PREFTO: REINFORCEMENT FINE-TUNING OF DIFFUSION PLANNERS WITH LOCAL EXECUTION FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Imitation-trained diffusion planners generate long reference trajectories, but deployed vehicles execute only a short segment before replanning. Reinforcement fine-tuning therefore needs feedback that connects generated plans to their long-term driving consequences. Scoring only a short execution can miss delayed hazards, whereas following an entire plan can penalize continuations that deployment would replace. Full rollouts with repeated replanning are costly, and local scores must also distinguish candidates under realistic execution conditions. We study how finite local execution feedback can improve a replanning planner while retaining its pretrained trajectory generator. Under a safety-preserving continuation model with a common expected quality decrement, our analysis identifies group-centered local scores as relative advantages with an additional safety-deficit penalty. A separate sufficient condition connects score gains to deployment improvement through candidate-dependent score–value discrepancy and execution discrepancy. We introduce Prefix-Rewarded Full-Trajectory Optimization (PreFTO). It compares plans from shared policy-visited states through surrogate controller execution, uses continuous safety margins with minimum aggregation, and applies group-relative feedback to the complete original plans with Diffusion-NFT. This separates the feedback window from both the generated trajectory and the deployment replanning interval. On reactive nuPlan, PreFTO improves the pretrained Diffusion Planner on all three tested splits, including a 4.58-point gain on Test14-hard. Ablations demonstrate that proper choices of feedback length, learning targets, and loss coverage can effectively improve the closed-loop behavior of an RL post-trained planner.

1 INTRODUCTION

Imitation learning trains diffusion planners to generate trajectories from demonstrations, capturing multiple plausible behaviors and dependencies across future actions (Janner et al., 2022; Chi et al., 2025; Zheng et al., 2025b). A long prediction provides context for the intent and motion of a maneuver. During execution, on the other hand, a planner usually replans from new observations before the full long prediction is executed. Long prediction and short execution thus serve different purposes, allowing structured trajectory generation together with responsive control.

Reinforcement learning can further improve an imitation-trained planner using feedback from execution. This introduces a different requirement: estimating the long-term consequences of a decision so that the policy can favor better plans. The generated reference, however, is transformed by controller tracking and environmental responses, and its remaining trajectory may be replaced at the next replanning step. The feedback must therefore connect a generated plan to the behavior it induces. Here, *evaluation* estimates an action chunk's long-term return, including consequences after the chunk. Its horizon specifies the duration of the action being valued.

Scoring only the interval before the next replanning step is a natural starting point, but may miss delayed consequences of the current decision. With a learned critic, short rollouts leave more of the return to a terminal-value estimate; with local safety scores alone, an apparently safe segment may already lead toward a later hazard. Looking farther ahead can reveal such consequences, motivating longer action chunks for value learning (Li et al., 2025b;a). Yet following the complete predicted plan introduces another difficulty: a later failure may result from a continuation that de-

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.

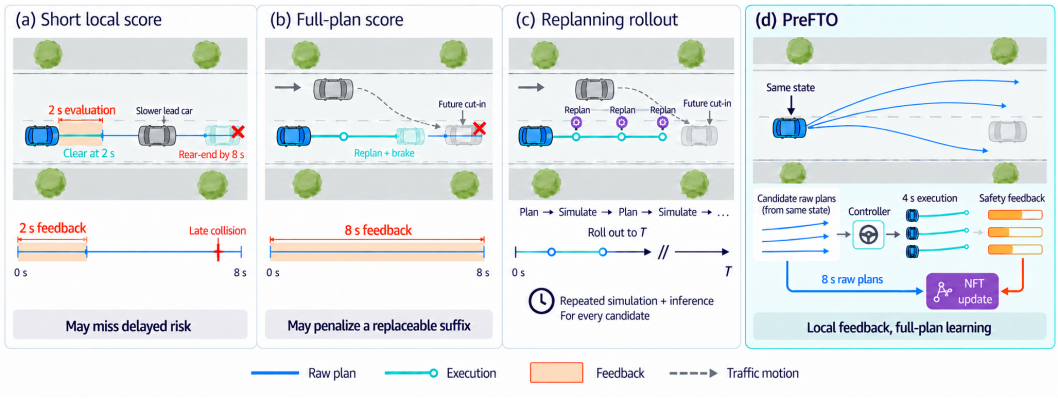

Figure 1: **Feedback choices for a replanning planner.** **(a)** Short local scoring may miss delayed hazards. **(b)** Scoring the full original plan may penalize a suffix that deployment replanning could correct. **(c)** Replanning rollouts repeatedly alternate simulation and planner inference. **(d)** PreFTO combines local execution feedback from shared-state candidates with full-plan learning.

ployment replanning would have corrected. Extending raw-plan following can therefore reveal more outcomes while introducing a discrepancy from deployment replanning. Continuing each candidate with replanning gives a more direct estimate of deployment consequences, but requires additional simulation and planner inference. As illustrated in Figure 1, useful feedback must balance delayed consequences, deployment relevance, and cost.

Local feedback also depends on where and how candidates are compared. Shared-state comparisons separate candidate quality from scene difficulty, and controller execution exposes tracking effects. Continuous safety margins distinguish collision-free candidates, while minimum aggregation retains brief hazards that averaging can conceal.

Our central question is *how finite local execution feedback can improve a replanning diffusion planner while adapting its pretrained trajectory generator*. Under a safety-preserving continuation model with a common expected quality decrement, we analyze that group-centered local scores equal relative continuation advantages with an additional safety-deficit penalty. A separate bound connects score gains to deployment improvement through candidate-dependent score–value and execution discrepancies.

