HARNESSING LEARNED PLANNERS: EFFICIENT REINFORCEMENT FINE-TUNING VIA GROUP-BASED RANK-TREE SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned multi-modal planners can be harnessed with inference-time search: wider search can recover useful alternatives, particularly on difficult cases, but increases inference latency. In this paper, we investigate how to transfer the gains from inference-time search back into the planner itself through reinforcement fine-tuning, while keeping the cheap deployment search configuration unchanged. The challenge is constructing informative rollout groups when planning involves both discrete continuation decisions and continuous proposal geometry. Naively enumerating initial proposals with greedy continuations can miss useful later decisions, while groups with identical returns provide no learning signal under group-relative optimization. We introduce group-based rank-tree sampling, which enumerates short top-K continuation-rank schedules under a frozen reference planner, combines them with independent geometry samples, and avoids global beam pruning during training. This provides structured coverage of discrete planning decisions while preserving variation in continuous proposal geometry. We further incorporate failure-aware reward shaping and a bounded imitation-gradient contribution to stabilize learning. On NAVSIM v1, fine-tuning improves the driving planner's PDMS Score from 93.7 to 93.9 under identical evaluation settings. On an internal benchmark of dead-end parking scenarios, fine-tuning improves the success rate from 84.7% to 92.2% using a fixed beam-width-8/top-6 evaluator, exceeding the 91.4% the original model reaches only with a wider beam-width-30/top-15 search. In zero-shot evaluation on the public ParkBench dataset, the fine-tuned planner also improves the success rate from 68.6% to 72.5%.

1 INTRODUCTION

Learned multimodal planners generate multiple candidate paths, while feasibility checks, ranking, and/or beam search turn these proposals into an executable plan (Kirby et al., 2026; Huang et al., 2025; Zhang et al., 2025; Zheng et al., 2025). We refer to this established generation–selection paradigm as a *planning harness*. Deployed performance depends on both the proposal model and the selection procedure. Wider search can uncover alternatives missed by greedy decisions, but its additional inference cost can conflict with real-time deployment constraints. This trade-off motivates our central question: *can reinforcement fine-tuning improve the proposal model enough to recover the benefits of wider search while keeping the deployment search unchanged?*

Reinforcement fine-tuning can improve the *selector* that ranks generated candidates (Huang et al., 2025; Ang et al., 2026) or the *proposal distribution* itself through policy gradients (Tang et al., 2026; Lu et al., 2026; Li et al., 2026). Updating only the selector can improve candidate selection while leaving the generated paths unchanged. We focus on improving the generator and investigate how to construct its training rollout groups when deployment uses a fixed search budget. This question is particularly relevant to sequential planners, where exploration varies both *discrete choices* and *continuous geometry*: each selected maneuver query generates a path whose endpoint determines the state for subsequent decisions. For example, a second-ranked reverse continuation in parking may leave room for a final turn that the greedy continuation obstructs. Repeated geometry perturbations around the greedy choice may leave this alternative unexplored. Our hypothesis is that, at a fixed rollout budget, exploring alternative continuations can provide more informative training signals

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.

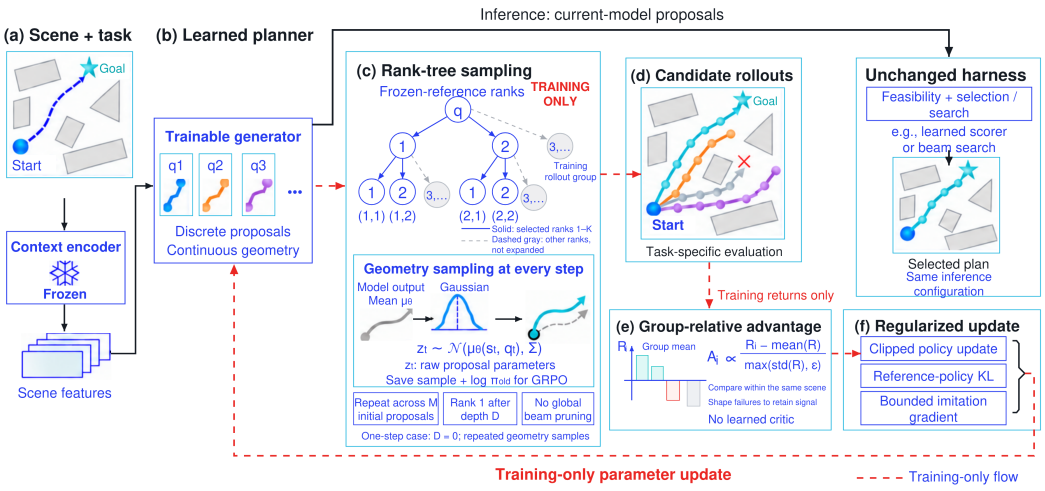


Figure 1: **Rank-tree GRPO: fine-tuning learned proposals with an unchanged deployment harness.** (a – b) A frozen context encoder provides scene features to a trainable generator that produces discrete proposals with continuous geometry. (c) Training-only rank-tree sampling explores a fixed set of continuation-rank schedules per initial proposal under a frozen reference, sampling geometry independently, with rank 1 after depth D and no global beam pruning (D=0 reduces to repeated geometry sampling). (d – f) Shaped task-specific returns are compared within each scene group to form critic-free advantages, which drive a clipped policy update regularized by a reference-policy KL and a bounded imitation gradient. At deployment, the updated generator supplies proposals to the existing checks and selection or search, with the inference configuration held fixed. Red dashed arrows indicate training-only flow.

for the generator's continuous predictions than allocating the same budget to additional geometry samples around greedy choices.

Group Relative Policy Optimization (GRPO) learns from return differences within each rollout group (Shao et al., 2024), making the alternatives included in that group central to the comparisons available for updating the generator. We introduce *rank-tree sampling*, which assigns each initial query a fixed collection of short continuation-rank schedules under a frozen reference planner. For instance, with $K = 2$ candidate ranks and $D = 2$ continuation steps, each root receives the schedules $(1, 1)$, $(1, 2)$, $(2, 1)$, and $(2, 2)$, followed by rank-1 continuations. Each rollout samples geometry independently and evaluates its scheduled ranks at its own reached states. Tree-structured policy optimization has been explored in prior work (Li et al., 2025a), and PaIR-Drive introduces a dedicated intention-conditioned trajectory sampler trained with GRPO (Lian et al., 2026). Our method instead allocates rollouts across the existing planner's multimodal continuations. The prescribed schedules balance exploration across local rank sequences without global score pruning or shared geometric prefixes. Generator updates use conditional continuous-action likelihoods, requiring neither differentiation through beam selection nor an additional categorical policy-gradient term. Failure shaping, damped advantages for all-failure groups, and a gradient-budgeted imitation anchor support the update. Figure 1 illustrates the resulting fine-tuning pipeline and its integration with the unchanged deployment harness.

Three empirical findings support the practical value of the proposed pipeline. First, on an internal test benchmark of 359 challenging dead-end parking scenarios, fine-tuned model increases its success from 84.7% to 92.2% under the unchanged beam-width-8/top-6 deployment search, exceeding the 91.4% achieved by the base planner with a wider beam-width-30/top-15 search. This comparison demonstrates improved performance at the original deployment budget. Second, the same checkpoint transfers zero-shot to the public ParkBench suite (Tao et al., 2026), increasing success from 68.6% to 72.5%. Third, a single-step formulation applied to DrivoR (Kirby et al., 2026) increases the PDM Score from 93.7 to 93.9 on NAVSIM v1, comparing the starting checkpoint and its fine-tuned counterpart under identical evaluation settings. This modest gain provides complemen-

tary evidence that the generator fine-tuning formulation also applies to a different architecture with generate-then-score selection; the single-step setting does not evaluate continuation branching.

Contributions.

- **Balanced rollout allocation over continuation ranks.** We introduce rank-tree sampling, a training-only strategy that assigns each initial query a fixed collection of reference-ranked continuation schedules. Each schedule is interpreted at the rollout's reached states, with independently sampled geometry. This construction balances exploration across the existing planner's continuation choices without global beam pruning or shared geometric prefixes.
- **Integration with conditional generator fine-tuning.** We combine rank-tree sampling with continuous-action GRPO. Prescribed ranks organize discrete continuation exploration, while conditional geometry likelihoods enable updates to the proposal generator. The formulation requires neither differentiation through deployment search nor an additional categorical policy-gradient term.
- **Empirical improvements under unchanged deployment harnesses.** We demonstrate improved sequential parking success under fixed search, exceeding a wider-search baseline and transferring zero-shot to public ParkBench scenes. A complementary single-step application to DrivoR on NAVSIM v1 evaluates generator fine-tuning under generate-then-score selection.

2 Related Work

Inference-time harnesses for planning. Classical approaches address constrained nonholonomic planning through geometric constructions, search, and optimization, including curvature-bounded shortest paths (Reeds & Shepp, 1990), Hybrid A* (Dolgov et al., 2008), state lattices (Pivtoraiko et al., 2009), and optimization-based collision avoidance (Zhang et al., 2021). Many learned planners also rely on search or selection at deployment, using beam search over sequential continuations or scoring a set of generated candidates (Kirby et al., 2026; Huang et al., 2025; Zhang et al., 2025; Zheng et al., 2025). This combination of learned proposals and search connects to broader work on search-guided learning and inference-time computation (Anthony et al., 2017; Snell et al., 2024). In autonomous driving, nuPlan (Caesar et al., 2021) and NAVSIM (Dauner et al., 2024) provide public benchmarks for evaluating planning performance. NAVSIM's PDMS Score draws on the evaluation criteria of the rule-based PDM planner (Dauner et al., 2023), while methods such as Hydra-MDP (Li et al., 2024), DiffusionDrive (Liao et al., 2025), and DrivoR (Kirby et al., 2026) illustrate the generation–selection paradigm. We adopt performance under the deployed harness as our evaluation target and fine-tune the proposal generator while keeping that harness unchanged. Our training surrogate uses rollout returns and continuous-action likelihoods, without differentiating through deployment search.

