# DRIVEBEYOND: RETHINKING GROUP ADVANTAGE FOR DRIVING POLICY OPTIMIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

End-to-end autonomous driving models are typically initialized through imitation learning (IL), after which reinforcement fine-tuning (RFT) is increasingly used to improve driving quality. Group Relative Policy Optimization (GRPO) has been adopted for this stage, estimating each trajectory's advantage from its reward relative to other trajectories sampled for the same scene. We revisit this group-relative construction and characterize two issues in its advantage estimation under driving RFT: **advantage shift**, where centering within each group discards the scene-level progress over training, and **advantage collapse**, where candidates receiving identical rewards yield vanishing advantages. Motivated by these findings, we propose **DriveBeyond**, a driving policy optimization method that incorporates references beyond the current sampling group through a shared memory module, calibrating advantages for groups with sufficient reward variation and helping recover learning signals for groups without. Experiments on NAVSIM v1 and v2 demonstrate that DriveBeyond consistently improves planning performance across both VLA- and WAM-based planners under three evaluation settings.

## 1 INTRODUCTION

End-to-end autonomous driving learns a direct mapping from sensor observations to driving trajectories or control actions (Hu et al., 2023; Jiang et al., 2023; Ma et al., 2025). Vision-language-action (VLA) models (Wang et al., 2026b;c; Luo et al., 2026) and world action models (WAMs) (Lu et al., 2026; Xu et al., 2026; Wang et al., 2026a) have emerged as two active directions in this field. Despite their architectural differences, both commonly learn driving policies through imitation learning (IL), using expert demonstrations as supervision. However, IL only encourages the policy to reproduce recorded expert behavior without directly optimizing the task rewards used to assess driving quality (Yang et al., 2026c). Reinforcement fine-tuning (RFT) (Zhou et al., 2025; Sheng et al., 2026; Wang et al., 2025) is therefore introduced as a post-IL stage that replaces trajectory matching with direct optimization of driving-quality rewards.

Among RFT algorithms, Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Guo et al., 2025) has become the dominant choice. GRPO samples a group of candidates for the same input and estimates each candidate's advantage from its reward relative to the group mean, eliminating the need to train a separate critic network. Motivated by this simplicity, a growing number of end-to-end planners (Fu et al., 2026; Zhou et al., 2025; 2026) adopt GRPO and report promising gains over their IL counterparts.

GRPO's design, however, is inherited from LLM mathematical reasoning, where the reward is a deterministic verifier of correctness (Shao et al., 2024), carrying the same meaning across problems. Even in this setting, recent work has identified limitations in its group-relative advantage estimation, including optimization bias (Liu et al., 2025) and vanishing gradients (Nan et al., 2025; He et al., 2026b). Driving RFT departs further from this setting as trajectory rewards are scene-dependent and the policy is initialized from an IL checkpoint already fitted to expert demonstrations. Despite these differences, existing driving methods adopt GRPO largely as-is, and how its advantage estimation behaves under these conditions has received little attention (Rawal et al., 2026; Zhou et al., 2026).

In this work, we revisit GRPO's advantage construction and analyze training rollouts. We find that, in driving RFT, a trajectory's group-relative advantage is not reliable for indicating whether it should be reinforced, which is caused by two issues discussed in Section 3. **Advantage shift**



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.

reflects a mismatch between advantages centered within the current group and scene-level progress. **Advantage collapse** concerns groups with little reward variation to guide learning. Both issues trace to the same root: the sampling group is the only reference in GRPO's advantage estimation, which is too narrow for the resulting advantages to reliably indicate which trajectories should be reinforced.

To address these issues, we introduce **DriveBeyond**, a driving RFT method that incorporates information beyond the current sampling group into advantage estimation through a shared memory module. The **Memory Module** stores a fixed IL mean reward and the latest mean reward for each training scene during RFT. Built on the memory, the **Progress Module** calibrates advantages using scene-specific IL rewards and recent training history for groups with sufficient reward variation, while the **Ghost Module** introduces a reference reward for low-variance groups, helping recover learning signals without altering the ranking among real candidates. Both modules operate only at training time and introduce no inference overhead.

We evaluate DriveBeyond on NAVSIM v1 (Dauner et al., 2024) and v2 (Cao et al., 2025) with both VLA-based (Xiong et al., 2026; Zhou et al., 2025) and WAM-based (Sheng et al., 2026) planners. Comprehensive experiments indicate that our proposed DriveBeyond consistently improves over GRPO and its variants. Our contributions are threefold:

- We revisit group-relative advantage estimation in driving RFT and, through an analysis of training rollouts, characterize and quantify advantage shift and advantage collapse in driving RFT, tracing both to the sampling group being GRPO's only reference.
- We propose DriveBeyond, which extends advantage estimation beyond the current group via a shared Memory Module, using a Progress Module to calibrate advantages against scene history and a Ghost Module to recover learning signals for low-variance groups.
- We demonstrate consistent gains over GRPO and GRPO-based variants across both VLA- and WAM-based planners on NAVSIM v1 and v2, supported by component ablations.

## 2 Preliminaries: Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is a PPO-style method that estimates advantages from sampled groups without training a separate value model. For each training scene $s \in \mathcal{D}_{\text{train}}$, the rollout policy $\pi_{\theta_{\text{old}}}$ samples $G$ candidate trajectories $\boldsymbol{o}_s = \{o_{s,1}, \ldots, o_{s,G}\}$. A simulator-based evaluator assigns each trajectory a reward $R_{s,i} \in [0, 1]$. The group mean and standard deviation are

$$\mu_s = \frac{1}{G}\sum_{i=1}^{G} R_{s,i}, \qquad \sigma_s = \sqrt{\frac{1}{G-1}\sum_{i=1}^{G}(R_{s,i}-\mu_s)^2}, \qquad G \geq 2. \tag{1}$$

Under outcome supervision, GRPO computes a shared trajectory-level advantage for the decisions within each sampled trajectory:

$$A_{s,i} = \frac{R_{s,i} - \mu_s}{\sigma_s + \epsilon}, \tag{2}$$

where $\epsilon > 0$ ensures numerical stability.

Each planner retains its original GRPO loss implementation. Let $p$ index the planner and $\mathcal{S}_p$ denote its group-level optimization surrogate, including its likelihood-ratio construction, clipping, aggregation, and any reference-policy regularization. Writing $\boldsymbol{A}_s = \{A_{s,1}, \ldots, A_{s,G}\}$, the policy objective is

$$\mathcal{J}_p(\theta) = \mathbb{E}_{\substack{s\sim\mathcal{D}_{\text{train}} \\ \boldsymbol{o}_s\sim\pi_{\theta_{\text{old}}}(\cdot|s)}} \left[\mathcal{S}_p\left(\theta; s, \boldsymbol{o}_s, \boldsymbol{A}_s\right)\right]. \tag{3}$$

Standard GRPO supplies the advantages in equation (2); DriveBeyond replaces them with the advantages defined in Section 4. Throughout, we say that a sampled trajectory is reinforced when it receives a positive advantage, i.e., when the clipped surrogate in equation (3) increases its likelihood ratio, and penalized otherwise; the sign of $A_{s,i}$ thus decides the direction in which each sampled trajectory is pushed. Appendix A presents the original token-level formulation and its policy-gradient background.

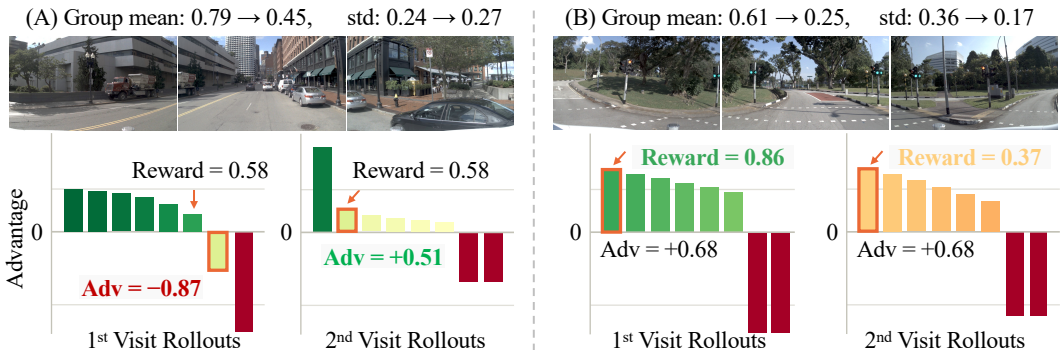



Figure 1: **Advantage shift across two visits to the same scene.** Rollouts are sorted by GRPO advantage and colored by reward. (A) The same reward, 0.58, receives $-0.87$ and then $+0.51$ as the group mean falls. (B) Rewards 0.86 and 0.37 receive the same $+0.68$ as the group mean and standard deviation fall together.

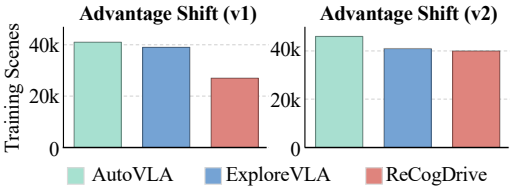



Figure 2: Shift samples during original GRPO post-training on NAVSIM v1 (left) and v2 (right) for three driving planners.

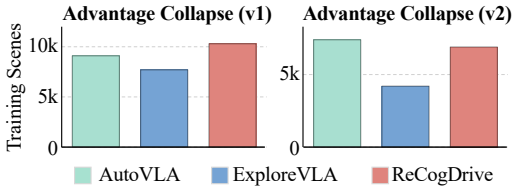



Figure 3: Collapse samples during original GRPO post-training on NAVSIM v1 (left) and v2 (right) for three driving planners.