Building on these analyses, we introduce PreFTO, Prefix-Rewarded Full-Trajectory Optimization, for an imitation-trained Diffusion Planner. It compares candidates from shared policy-visited states through controller execution without further planner calls. By using continuous safety scores, PreFTO is able to learn from the fine-grained differences in safety among these candidates. Group-relative feedback guides DiffusionNFT updates of the complete original plans(Zheng et al., 2025a): execution supplies the score, and the raw plan supplies the target.

On reactive nuPlan, PreFTO improves the pretrained baseline on all three tested splits, including a 4.58-point gain on Test14-hard. Under the official-model protocol, shorter scoring outperforms full-duration scoring, and original-plan targets outperform executed-motion targets. The benefit of full loss coverage varies across splits, supporting separate evaluation of these choices.

Our contributions are:

1. **Horizon separation and value analysis.** We distinguish prediction, evaluation, and execution horizons while treating loss coverage separately. Under explicit continuation assumptions, we interpret group-centered local scores as relative advantages with an additional safety preference. We further derive a sufficient condition for deployment improvement that accounts for score–value and execution discrepancies.
2. **Shared-state candidate comparison.** We compare candidate plans from the same policy-visited scene and time through controller execution in a surrogate simulator. Following each raw plan over a common local window makes the compared decision explicit and avoids subsequent planner calls during scoring.

3. **Continuous safety rewards with minimum aggregation.** We design continuous safety scores from road-boundary margins, collision clearance, and time to collision. Taking the minimum across time and safety dimensions retains brief hazards and low safety margins that averaging can obscure. This safety score is multiplied by driving quality to form the local training reward.

2 Related Work

Generative imitation learning and reinforcement fine-tuning. Diffuser, Diffusion Policy, and Diffusion Planner model action or trajectory distributions (Janner et al., 2022; Chi et al., 2025; Zheng et al., 2025b); Flow Planner and FlowDrive use flow matching (Tan et al., 2025; Wang et al., 2025), and Diffusion-ES searches generated trajectories using rewards (Yang et al., 2024). For reinforcement fine-tuning, DDPO and DPPO optimize denoising transitions (Black et al., 2023; Ren et al., 2024), whereas DiffusionNFT uses a negative-aware forward-process objective (Zheng et al., 2025a). We adopt the latter to learn complete reference plans from partial execution feedback while retaining the pretrained output length.

Action chunking, replanning, and value estimation. Model predictive control and action-chunking policies distinguish prediction from commitment (Rawlings et al., 2026; Chi et al., 2025). Q-chunking uses sequence-conditioned critics (Li et al., 2025b); Decoupled Q-Chunking separates critic and policy chunk lengths and uses an optimistic partial critic to extract shorter actions (Li et al., 2025a). Multi-step estimators and terminal-value planning account for future value beyond finite rollouts (Schulman et al., 2015; Sun et al., 2018). We distinguish the action duration being valued from the task objective and generation length, and analyze how rule-based local scores relate to chunk advantages and deployment improvement when tracking extends beyond the next replan (Section 4).

Driving feedback, candidate groups, and safety rewards. Gen-Drive fine-tunes generative driving planners (Huang et al., 2024); Plan-R1 applies group-relative optimization to motion tokens using simulated safety and feasibility feedback (Tang et al., 2025). PlannerRFT combines closed-loop state collection, controller-executed candidate groups, adaptive exploration, and diffusion-policy optimization, and compares 2-, 4-, and 6-second reward windows (Li et al., 2026). These precedents motivate our analysis of local feedback against a fixed deployment value and its use in full raw-trajectory NFT updates, together with continuous safety scores and minimum aggregation across time and dimensions. Our reactive-IDM candidate rollouts differ from PlannerRFT's log-replay training traffic; this protocol difference alone does not establish an algorithmic advantage.

3 MDP Formulation and Chunk Policies

Decision process and objective. Consider a finite-horizon MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0, T)$ with bounded rewards. At physical step k, an action a_k induces $s_{k+1} \sim P(\cdot \mid s_k, a_k)$ and reward $r_k = r(s_k, a_k, s_{k+1})$. Time is included in the state. Starting from ρ_0, a policy π maximizes

$$J(\pi) = \mathbb{E}_{\rho_0,\pi,P}\left[\sum_{k=0}^{T-1} r_k\right], \tag{1}$$

with $V^\pi(s)$ denoting its expected remaining return. All horizon comparisons retain this objective and the same terminal time T.

The trajectory scores considered in this paper are represented by a terminal reward: for a bounded functional $F(\tau_{0:T})$, set $r_k = 0$ for $k < T-1$ and $r_{T-1} = F(\tau_{0:T})$, so that $J(\pi) = \mathbb{E}_\pi[F(\tau_{0:T})]$. Augmenting the state with the history or metric summaries needed to compute F makes this representation Markov. It covers non-additive safety and quality scores; the relation between a finite local score and the resulting value is developed in Section 4.

Generation and deployment. At a planning decision, $p_\pi(\cdot \mid s)$ denotes the conditional distribution of complete plans $X = (a_0, \ldots, a_{H_p-1})$ generated by policy π at state s, including its

sampling procedure. The deployed policy samples X, executes $X_{[:H_a]}$, observes the next state, and replans. Candidate evaluation scores a fixed plan at a given state using a learned value function or another evaluator, without further planner calls. Prediction length H_p specifies the generated sequence, execution length H_a specifies the interval between planning decisions, and evaluation length H_r specifies the action chunk being valued. We mainly study $H_a \le H_r \le H_p$. Loss coverage $H_g \le H_p$ separately specifies which predicted coordinates receive the reinforcement objective. Horizons count physical steps, indexed by k; diffusion time is denoted by t. For reference trajectories, $\mathrm{execute}_{H_a}(X)$ denotes following the complete reference for H_a steps through a controller, as detailed in Section 5.