Imitation-learned multimodal planners. Multimodal imitation-learned policies (Shafiullah et al., 2022; Zhao et al., 2023; Chi et al., 2023), anchor- and query-based motion prediction and planning models (Chai et al., 2019; Shi et al., 2022; Cheng et al., 2024; Pini et al., 2023), register-token generator–scorer architectures (Kirby et al., 2026), and segment-level maneuver planners (Zheng et al., 2025) provide the foundations for the proposal generators considered here. Our parking implementation adopts a MultiPark-style factorized segment decoder (Zheng et al., 2025); our contribution concerns rollout allocation and generator fine-tuning rather than architectural design. Covariate shift during closed-loop execution (Ross et al., 2011; Bansal et al., 2019) motivates further optimization beyond imitation learning. In our experiments, teacher-forced imitation loss on challenging scenes has nearly converged, yet substantial closed-loop failures remain, highlighting a gap between imitation fit and deployed planning performance.

Reinforcement fine-tuning of planners. Policy-gradient methods, including group-relative optimization (Shao et al., 2024), have been applied to autoregressive trajectory policies, diffusion and flow policies (Ren et al., 2025; Li et al., 2026; Sheng et al., 2026; Pfrommer et al., 2025), and intent-conditioned driving policies (Lu et al., 2026). Plan-R1 uses variance-decoupled GRPO, replacing standard-deviation normalization with mean centering to preserve the influence of safety rewards (Tang et al., 2026). In constrained low-speed maneuvering, related approaches include hybrid policies with geometric takeover (Jiang et al., 2025b; Xue et al., 2026), diffusion refinement with RL

priors (Jiang et al., 2025a), and concurrent work on curvature-arc RL (Wang et al., 2026). These approaches can improve planning by learning a *selector* over generated candidates (Huang et al., 2025; Zhang et al., 2025; Ang et al., 2026), fine-tuning the *proposal distribution* (Tang et al., 2026; Lu et al., 2026; Li et al., 2026; Sheng et al., 2026; Ren et al., 2025; Pfrommer et al., 2025), or adapting the *sampling input* of a frozen generator (Wagenmaker et al., 2025). Our method fine-tunes the proposal distribution, focusing on rollout-group construction for sequential planners evaluated under unchanged deployment search.

Structured rollout collection has been studied in tree-based policy optimization (Li et al., 2025a; Ji et al., 2025; Yang et al., 2025; Li et al., 2025b; Ding & Ye, 2026). In driving, PaIR-Drive (Lian et al., 2026) introduces a dedicated intention-conditioned sampler that recurrently expands a trajectory tree. Our method instead prescribes reference-ranked continuation schedules for the existing planner's multimodal continuations: at each reached state the schedule selects a locally ranked candidate under a frozen reference, each rollout samples continuous geometry independently (shared rank prefixes need not share trajectory prefixes), and no shared-prefix trajectory tree or global beam pruning is maintained. The resulting rollouts support GRPO updates through conditional continuous-action likelihoods while the deployment search remains unchanged. Relative to DIAL's intent-balanced groups (Lu et al., 2026), our focus extends to subsequent decisions conditioned on reached states. CLOVER (Ang et al., 2026) analyzes candidate coverage and ranking; we examine how continuation choices organize training experience for generator fine-tuning. Expert iteration and search distillation (Anthony et al., 2017) use search-derived learning targets, whereas our method uses reference-ranked continuations to structure exploration and rollout returns to guide continuous-geometry updates. TOAD uses the Cross-Entropy Method to refine trajectories at inference time while keeping the planner frozen. Our approach instead updates the proposal generator during training and retains the existing deployment search, placing the additional optimization effort at a different stage.

3 Problem Setting

General interface. Given a scene or planning state, a generator proposes candidate motions, and a deployment harness selects a complete plan. A one-step planner produces complete trajectories, while a sequential planner produces motion segments whose endpoints determine subsequent planning states. We denote the deployment beam width by W and the number of candidates expanded per parent by k. During training, M denotes the number of initial queries, while K and D specify the number of continuation ranks explored and the number of scheduled continuation decisions, respectively. Since D counts decisions *after* the root, a one-step planner has a rollout horizon of $T = 1$, with $D = 0$ and $K^D = 1$. Multiple initial queries and independent geometry samples can still form a GRPO group containing multiple rollouts.

Structured segment planning. A scene x specifies surrounding obstacles, the ego vehicle's footprint, and, when applicable, a target pose. A plan consists of motion segments $a_{1:T}$, where each segment $a_t = (m_t, \kappa_t, \ell_t)$ combines a discrete maneuver mode m_t with continuous geometry: a curvature profile $\kappa_t \in \mathbb{R}^H$, where H is the number of curvature values per segment and each value satisfies $|\kappa_{t,h}| \le \kappa_{\max}$, and a segment length ℓ_t. Each segment is integrated under nonholonomic kinematics and checked for collisions along the swept vehicle footprint. In target-directed planning, success requires reaching the target within prescribed position and heading tolerances, without collision, using at most T segments. The policy, pre-trained by imitation on expert plans, factorizes as $\pi_\theta(a_t \mid s_t) = \pi_\theta(m_t \mid s_t) \cdot \pi_\theta(\kappa_t, \ell_t \mid s_t, m_t)$, separating the discrete mode distribution from the conditional geometry distribution. Mode probabilities provide candidate rankings. This discrete-continuous representation also accommodates anchor- and query-based planners with mode-conditioned geometry. Our sequential parking implementation uses alternating-direction maneuver segments (§5.1).

Sequential deployment harness. In a sequential planning task, the deployment harness constructs a plan using beam search $\mathrm{B}_{W,k}(\pi_\theta, x)$ with beam width W and per-parent expansion limit k. Search begins with the legal root modes. At each depth, the decoder scores the legal continuations of each branch and checks completed candidates against the task's success criterion. Each parent's top-k nonterminal candidates undergo collision checking, after which the best (at most) W feasible branches are retained according to cumulative mode score. If no successful plan is found within the segment budget, the harness returns the closest feasible branch and records a failure. Deployed

success is

$$\mathrm{Succ}(\theta) \;=\; \mathbb{E}_x[\,\mathbf{1}\{\,\mathrm{B}_{W,k}(\pi_\theta, x) \text{ returns a successful plan }\}\,]\,. \tag{1}$$

We keep (W, k) fixed when comparing the imitation and fine-tuned policies. Wider search provides a separate reference for the performance achievable with additional inference computation. For a fixed policy and deterministic geometry, let $\mathcal{E}_k(x)$ denote the proposal tree obtained by removing global beam-width pruning while preserving the segment budget, per-parent top-k nonterminal expansion, and terminal-candidate checks. The terminal checks are essential: a successful final segment may be evaluated even when it falls outside the nonterminal top-k. Deployed success then decomposes as

$$\mathrm{Succ}(\theta) = \underbrace{\Pr_x[\exists\,\tau \in \mathcal{E}_k(x) : \tau \text{ succeeds}]}_{\text{proposal coverage}} \times \underbrace{\Pr_x[\mathrm{B}_{W,k} \text{ succeeds} \mid \exists\,\tau \in \mathcal{E}_k(x) : \tau \text{ succeeds}]}_{\text{width/ranking conversion}}\,. \tag{2}$$

The identity holds because every successful plan returned by the beam belongs to $\mathcal{E}_k(x)$. Changing W and k can affect both factors, so the wider-search comparison measures a search-budget gap without isolating coverage from conversion; the stochastic training groups introduced below are a different proposal set, and their measured coverage does not bound deployed success after fine-tuning.

Group-relative fine-tuning and learning-signal availability. Group-relative methods sample a rollout group $G(x) = \{\tau_1, \ldots, \tau_n\}$ for each scene x and assign each rollout a return R_i, $i \in \{1, \ldots, n\}$. These returns define normalized advantages $A_i = (R_i - \bar{R})/\max(s_x, \varepsilon)$, where $\bar{R}$ and s_x are the group mean and population standard deviation, respectively, and $\varepsilon > 0$ ensures numerical stability. Restricting updates to groups containing at least one successful rollout is an optional gating rule. We distinguish the presence of successful rollouts from the availability of reward differences that support a group-relative update:

Definition 1 (Training coverage)**.** *For a rollout-group sampler G, training coverage is*

$$\mathrm{cov}(G) = \Pr[\exists\,\tau \in G(x) : \tau \textit{ succeeds}]\,,$$

where the probability is taken over scenes and rollout sampling.

If every rollout in a group receives the same return, all centered advantages are zero and the group contributes no reward-driven policy gradient—for all-failure and all-success groups alike—whereas an all-failure group can contribute when reward shaping differentiates its returns and the group is retained for learning. Let $m_x \in \{0, 1\}$ indicate whether the group is retained by the success/failure gate. We define the *signal-scene rate* as

$$\mathrm{sig}(G) = \mathbb{E}[\mathbf{1}\{s_x \geq \varepsilon,\; m_x = 1\}]\,, \tag{3}$$

with the expectation taken over scenes and rollout sampling. This diagnostic measures the fraction of retained groups whose return variation meets the threshold; it measures neither the magnitude of the resulting parameter gradients nor the learning signal supplied by auxiliary imitation.