## 3 Deep Look at GRPO in Driving Post-Training

We revisit GRPO's advantage construction to examine whether within-group comparison is sufficient to guide reinforcement in driving RFT. We analyze GRPO training rollouts of two VLA-based planners, ReCogDrive (Xiong et al., 2026) and AutoVLA (Zhou et al., 2025), and one WAM-based planner, ExploreVLA (Sheng et al., 2026), trained on the NAVSIM navtrain split (103k scenes) and evaluated on NAVSIM v1 and v2, and identify two issues: **advantage shift** and **advantage collapse**.

### 3.1 Advantage Shift

We define **advantage shift** as a mismatch between a trajectory's group-relative advantage and the policy's progress on the same scene: a trajectory can receive a positive advantage even though it scores below what the policy previously achieved on that scene, and an improvement over earlier visits is not necessarily rewarded with a larger advantage. The mismatch arises because equation (2) evaluates each trajectory relative to the current group: $A_{s,i}^{(k)} > 0$ if and only if $R_i > \mu_s^{(k)}$. The mean sets the zero point of the advantages and changes across visits. When it falls, a trajectory can receive a positive advantage without any increase in its reward. More generally, adding the same constant to all rewards in a group leaves equation (2) unchanged. Since the current group is its only reference, GRPO's advantages therefore do not reflect whether rewards have improved over earlier visits.

Figure 1(A) shows that trajectories with the same reward can receive advantages of opposite signs across visits to the same scene. The group mean falls from 0.79 to 0.45. A reward of 0.58 corresponds to an advantage of $-0.87$ in the earlier visit and $+0.51$ in the later one. The sign reverses because 0.58 is below the earlier mean but above the later one, showing that a positive advantage need not indicate high trajectory quality. In Figure 1(B), the best trajectories from two visits receive nearly identical advantages despite very different rewards. The highest reward falls from 0.86 to 0.37, while the group mean ($0.61 \rightarrow 0.25$) and standard deviation ($0.36 \rightarrow 0.17$) also decrease. Thus, both winners receive advantages of approximately $+0.68$, reinforcing the later winner as strongly as the earlier one despite scoring below the earlier group's mean.

Advantage shift is not limited to isolated cases. We count advantage shift by comparing rollout pairs from consecutive visits to the same scene. The current visit is counted if the lower-reward rollout in any pair receives an equal or higher group-relative advantage. Equal-reward rollouts with different

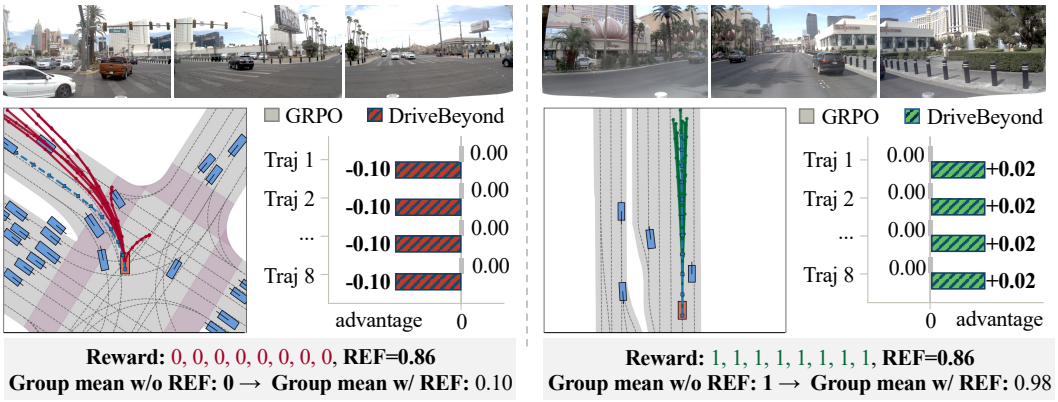



Figure 4: **Advantage collapse at different reward levels.** Each scene has a group of trajectories with identical rewards: 0 (left) and 1 (right). GRPO assigns zero advantage to every trajectory in both groups. Hatched bars show the nonzero advantages assigned by DriveBeyond, whose construction is described in Section 4.

group-relative advantages also qualify. Figure 2 shows the average number of training scenes per epoch that exhibit advantage shift under standard GRPO training across the three planners and both benchmarks. In driving, rewards encode graded trajectory quality, so a relative winner can still represent poor driving behavior. Assigning it a positive advantage can reinforce that behavior, even when higher-reward behavior has already been achieved in the same scene.

### 3.2 Advantage Collapse

We define **advantage collapse** as a condition in which the trajectories in a group receive identical or nearly identical rewards, leaving little within-group contrast for equation (2). In the limiting case where every trajectory receives the same reward $r$, the group mean equals $r$, so every advantage is zero. The group then contributes no reward-driven policy gradient.

In driving RFT, driving rewards usually combine hard sub-metrics multiplicatively: a single violation, such as a collision or leaving the drivable area, sets the reward to 0, while in easy scenes every trajectory can reach the maximum. Figure 4 visualizes this effect using two groups with identical rewards within each group. In the left scene, every sampled trajectory receives reward 0; in the right scene, every trajectory receives reward 1. These groups occupy opposite ends of the reward range, yet GRPO maps both to an all-zero advantage group, and the advantages retain no information about the reward level each group achieved. It therefore supplies no reward-driven signal to discourage the low-reward behaviors or reinforce the high-reward ones.

Advantage collapse recurs throughout training. Counting only scene visits whose rewards are exactly tied, a conservative criterion that excludes nearly tied groups, Figure 3 shows that under standard GRPO training, 4k–10k training scenes collapse per epoch across the three planners and both benchmarks. For each of these scenes, all rollouts are generated but yield no reward-driven gradient. In driving, all-zero groups can contain safety-critical failures such as collisions. Dynamic sampling in DAPO (Yu et al., 2025) filters out groups with uniformly correct or incorrect outcomes and continues sampling to fill the training batch. Analogous filtering in driving would exclude all-zero failure groups from the current update without recovering a reward-driven signal from them. This motivates extracting learning signals from collapsed groups, including those with safety-critical failures.

## 4 DriveBeyond

### 4.1 Overview

As illustrated in Figure 5, DriveBeyond keeps the rollout and policy update of GRPO unchanged and modifies only the advantage computation through three components: the Memory Module, the Progress Module, and the Ghost Module. The Memory Module stores two records for each training scene $s \in \mathcal{D}_{\text{train}}$: a fixed IL mean reward $\mu_s^{\text{IL}}$ obtained before RFT, and the group mean from its previous visit $\mu_s^{(k-1)}$, updated during training. Averaging $\mu_s^{\text{IL}}$ over all training scenes further gives a global IL reference reward REF (Section 4.2).

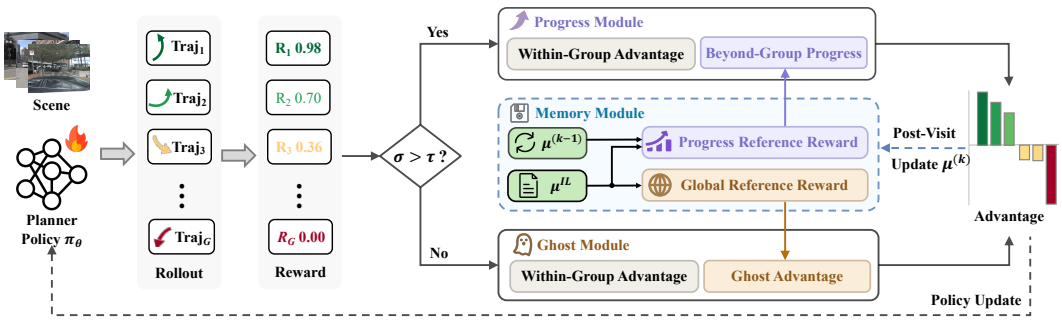



Figure 5: **Framework of DriveBeyond.** Each sampled group is routed by its reward spread. Groups with sufficient spread enter the Progress Module, which calibrates within-group advantages with a beyond-group progress term from the scene's stored IL and previous-visit means; the remaining groups enter the Ghost Module, which adds a ghost advantage from the global IL reference reward. The Memory Module provides both references and is updated with the current group mean after each visit.

On the $k$-th visit to scene $s$, the policy samples a group of $G$ trajectories with rewards $\{R_i\}_{i=1}^{G}$, whose mean and standard deviation are denoted by $\mu_s^{(k)}$ and $\sigma_s^{(k)}$. The group is then routed by its reward spread using a threshold $\tau$. If $\sigma_s^{(k)} > \tau$, the Progress Module calibrates within-group advantages with a beyond-group progress term computed from $\mu_s^{\mathrm{IL}}$ and $\mu_s^{(k-1)}$, so that the advantages also reflect the policy's progress on the same scene (Section 4.3). Otherwise, the group provides little within-group contrast, and the Ghost Module introduces REF as a reference reward to recover the learning signal (Section 4.4). After the advantages are computed, the stored previous-visit record of $s$ is replaced with $\mu_s^{(k)}$, which serves as $\mu_s^{(k-1)}$ at the next visit.

### 4.2 Memory Module

The Memory Module provides the beyond-group references used by the other two modules. For each training scene, it stores two records: an IL mean reward computed once before RFT and kept fixed, and the group mean from the previous visit, updated at every visit.