Chunk values. For $u = X_{[:H_r]}$, execute the prescribed chunk and then resume π. Its value is

$$Q^{\pi}_{H_r}(s,u) = \mathbb{E}\left[\sum_{j=0}^{H_r-1} r_{k+j} + V^{\pi}(s_{k+H_r}) \,\middle|\, s_k = s, \mathrm{execute}(u)\right]. \tag{2}$$

The reference continuation specifies the value target. A learned value function may estimate its contribution without explicitly simulating subsequent replanning. The continuation replans at the chunk endpoint and every H_a steps thereafter. Chunks are truncated at T, where $V^{\pi}(s_T) \equiv 0$. Thus H_r specifies the duration of the action being valued, while the value of $Q^{\pi}_{H_r}$ includes subsequent consequences or, when implemented with a practical estimator, an estimation of later consequences. This is the sequence-conditioned value perspective of chunked RL (Li et al., 2025b;a). For a controller-followed reference, $Q^{\pi}_{H_r}(s,X)$ uses $\mathrm{execute}_{H_r}(X)$ in the same definition, retaining the complete reference as the controller input (Appendix A). Figure 2 contrasts this value with local scoring and deployment replanning.

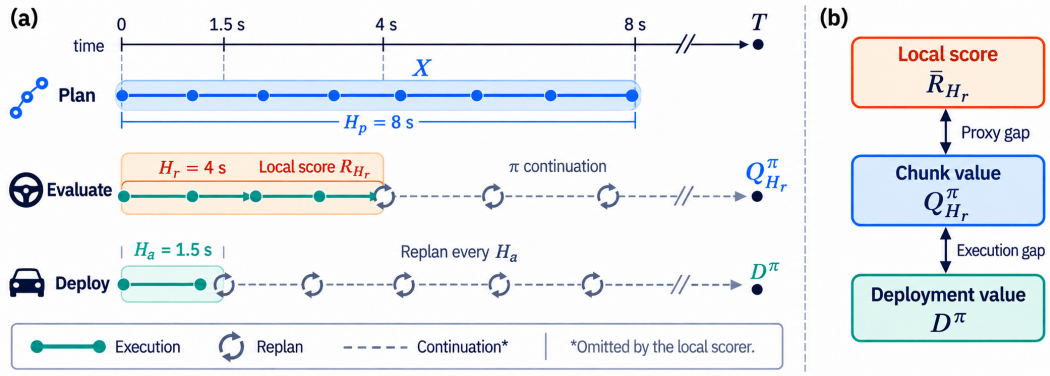

Figure 2: **Local scores, chunk values, and deployment values. (a)** Candidate scoring follows the reference X for H_r, whereas deployment replans after H_a. **(b)** The comparison separates score–value and execution discrepancies. Both $Q^{\pi}_{H_r}(s,X)$ and $D^{\pi}(s,X)$ include reference-policy continuation to T; dashed continuations are omitted by PreFTO's local scorer.

4 From Local Execution Feedback to Policy Improvement

We first characterize the relative value signal supplied by local scores, then identify when improving that signal supports the common deployment objective in Equation 1.

4.1 Chunk Values and Relative Advantages

For a reference policy π, let $b(\cdot \mid s)$ be the distribution used to sample a candidate group at state s. Define

$$A^{\mathrm{rel}}_{b}(s,X) = Q^{\pi}_{H_r}(s,X) - \mathbb{E}_{X' \sim b(\cdot|s)}[Q^{\pi}_{H_r}(s,X')]. \tag{3}$$

This relative chunk advantage compares candidates followed by the same reference policy. Its baseline equals $V^{\pi}(s)$ when $b(\cdot \mid s) = p_{\pi}(\cdot \mid s)$ and $H_r = H_a$; otherwise it is the mean evaluated-chunk value. Below, we distinguish this reference continuation from an optimistic model used to interpret the training signal.

4.2 Local Scores as Safety-Penalized Relative Advantages

Estimating chunk value by a complete Monte Carlo rollout requires simulating the specified continuation to the common endpoint. In order to avoid the high cost of repeated simulation rollouts, we study when and how local scores can serve as relative-value supervision under a continuation model.

At state s of physical time k, let $\ell = \min(k + H_r, T)$ and write $\tau_{a:b}$ for the trajectory segment from time a to b. Consider

$$F(\tau_{0:T}) = \sigma(\tau_{0:T})q(\tau_{0:T}), \qquad R_{H_r}(s, X) = \sigma(\tau_{k:\ell})q(\tau_{k:\ell}).$$

Here $\tau_{0:T}$ includes the history represented by s, candidate execution $\tau_{k:\ell}$, and subsequent continuation. Safety function $\sigma(\tau)$ and quality function $q(\tau)$ lie in $[0, 1]$. q may combine comfort, speed compliance, and progress (Section 5). Since state transitions and continuation policies may be random, the score $R_{H_r}(s, X)$ is a random variable; $\bar{R}_{H_r}(s, X)$ and $\bar{\sigma}_{H_r}(s, X)$ are the expectations of score and safety conditional on s and $\mathrm{execute}_{H_r}(X)$.