4 Method: Rank-tree GRPO

4.1 Group-based continuation-rank tree sampling

For each scene, we enumerate all M legal initial query IDs and allocate K^D rollout rows to each root, yielding a group of size $B = MK^D$. Each row corresponds to a leaf of a *fixed rank tree*, which specifies a continuation-rank schedule. Leaf l is assigned a rank $d_s(l) \in 1, \ldots, K$ at each continuation step $s \in 1, \ldots, D$. At that step, the rollout follows the $d_s(l)$-th ranked legal query under the frozen reference policy, evaluated at its current state. Subsequent steps use rank 1 (argmax). Geometry is sampled independently at every selected query. Because root IDs and continuation ranks are prescribed, the conditional rollout likelihood in Eq. (4) contains only continuous-geometry terms. Choosing $K \leq k$ keeps the explored local ranks within the deployment expansion limit. The sampler maintains no global beam and performs no cumulative-score pruning. Each row stops at its own terminal condition; success in one row does not terminate the others. For $K = 2, D = 2$, each root receives four schedules: $(1, 1)$, $(1, 2)$, $(2, 1)$, and $(2, 2)$. Enumerating these schedules

gives the same schedule coverage as drawing all four uniformly without replacement; their ordering is incidental to the method. A fixed rank schedule does not specify a fixed query-ID sequence: although the reference weights remain frozen, rankings are recomputed at each rollout's sampled state. Nor do different schedules necessarily produce distinct geometric paths or maneuver classes. At a fixed rollout budget $B = MK^D$, the relevant comparison is therefore between alternative allocations of the same B rows. Figure 7 in Appendix D contrasts this allocation with geometry-only sampling and with the deployment beam.

The prescribed schedules support a batched rollout loop with terminal masks, with advantages normalized across all MK^D rows in each scene. The tree represents *rank choices*; Gaussian geometry samples remain independent across rows, including those with identical root IDs and early rank assignments. Shared rank prefixes therefore need not correspond to shared trajectory prefixes. In the deterministic-geometry probe, leaves that terminate before their schedules diverge can produce duplicate outcomes; we retain these multiplicities in the group statistics. Setting $K = 1$ or $D = 0$ recovers the legacy initial-query fan with argmax continuations.

Rollout coverage evaluation. Before training, we evaluate each candidate (K, D) configuration in a single pass over a fixed set of scenes using deterministic geometry. We record the fraction of scenes with at least one successful rollout and the fraction whose groups are both retained and nonflat, as measured by the signal-scene rate in Eq. (3). All compared sampling configurations use the same checkpoint, reward shaping, and failure-group gate. Accordingly, "legacy" refers only to the sampling pattern, not to a success-only gating rule. This evaluation characterizes the learning opportunities exposed by each configuration at the current checkpoint; it neither predicts the final success rate after fine-tuning nor guarantees convergence. Measured coverage and signal-scene rates for the candidate configurations are reported in Table 2.

4.2 Group-relative objective on the continuous geometry

Fine-tuning optimizes the continuous geometry of the structured policy with a critic-free group-relative objective; discrete mode selection during rollouts is *exogenous*, supplied by the frozen imitation policy's ranking (§4.1), so trajectory log-probabilities contain no categorical action term. Per decision, the stochastic action is the selected query's *raw* geometry: the curvature profile follows a fixed-variance Gaussian $\kappa^{\text{raw}} \sim \mathcal{N}(\mu_\theta(s,m),\, \sigma_\kappa^2\, \Sigma_\rho)$ with AR(1) waypoint correlation Σ_ρ, and the raw length logit follows $\ell^{\text{raw}} \sim \mathcal{N}(\nu_\theta(s,m),\, \sigma_\ell^2)$. A deterministic parameter-free map converts raw actions to executable geometry, so likelihoods and KLs are evaluated for the latent raw actions, where the Gaussian density is exact and no physical-space Jacobian is required. Thus, for active decisions $t \in \mathcal{T}_i$,

$$\log \pi_\theta(\tau_i \mid x, m_{i,1:T}) = \sum_{t \in \mathcal{T}_i} \big[\log \mathcal{N}(\kappa_{i,t}^{\text{raw}}; \mu_\theta(s_{i,t}, m_{i,t}), \sigma_\kappa^2 \Sigma_\rho) + \log \mathcal{N}(\ell_{i,t}^{\text{raw}}; \nu_\theta(s_{i,t}, m_{i,t}), \sigma_\ell^2) \big]. \tag{4}$$

Here the conditional-log-density notation denotes the geometry factors along the recorded state/query sequence; it is not a categorical likelihood for the prescribed schedule. With group return R_i and trajectory-level advantage A_i from §4.3, the same A_i is broadcast to every active decision of trajectory i. The implementation follows the clipped surrogate of Schulman et al. (2017), using the *per-decision* ratio $r_{i,t} = \exp(\log \pi_\theta(a_{i,t} \mid s_{i,t}, m_{i,t}) - \log \pi_{\text{old}}(a_{i,t} \mid s_{i,t}, m_{i,t}))$ and averages decisions within each trajectory:

$$\mathcal{L}_{\text{RL}}(\theta) = -\, \mathbb{E}_i \left[\frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \min \left(r_{i,t} A_i,\ \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon) A_i \right) \right] + \sum_c \beta_c\, \mathbb{E}_i \left[\frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \text{KL}_c(s_{i,t}) \right], \tag{5}$$

Here $\text{KL}_c(s)$ is a state-conditioned reference divergence per component c (mode categorical, curvature, length, validity); the categorical term is a regularizer that protects the deployed ranking even though mode IDs are forced by the frozen selector, not a sampled-action likelihood. Groups masked by the signal gate contribute neither term. Remaining implementation details—raw-action storage, exact KL definitions, and rollout reuse—are given in Appendix D.

4.3 TAIL ACTIVATION: MARGIN-GUARDED SHAPED FAILURES AND FAILURE-GROUP ADVANTAGES

Some groups contain only failures. We retain informative comparisons among them using two supporting mechanisms. First, safe timeouts (no collision, no success) receive a shaped terminal reward. For N_i executed segments, terminal position error d_i, and wrapped heading error $|\Delta\psi_i|$, the implemented return is

$$R_i = \begin{cases} r_{\text{coll}}, & \text{if any swept-footprint collision occurs,} \\ r_{\text{succ}} + N_i r_{\text{step}}, & \text{if the goal is reached without collision,} \\ r_{\text{fail}} + w \exp[-(d_i/\sigma_d + |\Delta\psi_i|/\sigma_\psi)], & \text{otherwise.} \end{cases} \tag{6}$$

Collision overrides success and never accumulates a step cost; the efficiency penalty applies only to successful paths. The construction-time check $r_{\text{fail}} + w < r_{\text{succ}} + T r_{\text{step}}$ guarantees that *every* success outranks *every* shaped failure, while collisions remain at the flat r_{coll}; safe near-misses can therefore outrank collisions and distant wandering, but never a success. Second, for scene x with group mean $\bar{R}_x$ and population standard deviation s_x, the critic-free advantage is

$$A_i = \alpha_x \frac{R_i - \bar{R}_x}{\max(s_x, \varepsilon)}, \qquad \alpha_x = \begin{cases} 1, & \exists j : \tau_j \text{ succeeds,} \\ \lambda_{\text{fail}}, & \text{otherwise,} \end{cases} \tag{7}$$

and the group is masked if s_x is below a small threshold. Thus all-failure groups retain their normalized advantages at a damped scale rather than being zeroed: comparing shaped failures is a weaker teacher than comparing against a success, but does not establish that such updates improve deployed success. Flat all-collision groups still have zero centered advantages; retaining a group cannot manufacture reward contrast. Neither reward ordering nor shaping ensures collision-free deployment.

4.4 IMITATION ANCHOR WITH A RELATIVE GRADIENT BUDGET

We retain an auxiliary pure-imitation objective on hard training scenes, excluding constraint penalties that can conflict with demonstrated terminal contact. In the parking implementation, let $g_{\text{RL},r}$ denote the local gradient on rank r of Eq. 5, *including its KL terms*, and let $g_{\text{IL},r}$ denote the local anchor gradient. Separate backward passes produce

$$g_r = g_{\text{RL},r} + c_r w_{\text{IL}} g_{\text{IL},r}, \qquad c_r = \min\left(1, \frac{\eta_{\text{anc}} \|g_{\text{RL},r}\|_2}{\|w_{\text{IL}} g_{\text{IL},r}\|_2 + \varepsilon}\right). \tag{8}$$

For nonzero local RL norm, this caps the *local anchor-gradient contribution* by a fraction of that norm. If the local measured RL norm is zero, an absolute cap is used instead. The merged gradients are then averaged across ranks and globally norm-clipped before AdamW. Gradient cancellation means the same relative bound need not hold after averaging. It is also not a bound on parameter displacement under AdamW, nor a guarantee against performance regression. Algorithm 1 in Appendix D summarizes one complete training step.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Sequential parking. A MultiPark-style segment transformer (Zheng et al., 2025) emits curvature profiles and lengths from a fixed query bank. Each scene specifies the vehicle, obstacles, and target. Rollouts alternate driving direction, use swept-footprint collision checks, and stop at success, collision, or eight segments. The goal tolerance is 0.5 m and 10°. We retain the same width-8/top-6 beam evaluator for the original and fine-tuned models. The inherited architecture, teacher-forcing scheme, and imitation objective are in Appendix B. Our primary internal parking benchmark is a collection of dead-end scenarios (359 cases). We retain another normal parking benchmark Mini-5K (5,000) as an additional evaluation. We report success counts and collision outcomes, evaluating both checkpoints under the same beam settings. Fine-tuned checkpoints are selected by the best closed-loop evaluation result on the training corpus; the reported benchmarks are not used for checkpoint selection. Implementation-level hyperparameters and reward details are preserved in Appendix D.

