is

$$\mathcal{L}(p,\lambda) = \sum_\tau p(\tau)U(\tau) - \eta \sum_\tau p(\tau)\log\frac{p(\tau)}{q(\tau)} + \lambda\Big(\sum_\tau p(\tau) - 1\Big). \tag{20}$$

For any candidate trajectory with $p(\tau) > 0$, stationarity gives

$$U(\tau) - \eta\Big(\log\frac{p(\tau)}{q(\tau)} + 1\Big) + \lambda = 0, \tag{21}$$

and therefore

$$p(\tau) = q(\tau)\exp\left(\frac{U(\tau) + \lambda - \eta}{\eta}\right). \tag{22}$$

The normalizing constraint determines the common factor,

$$\exp\left(\frac{\lambda - \eta}{\eta}\right) = \left(\sum_{\tau'} q(\tau')\exp(U(\tau')/\eta)\right)^{-1}. \tag{23}$$

Substituting this factor yields Equation (19). Since the objective is strictly concave in p when $\eta > 0$ and q has full support, this stationary point is the unique maximizer. □

A.3 Variance Reduction

Proof. Fix s and suppress the dependence on (s, τ) for readability. Since $R_\alpha = Y - \alpha C$,

$$\begin{aligned} \mathrm{Var}[R_\alpha] &= \mathrm{Var}[Y - \alpha C] && (24)\\ &= \mathrm{Var}[Y] + \mathrm{Var}[\alpha C] - 2\mathrm{Cov}[Y, \alpha C] && (25)\\ &= \mathrm{Var}[Y] + \alpha^2 \mathrm{Var}[C] - 2\alpha \mathrm{Cov}[Y, C], && (26) \end{aligned}$$

which proves the variance identity. Differentiating with respect to α gives

$$\frac{\partial}{\partial \alpha} \mathrm{Var}[R_\alpha] = 2\alpha \mathrm{Var}[C] - 2\mathrm{Cov}[Y, C]. \tag{27}$$

Because $\mathrm{Var}[C] > 0$, the unique stationary point is $\alpha^\star = \mathrm{Cov}[Y, C]/\mathrm{Var}[C]$, and the second derivative is $2\mathrm{Var}[C] > 0$, so $\alpha^\star$ is the global minimizer.

For $\alpha = 1$,

$$\mathrm{Var}[Y - C] - \mathrm{Var}[Y] = \mathrm{Var}[C] - 2\mathrm{Cov}[Y, C]. \tag{28}$$

Thus $\mathrm{Cov}[Y, C] > \frac{1}{2}\mathrm{Var}[C]$ implies $\mathrm{Var}[Y - C] < \mathrm{Var}[Y]$. If the condition holds for $d^{\pi_\theta}_{\text{real}}$-almost every s, taking expectation over s preserves the strict inequality for the expected conditional variances. □

A.4 Dual Clipping

Proposition A.2 (Bounded Dual-Clipped Surrogate)**.** *Suppose $|\hat{A}^{\text{gr}}_t| \le A_{\max}$ for all t after clipping, $0 < \epsilon_{\text{clip}} < 1$, $c > 1$, and $\rho_t(\theta) > 0$. With $\ell_t(\theta)$ defined in Equation (8), define*

$$u_t(\theta) = \max(\ell_t(\theta), c\hat{A}^{\text{gr}}_t)\mathbf{1}_{\hat{A}^{\text{gr}}_t < 0} + \ell_t(\theta)\mathbf{1}_{\hat{A}^{\text{gr}}_t \ge 0}. \tag{29}$$

Then, for every t,

$$-cA_{\max} \le u_t(\theta) \le (1 + \epsilon_{\text{clip}})A_{\max}. \tag{30}$$

Proof. If $\hat{A}^{\text{gr}}_t \ge 0$, then $u_t(\theta) = \ell_t(\theta)$. Since $\mathrm{clip}(\rho_t(\theta), 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}}) \le 1 + \epsilon_{\text{clip}}$, the clipped surrogate satisfies

$$\ell_t(\theta) \le (1 + \epsilon_{\text{clip}})\hat{A}^{\text{gr}}_t \le (1 + \epsilon_{\text{clip}})A_{\max}. \tag{31}$$

Moreover $\ell_t(\theta) \ge 0$ in this case, and therefore $-cA_{\max} \le u_t(\theta) \le (1 + \epsilon_{\text{clip}})A_{\max}$.

