BEYOND IMITATION: PLANNER-AGNOSTIC REINFORCEMENT LEARNING POST-TRAINING FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-end autonomous driving has advanced rapidly on large-scale closed-loop benchmarks such as nuPlan, with planners commonly based on regression, diffusion, or flow matching. Although imitation learning (IL) provides strong behavioral priors, it primarily optimizes similarity to logged expert trajectories rather than task-level driving metrics such as safety, progress, comfort, and interaction quality. Reinforcement learning (RL) can address this limitation, but existing RL post-training methods are often coupled to planner-specific generation mechanisms, resulting in fragmented solutions across planner families. We propose Planner-Agnostic Reinforcement Learning (PARL), a unified post-training framework that isolates planner-specific differences in a lightweight log-probability adapter while sharing the remaining RL pipeline, including reward computation, KL regularization, and policy optimization. PARL supports regression-based, diffusion-based, and flow-matching-based planners without modifying their architectures, inference interfaces, or action spaces, and requires no privileged information. Under the full nuPlan closed-loop protocol, covering non-reactive and reactive settings on Val14, Test14-hard, and Test14, PARL consistently improves all six evaluated IL planners across all six settings, with the largest gains in interaction-heavy long-tail scenarios. These results demonstrate the feasibility and effectiveness of unified, planner-agnostic RL post-training for end-to-end autonomous driving. Code and pretrained checkpoints will be released upon acceptance.

1 INTRODUCTION

End-to-end autonomous driving has made rapid progress on large-scale benchmarks such as nuPlan (Caesar et al., 2021; Karnchanachari et al., 2024). Recent planners generally fall into three families: regression-based planners (Cheng et al., 2024b;a), diffusion-based planners (Zheng et al., 2025; Yao et al., 2026), and flow-matching-based planners (Tan et al., 2025; Wang et al., 2026a). Despite their different trajectory-generation mechanisms, these planners are predominantly trained by imitation learning (IL) on logged expert trajectories. IL is simple, scalable, and effective for learning strong behavioral priors. However, it primarily optimizes open-loop trajectory matching and does not directly optimize task-level driving metrics such as safety, progress, comfort, and interaction quality.

IL-only training also suffers from distribution shift: execution errors can move the planner away from expert-induced states, where recovery supervision is unavailable and failures may compound (Pomerleau, 1988). Moreover, key driving objectives such as collision avoidance, route completion, and comfort are not directly optimized by imitation losses. Reinforcement learning (RL) provides a natural way to improve this alignment by optimizing metric-based rewards computed from predicted trajectories. However, existing RL methods for end-to-end driving are often tightly coupled to planner-specific generation mechanisms. Diffusion-based methods optimize iterative denoising processes, while regression-based methods operate on directly predicted trajectories or multi-hypothesis outputs (Black et al., 2024; Zou et al., 2025), and flow-matching planners require yet another likelihood formulation. Some approaches also rely on privileged simulator information. Consequently, RL post-training remains fragmented across planner families.

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

This motivates the question: *Can RL post-training be made planner-agnostic?* We observe that the planner-specific differences relevant to policy optimization can be isolated in a lightweight log-probability adapter. The remaining training procedure, including trajectory sampling, reward computation, KL regularization, advantage normalization, and policy optimization, can then be shared across planner families. Based on this factorization, we propose **Planner-Agnostic Reinforcement Learning (PARL)**, a unified RL post-training framework for end-to-end driving planners. PARL provides family-specific likelihood adapters for categorical multi-hypothesis regression planners, diffusion planners, and flow-matching planners, while preserving each planner's original architecture, inference interface, and action space without requiring privileged information.

We evaluate PARL under the standard nuPlan closed-loop protocol, covering non-reactive and reactive settings on Val14, Test14-hard, and Test14. Across six imitation-learning planners spanning the three planner families, PARL consistently improves the IL initialization in all six evaluation settings, with the largest gains observed in interaction-heavy long-tail scenarios. These results demonstrate that a shared RL post-training recipe with lightweight family-specific adapters can improve heterogeneous end-to-end driving planners, and that the resulting improvements carry over to closed-loop evaluation.

Our contributions are threefold:

- We identify a factorization of RL post-training into a shared planner-agnostic policy optimization recipe and a lightweight planner-family-specific log-probability adapter.
- We derive practical likelihood adapters for regression-based, diffusion-based, and flow-matching-based planners, enabling a common RL post-training procedure across all three families.
- We evaluate PARL on the full nuPlan closed-loop protocol across six planners and six evaluation settings. PARL consistently improves every planner, demonstrating the effectiveness and practical viability of planner-agnostic RL post-training.

2 Related Work

End-to-end planners for autonomous driving. End-to-end autonomous driving has emerged as a unified alternative to modular perception-prediction-planning pipelines (Chen et al., 2024). Existing E2E planners can be broadly categorized into three representative families according to their trajectory parameterization, all of which are covered by our framework. Regression or multi-hypothesis planners, such as PlanTF (Cheng et al., 2024b), PLUTO (Cheng et al., 2024a), and HiP-AD (Tang et al., 2025), directly predict candidate trajectories. Diffusion-based planners, including DiffusionDrive (Liao et al., 2025), Diffusion Planner (Zheng et al., 2025), and ReflexDiffusion (Yao et al., 2026), formulate planning as iterative denoising. Flow-matching planners such as GoalFlow (Xing et al., 2025), FlowDrive (Wang et al., 2026a), and Flow Planner (Tan et al., 2025) instead learn continuous transformations from noise to trajectory distributions. Beyond these single-paradigm designs, recent hybrid planners further combine multiple parameterizations within a single model, e.g., ChainFlow-VLA (Wang et al., 2026b) couples an autoregressive proposal generator with a residual diffusion refiner conditioned on VLM hidden states. Such hybrids further broaden the space of trajectory parameterizations that any unified RL post-training recipe must eventually accommodate.

These families differ in action parameterization, sampling mechanisms, and likelihood formulation, making unified RL post-training non-trivial. Existing RL post-training methods, therefore, remain largely confined to individual planner families, and recipe-level unification across heterogeneous trajectory parameterizations remains underexplored.

Imitation learning in driving Imitation learning remains the dominant training paradigm for E2E driving planners. Classical behavioral cloning learns from expert demonstrations, while data augmentation and Dagger-style aggregation (Ross et al., 2011) improve robustness by broadening the training distribution. In autonomous driving, ChauffeurNet (Bansal et al., 2019), TransFuser (Chitta et al., 2023), UniAD (Hu et al., 2023), VAD (Jiang et al., 2023), PlanTF (Cheng et al., 2024b), and PLUTO (Cheng et al., 2024a) show that large-scale supervised imitation and strong scene representations can achieve competitive closed-loop performance.

However, IL optimizes expert matching rather than long-horizon driving objectives. Metrics such as collision avoidance, route progress, rule compliance, comfort, and interaction quality emerge only after policy rollout and are not directly optimized by imitation losses. Thus, IL provides a strong behavioral prior but remains limited as a standalone objective for closed-loop driving.

Reinforcement learning for autonomous driving Reinforcement learning enables direct optimization of closed-loop driving rewards beyond expert imitation. Early RL-based driving methods relied on simulators such as CARLA (Dosovitskiy et al., 2017); works such as Roach (Zhang et al., 2021) and Learning by Cheating (Chen et al., 2020) further demonstrated the value of RL experts and privileged simulator information. However, pure RL is often sample-inefficient for large-scale E2E planners, motivating recent IL-initialized RL post-training.