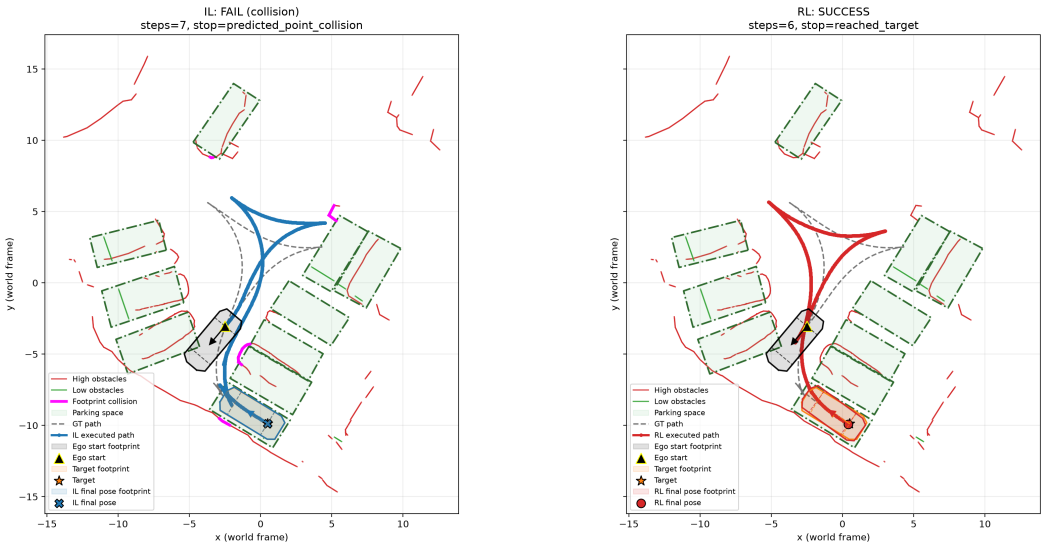


Figure 2: **Qualitative dead-end example under the same $B_{8,6}$ search.** The imitation planner (blue, left) ends in a predicted footprint collision (magenta), while the fine-tuned planner (red, right) parks collision-free. Additional examples are provided in Appendix A.

Public driving and parking evaluation. We evaluate the driving specialization on NAVSIM v1. Its single-step proposal-group construction is detailed in Appendix F. For public parking evaluation, we use the constrained-parking benchmark ParkBench (Tao et al., 2026), whose scenes are used for testing, without additional fine-tuning, hyperparameter adjustment, or domain adaptation. This evaluation assesses transfer beyond the internal parking benchmark.

5.2 Results

Driving results. Under a matched evaluation of the same starting checkpoint—identical proposal counts, scorer weights, and protocol—the original DrivoR planner attains a PDM Score of 93.7 on NAVSIM v1, and fine-tuning the generator raises it to 93.9. An evaluation-only oracle decomposition over the 12,147 scenes explains the modest deployed delta: the ground-truth-best of the 64 candidates already scores 99.3 PDMS before fine-tuning (99.1 after), so proposal coverage is near-saturated, while the oracle–selected gap narrows from 5.64 to 5.17 points—in the terms of Eq. (2), the realized gain is improved conversion under the frozen scorer, at a slight cost in best-case candidate quality (per-scene details in Appendix F). The single-step setting does not test continuation branching.

Public parking evaluation. Under the zero-shot ParkBench protocol, the imitation baseline succeeds on 68.6% of scenarios and the rank-tree fine-tuned checkpoint on 72.5%, a $+3.9$-point improvement whose direction is consistent with the internal dead-end comparison ($84.7\% \rightarrow 92.2\%$). The absolute level drops out of distribution, as expected for a planner trained on a different scenario distribution, while the fine-tuning gain survives the shift.

Fine-tuning under unchanged deployment search. Table 1 compares the reported checkpoints under the same $B_{8,6}$ evaluator, with the base planner's no-search result as a reference. Without search the base planner reaches only 49.0% on dead-end and 91.4% on Mini-5K, so the deployment beam is responsible for a large part of its success; our comparisons hold that beam fixed. Figure 2 shows one representative dead-end rescue: under the identical deployment search, the imitation planner terminates in a predicted collision, whereas the fine-tuned planner reaches the target. On the dead-end dataset, rank-tree GRPO increases successes from 304 to 331, a gain of 27 cases or 7.52 percentage points. The particular legacy run regresses on the dead-end suite. Hard-case data reweighting recovers the baseline count; this does not establish that hard-case sampling is generally ineffective.

Table 1: Success rate under the deployment search. Rows use the beam-width-8/top-6 evaluator except the no-search reference (a single greedy path, $W{=}1$, $k{=}1$). Counts in parentheses. Bold marks within-column maxima among the $\mathrm{B}_{8,6}$ checkpoints.

Planner	Search	Dead-end (359)	Mini-5K (5,000)
Base planner (IL)	none	49.03% (176)	91.40% (4,570)
Base planner (IL)	$\mathrm{B}_{8,6}$	84.68% (304)	99.32% (4,966)
GRPO, legacy sampler	$\mathrm{B}_{8,6}$	83.01% (298)	**99.34% (4,967)**
Legacy GRPO + hard-case data reweighting	$\mathrm{B}_{8,6}$	84.68% (304)	99.30% (4,965)
Rank-tree GRPO ($K = 2, D = 2$)	$\mathrm{B}_{8,6}$	**92.20% (331)**	**99.34% (4,967)**

Table 2: Pre-training rollout probe, kept separate from deployed success. Coverage is the fraction of scenes with at least one successful rollout; signal rate counts retained nonflat groups. Shared shaping permits nonflat all-failure groups to provide signal. The expanded-legacy and $K = 2, D = 2$ rows share a 120-row budget, stochastic geometry, and random seed, isolating allocation from sample count: the expanded legacy draws four geometry samples per root under argmax continuations, whereas the rank tree spends the same budget on four continuation-rank schedules.

Allocation	Rows/scene	Coverage	Signal rate
Legacy roots + argmax continuation	30	79.3%	90.3%
Expanded legacy (4 geometry samples/root)	120	85.9%	96.2%
Rank schedules $K = 2, D = 2$	120	89.7%	96.9%
Rank schedules $K = 2, D = 3$	240	91.9%	98.4%

Near-saturated general scenarios. Mini-5K remains near 99.3% for all reported checkpoints, with one additional success for rank-tree GRPO relative to the base. This is a limited non-regression check, not evidence of broad generalization or statistical equivalence. Details in Appendix E.

Inference search is a separate axis. Widening the deployment search improves the base model, from 84.7% with $\mathrm{B}_{8,6}$ to 91.4% with $\mathrm{B}_{30,15}$. Fine-tuning reaches 92.2% at the cheaper $\mathrm{B}_{8,6}$ setting, matching what the base model attains only with a roughly $9\times$ larger nonterminal expansion budget (450 versus 48 candidates per step). Fine-tuning thus delivers, under the unchanged deployment configuration, a capability the base model reaches only with substantially wider search. We compare nominal search budgets rather than measured wall-clock latency.

Pre-training rollout probe. Probes with the base checkpoint (Table 2) compare allocations at a matched 120-row budget under identical stochastic geometry and the same random seed. Spending the budget on four geometry samples per root under argmax continuations (expanded legacy) reaches 85.9% coverage and 96.2% signal-scene rate; spending the same budget on rank-tree schedules reaches 89.7% and 96.9%. The rank-tree allocation therefore adds 3.8 coverage points over equal-budget geometry-only sampling, with a smaller 0.7-point signal-rate gain; a second control seed (87.5% / 94.7%) places the coverage advantage in the 2.2–3.8 point range. Most of the improvement over the 30-row legacy reflects the larger budget, with allocation contributing this remainder. This probes training opportunity, not a final-checkpoint result; per-scene records have not been independently audited.

6 Discussion and Limitations

We introduced reference-ranked continuation schedules with independent geometry samples and scene-relative advantages. Under unchanged beam settings, rank-tree fine-tuning improves dead-end success, preserves near-saturated Mini-5K performance, and transfers zero-shot to ParkBench. Although proprietary parking data limit reproducibility, the strategy can be adapted to other learned-planner harnesses. Future work could extend our single-step driving formulation to closed-loop rollouts, using simulators or world models to account for future observations and other agents' responses.

REPRODUCIBILITY STATEMENT

Appendices B and C describe the inherited model, imitation objective, and preprocessing. Appendix D details the parking policy likelihood, reward formulation, and gradient-budget implementation. We specify the evaluated checkpoints and deployment search settings alongside the reported results. The internal parking training corpus is proprietary, while NAVSIM v1 and ParkBench provide public evaluation settings. ParkBench is used exclusively for testing, without additional fine-tuning. Fine-tuned checkpoints are selected through closed-loop evaluation on the training corpus, without using the reported evaluation benchmarks for checkpoint selection.

AI-USE STATEMENT

Large language model (LLM) assistants supported the implementation and preparation of this work. *Implementation:* Claude Code (Opus 5) assisted with implementing and editing portions of the training, evaluation, and analysis code, as well as analyzing experimental outputs. *Manuscript:* ChatGPT-6 Astra assisted with drafting and revising portions of the manuscript, including the related-work discussion, improving readability and concision, formatting references, and adjusting figure layouts. Qualitative figures were cropped for presentation without altering the plotted content. The research questions, method design, experiments, and conclusions are the authors' own. The authors reviewed all AI-assisted code and text and take full responsibility for the final methodology, code, citations, and experimental claims.

REFERENCES

Sining Ang, Yuguang Yang, Canyu Chen, and Yan Wang. Clover: Closed-loop value estimation and ranking for end-to-end autonomous driving planning. *arXiv preprint arXiv:2605.15120*, 2026.

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

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

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.

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning (CoRL)*, 2019.

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.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning (CoRL)*, 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Zheng Ding and Weirui Ye. Treegrpo: Tree-advantage grpo for online rl post-training of diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Practical search techniques in path planning for autonomous driving. In *AAAI Workshop: First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)*, 2008.

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

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning. *arXiv preprint arXiv:2509.21240*, 2025.

Mingyang Jiang, Yueyuan Li, Jiaru Zhang, Songan Zhang, and Ming Yang. A diffusion-refined planner with reinforcement learning priors for confined-space parking. *arXiv preprint arXiv:2510.14000*, 2025a.

Mingyang Jiang, Yueyuan Li, Songan Zhang, Siyuan Chen, Chunxiang Wang, and Ming Yang. Hope: A reinforcement learning-based hybrid policy path planner for diverse parking scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 2025b. arXiv:2405.20579.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, Anh-Quan Cao, Nermin Samet, Tuan-Hung Vu, and Matthieu Cord. Driving on registers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2601.12901.

Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, Zheng Zhang, Wei Shen, Qian Liu, Chenghua Lin, Jian Yang, Ge Zhang, and Wenhao Huang. TreePO: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv preprint arXiv:2508.17445*, 2025a.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. BranchGRPO: Stable and efficient GRPO with structured branching in diffusion models. *arXiv preprint arXiv:2509.06040*, 2025b.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, Yu-Gang Jiang, and Jose M. Alvarez. Hydra-MDP: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

Zhexi Lian, Haoran Wang, Xuerun Yan, Weimeng Lin, Xianhong Zhang, Yongyu Chen, and Jia Hu. Fine-tuning is not enough: A parallel framework for collaborative imitation and reinforcement learning in end-to-end autonomous driving. *arXiv preprint arXiv:2603.13842*, 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Hengtong Lu, Victor Shea-Jay Huang, Chengmin Yang, Pengfei Jing, Jifeng Dai, Yan Xie, and Benjin Zhu. Driving intents amplify planning-oriented reinforcement learning. *arXiv preprint arXiv:2605.12625*, 2026.

Samuel Pfrommer, Yixiao Huang, and Somayeh Sojoudi. Reinforcement learning for flow-matching policies. *arXiv preprint arXiv:2507.15073*, 2025.

Stefano Pini, Christian S. Perone, Aayush Ahuja, Ana Sofia Rufino Ferreira, Moritz Niendorf, and Sergey Zagoruyko. Safe real-world autonomous driving by learning to predict and plan with a mixture of experts. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

Mihail Pivtoraiko, Ross A. Knepper, and Alonzo Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333, 2009.

James A. Reeds and Lawrence A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

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. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2409.00588.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

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

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

Junhe Sheng, Ruofei Bai, Kuan Xu, Ruimeng Liu, Jie Chen, Shenghai Yuan, Wei-Yun Yau, and Lihua Xie. Beyond imitation: Reinforcement learning fine-tuning for adaptive diffusion navigation policies. *arXiv preprint arXiv:2603.12868*, 2026.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Xiaolong Tang, Meina Kan, Shiguang Shan, and Xilin Chen. Plan-r1: Safe and feasible trajectory planning as language modeling. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2505.17659.

Feng Tao, Luca Paparusso, Chenyi Gu, Robin Koehler, Chenxu Wu, Xinyu Huang, Christian Juette, David Paz, and Ren Liu. Adapting reinforcement learning for path planning in constrained parking scenarios. *arXiv preprint arXiv:2601.22545*, 2026.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025.

Zihan Wang, Bai Huang, Yang Guan, Xiao Li, Haoyu Xu, Naizheng Wang, and Shengbo Eben Li. Neuralparker: A reinforcement learning planner for irregular parking environments. *arXiv preprint arXiv:2608.24485*, 2026.

Yifan Xue, Toktam Mohammadnejad, Faizan M. Tariq, Sangjae Bae, David Isele, Yosuke Sakamoto, Nadia Figueroa, and Jovin D'sa. N3p: Accelerated automated parking via a learning-based naturalistic three-stage scheme. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2026. arXiv:2605.22722.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. TreeRPO: Tree relative policy optimization. *arXiv preprint arXiv:2506.05183*, 2025.

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

Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29(3):972–983, 2021.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

Han Zheng, Zikang Zhou, Guli Zhang, Zhepei Wang, Kaixuan Wang, Peiliang Li, Shaojie Shen, Ming Yang, and Tong Qin. Multipark: Multimodal parking transformer with next-segment prediction. *arXiv preprint arXiv:2508.11537*, 2025.

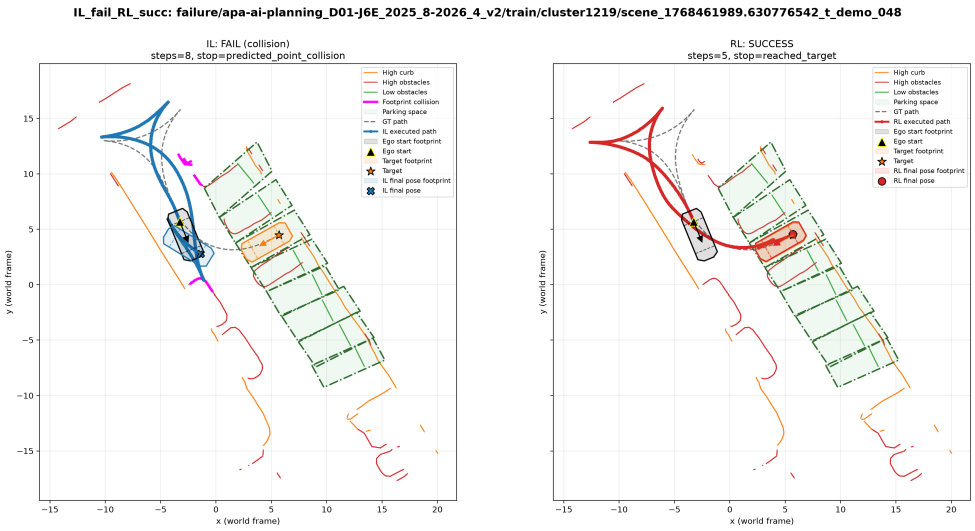


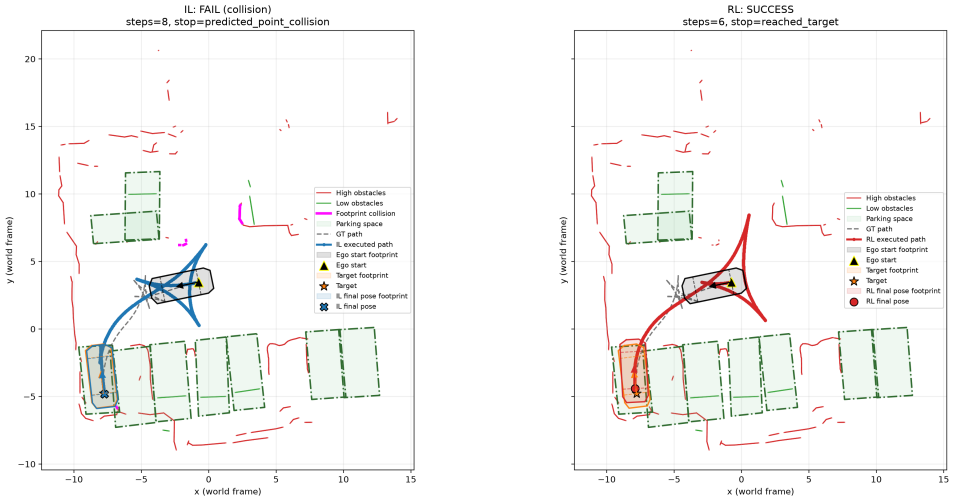
Figure 3: **Selected internal parking example.** The original IL planner (left, blue) terminates with a predicted-collision failure after eight steps; the RL-fine-tuned planner (right, red) reaches the target in five. The original evaluation plot is reproduced without redrawing trajectories. This selected rescue illustrates a changed maneuver and successful target arrival, not the average improvement or an isolated effect of the sampler. Additional cases appear below.

A Selected Internal Parking Examples

Figure 3 compares an IL rollout that ends in collision with an RL rollout that reaches the same parking target. Figure 4 presents two additional cases in which fine-tuning resolves collision failures, while Figure 5 shows a scene where the fine-tuned planner completes a maneuver that the IL planner fails to finish. These examples are selected from the internal dataset to illustrate behavioral changes and are not representative of all outcomes; aggregate performance is reported in Table 1.

The figures preserve the original trajectories, geometry, legends, and termination labels. Fine-tuning does not necessarily produce shorter rollouts. In the cluster1324 example (Figure 4b), the successful RL rollout takes seven steps, whereas the failed IL rollout terminates after five.

(a) Cluster1129: collision failure versus target reached



(b) Cluster1324: a successful maneuver can require more steps

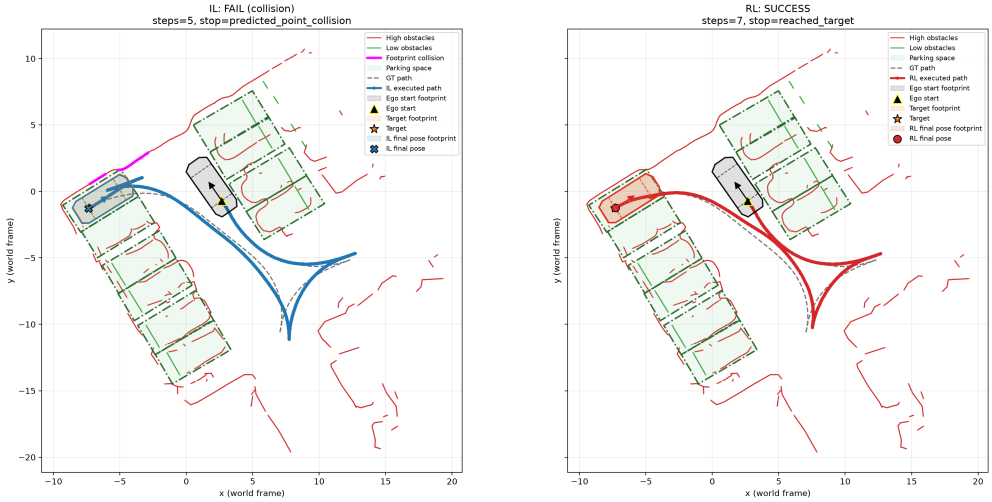


Figure 4: Two selected internal rescue cases. In (a), IL fails after eight steps and RL succeeds after six; in (b), IL fails after five and RL succeeds after seven. Blue and red denote the respective executed paths; magenta marks footprint-collision indications in the source plots. Blank margins and filename headers are cropped for layout; the plots are otherwise unchanged. These examples do not quantify the frequency of rescues or regressions.