**IL initialization.** Before RFT, we use the frozen IL policy $\pi_{\mathrm{IL}}$ to sample $G$ trajectories for each training scene $s$, using the same sampling configuration as in RFT, and compute their rewards $\{R_i^{\mathrm{IL}}\}_{i=1}^{G}$. Their mean, denoted by $\mu_s^{\mathrm{IL}}$, measures the initial performance of the policy on scene $s$. Averaging over all training scenes gives the global IL reference reward $\mathrm{REF} = \frac{1}{|\mathcal{D}_{\mathrm{train}}|}\sum_{s\in\mathcal{D}_{\mathrm{train}}} \mu_s^{\mathrm{IL}}$. Since $\mu_s^{\mathrm{IL}}$ remains fixed throughout RFT, so does REF. The Progress Module reads the scene-specific $\mu_s^{\mathrm{IL}}$, while the Ghost Module reads the global REF.

**Training history and updates.** The second record is the group mean from the previous visit to each scene, $\mu_s^{(k-1)}$, initialized as $\mu_s^{(0)} = \mu_s^{\mathrm{IL}}$ before the first visit. At the $k$-th visit, DriveBeyond first reads the stored records to compute the current group's advantages, and then replaces the previous-visit record with $\mu_s^{(k)}$. This read-before-write order ensures that each group is compared against the policy's preceding performance on the same scene rather than against itself. The update is applied at every visit, regardless of whether the group is routed to the Progress or the Ghost Module.

### 4.3 Progress Module

The Progress Module targets advantage shift and applies to groups with $\sigma_s^{(k)} > \tau$. It combines the within-group reward differences of GRPO with a measure of the policy's progress on the same scene. Because driving scenes differ in difficulty, progress is measured against the scene's own records in the Memory Module rather than a shared reference. Each trajectory's advantage is computed as:

$$A_{s,i}^{(k)} = \underbrace{w_a\left(R_i - \mu_s^{(k)}\right)}_{\text{within-group advantage}} + \underbrace{w_p\left(\mu_s^{(k)} - \max\left(\mu_s^{\mathrm{IL}}, \mu_s^{(k-1)}\right)\right)}_{\text{beyond-group progress}}, \quad \sigma_s^{(k)} > \tau, \tag{4}$$

where $w_a > 0$ and $w_p \geq 0$ weight the two terms.

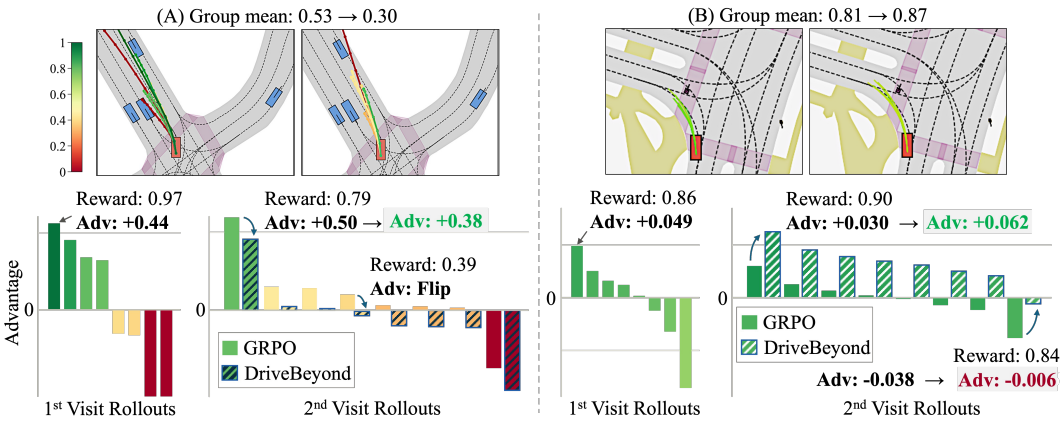



Figure 6: **Advantages from the Progress Module on two consecutive visits to the same scene.** Upper: sampled trajectories colored by reward. Lower: advantages under GRPO (solid) and DriveBeyond (hatched); the earlier visit provides the progress reference for the later one. (A) The group mean decreases ($0.53 \rightarrow 0.30$), lowering all advantages. (B) The group mean increases ($0.81 \rightarrow 0.87$), raising all advantages.

The within-group term compares each trajectory with the current group mean, while the progress term shifts all advantages in the group by the same amount according to the policy's progress on the scene. The progress reference $\max(\mu_s^{\mathrm{IL}}, \mu_s^{(k-1)})$ compares the current group with the policy's most recent performance on the scene, and taking the maximum with the IL mean prevents this reference from falling below the scene's initial reward level. On the first visit, the initialization $\mu_s^{(0)} = \mu_s^{\mathrm{IL}}$ reduces the progress term to $w_p(\mu_s^{(1)} - \mu_s^{\mathrm{IL}})$. When the current group mean falls below the larger of the two stored means, for $w_p > 0$, the progress term is negative and lowers the advantages of all trajectories in the group, including relative winners.

Figure 6 illustrates this effect with $w_a = 1$ and $w_p = 0.5$. In (A), the group mean drops from $0.53$ to $0.30$. The best reward falls from $0.97$ to $0.79$, yet the corresponding within-group advantage rises from $+0.44$ to $+0.50$ as the group deteriorates. The negative progress term lowers this advantage to $+0.38$ and changes the advantage of a trajectory scoring $0.39$, above the current group mean but below the previous one, from positive to negative. In (B), the group mean rises from $0.81$ to $0.87$. The best reward increases from $0.86$ to $0.90$, but the corresponding within-group advantage falls from $+0.049$ to $+0.030$ as peers also improve. The positive progress term raises it to $+0.062$. In both cases, the progress term aligns the advantages with whether the policy has improved on the scene, which within-group comparison alone cannot reflect. With the reference fixed at sampling, the progress term changes which trajectories are reinforced and the update variance (Appendix D).

### 4.4 Ghost Module

The Ghost Module targets advantage collapse and applies to groups with $\sigma_s^{(k)} \leq \tau$. It retrieves the global IL reference reward $\mathrm{REF}$ from the Memory Module and uses it as a ghost reward, providing a reference for groups with little within-group variation. Each real trajectory receives the advantage:

$$A_{s,i}^{(k)} = \underbrace{w_a \left(R_i - \mu_s^{(k)}\right)}_{\text{within-group advantage}} + \underbrace{\frac{\mu_s^{(k)} - \mathrm{REF}}{G+1}}_{\text{ghost advantage}}, \qquad \sigma_s^{(k)} \leq \tau. \tag{5}$$

The within-group term compares each trajectory with the current group mean and is scaled by $w_a$. The ghost advantage measures the gap between the current group mean and $\mathrm{REF}$, divided by $G+1$ since the ghost is counted together with the $G$ real rewards. It is positive when the group performs above the global IL level and negative when it performs below. Since this term is identical for every real trajectory, it shifts all advantages in the group by the same amount and leaves the ranking among them unchanged. Since rewards lie in $[0, 1]$, its magnitude is bounded by $1/(G+1)$.

For an exactly tied group with reward $r$, the within-group term vanishes and every trajectory receives $A_{s,i}^{(k)} = (r - \mathrm{REF})/(G+1)$. Tied groups above and below the global IL level therefore receive positive and negative advantages, respectively, even without within-group reward differences. We

Table 1: **Overall performance on NAVSIM.** We report PDMS on NAVSIM v1 and EPDMS on NAVSIM v2 (navtest and navhard). IL denotes the checkpoint before RFT, from which all other methods are trained.

| Planner | Method | PDMS↑ v1 navtest | EPDMS↑ v2 navtest | EPDMS↑ v2 navhard |
|---|---|---|---|---|
| VLA-based Planner | | | | |
| ReCogDrive (Xiong et al., 2026) | IL | 86.4 | 86.2 | 26.2 |
| | GRPO (2024) | 90.5 | 86.3 | 29.9 |
| | Dr.GRPO (2025) | 90.2 | 85.9 | 29.7 |
| | DAPO (2025) | 90.4 | 86.6 | 30.0 |
| | NGRPO (2025) | 90.5 | 86.7 | 31.7 |
| | AVSPO (2026b) | 90.6 | 86.8 | 32.2 |
| | **Ours** | **90.9** | **87.3** | **33.9** |
| AutoVLA (Zhou et al., 2025) | IL | 80.4 | 76.8 | 24.1 |
| | GRPO (2024) | 89.3 | 82.4 | 34.0 |
| | Dr.GRPO (2025) | 90.1 | 83.3 | 34.5 |
| | DAPO (2025) | 90.2 | 82.6 | 35.2 |
| | NGRPO (2025) | 89.5 | 83.0 | 35.8 |
| | AVSPO (2026b) | 89.7 | 83.3 | 36.0 |
| | **Ours** | **90.6** | **84.1** | **37.2** |
| WAM-based Planner | | | | |
| ExploreVLA (Sheng et al., 2026) | IL | 88.5 | 88.3 | 31.1 |
| | GRPO (2024) | 90.3 | 88.9 | 31.2 |
| | Dr.GRPO (2025) | 90.4 | 90.2 | 32.7 |
| | DAPO (2025) | 90.4 | 90.0 | 33.6 |
| | NGRPO (2025) | 90.3 | 89.4 | 34.0 |
| | AVSPO (2026b) | 90.5 | 90.1 | 34.7 |
| | **Ours** | **90.8** | **90.6** | **35.9** |

use the global REF rather than the scene-specific $\mu_s^{\mathrm{IL}}$ because tied groups typically arise in very easy or hard scenes, where the IL mean often takes the same value leaving the advantages at zero.