Existing RL post-training methods are typically architecture-specific. DDPO (Black et al., 2024) formulates diffusion generation as multi-step decision making, while DiffusionDriveV2 (Zou et al., 2025) and DIVER (Song et al., 2025) exploit diffusion denoising for reward optimization in driving planners. Regression-based methods operate on directly regressed trajectories or trajectory hypotheses, and autoregressive methods such as CarPlanner (Zhang et al., 2025) tailor RL to sequential trajectory-token generation. Consequently, existing RL approaches are often tied to a particular planner family, rely on privileged signals, or are evaluated only within a single architecture class.

PARL addresses this gap by factorizing RL post-training into a planner-agnostic policy-improvement recipe and a lightweight planner-native log-probability adapter, enabling unified reward-driven post-training for regression, diffusion, and flow-matching planners.

3 PRELIMINARY

3.1 PROBLEM FORMULATION

We formulate RL post-training as a one-step decision problem for each initial driving scene. Given the scene observation, a planner samples a complete future ego trajectory and receives a trajectory-level reward.

State. The state s aggregates (i) the ego kinematic vector $(x, y, \psi, v_x, v_y, a)$, where a denotes the longitudinal acceleration, (ii) neighbor-agent trajectories over the past 2 s with per-frame validity masks, (iii) static-object bounding boxes, (iv) a vectorized map containing lane centerlines, road boundaries, and crosswalks with per-point orientations, and (v) the ego route represented as an ordered set of reference lines. All quantities are expressed in the ego-centric frame at the decision time. These inputs are identical to those used by the corresponding IL planner and contain no privileged information.

Action. The action τ is a candidate future ego trajectory of $T = 80$ frames sampled at 10 Hz. We parameterize it as $\{(x_t, y_t, \cos\psi_t, \sin\psi_t)\}_{t=1}^{T} \in \mathbb{R}^{T\times 4}$.

Policy. A planner with parameters θ induces a stochastic policy $\pi_\theta(\tau \mid s)$ over candidate trajectories. The corresponding probability model depends on the planner family. In this work, we consider multi-hypothesis regression-based, diffusion-based, and flow-matching-based planners.

3.2 METRIC-BASED TRAJECTORY REWARD

For efficient RL training, the trajectory reward is computed directly from the predicted trajectory using a *PyTorch* reimplementation of the corresponding nuPlan scoring logic, without invoking the nuPlan simulator or performing closed-loop rollout. The score $\mathrm{Score}(\tau)$ consists of eight sub-scores $\{c_i(\tau)\}_{i=1}^{8}$. Four are multiplicative gates ($\mathcal{I}_{\text{mult}}$: no-fault-collision, drivable-area, progress-made, driving-direction), and four are additive weighted components ($\mathcal{I}_{\text{add}}$: progress-along-route, TTC, speed-limit, comfort) with weights $w = (5, 5, 4, 2)$.

$$\mathrm{Score}(\tau) = \left(\prod_{i\in\mathcal{I}_{\text{mult}}} c_i(\tau)\right) \cdot \frac{\sum_{j\in\mathcal{I}_{\text{add}}} w_j\, c_j(\tau)}{\sum_{j\in\mathcal{I}_{\text{add}}} w_j}. \tag{1}$$

The reward supplied to RL is a linear rescale of Eq. (1) into the symmetric range $[-1, 1]$, denoted $R(\tau) = 2\,\mathrm{Score}(\tau) - 1$. At evaluation time, we use the official nuPlan closed-loop protocol.

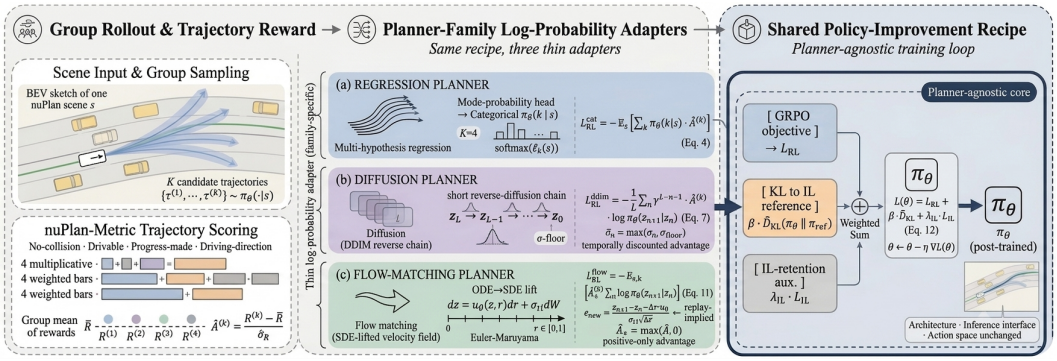

Figure 1: **Overall pipeline of PARL.** The framework separates planner-specific log-probability computation from a shared RL post-training pipeline, enabling unified optimization of regression-based, diffusion-based, and flow-matching-based planners.

3.3 Group Relative Policy Optimization

We build PARL on GRPO (Shao et al., 2024). For each scene s, the policy samples K candidate trajectories $\{\tau^{(k)}\}_{k=1}^{K}$, each scored by $R^{(k)} = R(\tau^{(k)})$. The group-relative advantage is

$$\begin{aligned} \hat{A}^{(k)} &= \frac{R^{(k)} - \bar{R}}{\hat{\sigma}_R + \epsilon}, \\ \bar{R} &= \frac{1}{K} \sum_{k'} R^{(k')}, \\ \hat{\sigma}_R^2 &= \frac{1}{K} \sum_{k'} \left(R^{(k')} - \bar{R}\right)^2. \end{aligned} \tag{2}$$

We adopt a strictly on-policy, single-update formulation and optimize the group-relative advantage with a REINFORCE-style (Williams, 1992) score-function estimator:

$$\mathcal{L}_{\mathrm{RL}}(\theta) = -\mathbb{E}_{s,k}\left[\hat{A}^{(k)} \cdot \log \pi_\theta(\tau^{(k)} \mid s)\right]. \tag{3}$$

The group mean serves as an implicit sample-based baseline, avoiding a separate value network. This objective is shared across planner families, with each family providing its own definition of $\log \pi_\theta(\tau \mid s)$.

4 Methodology

4.1 Framework Overview

As illustrated in Fig. 1, PARL factorizes RL post-training into three stages per iteration. Given an IL-pretrained planner π_θ, the left stage samples K candidate trajectories and computes their rewards $R^{(k)}$ using the PyTorch metric scorer described in Eq. (1). The center stage is the per-family log-probability adapter, which computes $\log \pi_\theta(\tau \mid s)$ for multi-hypothesis regression, diffusion, or flow-matching planners. The right stage applies the shared RL recipe, combining the RL objective $\mathcal{L}_{\mathrm{RL}}$ with a KL anchor to the frozen IL reference π_{ref} and an IL-retention auxiliary loss to update π_θ. Thus, sampling, reward computation, optimization, and shared regularizers are common across planner families, with differences confined to the log-probability adapter and its family-specific stabilization. The planner architecture, inference interface, and action space remain unchanged.

4.2 Shared Training Components

Two components are applied with the GRPO objective of Eq. (3), forming the shared training pipeline.

KL regularizer to the IL initialization. We anchor π_θ to the frozen IL reference π_{ref} with $\beta\, \widehat{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})$, where $\beta \geq 0$. For regression planners, the categorical KL is evaluated exactly. For continuous planners, we use the low-variance k3 estimator $\widehat{D}_{\text{KL}} = \mathbb{E}[\exp(\Delta) - \Delta - 1]$, where $\Delta = \log \pi_{\text{ref}}(\tau) - \log \pi_\theta(\tau)$ is clamped to $[-5, 5]$ for numerical stability.