If $\hat{A}^{\text{gr}}_t < 0$, then $u_t(\theta) = \max(\ell_t(\theta), c\hat{A}^{\text{gr}}_t)$. Since $c > 1$ and $\hat{A}^{\text{gr}}_t \ge -A_{\max}$, we have $c\hat{A}^{\text{gr}}_t \ge -cA_{\max}$, so

$$u_t(\theta) \ge c\hat{A}^{\text{gr}}_t \ge -cA_{\max}. \tag{32}$$

Both $\ell_t(\theta)$ and $c\hat{A}^{\text{gr}}_t$ are non-positive when $\hat{A}^{\text{gr}}_t < 0$, hence $u_t(\theta) \le 0 \le (1+\epsilon_{\text{clip}})A_{\max}$. Combining the two cases proves the bound. □

A.5 Residual Approximation Bias

Proof. From Equation (3), the real gradient can be written as

$$g(\theta) = \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(\tau \mid s)\left(\Phi_{\pi_\theta}(s, \tau) + \Delta_{\pi_\theta}(s, \tau)\right)\Big]. \tag{33}$$

Subtracting this expression from the surrogate gradient in Equation (10) yields

$$\widehat{g}(\theta) - g(\theta) = \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(\tau \mid s)\left(\widehat{\Delta}_{\pi_\theta}(s, \tau) - \Delta_{\pi_\theta}(s, \tau)\right)\Big]. \tag{34}$$

Let $X = \nabla_\theta \log \pi_\theta(\tau \mid s)$ and $e = \widehat{\Delta}_{\pi_\theta}(s, \tau) - \Delta_{\pi_\theta}(s, \tau)$. Jensen's inequality gives

$$\|\widehat{g}(\theta) - g(\theta)\|_2 = \|\mathbb{E}[Xe]\|_2 \le \mathbb{E}[\|X\|_2|e|]. \tag{35}$$

Applying Cauchy–Schwarz to the scalar random variables $\|X\|_2$ and $|e|$ gives

$$\mathbb{E}[\|X\|_2|e|] \le \left(\mathbb{E}\|X\|_2^2\right)^{1/2}\left(\mathbb{E}|e|^2\right)^{1/2}, \tag{36}$$

which proves Equation (11). If $\|X\|_2 \le G_\nabla$ almost surely, then $\left(\mathbb{E}\|X\|_2^2\right)^{1/2} \le G_\nabla$, yielding the bounded-gradient version. □

B Bench2Drive Protocol

B.1 Motivation for Adjustments

Closed-loop training imposes fundamentally different requirements on a benchmark than closed-loop evaluation. Beyond accurate assessment, training requires clear and informative failure signals, as well as visually and physically consistent environment dynamics across policy updates to avoid non-stationarity, together with high rollout efficiency. In particular, rollouts should terminate immediately upon meaningful failure events to provide concise supervision, and unnecessary criteria overhead should be minimized, as rollout time directly dictates training efficiency.

We therefore adopt a patched protocol, Bench2Drive*, which preserves the route set, ego-policy interface, and core metrics, while mitigating scenario artifacts that introduce non-policy noise into both training and evaluation.

B.2 Protocol-Level Adjustments

We apply six local adjustments to the official protocol.

- **Actor cleanup**. The official scenario manager can remove background or scenario-related actors at runtime, causing vehicles to disappear from the visual scene. We now clean up only background vehicles that block normal ego progress. Other nonblocking vehicles are kept in autopilot mode, and actors that would be removed immediately after spawning are skipped before they are spawned.
- **Timeout handling**. During both data collection and evaluation, we lower the upper bounds for scenario and route timeouts. This prevents a deadlocked route from consuming long simulator time after the policy has already produced the relevant failure signal.
- **BicycleFlow spawning**. We fix the BicycleFlow scenario so that cyclists are spawned only within the range specified by the scenario definition. The resulting flow follows normal behavior and avoids abnormal cyclist motion or collisions that are unrelated to the ego policy.
- **Route 3749 actor flow**. For route id 3749, we set the actor-flow speed to 12 m/s. This avoids an overly aggressive flow for the turning geometry, where the original setting could produce an invalid scene before the ego policy makes a meaningful decision.
- **Collision termination**. In both data collection and evaluation, a collision terminates the scene immediately. The original protocol may continue simulation after a collision; in BicycleFlow, for example, a cyclist can keep moving and generate repeated contacts from the same physical event. Early termination makes the failure state unambiguous and reduces noise in Driving Score. It may slightly lower Driving Score by removing post-collision driving segments, but it does not change Success Rate. We also ignore collision events with relative speed below 0.1 m/s to avoid duplicate contact records.
- **Town07 stop sign**. We ignore the invisible stop sign located at $(0, 0, 0)$ in Town07. This removes spurious stop-sign violations caused by a map artifact rather than by policy behavior.

B.3 Sanity Check Against Official Bench2Drive

The protocol adjustments aim to produce cleaner RL rollouts without altering the benchmark's qualitative semantics. We therefore compare the official protocol with Bench2Drive* across five representative policies.

- PDM-Lite [49] is a rule-based CARLA planner. It provides a nonlearned reference point whose behavior is driven by hand-designed planning logic rather than end-to-end perception and control.
- PlanT v2 [50] is a privileged planner with access to structured scene information. We include it to test whether the patched protocol remains consistent for high-performing planners that operate outside the sensor-only setting.
- UniAD [1] is a planning-oriented end-to-end driving system. We use it as a representative learned policy with integrated perception, prediction, and planning modules.

Table 7: **Evaluation comparison between official Bench2Drive and Bench2Drive*.** Bench2Drive* denotes the patched protocol used for cleaner and more efficient closed-loop training and evaluation. For Bench2Drive*, results are reported as mean ± standard deviation over three random seeds.