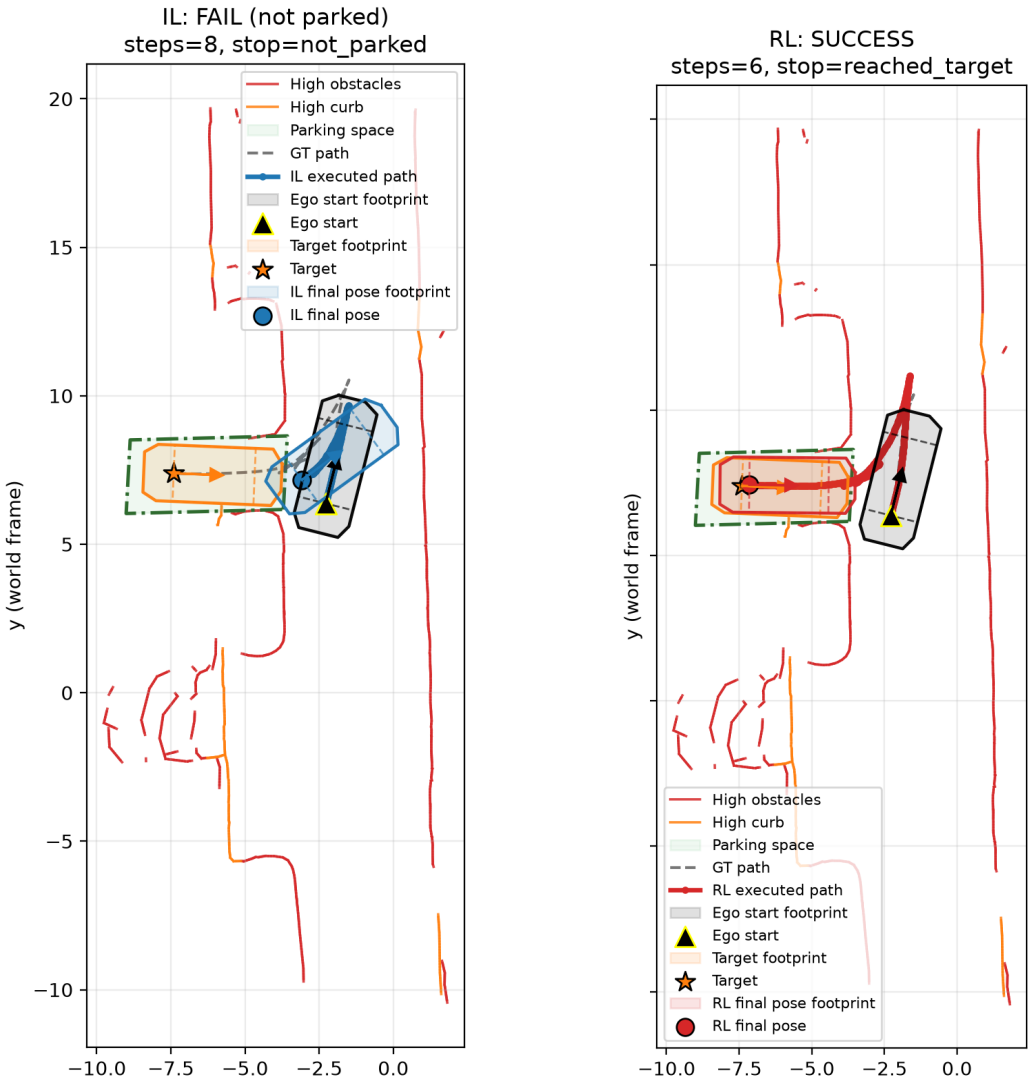


Figure 5: **Completing the parking maneuver.** In this selected cluster1216 example, the original IL planner remains not parked after eight steps, whereas the RL-fine-tuned planner reaches the target after six. This is a non-completion failure rather than the collision termination illustrated in Figure 3. Blank margins and the long filename header are omitted for readability; the two plotted panels are otherwise unchanged.

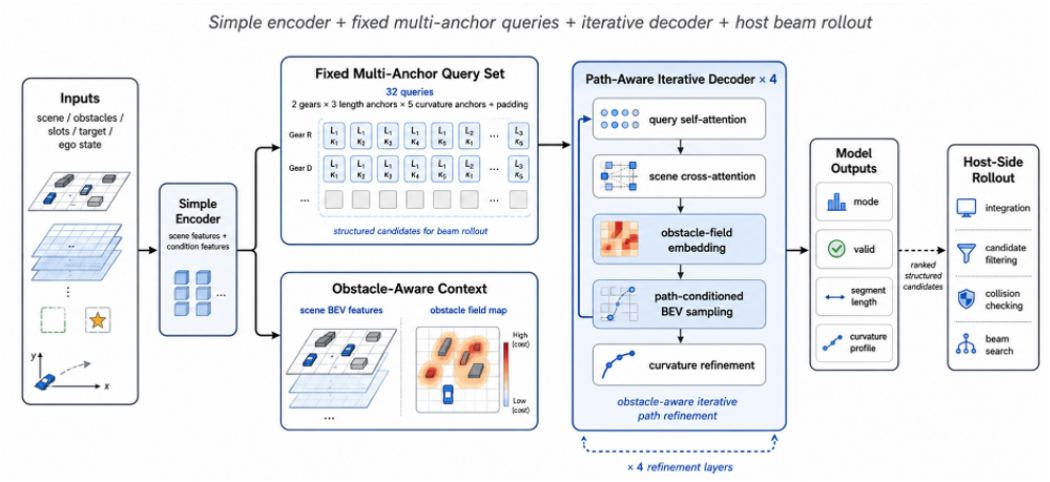


Figure 6: The base system: a structured segment policy deployed behind bounded test-time search. Encoders run once per scene, producing a BEV feature with 65 scene tokens and 2 condition tokens that are reused by every rollout branch. A fixed multi-anchor query bank — 2 gears $\times$ (3 length anchors $\times$ 5 curvature anchors + padding) $=$ 32 queries — defines the discrete candidate structure the deployed search expands. Four decoder layers refine each query with self-attention, scene/condition cross-attention, path-conditioned BEV and nearest-obstacle-field sampling (recomputed inside every layer along each query's current path), and curvature refinement; per-query heads emit mode score, validity, anchor-bounded length, and a 20-point curvature profile, integrated in FP32 into swept poses. The solid handoff into the host beam search $\mathrm{B}(W, k)$ — top-k expansion per parent, swept-footprint collision checking, width-W survival, first parked path wins — is the interface this paper ultimately evaluates. The rank-tree sampler uses beam-relevant local continuation ranks but deliberately omits this global pruning (§4.1). The dashed return edge is the autoregressive loop: decoding continues from each reached pose with the gear alternated.

B Inherited Parking Planner

The base policy (Figure 6) adapts MultiPark-style structured segment prediction (Zheng et al., 2025). We summarize it because the fine-tuning objective operates directly on this factorization; the architecture itself is not our contribution.

Demonstration construction. Each expert path is split at motion direction changes and adjacent pieces with the same gear are coalesced. Samples with more than $T{=}8$ alternating-gear segments are rejected. Retained geometry is expressed in the initial ego frame and resampled to $H{=}20$ equally spaced arc-length waypoints; headings are interpolated on the unit circle and signed curvature is recomputed from the resampled geometry. We export one initial state per scene containing the complete segment sequence. Importantly, all step-0 perception crops and target-conditioned maps are centered using the *requested final target*, not the expert's first pivot, preventing a teacher-only intermediate state from leaking into the policy input.

Encoders. Deterministic preprocessing produces a 20-channel obstacle raster and a 12-channel parking-slot raster on a 128×128 normalized BEV, plus a 4-channel nearest-obstacle field at resolution 256 for decoder-side geometric refinement. The BEV encoder fuses the two rasters with coordinate channels and uses an 8-D target embedding to FiLM-condition six residual blocks. It emits a pooled scene token and 8×8 pooled BEV tokens. A separate condition encoder emits vehicle-geometry and goal tokens; standstill ego-motion fields are omitted because they are constant or would reveal the first expert gear.

Structured decoder. The decoder carries a fixed query set: 2 gears $\times$ (3 length anchors $\{2.5, 6, 10\}$ m $\times$ 5 mean-curvature anchors $\{-0.2, -0.1, 0, 0.1, 0.2\}$ m^{-1} + one padding/EOS query) $=$ 32 queries. Four layers interleave query self-attention, scene cross-attention, direct nearest-obstacle-field injection, path-conditioned bilinear BEV sampling along each query's current path, and curvature refinement. Per query, parallel heads emit a mode logit, validity logit,

anchor-bounded segment length, and $H{=}20$ curvature values clamped by the sample's physical $\kappa_{\max}$. FP32 host-side integration produces the executable swept waypoints. Closed-loop decoding starts with both gears and then alternates gear after every executed segment.

Teacher forcing and matching. Training states are the expert segment starts plus an EOS state. Let L_t and $\bar{\kappa}_t$ be an expert segment's arc length and mean signed curvature. Within its expert gear g_t, the winner query is

$$q_t^* = \left(g_t, \arg\min_a |L_t - \bar{L}_a|, \arg\min_b |\bar{\kappa}_t - \bar{\kappa}_b| \right). \tag{9}$$

Only q_t^* receives continuous regression, preserving the other anchor cells as alternative modes (Chai et al., 2019; Shafiullah et al., 2022). Mode prediction is a 16-way classification problem *within each gear*; the expert gear uses a Gaussian soft target in anchor space with at least 0.7 mass on q_t^*, whereas the opposite gear and EOS target their padding queries. Validity uses an independently normalized focal BCE for positive anchor neighbours and hard negatives, avoiding dilution by roughly 31 negatives per state.