Proposition 1 (Relative advantage and safety preference)**.** *Suppose every candidate in the comparison support admits a nonempty class $\mathcal{C}(s, X)$ of continuation policies satisfying, for every $\nu \in \mathcal{C}(s, X)$,*

$$\sigma(\tau_{0:T}) = \sigma(\tau_{k:\ell}) \quad a.s., \qquad \mathbb{E}_\nu[q(\tau_{0:T}) \mid \mathcal{H}_\ell] = q(\tau_{k:\ell}) - \lambda(s), \tag{4}$$

where $\mathcal{H}_\ell$ is the endpoint history, $\lambda(s) \geq 0$ is a function of state and thus is common to the candidate actions, and $q(\tau_{k:\ell}) - \lambda(s) \in [0, 1]$ a.s. The value within this safety-preserving class is

$$\begin{aligned} Q^{\mathrm{opt}}_{H_r}(s, X) &:= \sup_{\nu \in \mathcal{C}(s,X)} \mathbb{E}_\nu[F(\tau_{0:T}) \mid s, \mathrm{execute}_{H_r}(X)] \\ &= \bar{R}_{H_r}(s, X) - \lambda(s)\bar{\sigma}_{H_r}(s, X). \end{aligned} \tag{5}$$

Define $A^{\mathrm{opt}}_b(s, X)$ by centering $Q^{\mathrm{opt}}_{H_r}(s, X)$ under $b(\cdot \mid s)$, as in Equation 3. Then

$$\begin{aligned} &\bar{R}_{H_r}(s, X) - \mathbb{E}_{X' \sim b(\cdot|s)}[\bar{R}_{H_r}(s, X')] \\ &\quad = A^{\mathrm{opt}}_b(s, X) + \lambda(s)\big(\bar{\sigma}_{H_r}(s, X) - \mathbb{E}_{X' \sim b(\cdot|s)}[\bar{\sigma}_{H_r}(s, X')]\big). \end{aligned} \tag{6}$$

Thus centered scores supply relative advantage under the model minus $\lambda(s)$ times the centered safety deficit $1 - \bar{\sigma}_{H_r}(s, X)$, giving training an additional safety preference. Equal expected safety cancels this correction. Binary deficits measure violation probability; continuous deficits also capture reduced margins. Appendix B gives the proof and finite-group relation.

4.3 From Local Feedback to Deployment Improvement

To connect local feedback to deployment return, let π and π_{new} denote the reference and updated deployment policies, with conditional plan distributions $p_\pi(\cdot \mid s)$ and $p_{\pi_{\mathrm{new}}}(\cdot \mid s)$, respectively. Both use the same execution interval H_a and controller. Candidate plans from either distribution are valued using the common reference continuation π:

$$\begin{aligned} Q^\pi_{H_r}(s, X) &\quad \text{execute } X \text{ for } H_r \text{ steps, then continue with } \pi, \\ D^\pi(s, X) &\quad \text{execute } X \text{ for } H_a \text{ steps, then continue with } \pi. \end{aligned}$$

Both use the same task score $F(\tau_{0:T})$ and terminal time T. The deployment chunk value $D^\pi(s, X)$ satisfies $\mathbb{E}_{X \sim p_\pi(\cdot|s)}[D^\pi(s, X)] = V^\pi(s)$.

Define two comparison discrepancies:

$$\begin{aligned} \epsilon_{\mathrm{proxy}}(s) &= \mathrm{span}_X[Q^\pi_{H_r}(s, X) - \bar{R}_{H_r}(s, X)], \\ \epsilon_{\mathrm{exec}}(s) &= \mathrm{span}_X[Q^\pi_{H_r}(s, X) - D^\pi(s, X)]. \end{aligned} \tag{7}$$

Here span_X is the supremum minus the infimum over the union of the supports of a probability measure. The first term measures candidate-dependent differences between local scores and value under the reference continuation. The second measures candidate-dependent differences between evaluation and deployment execution. Let $I_{\mathrm{score}}(s)$ and $I_{\mathrm{dep}}(s)$ be the new-minus-reference gains in expected score and deployment value:

$$\begin{aligned} I_{\mathrm{score}}(s) &= \mathbb{E}_{X \sim p_{\pi_{\mathrm{new}}}(\cdot|s)}[\bar{R}_{H_r}(s, X)] - \mathbb{E}_{X \sim p_\pi(\cdot|s)}[\bar{R}_{H_r}(s, X)], \\ I_{\mathrm{dep}}(s) &= \mathbb{E}_{X \sim p_{\pi_{\mathrm{new}}}(\cdot|s)}[D^\pi(s, X)] - \mathbb{E}_{X \sim p_\pi(\cdot|s)}[D^\pi(s, X)]. \end{aligned}$$

Proposition 2 (Local feedback and deployment improvement)**.** *For finite discrepancy spans and the common reference continuation π,*

$$I_{\mathrm{dep}}(s) \geq I_{\mathrm{score}}(s) - \mathrm{TV}\big(p_{\pi_{\mathrm{new}}}(\cdot \mid s), p_{\pi}(\cdot \mid s)\big)\big(\epsilon_{\mathrm{proxy}}(s) + \epsilon_{\mathrm{exec}}(s)\big), \tag{8}$$

where TV *is half the L^1 distance between probability measures.*

The finite-horizon performance-difference identity sums $I_{\mathrm{dep}}(s)$ over planning states visited by π_{new}; a positive expected sum of these lower bounds implies $J(\pi_{\mathrm{new}}) > J(\pi)$ (Appendix A).

Proposition 1 explains the local score signal's structure and analyzes why it is a feasible replacement of actual advantage. Proposition 2 then connects this signal to policy improvement by showing that as long as the improvement in local score exceeds a threshold dependent on policy and execution discrepancies, gain in closed-loop return is guaranteed.