IL-retention auxiliary loss. We retain the IL objective used to obtain π_{ref} as $\lambda_{\text{IL}}\mathcal{L}_{\text{IL}}$, with Smooth-L1 plus mode cross-entropy for regression, clean-trajectory ℓ_2 loss for diffusion, and velocity-field regression along the affine probability path for flow matching. This keeps RL post-training anchored to the original IL objective.

4.3 Per-Family Log-Probability Adapters

Each adapter defines how to compute $\log \pi_\theta(\tau \mid s)$ for a planner family and incorporates any family-specific stabilization. The resulting objective follows the shared GRPO template of Eq. (3).

Multi-hypothesis regression planners. A multi-hypothesis regression planner predicts M candidate trajectories $\{\tau^{(k)}\}_{k=1}^{M}$ with mode logits $\ell_k(s) \in \mathbb{R}$, inducing a categorical policy $\pi_\theta(k \mid s) \propto \exp(\ell_k(s))$ over M trajectory hypotheses. The planner is already a stochastic policy over the trajectory space; the GRPO update of Eq. (3) reduces to a categorical policy gradient on the mode logits:

$$\mathcal{L}_{\text{RL}}^{\text{cat}} = -\mathbb{E}_{s,k}\left[\hat{A}^{(k)} \log \pi_\theta(k \mid s)\right]. \tag{4}$$

The RL gradient flows through the categorical distribution $\pi_\theta(k \mid s)$ into the mode-scoring branch of the planner, updating both the mode head and its upstream shared backbone. The trajectory-emission head that produces the raw candidate coordinates $\{\tau^{(k)}\}$ is supervised by the IL auxiliary loss $\mathcal{L}_{\text{IL}}$. This division of labor requires no noise injection into the trajectory-generation path and no auxiliary stochastic perturbations, unlike the diffusion and flow-matching cases below. When candidates come from a scene-conditioned variable number of proposals, invalid entries are masked with a large negative logit before the softmax. The adapter applies unchanged to anchor-based, mixture-density, and query-based decoders.

Diffusion planners. A diffusion planner generates a trajectory through L DDIM (Song et al., 2021) reverse steps from an isotropic Gaussian prior. Let $z_n \in \mathbb{R}^{T\times d}$ denote the latent trajectory at denoising step $n \in \{0, \ldots, L-1\}$, and let $r_n \in [0, 1]$ denote its corresponding generative-time coordinate. We model the reverse process as an L-step Markov chain with Gaussian transitions

$$\pi_\theta(z_{n+1} \mid z_n) = \mathcal{N}\big(z_{n+1}; \mu_\theta(z_n, r_n), \sigma_n^2 I\big)\,, \tag{5}$$

where μ_θ is the DDIM next-step mean under the clean-sample parameterization and σ_n is the corresponding stochastic standard deviation. The trajectory log-probability decomposes as $\log \pi_\theta(\tau) = \sum_{n=0}^{L-1} \log \pi_\theta(z_{n+1} \mid z_n)$. We introduce two modifications for stable optimization. First, σ_n can approach zero near the end of the reverse process, causing the Gaussian log-density to become numerically unstable. We therefore impose a floor $\sigma_{\text{floor}} > 0$ when evaluating the log-probability, without modifying the sampling transition in Eq. (5):

$$\begin{aligned} \tilde{\sigma}_n &= \max(\sigma_n,\, \sigma_{\text{floor}})\,, \\ \log \pi_\theta(z_{n+1} \mid z_n) &= \log \mathcal{N}\big(z_{n+1};\ \mu_\theta(z_n, r_n),\ \tilde{\sigma}_n^2\big)\,, \end{aligned} \tag{6}$$

Second, we distribute the scene-level advantage $\hat{A}^{(k)}$ from Eq. (2) across denoising steps using γ^{L-n-1}, assigning less weight to early, high-noise transitions whose effect on the final trajectory is less direct:

$$\mathcal{L}_{\text{RL}}^{\text{ddim}} = -\mathbb{E}_{s,k}\left[\hat{A}^{(k)} \cdot \frac{1}{L}\sum_{n=0}^{L-1} \gamma^{L-n-1} \log \pi_\theta\left(z_{n+1}^{(k)} \mid z_n^{(k)}\right)\right] \tag{7}$$

Flow-matching planners. A flow-matching planner integrates a learned velocity field $u_\theta(z, r)$ from $r = 0$ to $r = 1$ according to the deterministic ODE $\mathrm{d}z = u_\theta(z, r)\,\mathrm{d}r$. Since a deterministic ODE does not provide a per-step transition density, we introduce an SDE lift for RL post-training:

$$\mathrm{d}z = u_\theta(z, r)\,\mathrm{d}r + \sigma_{\text{rl}}\,\mathrm{d}W, \tag{8}$$

where W is a standard Wiener process and $\sigma_{\mathrm{rl}} > 0$ is a constant diffusion coefficient. The lift serves only as an exploratory perturbation during post-training; inference retains the native deterministic ODE integration. Discretizing this SDE with Euler–Maruyama using L steps of size Δr yields the Gaussian transition

$$\pi_\theta(z_{n+1} \mid z_n) = \mathcal{N}\big(z_{n+1}; z_n + \Delta r\, u_\theta(z_n, r_n), \sigma_{\mathrm{rl}}^2 \Delta r\, I\big)\,, \tag{9}$$

so that the trajectory log-probability decomposes into a sum of L per-step Gaussian log-densities. This construction avoids the divergence-based likelihood computation required by an exact ODE change of variables, while providing a tractable transition density for policy optimization.

A key issue is preserving gradient dependence on the current velocity field. If the rollout noise $\varepsilon^{(n)}$ is reused as a detached quantity, the Gaussian log-probability is proportional to $-\|\varepsilon^{(n)}\|^2/2$ and has no gradient path to the current policy parameters. We therefore recompute the implied noise from the stored rollout trajectory:

$$\varepsilon_{\mathrm{new}}^{(n)} = \frac{z_{n+1}^{\mathrm{old}} - z_n^{\mathrm{old}} - \Delta r\, u_\theta\big(z_n^{\mathrm{old}}, r_n\big)}{\sigma_{\mathrm{rl}}\sqrt{\Delta r}}, \tag{10}$$

which restores gradient dependence through u_θ.

As a flow-specific stabilization, we rectify the group-relative advantage as

$$\hat{A}_+^{(k)} = \max\Big(\hat{A}^{(k)}, 0\Big)\,, \tag{11}$$

so that only above-average candidates contribute to the flow-matching update, since the SDE-lifted policy is highly stochastic early in training and low-reward candidates with large noise can produce disruptive gradient directions. The resulting objective is

$$\mathcal{L}_{\mathrm{RL}}^{\mathrm{flow}} = -\mathbb{E}_{s,k}\left[\hat{A}_+^{(k)} \sum_{n=0}^{L-1} \log \pi_\theta\Big(z_{n+1}^{(k)} \mid z_n^{(k)}\Big)\right]. \tag{12}$$

4.4 Overall Objective and Training Loop

Combining the family-specific RL loss with the shared KL and IL-retention terms, PARL minimizes

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{RL}}(\theta) + \beta\, \widehat{D}_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) + \lambda_{\mathrm{IL}}\, \mathcal{L}_{\mathrm{IL}}(\theta), \tag{13}$$

where $\mathcal{L}_{\mathrm{RL}}$ is instantiated by the corresponding regression, diffusion, or flow-matching adapter.

Starting from the IL initialization, each training iteration samples K candidate trajectories from the current policy π_θ, computes their metric-based rewards $R^{(k)}$, and updates θ by gradient descent on Eq. (13).

5 Experiment

We evaluate PARL on the nuPlan closed-loop benchmark; the full experimental setup, including benchmarks, baselines, and implementation details, is provided in Appendix A. We first compare PARL with six IL baselines across three planner families and six evaluation settings, and then examine its impact on driving behavior through qualitative case studies in interaction-heavy scenarios.