Imitation objective. The supervised objective is

$$\mathcal{L}_{\text{IL}} = \mathcal{L}_{\text{geom}} + \mathcal{L}_{\text{terminal}} + \mathcal{L}_{\text{class}} + \mathcal{L}_{\text{constraint}}, \tag{10}$$

with exact weights in Appendix C. Geometry uses Smooth-L1 on the winner's curvature, metric waypoint positions, and normalized length, plus the circular heading loss $1 - \cos(\hat{\psi} - \psi)$. Terminal supervision applies metric endpoint Smooth-L1, wrapped absolute heading error, and mean L1 displacement of the vehicle-contour corners. Constraint terms comprise an ego-centric signed-distance-field penalty on the four anchor-nearest candidates in the expert gear, an anchor-Voronoi hinge that prevents mode collapse across cells, and a steering-rate hinge. The training code selects checkpoints by validation validity-gated winner-decode recall rather than aggregate validation loss; this code rule does not by itself audit the reported checkpoint's data split.

Interpretation. Teacher-forced losses evaluate expert states, whereas closed-loop evaluation tests the states visited by generated plans. Their mismatch motivates rollout-based fine-tuning; a low aggregate imitation loss is not a direct measurement of covariate shift.

C Imitation-Learning Pipeline and Training Details

C.1 From expert paths to fixed-shape labels

The exporter pairs each scene JSON with its expert-path JSON and applies the same coordinate convention used by closed-loop inference. It validates the provided direction-change indices, assigns each raw piece a gear from its majority motion direction, and merges adjacent same-gear pieces because source control cusps can occur without a reversal. Samples requiring more than eight alternating strokes are rejected. Every retained stroke is transformed to the initial ego frame and resampled at 20 equal arc-length locations. We interpolate heading through sine and cosine and recompute $\kappa = d\psi/ds_{\text{signed}}$ from the resampled path, which preserves the decoder's sign convention for reverse motion. The resulting tensors contain segment-valid masks, gear, waypoints, curvature, and the terminal target, padded to eight segment positions; only the step-0 training item is exported.

The requested final target, rather than an expert intermediate pivot, defines the step-0 target features and all target-conditioned perception crops. This is an important anti-leakage contract: training and inference see identical obstacle regions, slot maps, and nearest-obstacle fields. We additionally keep metric obstacle points, chain connectivity, semantic route (high body, high curb, or low/tyre), per-scene safety distances, vehicle contour, and the exact reward target for deployment-faithful collision and RL evaluation.

C.2 Encoder and structured decoder

Preprocessing writes three dense geometric inputs: a 20-channel obstacle raster and 12-channel slot raster on a 128^2 normalized grid, and a 4-channel nearest-obstacle field on a 256^2 grid spanning $\pm 25\,\text{m}$. The first two are encoded by separate convolutional branches, concatenated with (x, y)

coordinate maps, and fused by a six-block BEV backbone FiLM-conditioned on the 8-D target embedding. Adaptive pooling produces one global scene token and 64 spatial tokens. The condition encoder produces two more 128-D tokens from six vehicle geometry scalars (length, width, overhangs, wheelbase, maximum curvature) and the goal. Ego motion is intentionally absent: scenes begin at standstill, initial heading is canonical in the ego frame, and feeding gear would leak the first expert action.

The 128-D decoder uses four layers, four attention heads, and a 256-D feedforward width. Its fixed query set is 2 gears $\times$ (3 length anchors $\{2.5, 6, 10\}$ m $\times$ 5 mean-curvature anchors $\{-0.2, -0.1, 0, 0.1, 0.2\}$ m^{-1} + padding) $= 32$ queries. Each layer performs query self-attention, scene/condition cross-attention, path-conditioned BEV sampling at 20 reference points with four bounded offsets, direct obstacle-field sampling, and curvature-delta refinement. The length head uses a piecewise-linear decode confined to the anchor's Voronoi interval; the curvature head is hard-clamped to the sample's maximum curvature ($\approx 1/5.4\,\mathrm{m}^{-1}$). Accurate FP32 integration maps the selected length and curvature profile to 20 poses and the next autoregressive state.

C.3 Teacher-forced objective

Teacher forcing packs only real expert segment starts and one EOS state. The matcher in Eq. 9 selects one winner in the expert gear; continuous targets supervise only this query. Each gear has a separate 16-way mode target. At a real state, the expert gear receives a Gaussian target over its anchor grid (σ_L=1.2 m, σ_κ=0.06 m^{-1}, winner mass at least 0.7), while the other gear targets padding. Both gears target padding at EOS. The validity soft band uses σ_L=2.0 m, σ_κ=0.06 m^{-1} and a minimum nonzero target 0.05.

Table 3: Imitation losses (winner-takes-all matched unless noted). Soft targets share probability across anchor-adjacent queries only (σ_ℓ=1.2, σ_κ=0.06, winner mass ≥ 0.7); the collision band uses a 2 cm ego-centric SDF with a 0.35 m soft safety margin on high obstacles.

Group	Term	Weight
Geometry	curvature profile Smooth-L1	1.0
	waypoint xy Smooth-L1 (1 m Huber scale)	0.5
	waypoint heading $1 - \cos(\Delta\psi)$	1.0
	segment length Smooth-L1 (25 m normalized)	1.0
Terminal	final xy Smooth-L1 (metric)	1.0
	final heading (wrapped absolute error)	2.0
	mean contour-corner L1 displacement	0.5
Classification	mode CE, anchor-soft targets, focal γ=1	0.2
	validity focal BCE (γ=2), per-population norm	1.0
Constraints	SDF collision band (top-4, high/low)	0.5 / 0.5
	anchor-Voronoi containment hinge	0.2
	steering-rate drivability hinge	1.0

The collision term queries an ego-centric signed-distance field constructed at the true vehicle body size. It takes the worst penetration over the swept path; high obstacles also receive a weak linear exterior band out to 0.35 m, while low obstacles are exempt on the terminal segment to match deployment. The anchor hinge keeps the length and mean curvature of each supervised band candidate within its own Voronoi cell, and the drivability term penalizes changes in bicycle steering angle per distance above the deployment judge's limit.

Imitation configuration. The configuration specifies AdamW, learning rate 5×10^{-4}, weight decay 10^{-4}, cosine decay with a 1,500-step linear warmup, BF16, batch 100 per rank, gradient-norm clipping at 1.0, and 200 epochs. Training data are shuffled; validation is not. The configured checkpoint-selection metric is validity-gated winner-decode recall on validation scenes rather than total loss, which can be dominated by mode-classification calibration.

D Group-Based Rank-Tree GRPO Implementation Details

Algorithm 1 Rank-tree GRPO: group-relative fine-tuning with rank-tree rollouts (one step)

1: **given** policy π_θ, frozen reference π_{ref} (imitation init), tree config (K, D), initial queries $1, \ldots, M$
2: collect group $G(x)$: M initial modes $\times$ K^D leaves with fixed continuation-rank digits from π_{ref}; sample raw geometry; roll out with swept-footprint collision checking
3: compute returns with shaped timeouts equation 6; advantages from equation 7 normalized over $G(x)$
4: PPO-clipped group-relative loss on geometry log-probs + per-component KL to π_{ref}
5: backprop RL loss; record $\|g_{\mathrm{RL}}\|$; backprop pure-imitation anchor on hard scenes; clip anchor gradient to $\eta_{\mathrm{anc}}\|g_{\mathrm{RL}}\|$ (absolute fallback for zero local norm); sum locally; average across ranks; global clip; optimizer step

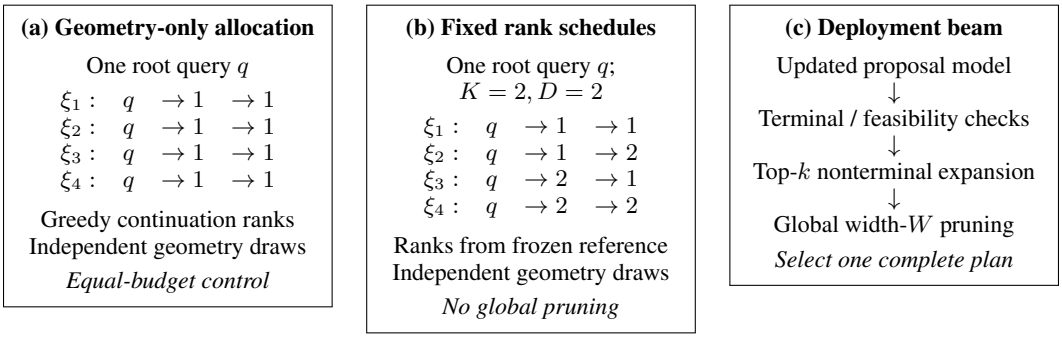


Figure 7: Training allocation is not deployment search. Panels (a,b) illustrate four rollout rows per root. Digits denote local continuation ranks, not query IDs. Each ξ_i denotes independent geometry draws throughout a row, including its root, so repeated digits do not imply shared geometric prefixes. After the two displayed decisions, rank 1 is used. Rows stop independently. Panel (c) instead uses the current model and global beam pruning; terminal candidates are checked under the existing evaluator's rules. The legacy sampler uses one greedy row per root, not the four shown in (a).

Algorithm 1 summarizes one complete training step of Rank-tree GRPO; the remainder of this appendix specifies its components. Figure 7 contrasts the training allocation with geometry-only sampling and with the deployment beam.

D.1 Raw-space policy, log probability, and KL

For curvature covariance $\Sigma_\rho = L_\rho L_\rho^\top$, one selected decision has

$$\log \pi_\theta(a_t \mid s_t, m_t) = \log \mathcal{N}(\kappa_t^{\mathrm{raw}}; \mu_{\theta,t}, \sigma_\kappa^2 \Sigma_\rho) + \log \mathcal{N}(\ell_t^{\mathrm{raw}}; \nu_{\theta,t}, \sigma_\ell^2), \tag{11}$$

$$D_\kappa = \frac{1}{2H\sigma_\kappa^2} \left\| L_\rho^{-1}(\mu_{\theta,t} - \mu_{\mathrm{ref},t}) \right\|_2^2, \qquad D_\ell = \frac{(\nu_{\theta,t} - \nu_{\mathrm{ref},t})^2}{2\sigma_\ell^2}. \tag{12}$$

The factor $1/H$ makes the 20-D curvature KL comparable to scalar components. The categorical KL is $D_{\mathrm{KL}}(p_\theta \| p_{\mathrm{ref}})$ over the 15 non-padding queries of each legal gear and is averaged across both gears at the root. Validity uses the exact Bernoulli KL. All four reference terms are computed at every active decision and averaged per trajectory. They are masked out, together with the PPO term, for groups with zero reward variance.