Driving Policy	Benchmark	Closed-loop Metrics		Multi-Ability (%)					
		DS ↑	SR (%) ↑	Merging ↑	Overtaking ↑	Emergency Brake ↑	Give Way ↑	Traffic Sign ↑	Mean ↑
Rule-based Planner									
PDM-Lite [49]	Bench2Drive	97.02	92.27	-	-	-	-	-	-
	Bench2Drive*	94.62 ± 0.59	88.48 ± 0.69	86.30 ± 0.00	86.67 ± 2.25	98.87 ± 0.98	73.33 ± 5.77	84.20 ± 2.10	85.83 ± 1.25
Privileged Planner									
PlanTv2 [50]	Bench2Drive	92.40 ± 1.70	83.80 ± 3.30	-	-	-	-	-	-
	Bench2Drive*	84.61 ± 0.74	76.06 ± 0.69	84.20 ± 0.69	56.30 ± 2.60	88.87 ± 0.98	16.67 ± 5.77	81.80 ± 1.21	65.53 ± 1.35
End-to-End method									
UniAD [1]	Bench2Drive	45.81	16.36	-	-	-	-	-	-
	Bench2Drive*	45.74 ± 0.60	20.00 ± 0.79	22.90 ± 0.69	26.67 ± 6.65	15.00 ± 0.00	30.00 ± 10.00	4.90 ± 1.21	19.90 ± 1.01
VAD [2]	Bench2Drive	42.35	15.00	-	-	-	-	-	-
	Bench2Drive*	44.25 ± 0.67	16.06 ± 0.95	12.13 ± 1.44	5.93 ± 3.42	30.53 ± 2.54	30.00 ± 10.00	2.47 ± 0.64	16.23 ± 1.38
SparseDrive [3]	Bench2Drive	44.54	16.71	-	-	-	-	-	-
	Bench2Drive*	44.21 ± 0.43	19.55 ± 0.45	14.63 ± 1.44	19.30 ± 6.41	28.33 ± 2.89	36.67 ± 5.77	3.53 ± 1.63	20.47 ± 0.95

- VAD [2] is an end-to-end method based on vectorized scene representations. It offers a learned baseline with a different intermediate representation from UniAD while staying in the same sensor-driven evaluation regime.
- SparseDrive [3] is an end-to-end method built around sparse scene representations. It is included because its closed-loop scores are close to the regime studied in this paper and because CRAFT is evaluated on SparseDrive-style policies.

Together, these policies cover rule-based, privileged, and end-to-end settings, allowing us to check whether the patched protocol changes efficiency and rollout cleanliness without altering the broad ordering of representative methods.

Table 7 shows the expected pattern. The adjusted protocol is slightly stricter for strong planners, mainly because terminal events are handled more directly and avoidable post-failure rollout time is removed. For end-to-end methods, the ranking remains broadly stable. UniAD remains ahead of VAD and SparseDrive. Moreover, Bench2Drive* produces substantially lower standard deviations than the official protocol, suggesting a more stable evaluation of closed-loop performance. This lower variance supports single-seed evaluation in the main experiments as a practical and cost-effective protocol for comparing relative method improvements under identical evaluation conditions. In practice, the shorter timeout and cleaner termination rules yield over 2× higher closed-loop training throughput and over 3× higher evaluation throughput. These efficiency gains are critical for reinforcement fine-tuning, where closed-loop rollout time is the primary bottleneck.

C Algorithm Framework

Algorithm 1 gives the training procedure that instantiates the objective in Equation (14). The loop follows the separation used throughout the method section. At each closed-loop state, the policy scores a finite set of candidate trajectories, and only the selected trajectory is passed to the controller to produce the action executed in the real environment. The counterfactual world later evaluates the stored candidate set from the same visited state, where counterfactual returns provide group-normalized counterfactual advantages without executing alternative candidates in CARLA. In parallel, the corrective reward is computed from the executed controller action and the resulting next state, then discounted, scaled, and clipped into the corrective advantage. The update combines the counterfactual advantage and corrective advantage with EMA teacher regularization, aligning the algorithm with the proxy-residual formulation in Section 3 and the implementation details reported below.

D Implementation Details

This section records the implementation choices used in the experiments in Section 4. The main paper keeps the method description abstract; here we specify which policy parameters are updated, how candidate futures are simulated, and how the reward and optimization terms are instantiated.

Algorithm 1 Counterfactual-to-Interactive Reinforcement Fine-Tuning (CRAFT)

Require: Pre-trained driving policy $\pi_{\theta_{\text{pre}}}$, controller κ, closed-loop environment $\mathcal{M}_{\text{real}}$, counterfactual world $\mathcal{E}_{\text{cp}}$, rollout buffer size B, candidate group size G, weights $\lambda_{\text{cp}}, \lambda_{\text{gr}}, \beta_r, \beta_f$, EMA momentum m