For the two groups in Figure 4, with $G = 8$ and $\mathrm{REF} = 0.86$, this gives $-0.10$ for each trajectory in the all-zero group and $+0.02$ in the all-one group, whereas GRPO assigns zero to both. For a tied group, this advantage yields a nonzero expected policy-gradient contribution that, on its own, pushes the policy away from uniformly failing and toward uniformly succeeding on the scene (Appendix D.4). The ghost is a scalar reward with no associated trajectory. It contributes no log-probability, probability ratio, or KL term, and is excluded from the policy-loss average.

### 4.5 Integration into Planners

Routing selects a module whose advantages for each group's $G$ real trajectories replace the GRPO advantages in equation (3). All other objective components, including the probability ratio, clipping, and KL regularization, remain unchanged; no value model is fitted. Using only scalar rewards, DriveBeyond is agnostic to policy architecture and output representation. It requires only GRPO-style group sampling and scene identification across repeated visits to index the Memory Module. Both hold for GRPO-based VLA planners using autoregressive token generation or diffusion decoding, and WAM planners using world-model decoding. The overhead is one IL policy pass over the training set before RFT and two scalars per scene, with no added network, loss term, or inference-time computation. We apply DriveBeyond to VLA- and WAM-based planners in Section 5.

## 5 Experiments

### 5.1 Experimental protocol

**Planners.** We apply DriveBeyond to two VLA-based planners, ReCogDrive (Xiong et al., 2026) and AutoVLA (Zhou et al., 2025), and one WAM-based planner, ExploreVLA (Sheng et al., 2026).

**Benchmarks and metrics.** All planners are trained on the navtrain split and evaluated on the navtest split of NAVSIM v1 (Dauner et al., 2024) and on the navtest and navhard splits of NAVSIM v2 (Cao et al., 2025). We report PDMS on NAVSIM v1 and EPDMS on NAVSIM v2.

**Baselines and protocol.** We compare DriveBeyond against standard GRPO (Shao et al., 2024), Dr. GRPO (Liu et al., 2025), which has also been adopted for driving RFT (Rawal et al., 2026), DAPO (Yu et al., 2025), NGRPO (Nan et al., 2025), AVSPO (He et al., 2026b), and additionally report the IL checkpoint before RFT. We reproduce these optimization methods in the driving RFT setting. For each planner, all methods start from the same IL checkpoint and use the same training data, reward function, and training settings; they differ only in the GRPO components specified by

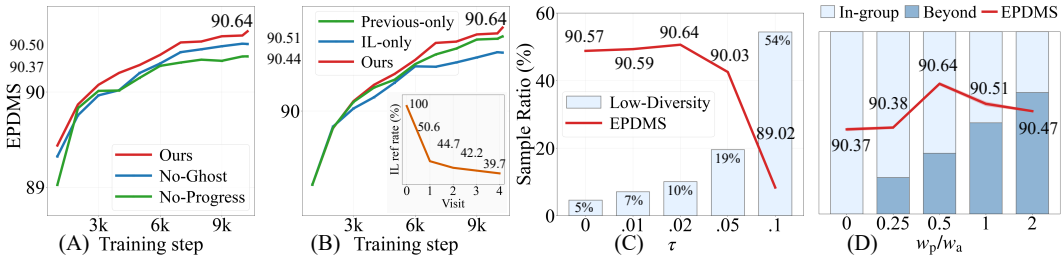



Figure 7: Ablations on ExploreVLA (EPDMS, NAVSIM v2 navtest). **(A) Component analysis:** training curves of DriveBeyond (Ours) and variants without the ghost advantage (No-Ghost) or the beyond-group progress term (No-Progress). **(B) Progress reference:** training curves using the fixed IL mean $\mu_s^{\mathrm{IL}}$ (IL-only), the previous-visit group mean $\mu_s^{(k-1)}$ (Previous-only), or their maximum (Ours). The inset shows the fraction of training scenes for which the IL mean supplies the reference at each visit. **(C) Routing threshold:** proportion of rollout samples in groups with $\sigma_s^{(k)} \leq \tau$ and the corresponding EPDMS for different $\tau$. **(D) Weight sensitivity:** EPDMS for different ratios $w_p/w_a$ with $w_a$ fixed. Stacked bars show the normalized shares of the within-group and beyond-group progress weights.

each method. DriveBeyond retains each planner's original GRPO loss implementation and modifies only the advantage construction. Implementation details are provided in Appendix B.1 to B.4.

### 5.2 Main Results

Table 1 reports the overall performance. DriveBeyond consistently achieves the best score across all planners and benchmarks. Relative to standard GRPO, it improves PDMS on NAVSIM v1 navtest by 0.4–1.3, EPDMS on NAVSIM v2 navtest by 1.0–1.7, and EPDMS on navhard by 3.2–4.7. By contrast, existing GRPO variants do not improve consistently in driving: Dr. GRPO falls below GRPO on all three benchmarks for ReCogDrive, and the strongest baseline varies across planners and benchmarks. These results support our analysis in Section 3 that the limitations of group-relative advantages in driving RFT are not resolved by existing modifications to GRPO.

The gains increase with benchmark difficulty. On navtest v1, where all RFT methods reach around 90 PDMS, the margins are modest. On navhard, DriveBeyond improves over GRPO by up to 4.7 EPDMS (ExploreVLA, 31.2 $\rightarrow$ 35.9). The gains also hold for all three planners, covering both VLA-based (ReCogDrive, AutoVLA) and WAM-based (ExploreVLA) architectures and IL starting points ranging from 76.8 to 88.3 EPDMS on NAVSIM v2 navtest. This suggests that the benefit of DriveBeyond is not tied to a specific planner design or initial performance level. Complete results, including all component metrics, are provided in Appendix C.1.

### 5.3 Ablation Studies

We conduct ablations with ExploreVLA on NAVSIM v2 navtest. Unless otherwise specified, we use $\tau = 0.02$, $w_a = 10$, and $w_p = 5$. Qualitative results are provided in Appendix E.

**Component analysis.** Figure 7(A) indicates the contributions of the two modules. The No-Progress variant sets $w_p = 0$, retaining the within-group advantage and the Ghost Module. The No-Ghost variant keeps the Progress Module but computes $A_{s,i}^{(k)} = w_a\big(R_i - \mu_s^{(k)}\big)$ for groups with $\sigma_s^{(k)} \leq \tau$, so that exactly tied groups receive zero advantages as in GRPO. Removing either module lowers the final EPDMS, from 90.64 to 90.37 without the progress term and to 90.50 without the ghost advantage, indicating that both modules contribute to planning performance.

**Progress reference.** Figure 7(B) compares three choices of reference for the beyond-group progress term. IL-only uses the scene's fixed IL mean $\mu_s^{\mathrm{IL}}$, Previous-only uses its previous-visit group mean $\mu_s^{(k-1)}$, and DriveBeyond uses their maximum, as in equation (4). Since $\mu_s^{(0)} = \mu_s^{\mathrm{IL}}$, all three start from the same reference, and all keep REF in the Ghost Module unchanged. The variants perform similarly early in training but diverge later, reaching 90.44 for IL-only, 90.51 for Previous-only, and 90.64 for DriveBeyond. The inset explains this trend by tracking the fraction of training scenes for which $\mu_s^{\mathrm{IL}} \geq \mu_s^{(k-1)}$, i.e., where the IL mean supplies the reference. This fraction starts at 100% by initialization and falls to 39.7% after four epochs, as the policy surpasses the IL level on most scenes. On these scenes, a fixed IL reference keeps assigning a positive progress term even when the

policy regresses relative to its previous visit, which accounts for the lower EPDMS of IL-only. On the remaining scenes, the policy still performs below the IL level, and taking the maximum prevents the reference from falling below it, which Previous-only does not.

**Routing threshold.** We vary the threshold $\tau$ that routes groups between the two modules, keeping other settings unchanged. Figure 7(C) reports the proportion of rollout samples in groups with $\sigma_s^{(k)} \leq \tau$ over the full RFT run, and the corresponding EPDMS. At $\tau = 0$, only exactly tied groups, about $5\%$ of the samples, use the Ghost Module, already giving an EPDMS of $90.57$. Extending it to groups with small nonzero reward spreads gives comparable results, with $90.59$ at $\tau = 0.01$ and a slightly higher $90.64$ at $\tau = 0.02$. Larger thresholds route groups with substantial reward differences to the Ghost Module, covering over $50\%$ of the samples at $\tau = 0.1$, and EPDMS drops to $90.03$ at $\tau = 0.05$ and $89.02$ at $\tau = 0.1$. This drop goes well below the No-Progress variant, suggesting that the global ghost shift, which does not account for scene difficulty, is harmful when applied to groups that already have meaningful within-group contrast. The Ghost Module is therefore effective as a fallback for low-diversity groups, but not as a replacement for the Progress Module.

**Weight sensitivity.** We fix $w_a$ and vary the ratio of the progress weight to the within-group weight, $w_p/w_a \in \{0, 0.25, 0.5, 1, 2\}$, where a ratio of 0 corresponds to the No-Progress variant. Figure 7(D) shows the corresponding normalized weight shares together with the mean EPDMS and its standard deviation over three seeds. The mean EPDMS increases from $90.37$ at a ratio of 0 to $90.64$ at $0.5$, and decreases to $90.51$ and $90.47$ at 1 and 2, respectively. A moderate progress weight thus works best, while larger weights bring no further gains. Notably, all nonzero ratios ($90.38$–$90.64$) outperform the strongest baseline on this benchmark in Table 1 (Dr. GRPO, $90.2$), indicating that DriveBeyond is not sensitive to the precise choice of $w_p$.