5.1 Main Results

Consistent Improvement across Planner Families. Figure 2 summarizes the average closed-loop score of each planner across the six evaluation settings, covering Val14, Test14-hard, and Test14 under both non-reactive (NR-CLS) and reactive (R-CLS) modes. Across all six IL baselines spanning regression (PlanTF, PLUTO), diffusion (Diffusion Planner, ReflexDiffusion, DIAMOND), and flow-matching (FlowDrive), PARL consistently yields positive improvements ranging from +0.81 to +2.45. The largest gains are achieved by PLUTO (+2.45) and Diffusion Planner (+2.37), demonstrating that the same RL post-training recipe transfers effectively across substantially different trajectory parameterizations. Even for FlowDrive, which has the strongest IL initialization with an

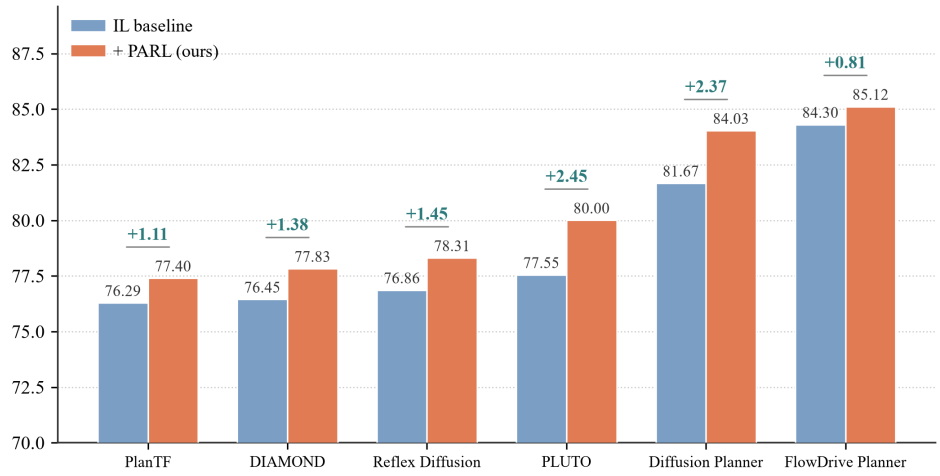

Figure 2: **Consistent improvement across planner families.** Average closed-loop score of six IL planners across the six nuPlan evaluation settings, including Val14, Test14-hard, and Test14 under both non-reactive (NR-CLS) and reactive (R-CLS) modes. PARL consistently improves every IL initialization across regression-based, diffusion-based, and flow-matching-based planners.

Table 1: Closed-loop planning results on the nuPlan benchmark. We report the closed-loop score (higher is better) on Val14, Test14-hard, and Test14 in both non-reactive (NR-CLS) and reactive (R-CLS) modes. Planners are labeled by parameterization family: REG for multi-hypothesis regression, DM for diffusion, and FM for flow matching. Each IL baseline is compared with its PARL post-trained counterpart, with the improvement over the IL initialization shown in parentheses.

Type	Family	Planner	Val14		Test14-hard		Test14	
			NR-CLS	R-CLS	NR-CLS	R-CLS	NR-CLS	R-CLS
Expert	–	Log-replay	93.53	80.32	85.96	68.80	94.03	75.86
IL	REG	PlanTF (Cheng et al., 2024b)	84.27	76.95	69.70	61.61	85.62	79.58
	REG	PLUTO (Cheng et al., 2024a)	88.89	78.11	70.03	59.74	89.90	78.62
	DM	Diffusion Planner (Zheng et al., 2025)	89.87	82.80	75.99	69.22	89.19	82.93
	DM	ReflexDiffusion (Yao et al., 2026)	87.23	74.55	76.18	61.53	86.40	75.28
	DM	DIAMOND (Ke et al., 2026)	86.29	78.30	68.46	59.17	86.55	79.90
	FM	FlowDrive (Wang et al., 2026a)	91.21	85.37	77.86	73.09	91.02	87.28
IL + PARL	REG	PlanTF-PARL	84.91 +0.64	78.39 +1.44	71.72 +2.02	61.75 +0.14	87.02 +1.40	80.60 +1.02
	REG	PLUTO-PARL	89.50 +0.61	79.98 +1.87	75.84 +5.81	60.86 +1.12	92.39 +2.49	81.45 +2.83
	DM	Diffusion Planner-PARL	91.40 +1.53	83.30 +0.50	81.60 +5.61	70.44 +1.22	92.65 +3.46	84.81 +1.88
	DM	ReflexDiffusion-PARL	87.75 +0.52	76.81 +2.26	77.29 +1.11	64.08 +2.55	88.30 +1.90	75.62 +0.34
	DM	DIAMOND-PARL	87.68 +1.39	79.93 +1.63	71.45 +2.99	59.69 +0.52	87.94 +1.39	80.27 +0.37
	FM	FlowDrive-PARL	91.87 +0.66	85.61 +0.24	79.42 +1.56	73.50 +0.41	92.29 +1.27	88.00 +0.72

average score of 84.30, PARL provides a further +0.81 improvement, indicating that metric-driven post-training remains beneficial even for a strong imitation-based planner.

Per-Benchmark Performance. Table 1 examines the performance gains across individual benchmarks and evaluation modes. PARL consistently improves every planner in all six settings, demonstrating robust gains across different scenario distributions and traffic dynamics. The largest improvements occur on Test14-hard NR-CLS, which emphasizes interaction-heavy and long-tail scenarios: PLUTO improves from 70.03 to 75.84 (+5.81), Diffusion Planner from 75.99 to 81.60 (+5.61), and DIAMOND from 68.46 to 71.45 (+2.99). The gains also persist under reactive traffic rather than being confined to non-reactive settings. On Test14-hard R-CLS, ReflexDiffusion and Diffusion Planner improve by +2.55 and +1.22, respectively, while all planners also achieve positive gains on Val14 R-CLS and Test14 R-CLS. FlowDrive exhibits relatively smaller gains (+0.24 to +1.56), consistent with its stronger IL initialization and consequently smaller improvement headroom.

Per-Sub-Score Analysis. To identify which aspects of driving behavior benefit most from RL post-training, Table 2 reports five closed-loop sub-scores on Test14-hard for one representative planner from each trajectory-parameterization family: PLUTO, Diffusion Planner, and FlowDrive. The largest gains are observed in TTC, with improvements of +4.27 for PLUTO, +4.78 for Diffusion Planner, and +3.31 for FlowDrive. Since TTC captures future interaction safety and is explicitly included in our reward, these gains indicate that PARL improves safety-related behavior beyond open-loop imitation. Collision scores also improve across all three planners (+1.22, +3.30, and +1.84), while Drivable and Comfort consistently increase, with gains of (+2.22, +0.37, +1.10) and (+0.07, +2.21, +0.74), respectively. Thus, the safety improvements do not come at the expense of drivable-area compliance or ride comfort. Progress is the only sub-score with a slight decrease, occurring for FlowDrive (-0.58), whereas PLUTO and Diffusion Planner improve by +1.18 and +4.18, respectively. This is consistent with a slight safety-progress trade-off: the post-trained policy adopts more conservative speed profiles and larger gaps to nearby agents, as illustrated in Figure 3, at a small cost in route progress. Overall, PARL improves all three safety-related metrics (Drivable, Collision, and TTC) for every planner while also improving Comfort.

5.2 Qualitative Analysis

To visualize the effect of PARL post-training beyond aggregate metrics, Figure 3 compares each of the three representative IL planners with its PARL counterpart in two interaction-heavy nuPlan scenarios. The examples highlight how metric-driven post-training changes the resulting driving behavior under the same initial conditions.