Ensure: Fine-tuned policy π_θ

Initialize $\theta \leftarrow \theta_{\text{pre}}$ and $\theta_T \leftarrow \theta_{\text{pre}}$
for training iteration $k = 1, 2, \ldots$ **do**
  Set $\theta_{\text{old}} \leftarrow \theta$ and clear $\mathcal{B}$
  while $|\mathcal{B}| < B$ **do**
    Observe s_t and candidates $\{\tau_{t,g}\}_{g=1}^{G}$
    Sample $\tau_t \sim \pi_{\theta_{\text{old}}}(\cdot \mid s_t)$ and execute $a_t = \kappa(s_t, \tau_t)$ ▷ closed-loop execution
    Observe s_{t+1} and $r_t^{\text{gr}} = r^{\text{gr}}(s_t, a_t, s_{t+1})$
    Store $(s_t, \{\tau_{t,g}\}_{g=1}^{G}, \tau_t, \log \pi_{\theta_{\text{old}}}(\tau_t \mid s_t), r_t^{\text{gr}})$ in $\mathcal{B}$
  end while
  for each visited state s_i in $\mathcal{B}$ **do**
    Run $\mathcal{E}_{\text{cp}}$ on $\{\tau_{i,g}\}_{g=1}^{G}$ to obtain counterfactual returns $\tilde{R}_{i,g}$ ▷ counterfactual reward
    Mask invalid candidates and normalize valid returns to $\tilde{A}_{i,g}$ as in Equation (6)
  end for
  Compute $\mathcal{L}_{\text{cp}}$ using Equation (7)
  Discount corrective rewards with γ_c, then scale and clip to $\hat{A}_t^{\text{gr}}$ ▷ corrective advantage
  Form $\rho_t(\theta) = \pi_\theta(\tau_t \mid s_t)/\pi_{\theta_{\text{old}}}(\tau_t \mid s_t)$ and compute $\mathcal{L}_{\text{gr}}$ using Equations (8) and (9)
  Compute $\mathcal{L}_{\text{dist}}$ and $\mathcal{L}_{\text{KL}}$ using Equation (13) ▷ on-policy self-distillation
  Update trainable policy parameters by minimizing $\mathcal{L}_{\text{all}}$ in Equation (14)
  Update $\theta_T \leftarrow m\theta_T + (1-m)\theta$
end for
return π_θ

D.1 Compute Resources

Our experiments are conducted on a single server equipped with eight NVIDIA RTX 4090D GPUs. CRAFT uses an asynchronous collect-and-train pipeline, where rollout workers interact with CARLA and write synchronized rollout snapshots to a shared buffer. The trainer then waits until enough rollout files are ready and aggregates them into a rollout batch, preprocesses the counterfactual and corrective signals, and performs one Lightning[1] training iteration before releasing the collectors to continue with the updated policy version. Each driving policy uses four GPUs for closed-loop rollout collection and four GPUs for policy training; under this configuration, one complete fine-tuning run takes 6–10 hours, depending on the policy's inference efficiency and the trainable modules. This design overlaps closed-loop data collection with policy optimization, which reduces idle time in the main training loop.

D.2 Trainable Modules of Driving Policies

We fine-tune only the modules that score candidate trajectories. All perception, map construction, detection, motion prediction, and trajectory-generation modules remain frozen. This restriction keeps sparse online rewards close to the policy decision boundary. The update can change which trajectory candidate is preferred, but it does not rewrite the visual or geometric driving stack. Figure 7 illustrates the operational paradigms of these driving policies.

- **HiP-AD [46].** The ego decision is represented as joint path-speed candidates with 48 ego modes. CRAFT and the non-PPO baselines update only the final classification branch in the last plan-refinement layer. When PPO is used, a lightweight critic is trained in addition to this branch.
- **MindDrive [24].** The high-level speed decision is chosen by a VLM decision expert before the policy generates the low-level path. Fine-tuning updates only the LoRA decision expert used for this high-level decision. The visual encoder, map and detection heads, language backbone, and action expert stay fixed. Under PPO, the critic is the only additional trainable component.