## 6 Related Work

**RL Post-Training for Autonomous Driving.** RL post-training refines IL-initialized driving policies with rewards from sampled trajectories (Xiong et al., 2026; Rowe et al., 2025; Li et al., 2026). VLM-based approaches also optimize planning reasoning and its link to driving decisions (Jiang et al., 2025b; Zhou et al., 2025; Wang et al., 2025; Ye et al., 2026); world models support representation learning, trajectory evaluation, and exploration (Yang et al., 2026a; Jiang et al., 2025a; Sheng et al., 2026; Yang et al., 2026b); negative and recovery examples give feedback on failures (Zhou et al., 2026; Tian et al., 2026). Other methods adapt advantages to trajectory modes and reward distributions (Zou et al., 2025; Bai et al., 2026) or distill RL teachers into policies (He et al., 2026a). DriveBeyond instead incorporates IL and historical rewards into advantage construction.

**Advantage Estimation in GRPO.** Critic-free estimators build baselines from sampled or greedily decoded outputs (Shao et al., 2024; Ahmadian et al., 2024; Li et al., 2024), and dropping the standard-deviation normalization preserves reward-gap magnitudes in language and driving post-training (Liu et al., 2025; Rawal et al., 2026; Zhou et al., 2026). For tied groups, existing methods filter them out (Yu et al., 2025; Bai et al., 2026), shape advantages with token entropy (Le et al., 2026), add goal-centered comparisons (Salmani-Zarchi et al., 2026), or append virtual rewards and re-standardize, using the maximum reward in NGRPO (Nan et al., 2025) or adaptive values in AVSPO (He et al., 2026b). DriveBeyond changes both references: its baseline is the scene's previous-visit mean with a fixed IL floor, so advantages measure progress on the scene, and its ghost reward is the fixed IL mean over training scenes, added without re-normalization, so all-success groups still receive a positive advantage bounded by $1/(G+1)$.

## 7 Conclusion

We presented DriveBeyond, which incorporates rewards from IL initialization and training history into advantage construction through a scene-specific progress term and a global IL ghost reward. Across three VLA- and WAM-based driving planners and three NAVSIM evaluation settings, DriveBeyond consistently improves aggregate driving scores over original GRPO.

**Limitations** DriveBeyond uses scene-specific reward memory across visits; its effectiveness in streaming RFT remains unexplored. Sharing references across related scenes is a future direction. The Ghost Module's shared advantage adjustment adds no relative-quality information within groups with low reward diversity. Better trajectories still depend on policy sampling; targeted exploration could improve learning when successful trajectories are rarely sampled.

## AI USE STATEMENT

We used Codex and Claude for grammar correction and language polishing. All research content, literature analysis, and writing were conducted independently by the authors.

## ETHICS STATEMENT

This work studies reinforcement fine-tuning of autonomous driving planners on NAVSIM v1 and v2. Since autonomous driving is safety-critical, improvements in benchmark performance do not establish safety in real-world deployment. Applying the proposed method to real vehicles would require further validation under diverse driving conditions and assessment of potential safety risks.

## REPRODUCIBILITY STATEMENT

Section 4 describes DriveBeyond, including memory initialization and updates and the advantage formulations of the progress and ghost modules. Section 5.1 describes the benchmarks, planners, and experimental protocol. Appendix B provides additional implementation details and hyperparameter settings for reproducing training and evaluation. We provide anonymous code for reproducing our experimental results at https://anonymous.4open.science/r/drivebeyond-915B/.

## REFERENCES


Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *ACL*, pp. 12248–12267, 2024.

Haojie Bai, Aimin Li, Ruoyu Yao, Xiongwei Zhao, Tingting Zhang, Xing Zhang, Lin Gao, et al. SCORP: Scene-consistent multi-agent diffusion planning with stable online reinforcement post-training for cooperative driving. *arXiv preprint arXiv:2604.11734*, 2026.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. In *CoRL*, pp. 4709–4722, 2025.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *NeurIPS*, volume 37, pp. 28706–28719, 2024.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Hongwei Xie, Bing Wang, Guang Chen, Hangjun Ye, Dingkang Liang, and Xiang Bai. MindDrive: A vision-language-action model for autonomous driving via online reinforcement learning. In *ECCV*, pp. 503–522, 2026.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

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

Xixiang He, Qiyao Sun, Ao Cheng, Xingming Li, Xuanyu Ji, Hailun Lu, Runke Huang, and Qingyong Hu. Advantage collapse in group relative policy optimization: Diagnosis and mitigation. In *ICML*, 2026b.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, pp. 17853–17862, 2023.

Anqing Jiang, Yu Gao, Yiru Wang, Zhigang Sun, Shuo Wang, Yuwen Heng, Hao Sun, Shichen Tang, Lijuan Zhu, Jinhao Chai, et al. IRL-VLA: Training an vision-language-action policy via reward world model. *arXiv preprint arXiv:2508.06571*, 2025a.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized scene representation for efficient autonomous driving. In *ICCV*, pp. 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. AlphaDrive: Unleashing the power of VLMs in autonomous driving via reinforcement learning and reasoning. *arXiv preprint arXiv:2503.07608*, 2025b.

Ruinan Jin, Shuai Li, and Baoxiang Wang. On stationary point convergence of PPO-Clip. In *ICLR*, 2024.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, pp. 267–274, 2002.

Thanh-Long V Le, Myeongho Jeon, Kim Vu, Viet Lai, and Eunho Yang. No prompt left behind: Exploiting zero-variance prompts in LLM reinforcement learning via entropy-guided advantage shaping. In *ICLR*, 2026.

Hongchen Li, Tianyu Li, Jiazhi Yang, Mingyang Shang, Gaoqiang Wu, Caojun Wang, Haochen Tian, Zengrong Lin, Zhihui Hao, XianPeng Lang, et al. PlannerRFT: Reinforcing diffusion planners through closed-loop and sample-efficient fine-tuning. In *CVPR*, pp. 24929–24938, 2026.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *ICML*, pp. 29128–29163, 2024.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Yiren Lu, Xin Ye, Jiaming Liu, Philip Jacobson, Jin Yao, Yi-chung Chen, Liam Merino, Dhruva Dixith Kurra, Min Cai, Tom Lampo, et al. GeoWAM: Visual geometry world action models for autonomous driving. *arXiv preprint arXiv:2608.23486*, 2026.

Yuechen Luo, Fang Li, Qimao Chen, Shaoqing Xu, Jiaxin Liu, Ziying Song, Zhi-xin Yang, and Fuxi Wen. Unleashing VLA potentials in autonomous driving via explicit learning from failures. In *CVPR*, pp. 24833–24842, 2026.

Yukai Ma, Tiantian Wei, Naiting Zhong, Jianbiao Mei, Tao Hu, Licheng Wen, Xuemeng Yang, Botian Shi, and Yong Liu. LeapVAD: A leap in autonomous driving via cognitive perception and dual-process thinking. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

Gongrui Nan, Siye Chen, Jing Huang, Mengyu Lu, Dexun Wang, Chunmei Xie, Weiqi Xiong, Xianzhou Zeng, Qixuan Zhou, Yadong Li, et al. NGRPO: Negative-enhanced group relative policy optimization. *arXiv preprint arXiv:2509.18851*, 2025.

Ishaan Rawal, Shubh Gupta, Yihan Hu, and Wei Zhan. NoRD: A data-efficient vision-language-action model that drives without reasoning. In *CVPR*, 2026.

Luke Rowe, Rodrigue de Schaetzen, Roger Girgis, Christopher Pal, and Liam Paull. Poutine: Vision-language-trajectory pre-training and reinforcement learning post-training enable robust end-to-end autonomous driving. *arXiv preprint arXiv:2506.11234*, 2025.

Mohammad Mahdi Salmani-Zarchi, Zahra Rahimi, Heshaam Faili, and Mohammad Javad Dousti. MDP-GRPO: Stabilized group relative policy optimization for multi-constraint instruction following. In *ACL*, pp. 42784–42797, 2026.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, pp. 1889–1897, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *ICLR*, 2016.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Zihao Sheng, Xin Ye, Jingru Luo, Sikai Chen, and Liu Ren. ExploreVLA: Dense world modeling and exploration for end-to-end autonomous driving. In *ECCV*, pp. 266–284, 2026.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NeurIPS*, volume 12, 2000.

Kefei Tian, Yuansheng Lian, Kai Yang, Xiangdong Chen, and Shen Li. SafeAlign-VLA: A negative-enhanced safe alignment framework for risk-aware autonomous driving. *arXiv preprint arXiv:2605.19524*, 2026.

Linbo Wang, Yupeng Zheng, Qiang Chen, Shiwei Li, Yichen Zhang, Zebin Xing, Qichao Zhang, Xiang Li, Deheng Qian, Pengxuan Yang, et al. Latent-WAM: Latent world action modeling for end-to-end autonomous driving. *arXiv preprint arXiv:2603.24581*, 2026a.

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

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-R1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

Yiru Wang, Zichong Gu, Yu Gao, Anqing Jiang, Zhigang Sun, Shuo Wang, Yuwen Heng, and Hao Sun. HiST-VLA: A hierarchical spatio-temporal vision-language-action model for end-to-end autonomous driving. *arXiv preprint arXiv:2602.13329*, 2026c.

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

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *ICLR*, volume 2025, pp. 28240–28264, 2025.