The curvature correlation is AR(1), $[\Sigma_\rho]_{ij} = \rho^{|i-j|}$ over the H waypoints. Raw actions collected from π_{old} are stored and detached; updates evaluate those same raw actions at the current means, not the density of a fixed noise vector alone. A deterministic parameter-free map (curvature clamp to $\pm\kappa_{\max}$ and anchor-bounded length decode) converts raw actions to executable geometry, so no physical-space Jacobian enters the likelihood. Rollouts are reused for two clipped passes with an approximate behavior-KL early stop; the entire decoder is trainable while both encoders remain frozen.

D.2 ROLLOUT, REWARD, AND COLLISION CONTRACT

The frozen reference decoder chooses every discrete query ID. Step 0 enumerates all 30 legal queries; for tree depth $s \leq D$, a precomputed base-K digit chooses reference rank $1, \ldots, K$; later steps use rank 1. The trainable decoder samples raw geometry for that query at every active step. Terminated leaves retain fixed tensor shape and may duplicate siblings if they finish before their assigned branch depth. The rollout stops at success, collision, or eight segments. Collision checking re-integrates poses at no more than $0.10\,\mathrm{m}$ spacing, tests connected obstacle segments as well as isolated points, uses separate high/high-curb/low footprints, and gives the final executed segment the same low-obstacle exemption as deployment.

Table 4: Parking fine-tuning configuration described in the implementation.
These values specify the recipe, not a verified manifest for every reported run.

Setting	Value
Rollout tree (M, K, D); group size	$(30, 2, 2)$; 120
Raw curvature noise σ_κ; AR(1) ρ	0.005; 0.9
Length-logit noise σ_ℓ	0.05
PPO clip; update passes; behavior-KL stop	0.2; 2; 0.05
Reference-KL weight (each component)	0.1
$r_{\mathrm{succ}}, r_{\mathrm{step}}, r_{\mathrm{coll}} = r_{\mathrm{fail}}$	$3, -0.2, -3$
Timeout shaping w, σ_d, σ_ψ	$2, 5\,\mathrm{m}, 45^\circ$
Failure-group scale λ_{fail}	0.5
Anchor budget; weight; excluded terms	0.25; 0.2; constraints
Learning rate; length-head multiplier	2×10^{-6}; 0.1
Global gradient-norm clip	1.0
Segment budget; goal tolerance	8; $0.5\,\mathrm{m}/10^\circ$
Deployment beam (W, k)	$(8, 6)$

Reward margin (Eq. 6). With $r_{\mathrm{fail}} = -3$, w=2, the shaped-timeout return is upper-bounded by -1; the worst success is $r_{\mathrm{succ}} + T\, r_{\mathrm{step}} = 3 - 1.6 = 1.4 > -1$, so every success outranks every failure for all allowed trajectory lengths; the trainer checks this inequality at construction. Advantages use population standard deviation and $\varepsilon = 10^{-6}$. A group with standard deviation below 10^{-6} is skipped; a nonflat all-failure group is retained at scale 0.5.

Imitation anchor. On training items marked as dead ends, the current decoder is teacher-forced on the original expert sequence. We sum the mode, validity, winner geometry, and terminal-pose losses, excluding collision-band, anchor-containment, and drivability terms because they can conflict with expert terminal contact against low obstacles. After multiplying by 0.2, this gradient is capped to $0.25\|g_{\mathrm{RL}}\|_2$ before addition. If the local RL gradient norm is zero, the fallback absolute cap is 0.3. This clipping precedes distributed averaging; the averaged merged gradient is then clipped to norm 1.0. The bound is local, not a bound on the averaged contribution or AdamW parameter movement.

Checkpoint evaluation. The online deterministic proxy uses one noise-free trajectory and is intentionally cheap; the 30-root deterministic and sampled diagnostics measure coverage and signal rate. Neither is the paper's acceptance metric. We retain epoch checkpoints and evaluate the reported epoch-15 model with the full beam-width-8/top-6 closed-loop evaluator. This matters because the proxy is top-1 while deployment is designed to recover from a poor top-1 candidate.

E TRAINING-SET DIAGNOSTICS

The reported success counts on the approximately 6K-scene training corpus are 3,094 for the base planner, 3,418 for legacy GRPO, 3,185 for legacy GRPO with hard-case data reweighting, and 3,996 for rank-tree GRPO. We retain counts because an exact common denominator has not been established. These are optimization diagnostics, not generalization evidence. In particular, training-set improvement in the legacy run coexists with regression on both dead-end evaluation suites.

An earlier train-on-test diagnostic reportedly increased success on the 359-scene suite from $304/359$ to $319/359$ (88.9%). It directly optimized that suite and used a different run configuration. It is neither a generalization result nor a bound on the separate 6K-trained model.

F SINGLE-STEP PROPOSAL FINE-TUNING: DRIVOR

This appendix describes the single-step driving specialization evaluated on NAVSIM v1, where fine-tuning improves the PDMS Score from 93.7 (matched re-evaluation of the starting checkpoint) to 93.9. DrivoR (Kirby et al., 2026) generates complete trajectories and applies a learned scorer. In our interface this gives $T = 1, D = 0$: selecting M reference-ranked proposal slots and taking S geometry samples per slot yields MS alternatives but no continuation branching. Top-M selection and sample allocation change the group; permuting its rows alone does not change an order-invariant GRPO loss. Frozen scorer weights do not freeze predicted scores or the selected trajectory, because the generator changes the scorer's inputs.

Stored-action likelihood. For each spatial axis, let z_i be the sampled waypoint vector and $\mu_{\theta,i}$ its current proposal mean. The implemented covariance is

$$C_i = r_i^2\big(D_\sigma \Sigma_\rho D_\sigma + 10^{-10} I\big), \qquad D_\sigma = \mathrm{diag}(\sigma_{xy} e_1, \ldots, \sigma_{xy} e_H).$$

The default ramp e_t increases from 0.3 to 1, with $\sigma_{xy} = 0.2$, $\rho = 0.9$, and absolute scale $r_i = 1$. The optional relative scale is detached and stored at collection. Actions $z_i = \mu_{\mathrm{old},i} + L_i \epsilon_i$ are collected without gradients and stored; the update evaluates $\sum_{a\in\{x,y\}} \log \mathcal{N}(z_i^a; \mu_{\theta,i}^a, C_i)$. It does not differentiate the density of fixed ϵ_i alone. Headings are recomputed deterministically from sampled positions. The positive default ramp and diagonal jitter avoid a singular covariance; a hypothetical zero-variance, no-jitter formulation would require a density restricted to nondegenerate coordinates.

Training return versus official metric. Let $p_i \in [0, 1]$ be the final PDMS score, and let $e_i, t_i, c_i \in [0, 1]$ be its progress, time-to-collision, and comfort components. The implementation's passing predicate is $p_i > 0$. Its shaped training return is

$$R_i^{\mathrm{drive}} = \begin{cases} 1 + p_i, & p_i > 0, \\ \gamma\ \mathrm{clip}((5e_i + 5t_i + 2c_i)/12, 0, 1), & p_i = 0, \end{cases} \qquad 0 < \gamma < 1.$$

The default $\gamma = 0.99$ gives passing returns in $(1, 2]$ and failing returns in $[0, 0.99]$. This ordering concerns the training return, not an altered official evaluation PDMS or a deployment safety guarantee. Multiplicative PDMS gates can otherwise make failed candidates share zero return. The generator is updated with clipped Gaussian ratios, reference KL, and an imitation anchor; backbone and scorer weights remain fixed.

Distributed anchor convention. Unlike the parking implementation, this module computes root-mean-square local gradient norms across ranks and uses one shared anchor coefficient. For gradients including the RL reference-KL term, define $n_{\mathrm{RL}}^2 = |\mathcal{R}|^{-1} \sum_r \|g_{\mathrm{RL},r}\|^2$ and $n_{\mathrm{IL}}^2 = |\mathcal{R}|^{-1} \sum_r \|w_{\mathrm{IL}} g_{\mathrm{IL},r}\|^2$. The coefficient is $\min(1, \eta n_{\mathrm{RL}}/(n_{\mathrm{IL}} + \varepsilon))$ when at least one rank reports a retained RL group, with an absolute-budget fallback otherwise. This bounds the RMS contribution, not each rank's contribution or the norm after gradient averaging.

Oracle decomposition. Scoring all 64 candidates with the ground-truth PDM evaluator yields an evaluation-only reference unavailable at deployment. The oracle-best candidate scores 99.33 PDMS for the starting checkpoint and 99.08 after fine-tuning (oracle-best equals 1.0 on 86.8% and 85.8% of scenes, respectively), while the scorer-selected score rises from 93.69 to 93.91; proposal coverage is therefore near-saturated, and fine-tuning instead narrows the oracle–selected conversion gap from 5.64 to 5.17 points. On the 12,147 matched scenes, the selected score improves on 3,908 scenes and worsens on 2,604 (5,635 ties), while the oracle-best improves on 709 and worsens on 1,165 (10,273 ties). Among the selected trajectory's subscores, ego progress ($0.899 \to 0.903$) and drivable-area compliance ($0.989 \to 0.992$) account for most of the gain, with the remaining components approximately unchanged. The scorer selects the ground-truth-best slot on only about 2.5% of scenes for both checkpoints, so realized performance depends on score alignment across the candidate set rather than exact top-1 agreement. The slight oracle decrease is consistent with mild distribution sharpening under RL fine-tuning: probability mass concentrates on proposals the frozen scorer selects, at a small cost in best-case candidate diversity.