[1] https://github.com/Lightning-AI/pytorch-lightning

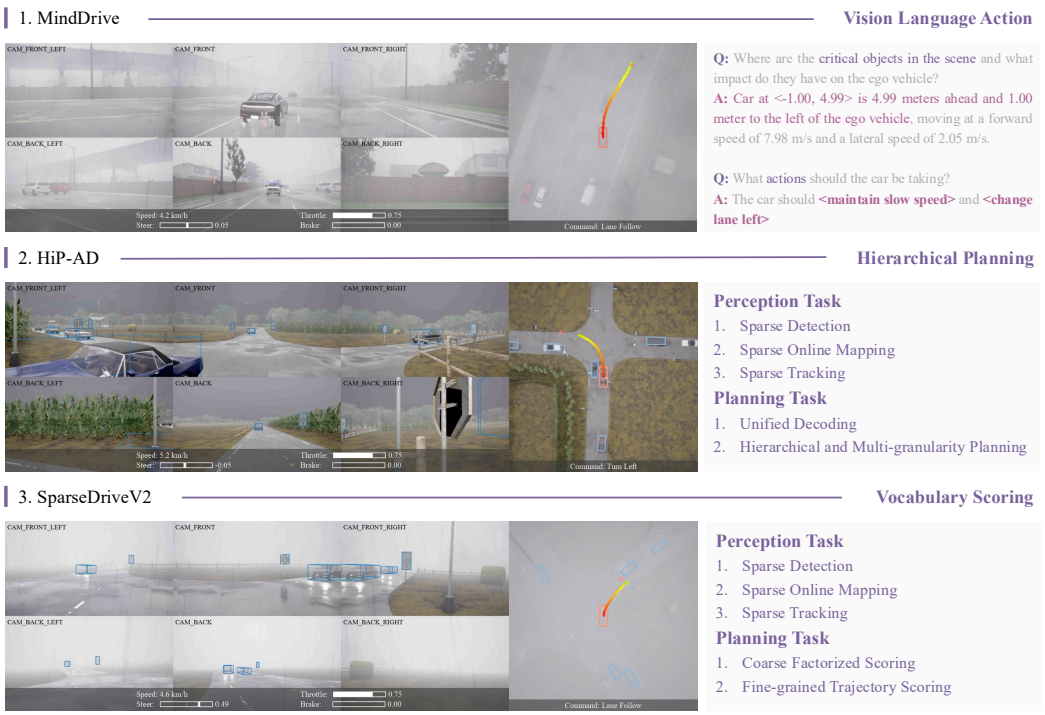


Figure 7: **Operational paradigms of driving policies.**

- **SparseDriveV2**† **[22].** The policy scores a 512-mode trajectory vocabulary. Fine-tuning updates only the final trajectory classification branch that assigns probabilities to vocabulary candidates. The trajectory vocabulary, temporal queue, perception backbone, map head, detection head, and trajectory decoder remain frozen. PPO adds the critic, while all other fine-tuning methods update only the classification branch.

D.3 Counterfactual World

The counterfactual world evaluates the policy candidate set at states visited by real closed-loop rollouts. For each observation, the Bench2Drive agent stores an auxiliary record containing candidate trajectories $\tau_{1:G}$ in the local ego frame, candidate logits, the current ego state, nearby-agent states and actions, route geometry, scenario flags, and nearby map polygons. Ragged candidate counts, route lengths, and agent counts are padded into dense tensors with validity masks before simulation.

By default, each candidate is simulated with a batched trajectory tracker. The tracker prepends the current ego pose to each candidate trajectory, transforms the local trajectory into the global CARLA frame, and flattens the batch and candidate dimensions into independent tracking problems. At each virtual step, a PID controller tracks a short reference segment, and a kinematic bicycle model advances the ego state. Nearby agents use a decay rollout model in which their current actions are held for a short decision window. The model then reduces throttle and ramps up braking, with a shorter ramp when an agent is on a lane connector. Collision checks use rectangular vehicle footprints and batched separating-axis tests.

The counterfactual return for candidate g at real state s_i is

$$\tilde{R}_{i,g} = \sum_{t=0}^{H-1} \gamma^t r^{\mathrm{cp}}_{i,g,t} - \lambda^{\mathrm{cp}}_{\mathrm{red}} I^{\mathrm{red}}_{i,g} - \lambda^{\mathrm{cp}}_{\mathrm{stop}} I^{\mathrm{stop}}_{i,g}. \tag{37}$$

The rollout horizon H, simulator step Δt, and discount γ are listed in Table 8; in this setting, the simulated rollout contains the current state plus H future states. Candidate returns are normalized

Table 8: **Main implementation hyperparameters.** The settings are shared across policies unless the table states otherwise. CRAFT uses the counterfactual reward for candidate scoring and the corrective reward for residual correction.

Component	Setting	Value
Trainer	Epochs per update round	4
	Optimizer	AdamW
	Initial / minimum learning rate	10^{-4} / 5×10^{-6}
	Weight decay	10^{-5}
	Gradient clipping	Global norm 0.5
	Precision / training devices	32-bit / 4 GPUs
	Data decode workers	8
	Shared rollout discount	$\gamma = 0.98$
	GAE trace parameter	$\lambda_{\mathrm{GAE}} = 0.95$
	Value clip range	$\epsilon_v = 2.0$
	Critic learning-rate multiplier	$\kappa_v = 2.0$
	Return-normalization epsilon	$\varepsilon_{\mathrm{norm}} = 10^{-8}$
	EMA teacher momentum	$m = 0.99$
	Rollout buffer size	$B = 2048$ transitions
counterfactual world	Tracker	Batched PID trajectory tracker
	Virtual time step	$\Delta t = 0.1$ s
	Rollout horizon	$H = 20$ steps, 21 states including the current state
	Inference interval	5 simulator frames
	Chunk size	500 candidate rollouts
	Other-agent model	Decay rollout with kinematic bicycle dynamics
	Group normalization	Mean/std within valid candidates
	Std floor for group advantages	$\sigma_{\min} = 5.0$
	Collision speed reference	$v_{\mathrm{ref}} = 5.0$ m/s
	Collision speed weight	$\alpha_v = 0.5$
	Collision speed clip bounds	$[\nu_{\min}, \nu_{\max}] = [0, 1]$
Fine-tuning objectives	Counterfactual proxy loss weight λ_{cp}	1.0
	Grounded residual loss weight λ_{gr}	0.5
	Distillation weight β_r	0.5
	Forward-KL weight β_f	0.1
	PPO clip range ϵ_{clip}	0.2
	Dual-clip constant c	2.0
	Corrective discount γ_c	0.8
	Corrective return scale	$S_{\mathrm{gr}} = 8.0$
	Corrective advantage clip	$[A_{\min}, A_{\max}] = [-1, 1]$