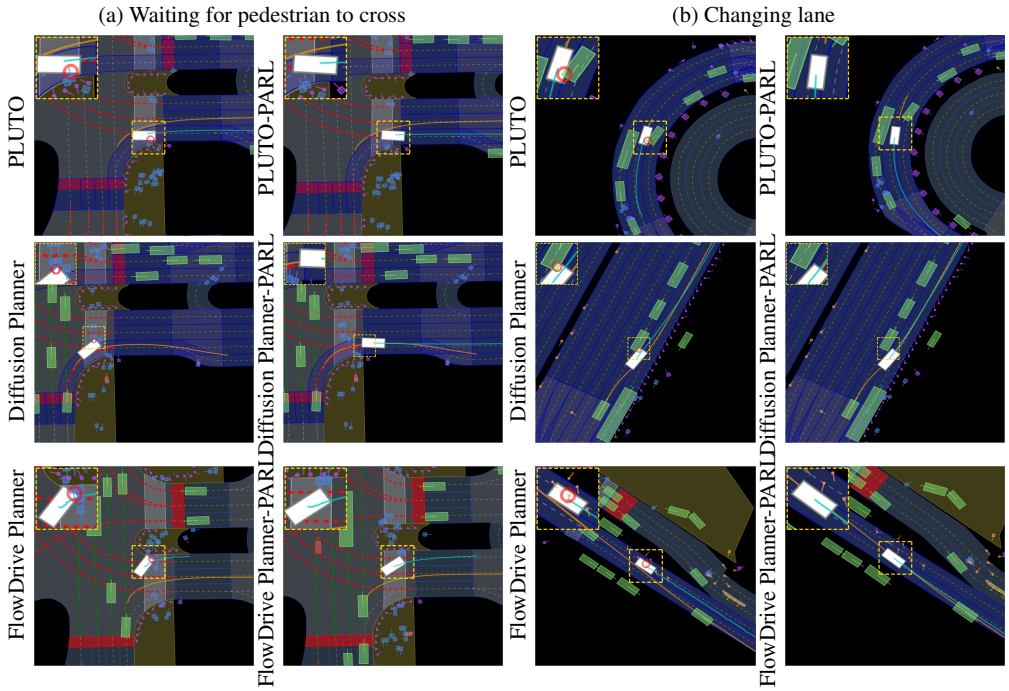

Figure 3: **Interaction-Aware Closed-Loop Behavior.** Qualitative comparison of IL initialization and PARL post-training on two interaction-heavy nuPlan scenarios: (a) waiting for pedestrian to cross and (b) lane changing. PLUTO (regression), Diffusion Planner (diffusion), and FlowDrive (flow matching) are compared under identical initial conditions, with IL and PARL shown side by side. The ego vehicle is drawn as a white rectangle, surrounding vehicles as green rectangles, pedestrians as small blue rectangles, and traffic cones as yellow dots. Red circles indicate collision events observed in the IL cases and avoided after PARL post-training.

In scenario (a), a pedestrian crosses in front of the ego vehicle at an intersection. All three IL baselines, PLUTO, Diffusion Planner, and FlowDrive Planner, continue toward the crossing and collide with the pedestrian, as indicated by the red circles in the first column. After PARL post-training, all

three planners respond earlier by reducing their longitudinal speed and avoid the collision, demonstrating a more conservative response to a vulnerable road user. Notably, this behavior emerges consistently across regression-based, diffusion-based, and flow-matching-based planners despite their substantially different trajectory generation mechanisms.

In scenario (b), the ego vehicle performs a lane change on a curved multi-lane road with multiple nearby hazards. PLUTO and Diffusion Planner collide with a neighboring vehicle in the target lane, while FlowDrive Planner avoids the vehicle but strikes a traffic cone placed further along the planned path. After PARL post-training, PLUTO-PARL and Diffusion Planner-PARL adjust their trajectories to maintain a sufficient longitudinal gap from the neighboring vehicle and complete the lane change safely. FlowDrive Planner-PARL additionally modifies its lateral trajectory to avoid the traffic cone. These examples illustrate that the improvement is not limited to reducing a single type of collision: PARL enables the planners to adapt both longitudinally and laterally to different interacting hazards.

Table 2: Per-sub-score comparison of IL initialization and PARL post-training on Test14-hard under non-reactive (NR-CLS) traffic. We report Drivable (drivable-area compliance), Comfort (ego comfort), Progress (progress along expert route), Collision (no-fault collision), and TTC (time-to-collision within bound) for PLUTO (regression), Diffusion Planner (diffusion), and FlowDrive (flow matching). Parentheses indicate improvement over the corresponding IL baseline.

Type	Planner	Drivable	Comfort	Progress	Collision	TTC
IL	PLUTO	95.90	92.48	86.28	86.84	77.82
	Diffusion Planner	96.69	93.75	88.14	88.24	77.94
	FlowDrive	95.22	95.22	91.97	88.42	77.57
IL + PARL	PLUTO-PARL	98.12 +2.22	92.55 +0.07	87.46 +1.18	88.06 +1.22	82.09 +4.27
	Diffusion Planner-PARL	97.06 +0.37	95.96 +2.21	92.32 +4.18	91.54 +3.30	82.72 +4.78
	FlowDrive-PARL	96.32 +1.10	95.96 +0.74	91.39 -0.58	90.26 +1.84	80.88 +3.31

The qualitative observations are consistent with the improvements in the Collision and TTC subscores reported in Table 2. Rather than simply following the imitation-derived trajectory, the post-trained planners exhibit earlier speed adaptation, larger separation from nearby agents, and more appropriate lateral path adjustments when conflicts arise. Together with the quantitative results, these cases suggest that metric-based RL post-training can refine the interaction behavior of heterogeneous IL planners without changing their underlying architectures or trajectory parameterizations. Additional qualitative comparisons on further interaction-heavy scenarios and ablation studies of individual PARL components are provided in Appendix B.

6 Conclusion

We presented PARL, a planner-agnostic reinforcement learning post-training framework for end-to-end autonomous driving. Across six representative IL planners spanning regression, diffusion, and flow-matching parameterizations, PARL consistently improves closed-loop performance in all six evaluation settings on Val14, Test14-hard, and Test14 under both non-reactive and reactive traffic, with the largest gains observed in interaction-heavy and long-tail scenarios. These results demonstrate that reward-driven refinement can improve heterogeneous planners without planner-specific RL co-design, narrowing the gap between imitation learning and closed-loop driving performance. PARL is enabled by a training-level factorization that isolates planner-specific behavior in a lightweight log-probability adapter while sharing the remaining policy-improvement pipeline, including trajectory sampling, metric-based reward computation, KL anchoring, advantage normalization, and optimization. This factorization provides a unified RL post-training recipe across substantially different trajectory-parameterization families. More broadly, PARL offers a practical approach to transforming imitation-based driving policies into reward-aligned and interaction-aware agents without redesigning the underlying planner. Future work will extend this recipe to hybrid planners that combine multiple generation mechanisms, richer reward formulations for fine-grained safety and comfort preferences, and broader planner classes such as autoregressive and vision-language-conditioned trajectory generators.

AI USE STATEMENT

In this work, we used generative AI tools for English-language editing, LaTeX formatting, citation-key conversion, and some coding tasks. We have not used generative AI tools to generate claims, design or conduct experiments, generate or process experimental data, or analyze and interpret the reported results. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 3 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Robotics: Science and Systems*, 2019.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 66–75, 2020.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024. doi: 10.1109/TPAMI.2024.3435937.

Jie Cheng, Yingbing Chen, and Qifeng Chen. Pluto: Pushing the limit of imitation learning-based planning for autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024a.