5 PreFTO: Execution Feedback and Full-Trajectory Fine-Tuning

PreFTO retains Diffusion Planner's 8-second prediction (H_p), evaluates 4 seconds of candidate execution (H_r), replans every 1.5 seconds (H_a), and applies NFT over the full 8-second output (H_g). Figure 3 summarizes the collection, scoring, and learning paths.

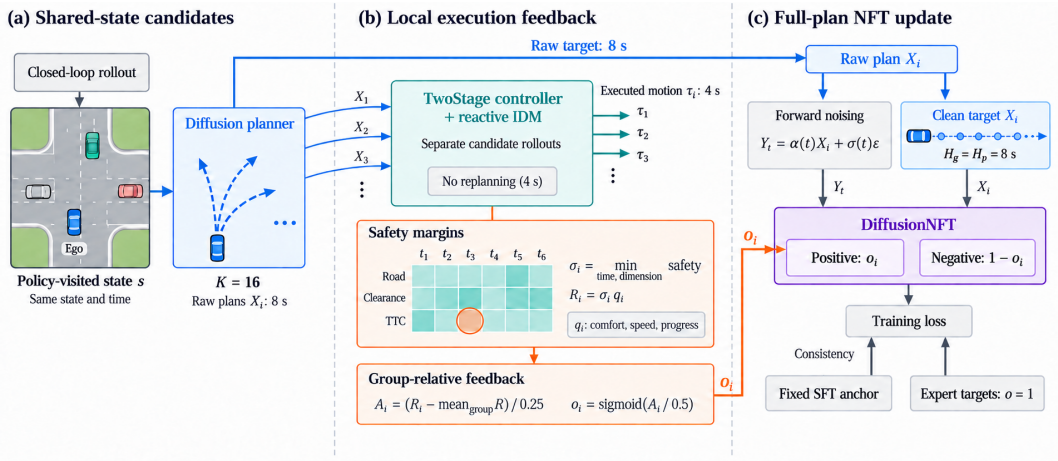

Figure 3: **PreFTO overview. (a)** Closed-loop rollouts supply policy-visited states for shared-state candidate generation. **(b)** Each candidate is followed for H_r without replanning, and safety–quality scores are compared within the group. **(c)** The resulting weights guide negative-aware NFT updates using the original complete plans as targets.

5.1 Comparing Candidates at Policy-Visited States

We initialize from the official imitation-trained Diffusion Planner (Zheng et al., 2025b). The current behavior policy collects decision states through closed-loop rollout. At each captured scene and time, an exploration sampler defines $b(\cdot \mid s)$ and draws 16 complete references with shared model conditioning. Each candidate starts from the same saved ego state and reconstructed context. A collection cycle covers 1,088 scenarios with five captured states sampled from the episode in each scenario, giving 5,440 groups and 87,040 policy candidates before filtering. Groups remain separate across states and collection cycles.

PreFTO implements candidate evaluation through controller execution and rule-based local scoring. The TwoStage controller follows each complete reference X_i for H_r while surrounding vehicles respond using reactive IDM. Low-level tracking remains active, and no new plan is generated within the scoring window. The controller can read waypoints beyond the executed segment, so $\mathrm{execute}_{H_r}(X_i)$ retains the complete reference. Deployment instead invokes $\mathrm{execute}_{H_a}(X_i)$ before replanning, giving the execution discrepancy analyzed in Section 4.3.

We retain (s, X_i, R_i). Let τ_i be the realized execution segment from time k of state s to $\ell = \min(k+H_r, T)$; its score R_i is a realization of $R_{H_r}(s, X_i)$. The raw reference supplies the learning target; executed targets instead fit a controller-transformed distribution (Appendix C.3), tested in Section 6.3. Appendix D specifies capture times, sampler settings, and reference refresh.

5.2 Continuous Safety Scores and Minimum Aggregation

To distinguish locally safe candidates with different risk margins, we map road-boundary margin, collision clearance, and time-to-collision (TTC) at physical step j to scores $s_{\mathrm{road,i}}(j)$, $s_{\mathrm{col,i}}(j)$, and $s_{\mathrm{TTC,i}}(j)$ in $[0, 1]$. Hard events retain explicit handling, including zero collision safety for an at-fault collision. We aggregate by the minimum across executed steps and safety dimensions:

$$\sigma(\tau_i) = \min_{k<j\le\ell} \min\{s_{\mathrm{road,i}}(j), s_{\mathrm{col,i}}(j), s_{\mathrm{TTC,i}}(j)\}. \tag{9}$$

The time minimum retains brief hazards, while the dimension minimum preserves a low margin on any safety axis. Safety then modulates quality:

$$q(\tau_i) = \frac{2c_i + 2v_i + p_i}{5}, \qquad R_i = \sigma(\tau_i) q(\tau_i). \tag{10}$$

Here c_i and v_i are mean comfort and speed compliance over τ_i, and p_i is its clipped progress relative to the expert route. TTC appears only in safety. Section 4.2 interprets this score through a safety-preserving continuation model; Appendix D specifies its thresholds and normalization.

5.3 Group-Relative Feedback for Full-Plan NFT

Within each captured state, let $\overline{R}_g$ be the mean policy-candidate score; groups with zero score variance are discarded. We form a group-relative surrogate advantage and an optimality weight:

$$A_i = \frac{R_i - \overline{R}_g}{0.25}, \qquad o_i = \mathrm{sigmoid}(A_i/0.5). \tag{11}$$

Under the continuation model, centering estimates relative advantage with a safety-deficit penalty (Equation 6); Appendix B accounts for finite groups and execution noise. The exploration distribution $b(\cdot \mid s)$ defines the baseline. The fixed scale preserves score magnitudes across groups, and the sigmoid maps the signal to branch weights. No separate safety penalty is added.