within each valid group as follows:

$$\mu_i = \frac{1}{|G_i|} \sum_{g \in G_i} \tilde{R}_{i,g}, \qquad \sigma_i = \max\left(\sqrt{\frac{1}{|G_i|} \sum_{g \in G_i} (\tilde{R}_{i,g} - \mu_i)^2}, \sigma_{\min} \right), \tag{38}$$

$$\tilde{A}_{i,g} = \frac{\tilde{R}_{i,g} - \mu_i}{\sigma_i}. \tag{39}$$

Padded candidates are masked out and receive zero advantage. This is the group advantage used by GRPO and by the counterfactual proxy of CRAFT.

D.4 Reward Design

The implementation uses two rewards. The counterfactual reward is a dense progress-and-infraction signal used to score candidate futures in the counterfactual world; PPO and REINFORCE++ use the closed-loop instance of the same dense design as their rollout reward. The corrective reward is a sparse safety signal used to instantiate grounded residual correction in CRAFT. The two rewards implement the counterfactual proxy and grounded residual correction, respectively. Both designs are inspired by the progress, route-adherence, and safety terms of CaRL [25].

Table 9: **Reward hyperparameters.** Superscripts distinguish the counterfactual proxy reward from the grounded residual corrective reward; these instantiate the dense counterfactual proxy and grounded residual correction, respectively.

Reward	Symbol	Value	Role
Counterfactual reward	$p_{\min}$ / $p_{\max}$	0.0 / 1.2	Progress clipping bounds
	w_{prog}	8.0	Progress weight
	w_g / w_c / w_h	3.0 / 0.8 / 2.0	Route, centerline, and heading efficiency weights
	$\eta_{\min}$	0.0	Minimum route-efficiency multiplier
	d_{clip}	0.10	Deviation-delta clipping bound
	δ_g^{rec} / δ_c^{rec}	0.08 / 0.08	Recovery activation thresholds
	k_g / k_c / k_h	0.4 / 0.2 / 0.4	Recovery weights
	c_{clip}	0.5	Recovery reward clipping bound
	a_{rec} / b_{rec}	0.5 / 0.5	Progress-dependent recovery scaling
	$\lambda_{\text{offroad}}^{\text{cp}}$	1.5	Off-road step cost
	$\lambda_{\text{opp}}^{\text{cp}}$	0.1	Opposite-lane step cost
	$\lambda_{\text{offroute}}^{\text{cp}}$	1.5	Off-route step cost
	$\lambda_{\text{emg}}^{\text{cp}}$	1.0	Emergency-lane step cost
	$\lambda_{\text{coll}}^{\text{cp}}$	40.0	First-collision penalty
	$\lambda_{\text{red}}^{\text{cp}}$	40.0	Red-light trajectory penalty
	$\lambda_{\text{stop}}^{\text{cp}}$	40.0	Stop-sign trajectory penalty
Corrective reward	$\lambda_{\text{offroad}}^{\text{gr}}$	0.5	Off-road penalty
	$\lambda_{\text{emg}}^{\text{gr}}$	0.2	Emergency-lane penalty
	$\lambda_{\text{offroute}}^{\text{gr}}$	0.5	Off-route penalty
	$\lambda_{\text{red}}^{\text{gr}}$	2.0	Red-light violation penalty
	$\lambda_{\text{stop}}^{\text{gr}}$	2.0	Stop-sign violation penalty
	$\lambda_{\text{coll}}^{\text{gr}}$	5.0	Collision penalty
	v_{stop}	0.1 m/s	Stop-required motion threshold
	v_{go}	2.0 m/s	Go-required slow-motion threshold

Counterfactual reward. The counterfactual reward scores efficiency, route adherence, traffic-rule compliance, and collision safety for each simulated candidate. Progress is clipped and normalized as

$$\bar{\Delta p}_t = \text{clip}(\Delta p_t, p_{\min}, p_{\max}), \qquad \Delta \hat{p}_t = \frac{\bar{\Delta p}_t}{p_{\max}}. \tag{40}$$

The route-efficiency multiplier penalizes route, lane-center, and heading deviations as

$$\eta_t = \max\left(\exp(-w_g d_t^g)\exp(-w_c d_t^c)\exp(-w_h d_t^h), \eta_{\min}\right). \tag{41}$$

The recovery term rewards reductions in existing deviations through

$$c_t = \mathbf{1}[d_t^g > \delta_g^{\text{rec}}]k_g(-\bar{\Delta} d_t^g) + \mathbf{1}[d_t^c > \delta_c^{\text{rec}}]k_c(-\bar{\Delta} d_t^c) + \mathbf{1}[d_t^h > \delta_c^{\text{rec}}]k_h(-\bar{\Delta} d_t^h), \tag{42}$$

$$r_t^{\text{rec}} = \text{clip}(c_t, -c_{\text{clip}}, c_{\text{clip}})\,(a_{\text{rec}} + b_{\text{rec}}\Delta \hat{p}_t)\,, \tag{43}$$

where each deviation delta is clipped to $[-d_{\text{clip}}, d_{\text{clip}}]$. Table 9 reports separate route and centerline recovery thresholds; the heading recovery term intentionally reuses the centerline threshold because both deviations are activated at the same local-lane recovery boundary. The collision penalty is scaled by

$$m_t^{\text{coll}} = 1 + \alpha_v\,\text{clip}\left(\frac{v_t}{v_{\text{ref}}}, \nu_{\min}, \nu_{\max}\right). \tag{44}$$