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

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Proceedings of the Conference on Robot Learning (CoRL)*, Proceedings of Machine Learning Research, pp. 1268–1281. PMLR, 2023.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16, 2017.

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 (CVPR)*, 2023.

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 (ICCV)*, pp. 8340–8350, 2023.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 629–636. IEEE, 2024.

Wenlong Ke, Weiliang Lu, Yang Xiao, Liang Hu, Xiang Li, and Yong Zhang. Diamond: Diffusion-augmented imitation learning for motion planning in autonomous driving. In *2026 5th Conference on Fully Actuated System Theory and Applications (FASTA)*, pp. 1–8. IEEE, 2026.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047, 2025.

Dean A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, volume 1, 1988.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 627–635, 2011.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Breaking imitation bottlenecks: Reinforced diffusion powers diverse trajectory generation. *arXiv preprint arXiv:2507.04049*, 2025.

Tianyi Tan, Yinan Zheng, Ruiming Liang, Zexu Wang, Kexin Zheng, Jinliang Zheng, Jianxiong Li, Xianyuan Zhan, and Jingjing Liu. Flow matching-based autonomous driving planning with advanced interactive behavior modeling. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025.

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 (ICCV)*, pp. 25605–25615, 2025.

Lingguang Wang, Ömer Şahin Taş, Marlon Steiner, and Christoph Stiller. Flowdrive: Moderated flow matching with data balancing for trajectory planning. *arXiv preprint arXiv:2509.21961*, 2026a.

Xiyang Wang, Xinlin Wang, Tingguang Zhou, Gong Chen, Xingtai Gui, Zhi Xu, Xiaolei Wu, Feiyang Tan, Hangning Zhou, and Mu Yang. Chainflow-vla: Causal flow planning with vision-language models. *arXiv preprint arXiv:2605.23270*, 2026b.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1602–1611, 2025.

Xuemei Yao, Xiao Yang, Jianbin Sun, Liuwei Xie, Xuebin Shao, Xiyu Fang, Hang Su, and Kewei Yang. Reflexdiffusion: Reflection-enhanced trajectory planning for high-lateral-acceleration scenarios in autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18701–18709, 2026. doi: 10.1609/aaai.v40i22.38938.

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

Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-based planning for autonomous driving with flexible guidance. In *International Conference on Learning Representations (ICLR)*, 2025.

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.

A Experimental Setup Details

A.1 Benchmarks

We conduct all experiments on nuPlan (Caesar et al., 2021; Karnchanachari et al., 2024), a large-scale benchmark for closed-loop autonomous driving, which contains approximately 1,300 hours of urban driving across four cities and 75 scenario types. Following prior work (Cheng et al., 2024a; Zheng et al., 2025), all planners are trained on the standard 1M-scenario training split. Evaluation uses the official nuPlan closed-loop simulator with a 10 Hz LQR controller and a 15 s planning horizon. We consider three commonly used benchmarks: (i) Val14 (Dauner et al., 2023), a validation split containing 1,118 scenarios across 14 scenario types; (ii) Test14 (Cheng et al., 2024b), a held-out test split of representative scenarios; and (iii) Test14-hard (Cheng et al., 2024b), a challenging subset containing the worst-performing cases for the rule-based PDM planner and emphasizing interaction-heavy and long-tail scenarios. Each benchmark is evaluated in both non-reactive (NR) and reactive (R) modes, resulting in six evaluation settings. In reactive mode, surrounding vehicles are controlled by an IDM policy conditioned on the ego trajectory, thereby closing the interaction loop between the planner and surrounding traffic.

A.2 Baseline Planners

We evaluate PARL on six representative imitation-learning planners spanning three trajectory-parameterization families: regression, diffusion, and flow matching. The regression-based planners are PlanTF (Cheng et al., 2024b) and PLUTO (Cheng et al., 2024a); the diffusion-based planners are Diffusion Planner (Zheng et al., 2025), ReflexDiffusion (Yao et al., 2026), and DIAMOND (Ke et al., 2026); and the flow-matching-based planner is FlowDrive (Wang et al., 2026a). For each planner X, we denote the released IL checkpoint by X and its PARL-post-trained counterpart by X-PARL. PARL post-training preserves the original architecture and inference interface.

We compare the following baseline planners:

- *PlanTF* (Cheng et al., 2024b): a transformer-based imitation-learning planner that studies input representations and data augmentation strategies for closed-loop planning and serves as a strong pure-IL baseline.
- *PLUTO* (Cheng et al., 2024a): a query-based imitation-learning planner with a decoupled longitudinal–lateral decoder, differentiable auxiliary losses, and contrastive imitation learning; it was the first learning-based method reported to outperform the rule-based PDM planner on nuPlan.
- *Diffusion Planner* (Zheng et al., 2025): a diffusion-transformer planner that jointly models prediction and planning through trajectory score learning and employs classifier guidance for safety and comfort adjustment without rule-based refinement.

- *ReflexDiffusion* (Yao et al., 2026): an inference-time, architecture-agnostic reflection module that applies physics-aware conditional-versus-unconditional gradient adjustments during denoising to improve behavior in high-lateral-acceleration scenarios.
- *DIAMOND* (Ke et al., 2026): a diffusion-augmented imitation-learning planner that uses a diffusion model to synthesize interaction-rich closed-loop rollouts for recovery training, together with collision, off-road, and comfort auxiliary losses to mitigate causal confusion.
- *FlowDrive* (Wang et al., 2026a): a flow-matching trajectory planner that maps noise to trajectory distributions in a few integration steps, combined with cluster-based data reweighting and moderated in-the-loop guidance to improve performance in long-tailed scenarios.

A.3 Implementation Details

All PARL post-training experiments are conducted on $8 \times$ NVIDIA A800 GPUs. We select 100K scenarios from the full 1M-scenario nuPlan training split used for IL pretraining, while preserving the original scenario distribution of the full training set. Each planner is post-trained with the same PARL recipe: at each iteration, $K = 4$ candidate trajectories are sampled per scene from the current policy, scored by the PyTorch metric-based reward described in Sec. 3.2, and used to compute the group-relative advantage for policy optimization. We use the AdamW optimizer with a cosine learning-rate schedule and one epoch of warmup, a weight decay of 1×10^{-4}, and gradient clipping with a maximum norm of 5.0. The KL regularizer to the frozen IL reference and the IL-retention auxiliary loss are weighted by β and λ_{IL}, respectively, and are tuned within small ranges for each planner family (see the ablations in Appendix B.2).

The adapter-specific parameters σ_{floor} and γ for the diffusion adapter and σ_{rl} for the flow-matching adapter follow the formulations in the Methodology section. Both the DDIM and Euler–Maruyama processes use $L = 4$ steps. Model-specific capacity parameters, including learning rate, batch size, and total training epochs, follow the corresponding planner's original IL training configuration. The complete configuration is summarized in Table 3.

Table 3: Hyperparameters used in PARL.

Type	Parameter	Symbol	Value
Training	Optimizer	–	AdamW
	LR schedule	–	Cosine w/ warmup
	Warmup epochs	–	1
	Weight decay	–	1×10^{-4}
	Gradient clipping (max norm)	–	5.0
RL objective	Group size (candidates per scene)	K	4
	KL coefficient	β	per-family (see B.2)*
	IL-retention weight	λ_{IL}	per-family (see B.2)*
	Reward normalization	–	Group-relative
	KL estimator (continuous π)	–	k3 (Schulman)
	KL estimator (categorical π)	–	Exact
	KL-Δ clamp range	–	$[-5.0, 5.0]$
Adapter	DDIM / EM steps	L	4
	DDIM sigma floor (log-prob)	σ_{floor}	$0.04^{\dagger}$
	Temporal advantage discount	γ	$0.8^{\dagger}$
	SDE lift coefficient	σ_{rl}	$0.3^{\ddagger}$

* β and λ_{IL} are tuned per family; the specific values used for each family are reported in Appendix B.2, with $\beta \in \{0.001, 0.05\}$ and $\lambda_{\mathrm{IL}} \in \{0.05, 0.1, 0.5\}$ across the three families.
† Applies to the diffusion adapter only (Diffusion Planner, ReflexDiffusion, DIAMOND).
‡ Applies to the flow-matching adapter only (FlowDrive).