We use DiffusionNFT's negative-aware forward-process objective (Zheng et al., 2025a). For sample i, let Y_{i,t_i} be the noisy raw target at diffusion time t_i, conditioned on state s_i. Trainable and old predictors receive the same inputs. The old reference is fixed within an update cycle; historical samples use their collection reference. For generic inputs (Y_t, s, t), define

$$f_\theta^{\pm}(Y_t, s, t) = f_{\mathrm{old}}(Y_t, s, t) \pm \beta\big(f_\theta(Y_t, s, t) - f_{\mathrm{old}}(Y_t, s, t)\big). \tag{12}$$

With $\beta = 0.1$ and $H_g = H_p$, the core loss is

$$\ell_i = o_i \| f_\theta^+(Y_{i,t_i}, s_i, t_i) - X_i \|_{H_g}^2 + (1 - o_i) \| f_\theta^-(Y_{i,t_i}, s_i, t_i) - X_i \|_{H_g}^2. \tag{13}$$

The squared norm sums normalized ego-coordinate residuals and averages over the covered steps. Thus 4-second execution feedback weights an 8-second learning target, with gradients through the predictor.

Each microbatch $\mathcal{B}$ contains 24 policy samples and eight expert examples. Experts supply complete demonstration targets with $o_i = 1$ outside the candidate group statistics. For policy samples, let $m_i = \mathrm{clip}(|A_i|, 1, 5)$; experts use $m_i = 1$. Normalize $w_i = m_i / (|\mathcal{B}|^{-1} \sum_{j\in\mathcal{B}} m_j + 10^{-8})$. The complete loss is

$$\mathcal{L}_{\text{PreFTO}} = \frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \left[\frac{8 w_i}{d_i} \ell_i + \frac{0.25}{d_i} \| f_\theta(Y_{i,t_i}, s_i, t_i) - f_{\mathrm{SFT}}(Y_{i,t_i}, s_i, t_i) \|_{H_p}^2 \right]. \tag{14}$$

Here f_{SFT} is the fixed pretrained predictor on the same noisy input, and d_i is the detached mean absolute prediction residual over ego and neighbor slots, floored at 10^{-3}. Only ego regression is optimized. The anchor shares d_i but receives neither the magnitude weight nor the NFT multiplier.

Table 1: NuPlan closed-loop scores (0–100). NR/R denotes non-reactive/reactive traffic; the ego planner replans in both modes. Published baseline protocols are retained; the last two rows are evaluated locally. Bold and underline mark the two highest reported learning-based scores per column. Dashes indicate unavailable or split-unverified results.

Method	Val14		Random		Hard	
	NR	R	NR	R	NR	R
Rule-based						
IDM	75.60	77.33	70.39	74.42	56.15	62.26
PDM-Closed	92.84	92.12	90.05	91.63	65.08	75.19
Learning-based						
PDM-Open	53.53	54.24	52.81	57.23	33.51	35.83
GameFormer (w/o refinement)	13.32	8.69	11.36	9.31	7.08	6.69
PlanTF	84.27	76.95	85.62	79.58	69.70	61.61
PLUTO (w/o refinement)	88.89	78.11	89.90	78.62	70.03	59.74
G2DP (Yu et al., 2026)	**92.26**	83.91	**91.58**	85.26	**80.05**	**73.91**
DFP (Zhang et al., 2026)	<u>90.33</u>	79.97	90.69	81.96	76.91	63.56
KnowDiffuser (Ding et al., 2026)	87.50	81.25	—	—	—	—
PlannerRFT (Li et al., 2026)	89.96	**84.46**	<u>90.76</u>	**85.80**	<u>77.16</u>	72.21
M-Diffusion Planner (Aggressive)	82.63	75.11	—	—	68.21	63.83
M-Diffusion Planner (Conservative)	85.51	78.69	—	—	69.72	65.71
M-Diffusion Planner (Comfortable)	88.72	83.31	—	—	70.35	67.83
Diffusion Planner (local IL)	88.55	83.40	84.77	84.77	74.28	68.21
PreFTO	89.57	<u>84.27</u>	86.14	<u>85.38</u>	75.38	<u>72.79</u>

Complete-plan denoising lets local preferences update the joint trajectory distribution. Appendix C derives this connection and shows that NFT loss masking changes both reward-directed and old-reference quadratic terms. Full loss coverage is assessed through the corresponding ablation. Every six epochs, the behavior reference is refreshed by averaging its weights equally with the trainable model. The second cycle mixes fresh and historical data with their matched references. Deployment uses a separate EMA with decay 0.99 after each optimizer step. Appendix D gives the full training and replay specification.

6 Experiments

6.1 Protocol and Overall Closed-Loop Performance

We evaluate on reactive and non-reactive nuPlan with TwoStage tracking (Caesar et al., 2021): Val14 (1,118 scenarios), Test14-random (261), and Test14-hard (272). Both modes use the same IL and PreFTO checkpoints and settings, changing only surrounding-agent behavior. All subsequent ablations use reactive traffic. Unless varied, $H_p = H_g = 8$ s, $H_r = 4$ s, and $H_a = 1.5$ s; loss coverage denotes the reward-directed term. Training uses 1,088 scenarios; deployment draws one DPM proposal at temperature 0.5. Appendices D and E give the training recipe and result aggregation. Bold and underlined values denote best and second-best results within each comparison, with ties marked alike.

Table 1 includes PlannerRFT and M-Diffusion Planner (Ding et al., 2025). PreFTO improves all splits over our IL baseline in both modes, gaining 4.58 on reactive Hard. Non-reactive gains are 1.02, 1.37, and 1.10 on Val14, Random, and Hard, respectively. Appendices E.3 and E.2 provide reactive components and examples.