The counterfactual reward parameters are summarized in Table 9. The per-step counterfactual reward is

$$\begin{aligned} r_t^{\text{cp}} = {} & w_{\text{prog}}\Delta \hat{p}_t \eta_t + r_t^{\text{rec}} - \lambda_{\text{offroad}}^{\text{cp}} I_{\text{offroad}} - \lambda_{\text{opp}}^{\text{cp}} I_{\text{opposite}} - \lambda_{\text{offroute}}^{\text{cp}} I_{\text{offroute}} \\ & - \lambda_{\text{emg}}^{\text{cp}} I_{\text{emergency}} - \lambda_{\text{coll}}^{\text{cp}} m_t^{\text{coll}} I_{\text{collision}}. \end{aligned} \tag{45}$$

In the counterfactual world, red-light and stop-sign violations are trajectory-level flags, giving the return-level penalties in Equation (37). The implementation applies these penalties at the first reward index of the simulated rollout. Collision cost is applied only at the first collision event for each candidate, so high-speed first impacts receive larger penalties.

Corrective reward. The corrective reward is intentionally sparse and safety centered. It is computed after the selected trajectory is converted by the controller into the action actually executed in CARLA, not from alternative virtual candidates:

$$r_t^{\mathrm{gr}} = -\lambda_{\mathrm{offroad}}^{\mathrm{gr}} I_{\mathrm{offroad}} - \lambda_{\mathrm{emg}}^{\mathrm{gr}} I_{\mathrm{emergency}} - \lambda_{\mathrm{offroute}}^{\mathrm{gr}} I_{\mathrm{offroute}} - \lambda_{\mathrm{red}}^{\mathrm{gr}} I_{\mathrm{red}} - \lambda_{\mathrm{stop}}^{\mathrm{gr}} I_{\mathrm{stop}} - \lambda_{\mathrm{coll}}^{\mathrm{gr}} I_{\mathrm{collision}}. \tag{46}$$

The traffic-light and stop-sign terms distinguish uncontrolled zones, stop-required zones, and go-required zones. In a stop-required zone, moving faster than v_{stop} is penalized; in a go-required zone, the wrapper checks whether the path ahead is clear before penalizing motion below v_{go}.

D.5 Fine-tuning Baselines

The fine-tuners differ in how they consume closed-loop and counterfactual signals. PPO and REINFORCE++ use the dense progress-and-infraction reward on executed rollouts, GRPO uses only counterfactual advantages, and CRAFT combines the counterfactual proxy with the corrective reward.

PPO baseline. PPO is the only compared fine-tuner that trains a critic. The datamodule computes generalized advantage estimates with discount γ and trace parameter λ_{GAE}:

$$\delta_t = r_t + \gamma u_t^{\mathrm{term}} V(s_{t+1}) - V(s_t), \qquad A_t = \delta_t + \gamma \lambda_{\mathrm{GAE}} u_t^{\mathrm{done}} A_{t+1}. \tag{47}$$

Here u_t^{term} is the continuation mask for value bootstrapping after terminal states, and u_t^{done} is the continuation mask for recursive return or advantage computation after episode boundaries. The policy loss uses the standard clipped ratio with range ϵ_{clip}, and the value loss uses a clipped SmoothL1 target with value-clip range ϵ_v. The critic parameter group uses the actor learning rate multiplied by κ_v.

REINFORCE++ baseline. This value-free baseline removes the critic and uses normalized Monte Carlo returns from the executed rollout,

$$G_t = r_t + \gamma u_t^{\mathrm{done}} G_{t+1}, \qquad A_t = \frac{G_t - \mu_G}{\sigma_G + \varepsilon_{\mathrm{norm}}}. \tag{48}$$

It then uses the same selected-trajectory clipped ratio as PPO, with the shared clip range ϵ_{clip}.

GRPO baseline. This group-relative baseline discards the closed-loop rollout reward and optimizes the normalized counterfactual advantages in Equation (39). For valid candidates, its objective is

$$\mathcal{L}_{\mathrm{GRPO}} = -\mathbb{E}_i \left[\sum_{g \in G_i} \pi_\theta(\tau_g \mid s_i) \tilde{A}_{i,g} \right] + \beta_r \mathcal{L}_{\mathrm{dist}} + \beta_f \mathcal{L}_{\mathrm{KL}}, \tag{49}$$

where β_r and β_f are the regularization weights in Table 8.