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. ReCogDrive: A reinforced cognitive framework for end-to-end autonomous driving. In *ICLR*, 2026.

Yifang Xu, Jiahao Cui, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Feipeng Cai, Neng Zhang, Yaoyi Li, Jia Cai, et al. WAM-Flow: Parallel coarse-to-fine motion planning via discrete flow matching for autonomous driving. In *CVPR*, pp. 24918–24928, 2026.

Pengxuan Yang, Ben Lu, Zhongpu Xia, Chao Han, Yinfeng Gao, Teng Zhang, Kun Zhan, XianPeng Lang, Yupeng Zheng, and Qichao Zhang. WorldRFT: Latent world model planning with reinforcement fine-tuning for autonomous driving. In *AAAI*, volume 40, pp. 11649–11657, 2026a.

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

Ruining Yang, Muxing Wang, Yixiao Chen, Tongfei Guo, Yi Xu, Can Cui, Zichong Yang, Yitian Zhang, Ziran Wang, Yun Fu, et al. Post-training in end-to-end autonomous driving. *arXiv preprint arXiv:2607.08072*, 2026c.

Yuqi Ye, Zijian Zhang, Junhong Lin, Shangkun Sun, Changhao Peng, and Wei Gao. AutoDrive-P$^3$: Unified chain of perception–prediction–planning thought via reinforcement fine-tuning. In *ICLR*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *NeurIPS*, volume 38, pp. 113222–113244, 2025.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. AutoVLA: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. In *NeurIPS*, volume 38, pp. 27920–27956, 2025.

Zewei Zhou, Ruining Yang, Xuewei Qi, Yiluan Guo, Sherry X Chen, Tao Feng, Kateryna Pistunova, Yishan Shen, Lili Su, Jiaqi Ma, et al. SpanVLA: Efficient action bridging and learning from negative-recovery samples for vision-language-action model. *arXiv preprint arXiv:2604.19710*, 2026.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. DiffusionDriveV2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

# APPENDIX

**Contents.**


**A Additional Background on GRPO**
- A.1 Policy Gradients and Baselines
- A.2 Policy Improvement and Surrogate Objectives
- A.3 Proximal Policy Optimization
- A.4 Token-Level GRPO Formulation

**B Experimental and Implementation Details**
- B.1 Benchmark Details
- B.2 Planner Details
- B.3 Baseline Details
- B.4 Experimental Details

**C Additional Experimental Results**
- C.1 Main Results on NAVSIM v1 and NAVSIM v2
- C.2 Component Analysis on ReCogDrive

**D Theoretical Analysis**
- D.1 Setup
- D.2 Calibration by the Historical Reference
- D.3 Policy-Gradient Interpretation of Reference Calibration
- D.4 Expected Update from the Ghost Term on Tied Groups

**E Qualitative Results**


## A ADDITIONAL BACKGROUND ON GRPO

### A.1 POLICY GRADIENTS AND BASELINES

Consider a discounted Markov decision process (MDP) $(\mathcal{X}, \mathcal{A}, P, r, \gamma, d_0)$, with state space $\mathcal{X}$, action space $\mathcal{A}$, transition kernel $P$, bounded reward function $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, discount factor $\gamma \in (0, 1)$, and initial-state distribution $d_0$. We use $x$ for an MDP state, reserving $s$ for a driving scene as in the main text. The expected discounted return of a policy $\pi$ is

$$J(\pi) = \mathbb{E}_{d_0,\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)\right]. \tag{6}$$

The normalized discounted state distribution is $d^\pi(x) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr_\pi(x_t = x)$, with $x_0 \sim d_0$.

The action-value function $Q^\pi(x, a)$ is the expected discounted return starting from state $x$ and action $a$, and then following $\pi$. The state-value function is $V^\pi(x) = \mathbb{E}_{a\sim\pi(\cdot|x)}[Q^\pi(x, a)]$, giving the advantage

$$A^\pi(x, a) = Q^\pi(x, a) - V^\pi(x). \tag{7}$$

For a differentiable policy parameterization $\pi_\theta$, policy-gradient methods optimize $J(\theta) = J(\pi_\theta)$. The policy-gradient theorem (Sutton et al., 2000) gives

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{x\sim d^{\pi_\theta},\, a\sim\pi_\theta(\cdot|x)}\Big[\nabla_\theta \log \pi_\theta(a \mid x) \cdot \big(Q^{\pi_\theta}(x, a) - b(x)\big)\Big], \tag{8}$$

where $b(x)$ is an action-independent baseline treated as fixed during differentiation. For each state $x$, its contribution to the expected score-function gradient is zero:

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|x)}\left[b(x)\nabla_\theta \log \pi_\theta(a \mid x)\right] = b(x)\nabla_\theta \sum_a \pi_\theta(a \mid x) = 0. \tag{9}$$

Sums over states or actions are replaced by integrals in continuous spaces. Choosing $b(x) = V^{\pi_\theta}(x)$ yields the advantage in Eq. (7).

In practice, policy gradients are estimated from sampled trajectories. At optimization step $n$, the parameters are updated as

$$\theta_{n+1} = \theta_n + \eta_n \widehat{\nabla J(\theta_n)}, \tag{10}$$

where $\eta_n > 0$ is the step size and $\widehat{\nabla J(\theta_n)}$ is a policy-gradient estimate. The objective is generally nonconcave under nonlinear policy parameterizations. Convergence and approximation guarantees depend on the policy class, estimation error, and state-distribution coverage (Agarwal et al., 2021).

### A.2 Policy Improvement and Surrogate Objectives

Policy iteration alternates between evaluating the current policy and improving it using its value function. The performance-difference identity expresses the return difference between two policies through the old policy's advantage (Kakade & Langford, 2002; Schulman et al., 2015):

$$J(\pi) - J(\pi_{\text{old}}) = \frac{1}{1-\gamma} \mathbb{E}_{x \sim d^{\pi},\, a \sim \pi(\cdot|x)} \left[ A^{\pi_{\text{old}}}(x, a) \right]. \tag{11}$$

Thus, a positive expected old-policy advantage under the new policy's discounted state distribution corresponds to an increase in return. Exact policy evaluation and greedy improvement are generally impractical in large or parameterized policy spaces, motivating approximate policy-improvement methods.

The dependence of $d^{\pi}$ on the new policy makes the performance-difference expression difficult to optimize directly. Replacing it with $d^{\pi_{\text{old}}}$ gives the surrogate

$$\mathcal{L}_{\pi_{\text{old}}}(\pi) = J(\pi_{\text{old}}) + \frac{1}{1-\gamma} \mathbb{E}_{x \sim d^{\pi_{\text{old}}},\, a \sim \pi(\cdot|x)} \left[ A^{\pi_{\text{old}}}(x, a) \right]. \tag{12}$$

For a differentiable policy parameterization, this surrogate agrees with $J$ in value and gradient at the old policy. It underlies the policy-improvement analysis of conservative policy iteration and TRPO (Kakade & Langford, 2002; Schulman et al., 2015). With the old-policy state distribution fixed, replacing $V^{\pi_{\text{old}}}(x)$ by another fixed state-only baseline changes this unclipped surrogate only by a constant independent of the new policy.

PPO collects rollout data under $\pi_{\theta_{\text{old}}}$ and estimates advantages for these samples. It then keeps the data and advantages fixed while performing multiple epochs of minibatch optimization (Schulman et al., 2017). Likelihood-ratio clipping discourages large policy changes during these updates. These optimization epochs are distinct from the training-scene epochs used in Sec. 4.

### A.3 Proximal Policy Optimization

Practical policy optimization uses sampled or learned estimates of values and advantages. Common approaches include Monte Carlo return estimation (Williams, 1992), temporal-difference learning with actor-critic models (Sutton et al., 2000; Sutton & Barto, 2018), and generalized advantage estimation (GAE) (Schulman et al., 2016).

PPO combines an outer policy-improvement loop with inner stochastic-gradient updates (Schulman et al., 2017). At each outer iteration, it collects trajectories under $\pi_{\theta_{\text{old}}}$, estimates advantages, and constructs a surrogate objective. During the inner updates, PPO keeps the rollout data and advantage estimates fixed while performing multiple epochs of minibatch optimization on $\pi_{\theta}$. Likelihood-ratio clipping discourages large policy changes relative to $\pi_{\theta_{\text{old}}}$ during these repeated updates. These inner optimization epochs are distinct from the scene visits indexed by $k$ in Sec. 4.

PPO's clipped objective does not impose TRPO's explicit trust-region constraint and therefore does not directly inherit its monotonic-improvement guarantee. Stationary-point convergence results for PPO-Clip hold under specified regularity, step-size, and advantage-estimation conditions (Jin et al., 2024).

### A.4 Token-Level GRPO Formulation

GRPO uses group-relative reward advantages within a clipped policy-update framework without fitting a value model (Shao et al., 2024). In the original LLM setting, it samples $G$ responses to a

prompt $q \sim \mathcal{D}_Q$ using $\pi_{\theta_{\text{old}}}$. Response $o_{q,i} = (o_{q,i,1}, \ldots, o_{q,i,T_i})$ contains $T_i$ tokens. At step $t$, the state is the prompt and generated prefix, $h_{q,i,t} = (q, o_{q,i,<t})$, and the action is the next token. Under outcome supervision, Eq. (2), with $q$ replacing $s$, gives a response-level advantage $A_{q,i}^{\text{GRPO}}$ shared by all its tokens.