6.2 Horizon Choices and Local Value Feedback

Table 2 separates evaluation length and loss coverage. With full coverage, 4-second evaluation scores highest; 1.5 seconds exceeds 8 seconds. Four-second loss coverage improves Random but lowers Val and Hard.

Native-generation and execution-interval studies are reported in Appendices E.4 and E.5.

Table 2: Horizon ablations. Loss coverage denotes only the reward-directed term. Old-reference and SFT-anchor coverage are fixed at 8 s; $H_p = 8$ s and $H_a = 1.5$ s.

Setting	H_r (s)	Loss coverage (s)	Val14	Random	Hard
PreFTO	4	8	**84.27**	85.38	**72.79**
Short evaluation	1.5	8	84.06	85.27	71.19
Full-length evaluation	8	8	82.68	84.22	70.42
Prefix reward direction	4	4	84.07	**86.37**	70.17

6.3 Candidate Comparisons at Shared States

In Table 3, expert states and executed targets lower all scores; a buffer-wide baseline or perfect tracking lowers Val/Hard but raises Random. Tracking controls retain reactive traffic and raw targets; executed-target learning retains controller-based scoring.

Table 3: Candidate comparison ablations. Each row changes the named choice. State, scoring, and target definitions are in Appendix E.9.

Setting	Val14	Random	Hard
PreFTO	**84.27**	85.38	**72.79**
Expert-log states	83.43	85.02	70.11
Buffer-wide reward baseline	82.82	85.61	70.67
Perfect tracking for scoring	83.33	**85.69**	70.72
Executed trajectory target	83.61	84.93	71.72

6.4 Continuous Safety and Minimum Aggregation

Table 4 isolates continuous versus binary safety and each minimum versus mean operation, with quality fixed and TTC only in safety. The reported reference scores highest on all three splits.

Table 4: Safety design ablations. Safety aggregation uses dimension d and time t. Result aggregation and feedback statistics are in Appendix E.10.

Safety setting	Val14	Random	Hard
PreFTO: continuous, $\min_d \min_t$	**84.27**	**85.38**	**72.79**
Binary, $\min_d \min_t$	83.88	85.08	70.92
Continuous, $\min_d \mathrm{mean}_t$	83.78	84.66	69.62
Continuous, $\mathrm{mean}_d \min_t$	83.69	84.96	69.68

Aggregation changes reward gaps and filtering, so results assess the scoring–update procedure. Appendices E.6 and E.7 give additional controls.

7 Discussion and Conclusion

PreFTO learns complete raw plans from local execution feedback. Experiments support separate horizons, shared-state comparisons, and continuous minimum-aggregated safety. Loss coverage and execution mapping have split-dependent effects; safety gains can reduce progress. Four-second evaluation performs best here, without establishing universal value-estimation accuracy.

The theory gives conditional optimism and sufficient conditions for deployment improvement. Experiments do not verify continuation assumptions or support-wise discrepancy bounds. Evidence covers one planner family, and Hard informed checkpoint selection.

AI USE STATEMENT

AI tools assisted with literature discovery and comparison, review of implementation and experiment records, mathematical formulation, interpretation, manuscript drafting, and figure preparation. The authors have reviewed the AI-assisted material and take responsibility for the final text, claims, derivations, citations, and figures. AI tools also assisted in rerunning existing checkpoints and visualizing simulator records for Figure 4; the plotted states come from the simulator.

REPRODUCIBILITY STATEMENT

Sections 3, 5, and 6 specify the horizons, training objective, and evaluation scope. The supplementary material provides proofs of the theoretical results, the training recipe and hyperparameters in Appendix D, and additional experimental protocols and ablations. These details document the method and reported evaluation procedures; they do not imply bit-for-bit reproduction of historical runs.

ETHICS STATEMENT

This study evaluates planner adaptation in simulation. Improvements in selected aggregate safety metrics do not establish safety in deployment. The observed reduction in route progress must also be evaluated explicitly, since an aggregate score can obscure this tradeoff. We report no real-vehicle deployment.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training Diffusion Models with Reinforcement Learning. *arXiv preprint arXiv:2305.13301*, 2023. URL https://arxiv.org/abs/2305.13301.

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. URL https://arxiv.org/abs/2106.11810.

Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking Imitation-based Planner for Autonomous Driving. *arXiv preprint arXiv:2309.10443*, 2023. URL https://arxiv.org/abs/2309.10443.

Jie Cheng, Yingbing Chen, and Qifeng Chen. PLUTO: Pushing the Limit of Imitation Learning-based Planning for Autonomous Driving. *arXiv preprint arXiv:2404.14327*, 2024. URL https://arxiv.org/abs/2404.14327.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *The International Journal of Robotics Research*, 2025. doi: 10.1177/02783649241273668. URL https://diffusion-policy.cs.columbia.edu/diffusion_policy_ijrr.pdf.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with Misconceptions about Learning-based Vehicle Motion Planning. *arXiv preprint arXiv:2306.07962*, 2023. URL https://arxiv.org/abs/2306.07962.

Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Vincent Dutordoir, Riccardo Barbano, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. DEFT: Efficient Fine-Tuning of Diffusion Models by Learning the Generalised h-transform. *arXiv preprint arXiv:2406.01781*, 2024. URL https://arxiv.org/abs/2406.01781.

Fan Ding, Xuewen Luo, Fucai Ke, Hwa Hui Tew, Susilawati, Vishnu Monn Baskaran, and Junn Yong Loo. Drive As You Like: Multi-Head Diffusion with Reinforcement Learning for Personalized Driving. *arXiv preprint arXiv:2508.16947*, 2025. URL https://arxiv.org/abs/2508.16947.