B Additional Experimental Results

B.1 Additional Qualitative Comparisons

Figure 4 provides additional qualitative comparisons in two interaction-heavy nuPlan scenarios. In scenario (a), the ego vehicle initiates a right turn at an intersection. All three IL baselines fail to maintain sufficient lateral clearance and collide with different obstacles: PLUTO drifts toward the adjacent right lane and collides with a vehicle (first column, first row); Diffusion Planner turns too tightly and contacts the road curb (first column, second row); and FlowDrive Planner swings too far left and collides with a vehicle on the opposite side (first column, third row). After PARL post-training, all three planners adjust their steering trajectories and maintain adequate clearance from surrounding vehicles and road boundaries, completing the turn without collision (second column). In scenario (b), the ego vehicle travels straight with multiple nearby vehicles. PLUTO progresses too slowly and is caught by a following vehicle, resulting in a rear-end collision (third column, first row). Diffusion Planner and FlowDrive Planner fail to maintain sufficient lateral clearance from a neighboring vehicle on the left and collide with it (third column, second and third rows). After PARL post-training, PLUTO-PARL maintains an appropriate longitudinal speed, while Diffusion Planner-PARL and FlowDrive Planner-PARL preserve a sufficient lateral gap from the left-side neighbor, avoiding collision in all cases (fourth column). Together with Figure 3, these examples show that PARL improves the planner's responses to diverse interaction and collision scenarios that are not directly corrected by open-loop imitation.

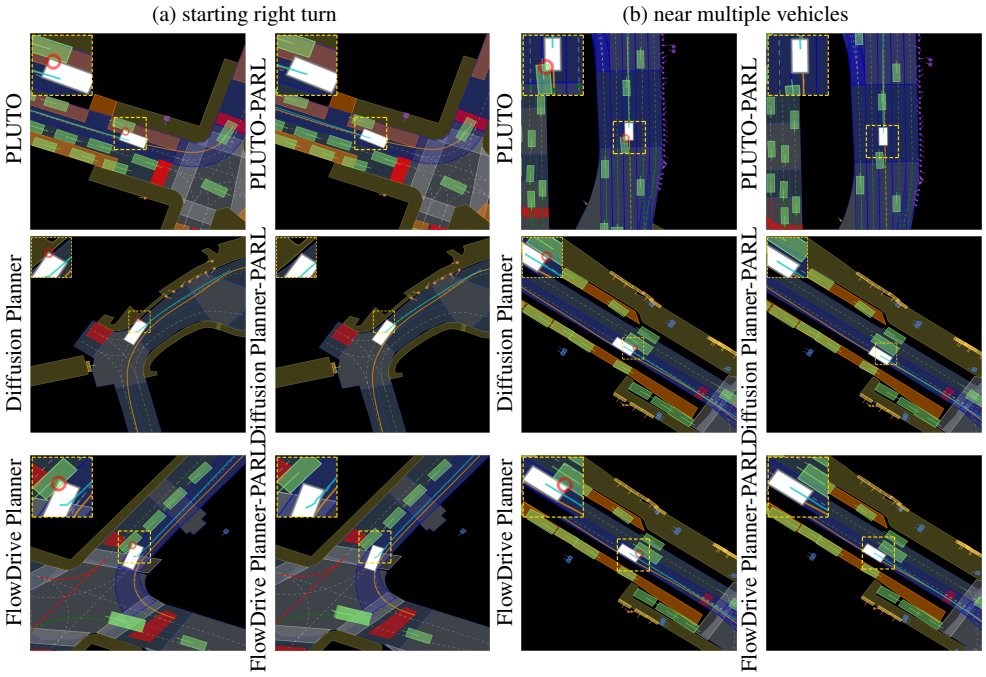

Figure 4: **Additional Interaction-Aware Closed-Loop Behaviors.** Qualitative comparison on two additional interaction-heavy nuPlan scenarios: (a) right-turning at an intersection and (b) navigating among multiple neighboring vehicles. As in Figure 3, red circles indicate collision events observed in the IL cases and avoided after PARL post-training.

B.2 Ablation Studies

To identify which design choices in PARL are essential and which are marginal refinements, we conduct a systematic ablation on one representative planner from each parameterization family: PLUTO (regression), Diffusion Planner (diffusion), and FlowDrive (flow-matching). All ablations follow the standard evaluation protocol in Appendix A on Val14, Test14-hard, and Test14 under both non-reactive (NR-CLS) and reactive (R-CLS) modes. We report the closed-loop score and its average across the six evaluation settings. The ablations cover two axes. The first examines

the shared regularization coefficients β (KL anchor) and λ_{IL} (IL-retention), reported in Tables 4, 5, and 7. The second examines family-specific adapter choices, reported in Tables 6 (diffusion) and 8 (flow matching). In each table, the last row corresponds to the full PARL configuration used in the main results. For the family-specific ablations in Tables 6 and 8, the shared regularization coefficients are fixed to the best setting from the preceding table.

Regression Family (PLUTO)

Table 4 examines the effects of the two shared regularizers, β and λ_{IL}, on PLUTO. Under the fully unregularized setting (Row 1, $\beta = 0$, $\lambda_{\text{IL}} = 0$), the metric-based RL objective alone improves PLUTO from its IL initialization of 77.55 to 78.99, a $+1.44$ gain from the categorical policy-gradient update on the mode logits. Adding the KL anchor alone (Row 2, $\beta = 0.05$) yields 78.77, which is -0.22 below the unregularized setting, whereas adding IL retention alone (Row 3, $\lambda_{\text{IL}} = 0.05$) yields 79.52, a further $+0.53$ improvement. Combining both regularizers (Row 4, the full PARL setting with $\beta = 0.05$ and $\lambda_{\text{IL}} = 0.05$) achieves 80.00, corresponding to the $+2.45$ improvement over IL reported in the main results. Thus, the KL anchor provides limited benefit in isolation but becomes beneficial when combined with IL retention, suggesting that the two regularizers are complementary for the regression family.

Table 4: Ablation on the two shared regularization coefficients β and λ_{IL} for PLUTO.

β	λ_{IL}	Val14		Test14-hard		Test14		Avg
		NR-CLS	R-CLS	NR-CLS	R-CLS	NR-CLS	R-CLS	
0	0	88.47	79.12	74.04	60.41	90.00	81.88	78.99
0.05	0	88.20	79.03	74.09	60.59	90.13	80.58	78.77
0	0.05	89.13	79.35	75.11	60.65	91.62	81.24	79.52
0.05	0.05	89.50	79.98	75.84	60.86	92.39	81.45	80.00

Diffusion Family (Diffusion Planner)

Table 5 shows that Diffusion Planner exhibits a regularization profile broadly consistent with the regression case in Table 4. Even without regularization (Row 1, $\beta = 0$, $\lambda_{\text{IL}} = 0$), PARL post-training improves Diffusion Planner from its IL initialization of 81.67 to 83.17 average score, yielding a $+1.50$ gain from the metric-based RL objective alone. This closely matches the $+1.44$ gain observed for PLUTO under the same setting, indicating that the DDIM-chain likelihood adapter can support effective policy updates without explicit regularization. The two regularizers again contribute asymmetrically: adding the KL anchor alone (Row 2) yields a marginal $+0.08$ improvement, whereas adding IL retention alone (Row 3) produces a larger $+0.63$ gain. Unlike PLUTO, for which the KL anchor alone reduces performance by -0.22, the KL term is mildly beneficial for Diffusion Planner even in isolation. Combining both regularizers (Row 4, the full PARL setting with $\beta = 0.001$ and $\lambda_{\text{IL}} = 0.5$) achieves 84.03, corresponding to a $+0.86$ improvement over the unregularized setting and a total $+2.37$ gain over the IL initialization. Consistent with the regression results, the metric-based RL objective accounts for a substantial portion of the improvement, while IL retention and KL anchoring provide complementary refinements whose relative contributions depend on the planner family.