The token-level likelihood ratio is

$$\rho_{q,i,t}(\theta) = \frac{\pi_\theta(o_{q,i,t} \mid h_{q,i,t})}{\pi_{\theta_{\text{old}}}(o_{q,i,t} \mid h_{q,i,t})}. \tag{13}$$

Define the clipped surrogate term as

$$\ell_{\text{clip}}(\rho, A) = \min\Big\{\rho A, \text{clip}\left(\rho, 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}}\right)A\Big\}. \tag{14}$$

The sampled KL penalty in the original formulation is

$$\widehat{D}_{q,i,t}(\theta) = \frac{\pi_{\text{ref}}(o_{q,i,t} \mid h_{q,i,t})}{\pi_\theta(o_{q,i,t} \mid h_{q,i,t})} - \log \frac{\pi_{\text{ref}}(o_{q,i,t} \mid h_{q,i,t})}{\pi_\theta(o_{q,i,t} \mid h_{q,i,t})} - 1. \tag{15}$$

The objective is

$$\mathcal{J}_{\text{token}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\left(\ell_{\text{clip}}\big(\rho_{q,i,t}(\theta), A_{q,i}^{\text{GRPO}}\big) - \beta\widehat{D}_{q,i,t}(\theta)\right)\right], \tag{16}$$

where the expectation is over prompts and responses sampled from the rollout policy. The parameters $\epsilon_{\text{clip}} > 0$ and $\beta \geq 0$ control clipping and reference-policy regularization, respectively.

GRPO's group mean includes the reward of the response whose advantage is being computed. It therefore depends on that sampled response, unlike the action-independent baseline in Eq. (9). This dependence, together with group-dependent scaling and clipping, must be accounted for when analyzing the expected update.

# B Experimental and Implementation Details

## B.1 Benchmark Details

**Training reward.** Rewards are NAVSIM simulator scores of each sampled trajectory. For the NAVSIM v1 benchmark we use the Predictive Driver Model Score (PDMS) (Dauner et al., 2024) as the driving reward,

$$r_{\text{Driving}} = \text{PDMS}, \qquad \text{PDMS} = \text{NC} \times \text{DAC} \times \left(\frac{5\,\text{TTC} + 2\,\text{C} + 5\,\text{EP}}{12}\right), \tag{17}$$

where NC denotes No at-fault Collision, DAC Drivable Area Compliance, EP Ego Progress, C Comfort, and TTC Time-to-Collision. Following the standard definition, these sub-scores are normalized to $[0, 1]$, and their weighted combination yields a comprehensive measure of driving quality. If the planner fails to produce a valid output, we assign a reward of 0.

For the NAVSIM v2 benchmark we use its extended version, the Extended Predictive Driver Model Score (EPDMS) (Cao et al., 2025), which introduces additional rule-compliance metrics. Compared with PDMS, EPDMS further incorporates Driving Direction Compliance (DDC) and Traffic Light Compliance (TLC), and replaces the single comfort term with finer-grained comfort metrics, Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC). Each metric is filtered against the human driving log,

$$\text{filter}_m(\text{agent}, \text{human}) = \begin{cases} 1.0, & \text{if } m(\text{human}) = 0, \\ m(\text{agent}), & \text{otherwise}, \end{cases} \tag{18}$$

and the training reward is the EPDMS *without the EC term*,

$$\begin{aligned} r_{\text{Driving}} = &\left(\prod_{m\in\{\text{NC,DAC,DDC,TLC}\}} \text{filter}_m(\text{agent}, \text{human})\right) \\ &\cdot \left(\frac{\sum_{m\in\{\text{TTC,EP,HC,LK}\}} w_m\, \text{filter}_m(\text{agent}, \text{human})}{\sum_{m\in\{\text{TTC,EP,HC,LK}\}} w_m}\right), \end{aligned} \tag{19}$$

with $w_{\text{TTC}} = 5$, $w_{\text{EP}} = 5$, $w_{\text{HC}} = 2$, and $w_{\text{LK}} = 2$. The EPDMS we *report* is the full metric of Cao et al. (2025), which additionally includes EC in the weighted sum with $w_{\text{EC}} = 2$.

### B.2 Planner Details

**Planners and IL checkpoints.** For each planner, all compared methods start from the same IL/SFT checkpoint and follow the RL training recipe of the corresponding planner, replacing only the advantage computation; all advantage estimators in this paper therefore share the same rollout budget, optimizer, regularizers and reward for a given planner. All baseline numbers in this paper, including the GRPO rows, are our own reproductions from the released code[1] under this protocol.

*ReCogDrive* (Xiong et al., 2026) pairs a frozen InternVL3-2B VLM with a DDPM diffusion planner that outputs future waypoints and headings $(x, y, \theta)$; RL follows its Diffusion-GRPO recipe: the trajectory log-probability is the sum of per-step Gaussian log-densities along the denoising chain, $\log \pi_\theta(x_{0:T}) = \sum_t \log \pi_\theta(x_{t-1} \mid x_t)$, discounted with $\gamma = 0.6$ over denoising steps, and the loss adds a behavior-cloning term. We keep the BC term for all estimators, with a weight of $\lambda = 0.1$ (NAVSIM v1 reward) and $\lambda = 0.3$ (NAVSIM v2 reward), the values used by the released code (https://github.com/xiaomi-research/recogdrive/issues/74).

*AutoVLA* (Zhou et al., 2025) is a Qwen2.5-VL-3B VLA that emits $K$=2048 discrete action tokens; RL follows its GRPO reinforced fine-tuning recipe with LoRA, with token-level log-probabilities summed over the action sequence and a token-level KL penalty to the SFT policy with $\beta = 0.04$. We use the action-only variant of AutoVLA, trained without chain-of-thought reasoning data, so the CoT-length penalty of the original reward does not apply and the reward is the plain simulator score.

*ExploreVLA* (Sheng et al., 2026) is a Show-o (Xie et al., 2025; Li et al., 2023) world action model that outputs continuous 4-s waypoints $(x, y, \theta)$ through an MLP head; RL follows its GRPO recipe with LoRA rank 32, including its world-model exploration bonus.

**Policy Optimization across Planners.** We retain each planner's original GRPO loss implementation. The surrogate $\mathcal{S}_p$ in equation (3) encompasses the planner's likelihood-ratio construction, clipping, aggregation, and any regularization terms. DriveBeyond supplies an advantage $A_{s,i}^{(k)}$ for each real trajectory through the Progress Module or the Ghost Module (Section 4.5); these advantages enter the same loss implementation used by the corresponding planner.

The Memory Module (Section 4.2) provides scalar reward references for advantage computation. These references serve a different role from a reference policy used for KL regularization. The ghost reward REF affects the advantage calculation but introduces no additional trajectory or loss entry, so policy optimization uses only the $G$ real trajectories.

### B.3 Baseline Details

**GRPO** (Shao et al., 2024) computes advantages by centering each reward with the group mean and normalizing it by the group standard deviation: $A_{s,i}^{(k)} = (R_i - \mu_s^{(k)})/(\sigma_s^{(k)} + \epsilon)$. The advantage therefore reflects a trajectory's reward relative to other trajectories in the same group.

**Dr.GRPO** (Liu et al., 2025) removes the group standard-deviation normalization while retaining mean centering, giving $A_{s,i}^{(k)} = R_i - \mu_s^{(k)}$. Its advantages preserve the magnitude of within-group reward differences without rescaling them according to the group's reward variability.

**DAPO** (dynamic sampling) (Yu et al., 2025) retains the standardized group-relative advantage used by GRPO, $A_{s,i}^{(k)} = (R_i - \mu_s^{(k)})/(\sigma_s^{(k)} + \epsilon)$. It filters out groups with identical rewards, whose advantages are zero, and constructs training batches from groups with nonzero within-group reward variation.

**NGRPO** (Nan et al., 2025) augments each reward group with a virtual sample assigned the maximum attainable reward. It recomputes the mean and standard deviation over the augmented group

[1] Official repositories: ReCogDrive: https://github.com/xiaomi-research/recogdrive; AutoVLA: https://github.com/ucla-mobility/AutoVLA; ExploreVLA: https://github.com/zihaosheng/ExploreVLA.

and uses these statistics to normalize the advantages of the real samples. This virtual reference assigns negative advantages to all-zero-reward groups that would otherwise receive zero advantages under GRPO.

**AVSPO** (He et al., 2026b) uses the Advantage Collapse Rate (ACR) to guide virtual reward augmentation. When the batch-level ACR exceeds an adaptive threshold, it augments low-variance groups with virtual rewards whose number depends on the ACR and whose values depend on the observed group rewards. Under binary rewards, all-correct groups receive lower-valued virtual rewards, while all-incorrect groups receive positive anchor rewards. Advantages for the real samples are then standardized using the augmented reward mean and standard deviation.