CRAFT. The full method keeps the same counterfactual proxy as GRPO, but adds a selected-trajectory update weighted by the corrective reward observed after closed-loop execution. The corrective return is

$$G_t^{\mathrm{gr}} = r_t^{\mathrm{gr}} + \gamma_c u_t^{\mathrm{done}} G_{t+1}^{\mathrm{gr}}, \tag{50}$$

and the implementation scales and clips it as

$$\hat{A}_t^{\mathrm{gr}} = \mathrm{clip}\left(\frac{G_t^{\mathrm{gr}}}{S_{\mathrm{gr}}}, A_{\min}, A_{\max} \right). \tag{51}$$

Let τ_t be the selected trajectory at s_t and $a_t = \kappa(s_t, \tau_t)$ be the executed controller action. The likelihood ratio is $\rho_t = \exp(\log \pi_\theta(\tau_t \mid s_t) - \log \pi_{\mathrm{old}}(\tau_t \mid s_t))$. The grounded residual correction objective uses PPO clipping followed by dual clipping:

$$o_t^{\min} = \min\left(\rho_t \hat{A}_t^{\mathrm{gr}},\ \mathrm{clip}(\rho_t, 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}}) \hat{A}_t^{\mathrm{gr}} \right), \tag{52}$$

$$o_t = \begin{cases} \max(o_t^{\min}, c\hat{A}_t^{\mathrm{gr}}), & \hat{A}_t^{\mathrm{gr}} < 0, \\ o_t^{\min}, & \hat{A}_t^{\mathrm{gr}} \geq 0, \end{cases} \tag{53}$$

The implemented CRAFT loss is

$$\mathcal{L}_{\mathrm{CRAFT}} = \lambda_{\mathrm{cp}} \mathcal{L}_{\mathrm{cp}} + \lambda_{\mathrm{gr}} \mathcal{L}_{\mathrm{gr}} + \beta_r \mathcal{L}_{\mathrm{dist}} + \beta_f \mathcal{L}_{\mathrm{KL}}, \tag{54}$$

where λ_{cp}, λ_{gr}, β_r, and β_f are set as in Table 8.

Teacher regularization. All reported fine-tuners keep a teacher initialized from the pre-trained actor. For the trainable actor subset, the teacher is updated by EMA,

$$\theta_T \leftarrow m\,\theta_T + (1-m)\theta. \tag{55}$$

The distillation term is $D_{\mathrm{KL}}(\pi_\theta \| \pi_T)$, and the optional forward-KL term is $D_{\mathrm{KL}}(\pi_T \| \pi_\theta)$. The ego configurations enable the forward-KL term for all reported fine-tuners.

E Broader Impacts

Positive societal impacts. This work studies an efficient approach to fine-tuning autonomous driving policies with closed-loop feedback. CRAFT uses dense counterfactual supervision for common driving decisions and grounded residual correction for interaction-dependent events, such as collisions, off-route behavior, and traffic-rule violations. This separation reduces the dependence of policy improvement on large numbers of expensive closed-loop rollouts. The same idea can be applied to different closed-loop platforms, including interactive simulators and learned world models, provided that the platform can score candidate futures and provide feedback from executed actions. The method also keeps perception and most policy modules frozen in the reported experiments, which can make safety-oriented fine-tuning more accessible without retraining a full driving stack.

Negative societal impacts. The main potential risk is over-reliance on a particular closed-loop platform. A policy improved in a simulator or a world model may still inherit limitations from that platform, including incomplete scenario coverage, simplified traffic behavior, or reward choices that fail to capture uncommon cases. More efficient fine-tuning may also lower the barrier to adapting a driving policy before sufficiently broad validation has been completed. This paper treats CRAFT as a post-training research method rather than a deployment procedure. Any real-world use would require separate safety validation, stress testing, and checks against local driving rules before operation outside controlled environments.

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: It accurately reflects the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses several limitations of the work performed by the authors in Section 5.

   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: See Appendix A for the full set of assumptions and complete proofs.

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: See Appendix D for detailed experimental setup and procedures.

   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: [Yes]

Justification: Code, environment setup, data access instructions, and training/evaluation commands are available at https://github.com/CurryChen77/CraftPolicy. The project page is linked in the abstract.

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: See Appendix D.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Main method comparisons use a single random seed. Appendix B reports three-seed protocol checks, but uncertainty across seeds is not reported for the main method comparisons.

   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: See Appendix D.1.

   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: We have reviewed the Code of Ethics and confirm that we conform to the guidelines.

   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: See Appendix E for a discussion of positive and negative societal impacts.

    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 work evaluates driving policies in simulation and does not release personal data or a new general-purpose generative model. Risks of real-world deployment are discussed in Appendix E.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [No]

    Justification: The original sources of the driving policies, simulator, and benchmark are cited. A consolidated account of all third-party asset licenses and checkpoint terms is not provided in the paper.

    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: [Yes]

    Justification: The CRAFT code is available under the Apache License 2.0 at https://github.com/CurryChen77/CraftPolicy, with environment setup, data/checkpoint access instructions, and training/evaluation examples.

    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 paper does not involve crowdsourcing nor research with human subjects.

    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 paper does not involve crowdsourcing nor research with human subjects.

    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: CRAFT introduces no LLM-specific mechanism. MindDrive is an existing vision-language-action policy used for evaluation; its pre-trained language backbone remains frozen, as described in Appendix D.2.

    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.