Table 5: Ablation on the two shared regularization coefficients β and λ_{IL} for Diffusion Planner.

β	λ_{IL}	Val14		Test14-hard		Test14		Avg
		NR-CLS	R-CLS	NR-CLS	R-CLS	NR-CLS	R-CLS	
0	0	90.19	83.44	78.52	69.30	91.62	85.97	83.17
0.001	0	90.35	83.95	78.54	69.78	92.13	84.77	83.25
0	0.5	91.65	83.30	81.57	69.29	92.74	84.24	83.80
0.001	0.5	91.40	83.30	81.60	70.44	92.65	84.81	84.03

Table 6: Ablation on the two diffusion-adapter hyperparameters, sigma flooring σ_{floor} (Eq. 6) and temporal advantage discount γ (Eq. 7), for Diffusion Planner (β = 0.001, λ_{IL} = 0.5).

σ_{floor}	γ	Val14		Test14-hard		Test14		Avg
		NR-CLS	R-CLS	NR-CLS	R-CLS	NR-CLS	R-CLS	
0	1.0	91.40	83.61	80.33	69.31	91.74	84.66	83.51
0.04	1.0	90.71	83.19	79.81	69.66	91.95	84.57	83.32
0	0.8	91.13	83.30	81.71	69.49	92.21	84.41	83.71
0.04	0.8	91.40	83.30	81.60	70.44	92.65	84.81	84.03

Table 6 ablates the two diffusion-adapter hyperparameters under the best shared regularization setting identified in Table 5. All four configurations fall within a narrow 0.71-point range, from 83.32 to 84.03 average score, indicating that the DDIM-chain adapter is relatively robust to these choices once the shared regularizers are properly configured. The temporal advantage discount γ has the more consistent effect: setting $\gamma = 0.8$ improves performance over $\gamma = 1.0$ by $+0.20$ without sigma flooring (Row 3 versus Row 1) and by $+0.71$ with sigma flooring (Row 4 versus Row 2). This supports the use of temporal discounting to reduce the contribution of early, high-noise denoising steps, whose influence on the final trajectory is more diffuse. Sigma flooring, in contrast, exhibits an interaction with γ rather than a uniform effect. With $\gamma = 1.0$, enabling σ_{floor} slightly reduces performance from 83.51 to 83.32 (-0.19), whereas with $\gamma = 0.8$ it improves performance from 83.71 to 84.03 ($+0.32$). The full PARL configuration (Row 4, $\sigma_{\text{floor}} = 0.04$ and $\gamma = 0.8$) achieves the best average score of 84.03. Overall, the results suggest that the two adapter parameters are complementary: sigma flooring controls the numerical behavior of the per-step Gaussian log-density near the end of the denoising chain, while temporal discounting prioritizes later denoising steps in the policy-gradient signal.

Flow-Matching Family (FlowDrive)

Table 7 shows that the IL-retention regularizer λ_{IL} is essential for stable post-training with the flow-matching adapter. Removing λ_{IL} (Rows 1 and 2) leads to substantial degradation below the IL initialization: the average closed-loop score drops to 68.92 (Row 1, no regularization) and 67.91 (Row 2, KL anchor only), approximately 15 points below the FlowDrive IL baseline of 84.30. Moreover, adding the KL anchor without IL retention further reduces performance by 1.01 points relative to the unregularized setting, indicating that a KL-based trust region alone cannot provide sufficient stabilization for the stochastic SDE-lifted policy. In contrast, enabling λ_{IL} restores and surpasses the IL performance: Row 3 (λ_{IL} only) reaches 84.68, while Row 4 (full PARL, $\beta = 0.001$, $\lambda_{\text{IL}} = 0.1$) further improves to 85.12. Thus, IL retention serves as the dominant stabilizing component for flow-matching post-training, with the KL anchor providing an additional $+0.44$ gain once λ_{IL} is present. This stronger dependence on IL-based stabilization, compared with the diffusion adapter in Table 5, is consistent with the additional stochasticity introduced by the SDE lift and highlights the importance of retaining the pretrained behavior during flow-matching policy optimization.

Table 7: Ablation on the two shared regularization coefficients β and λ_{IL} for FlowDrive.

β	λ_{IL}	Val14		Test14-hard		Test14		Avg
		NR-CLS	R-CLS	NR-CLS	R-CLS	NR-CLS	R-CLS	
0	0	82.56	65.00	71.61	48.98	83.27	62.11	68.92
0.001	0	81.65	63.63	69.73	47.86	83.19	61.42	67.91
0	0.1	91.82	85.46	79.16	72.65	92.19	86.77	84.68
0.001	0.1	91.87	85.61	79.42	73.50	92.29	88.00	85.12

Table 8 ablates the two flow-matching adapter design choices under the best regularization setting identified in Table 7. All four configurations remain within a narrow 0.75-point range (84.37 to 85.12 average score), in marked contrast to the shared-regularization ablation, where removing λ_{IL} led to substantial performance degradation. Disabling the replay-implied-noise reparameterization

Table 8: Ablation of the two flow-matching-adapter design choices for FlowDrive: the SDE lift coefficient σ_{rl} (Eq. 8) and the replay-implied-noise reparameterization (Eq. 10). The KL and IL-retention coefficients are fixed at $\beta = 0.001$ and $\lambda_{\text{IL}} = 0.1$.

σ_{rl}	implied-noise	Val14		Test14-hard		Test14		Avg
		NR-CLS	R-CLS	NR-CLS	R-CLS	NR-CLS	R-CLS	
0.3	off	91.60	85.13	79.03	72.74	92.66	86.86	84.67
0.1	on	91.45	85.44	77.74	73.59	91.32	88.17	84.62
1.0	on	91.11	84.94	78.37	72.89	91.49	87.40	84.37
0.3	on	91.87	85.61	79.42	73.50	92.29	88.00	85.12

(Row 1) at $\sigma_{\text{rl}} = 0.3$ reduces performance by 0.45, consistent with our formulation: directly reusing the noise sampled during rollout removes the dependence of the evaluated log-probability on the current policy parameters, resulting in a vanishing policy-gradient signal through the velocity network. Recomputing the implied noise according to Eq. (10) restores this parameter dependence and yields a nonzero gradient. The SDE lift coefficient exhibits a clear optimum at $\sigma_{\text{rl}} = 0.3$ (Row 4), achieving 85.12 average score, compared with 84.62 at $\sigma_{\text{rl}} = 0.1$ (Row 2) and 84.37 at $\sigma_{\text{rl}} = 1.0$ (Row 3). This trend reflects the trade-off between exploration and gradient stability: a smaller diffusion coefficient provides limited stochastic exploration, whereas excessive diffusion introduces larger perturbations into the sampled trajectories and weakens the reward-aligned learning signal. Overall, the replay-implied-noise reparameterization and SDE lift coefficient provide complementary fine-grained refinements, while the dominant stability requirement for flow-matching post-training remains the IL-retention regularization identified in Table 7.