Fan Ding, Xuewen Luo, Fengze Yang, Bo Yu, HwaHui Tew, Ganesh Krishnasamy, and Junn Yong Loo. KnowDiffuser: A Knowledge-Guided Diffusion Planner with LLM Reasoning. *arXiv preprint arXiv:2603.10441*, 2026. URL https://arxiv.org/abs/2603.10441.

Carson Eisenach, Haichuan Yang, Ji Liu, and Han Liu. Marginal Policy Gradients: A Unified Family of Estimators for Bounded Action Spaces with Applications. *arXiv preprint arXiv:1806.05134*, 2018. URL https://arxiv.org/abs/1806.05134.

Zhiyu Huang, Haochen Liu, and Chen Lv. GameFormer: Game-theoretic Modeling and Learning of Transformer-based Interactive Prediction and Planning for Autonomous Driving. *arXiv preprint arXiv:2303.05760*, 2023. URL https://arxiv.org/abs/2303.05760.

Zhiyu Huang, Xinshuo Weng, Maximilian Igl, Yuxiao Chen, Yulong Cao, Boris Ivanovic, Marco Pavone, and Chen Lv. Gen-Drive: Enhancing Diffusion Generative Driving Policies with Reward Modeling and Reinforcement Learning Fine-tuning. *arXiv preprint arXiv:2410.05582*, 2024. URL https://arxiv.org/abs/2410.05582.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with Diffusion for Flexible Behavior Synthesis. *arXiv preprint arXiv:2205.09991*, 2022. URL https://arxiv.org/abs/2205.09991.

Sergey Levine. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. *arXiv preprint arXiv:1805.00909*, 2018. URL https://arxiv.org/abs/1805.00909.

Hongchen Li, Tianyu Li, Jiazhi Yang, Haochen Tian, Caojun Wang, Lei Shi, Mingyang Shang, Zengrong Lin, Gaoqiang Wu, Zhihui Hao, Xianpeng Lang, Jia Hu, and Hongyang Li. PlannerRFT: Reinforcing Diffusion Planners through Closed-Loop and Sample-Efficient Fine-Tuning. *arXiv preprint arXiv:2601.12901*, 2026. URL https://arxiv.org/abs/2601.12901.

Qiyang Li, Seohong Park, and Sergey Levine. Decoupled Q-Chunking. *arXiv preprint arXiv:2512.10926*, 2025a. URL https://arxiv.org/abs/2512.10926.

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement Learning with Action Chunking. *arXiv preprint arXiv:2507.07969*, 2025b. URL https://arxiv.org/abs/2507.07969.

James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, second edition, 2026. URL https://sites.engineering.ucsb.edu/~jbraw/mpc/MPC-book-2nd-edition-6th-printing.pdf. Sixth printing.

Allen Z. Ren, Justin Lidard, Lars L. Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion Policy Policy Optimization. *arXiv preprint arXiv:2409.00588*, 2024. URL https://arxiv.org/abs/2409.00588.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv preprint arXiv:1506.02438*, 2015. URL https://arxiv.org/abs/1506.02438.

Wen Sun, J. Andrew Bagnell, and Byron Boots. Truncated Horizon Policy Search: Combining Reinforcement Learning & Imitation Learning. *arXiv preprint arXiv:1805.11240*, 2018. URL https://arxiv.org/abs/1805.11240.

Tianyi Tan, Yinan Zheng, Ruiming Liang, Zexu Wang, Kexin Zheng, Jinliang Zheng, Jianxiong Li, Xianyuan Zhan, and Jingjing Liu. Flow Matching-Based Autonomous Driving Planning with Advanced Interactive Behavior Modeling. *arXiv preprint arXiv:2510.11083*, 2025. URL https://arxiv.org/abs/2510.11083.

Xiaolong Tang, Meina Kan, Shiguang Shan, and Xilin Chen. Plan-R1: Safe and Feasible Trajectory Planning as Language Modeling. *arXiv preprint arXiv:2505.17659*, 2025. URL https://arxiv.org/abs/2505.17659.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*, 62(2):1805–1824, 2000. doi: 10.1103/PhysRevE.62.1805. URL https://arxiv.org/abs/cond-mat/0002177.

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*, 2025. URL https://arxiv.org/abs/2509.21961.

Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-ES: Gradient-free Planning with Diffusion for Autonomous Driving and Zero-Shot Instruction Following. *arXiv preprint arXiv:2402.06559*, 2024. URL https://arxiv.org/abs/2402.06559.

Hang Yu, Ye Jin, Alessandro Canevaro, Julian Schmidt, Julian Jordan, Peizheng Li, Marc Kaufeld, Silvan Lindner, Johannes Betz, and Wilhelm Stork. G2DP: Diffusion Planning with Spatio-Temporal Grid Guidance. *arXiv preprint arXiv:2606.26017*, 2026. URL https://arxiv.org/abs/2606.26017.

Zehan Zhang, Neng Zhang, Yaoyi Li, Jia Cai, and Zhiling Wang. Diffusion Forcing Planner: History-Annealed Planning with Time-Dependent Guidance for Autonomous Driving. *arXiv preprint arXiv:2606.11019*, 2026. URL https://arxiv.org/abs/2606.11019.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online Diffusion Reinforcement with Forward Process. *arXiv preprint arXiv:2509.16117*, 2025a. URL https://arxiv.org/abs/2509.16117.

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. *arXiv preprint arXiv:2501.15564*, 2025b. URL https://arxiv.org/abs/2501.15564.