**Algorithm 1** GRPO advantage computation

**for** each scene visit $(s, k)$ **do**
  sample $G$ trajectories from policy $\pi$; obtain rewards $\{R_i\}_{i=1}^{G}$
  compute $\mu_s^{(k)}$ and $\sigma_s^{(k)}$
  for each $i = 1, \ldots, G$:
  $A_{s,i}^{(k)} \leftarrow \dfrac{R_i - \mu_s^{(k)}}{\sigma_s^{(k)} + \epsilon}$
**end for**
update $\theta$ with equation (3) using $\{A_{s,i}^{(k)}\}$

**Algorithm 2** DriveBeyond advantage computation

**Require:** Fixed IL mean rewards $\{\mu_s^{\mathrm{IL}}\}_{s\in\mathcal{D}_{\mathrm{train}}}$ from $G$ rollouts per scene of the frozen IL policy $\pi_{\mathrm{IL}}$, using the RFT sampling configuration; fixed global IL reference reward $\mathrm{REF} = \frac{1}{|\mathcal{D}_{\mathrm{train}}|}\sum_{s\in\mathcal{D}_{\mathrm{train}}} \mu_s^{\mathrm{IL}}$; stored previous-visit means initialized as $\mu_s^{(0)} = \mu_s^{\mathrm{IL}}$; threshold $\tau$; weights $w_a > 0$ and $w_p \geq 0$

**for** each scene visit $(s, k)$ **do**
  sample $G$ trajectories from policy $\pi$; obtain rewards $\{R_i\}_{i=1}^{G}$
  compute $\mu_s^{(k)}$ and $\sigma_s^{(k)}$
  **if** $\sigma_s^{(k)} > \tau$ **then**
    for each $i = 1, \ldots, G$:
    $A_{s,i}^{(k)} \leftarrow w_a \left(R_i - \mu_s^{(k)}\right) + w_p \left(\mu_s^{(k)} - \max\left(\mu_s^{\mathrm{IL}}, \mu_s^{(k-1)}\right)\right)$
  **else**
    for each $i = 1, \ldots, G$:
    $A_{s,i}^{(k)} \leftarrow w_a \left(R_i - \mu_s^{(k)}\right) + \dfrac{\mu_s^{(k)} - \mathrm{REF}}{G+1}$
  **end if**
  replace the stored previous-visit mean of $s$ with $\mu_s^{(k)}$ for its next visit
**end for**
update $\theta$ with equation (3) using $\{A_{s,i}^{(k)}\}$ for the $G$ real trajectories in each group

### B.4 Experimental Details

Algorithms 1 and 2 summarize the advantage-computation procedures of original GRPO and DriveBeyond, respectively. Table 2 summarizes the planner-specific training settings and the hyperparameters of DriveBeyond. For the main comparisons, we report results averaged over three random seeds. The corresponding standard deviations and all component metrics are provided in Appendix C.1.

**IL initialization.** For each planner, we initialize the Memory Module by sampling $G$ trajectories for each training scene $s \in \mathcal{D}_{\mathrm{train}}$ from the frozen IL policy $\pi_{\mathrm{IL}}$, using the same sampling configuration as in RFT. The rollouts are evaluated using PDMS for NAVSIM v1 and EPDMS for NAVSIM v2. We store the mean of their rewards $\{R_i^{\mathrm{IL}}\}_{i=1}^{G}$ as the fixed IL mean reward $\mu_s^{\mathrm{IL}}$ and compute the global IL reference reward REF by averaging over $\mathcal{D}_{\mathrm{train}}$. Both references are computed separately for each planner and benchmark and remain fixed throughout RFT. The previous-visit group mean is initialized as $\mu_s^{(0)} = \mu_s^{\mathrm{IL}}$ and updated after advantage computation at every visit, as described in Section 4.2.

**Choice of $w_a$.** We use two forms of the within-group weight $w_a$ in equations (4) and (5): a constant, or GRPO's group-normalized scaling $w_a = 1/\sigma_s^{(k)}$. Both are shared by the $G$ trajectories of a group,

Table 2: **Hyperparameters.** RL settings follow each planner's released recipe. For ReCogDrive, the within-group advantage term is $(R_i - \mu_s^{(k)})/\sigma_s^{(k)}$ when $\sigma_s^{(k)} > 0$. Only exactly tied groups ($\tau = 0$) use the Ghost Module; their within-group advantage term is set directly to zero, giving $A_{s,i}^{(k)} = (\mu_s^{(k)} - \mathrm{REF})/(G+1)$.

| | **ReCogDrive** | **AutoVLA** | **ExploreVLA** |
|---|---|---|---|
| Initialization | released IL checkpoint | reproduced action-only SFT checkpoint | reproduced Stage-1 checkpoint |
| Trainable parameters in RL | diffusion planner | LoRA | LoRA |
| Group size $G$ | 8 | 8 | 8 |
| Epochs | 10 | 5 | 5 |
| Learning rate | $10^{-4}$ (cosine) | $3 \times 10^{-5}$ | $3 \times 10^{-6}$ |
| KL weight $\beta$ | BC term | 0.04 | 0.01 |
| Planner-specific terms | BC weight 0.1 (v1) / 0.3 (v2), $\gamma = 0.6$ | – | exploration bonus ($\lambda = 0.5$, $\delta = 0.9$) |
| DriveBeyond: threshold $\tau$ | 0 | 0.02 | 0.02 |
| DriveBeyond: within-group advantage weight $w_a$ | $1/\sigma_s^{(k)}$ for $\sigma_s^{(k)} > 0$ | 10 | 10 |
| DriveBeyond: beyond-group progress weight $w_p$ | 1 | 5 | 5 |

Table 3: **Performance on the NAVSIM v1 navtest.**

| **Planner** | **Method** | **NC↑** | **DAC↑** | **EP↑** | **TTC↑** | **Comf↑** | **PDMS↑** |
|---|---|---|---|---|---|---|---|
| *VLA-based Planner* | | | | | | | |
| ReCogDrive (2026) | GRPO | 98.1 | 97.7 | 85.9 | 95.1 | 100.0 | $90.5_{\pm 0.19}$ |
| | Dr.GRPO | 98.5 | 97.9 | 83.7 | 96.0 | 100.0 | $90.2_{\pm 0.20}$ |
| | DAPO | 98.7 | 98.0 | 83.8 | 96.4 | 100.0 | $90.4_{\pm 0.16}$ |
| | NGRPO | 98.6 | 98.2 | 84.0 | 96.1 | 100.0 | $90.5_{\pm 0.19}$ |
| | AVSPO | 98.2 | 97.5 | 86.1 | 95.1 | 100.0 | $90.6_{\pm 0.18}$ |
| | **Ours** | 98.4 | 97.9 | 86.2 | 95.7 | 100.0 | $\mathbf{90.9}_{\pm 0.18}$ |
| AutoVLA (2025) | GRPO | 99.4 | 98.7 | 79.6 | 96.7 | 100.0 | $89.3_{\pm 0.10}$ |
| | Dr.GRPO | 99.3 | 98.7 | 80.4 | 97.2 | 100.0 | $90.1_{\pm 0.10}$ |
| | DAPO | 99.6 | 98.6 | 80.6 | 97.7 | 99.9 | $90.2_{\pm 0.09}$ |
| | NGRPO | 99.5 | 99.0 | 78.4 | 97.8 | 99.8 | $89.5_{\pm 0.11}$ |
| | AVSPO | 99.5 | 99.2 | 79.1 | 97.4 | 99.9 | $89.7_{\pm 0.11}$ |
| | **Ours** | 99.3 | 98.9 | 81.8 | 97.3 | 100.0 | $\mathbf{90.6}_{\pm 0.10}$ |
| *WAM-based Planner* | | | | | | | |
| ExploreVLA (2026) | GRPO | 98.8 | 97.9 | 83.5 | 96.6 | 100.0 | $90.3_{\pm 0.08}$ |
| | Dr.GRPO | 98.8 | 98.1 | 83.9 | 96.0 | 100.0 | $90.4_{\pm 0.09}$ |
| | DAPO | 98.5 | 98.0 | 83.7 | 96.3 | 100.0 | $90.4_{\pm 0.10}$ |
| | NGRPO | 98.6 | 97.9 | 83.7 | 96.2 | 100.0 | $90.3_{\pm 0.10}$ |
| | AVSPO | 98.9 | 98.0 | 83.0 | 97.2 | 100.0 | $90.5_{\pm 0.09}$ |
| | **Ours** | 98.9 | 98.3 | 84.1 | 96.7 | 100.0 | $\mathbf{90.8}_{\pm 0.08}$ |

so neither changes within-group signs or rankings. AutoVLA and ExploreVLA use the constant form with $w_a = 10$, $w_p = 5$, and $\tau = 0.02$. ReCogDrive uses the normalized form with $w_p = 1$ and $\tau = 0$, because removing the normalization degrades it under its Diffusion-GRPO recipe (Table 1, Dr.GRPO vs. GRPO). For its exactly tied groups, $\sigma_s^{(k)} = 0$ and the within-group term is set to zero, giving $A_{s,i}^{(k)} = (\mu_s^{(k)} - \mathrm{REF})/(G+1)$.

# C Additional Experimental Results

## C.1 Main Results on NAVSIM v1 and NAVSIM v2

Tables 3, 4, and 5 report the complete results with all component metrics on NAVSIM v1 navtest, NAVSIM v2 navtest, and NAVSIM v2 navhard. All entries are averaged over three seeds, and standard deviations are given for the aggregate scores.

**NAVSIM v1 navtest (Table 3).** All RFT methods reach 89–91 PDMS, and DriveBeyond attains the highest PDMS for every planner (90.9, 90.6, and 90.8), improving over GRPO by 0.4, 1.3, and 0.5 and over the strongest baseline by 0.3–0.4.

**NAVSIM v2 navtest (Table 4).** DriveBeyond attains the highest EPDMS for every planner (87.3, 84.1, and 90.6), improving over GRPO by 1.0, 1.7, and 1.7 and over the strongest baseline by 0.5, 0.8, and 0.4.

**NAVSIM v2 navhard (Table 5).** Compared with GRPO, the EPDMS increases from 29.9 to 33.9 for ReCogDrive, from 34.0 to 37.2 for AutoVLA, and from 31.2 to 35.9 for ExploreVLA, corresponding to gains of 4.0, 3.2, and 4.7 points. EC is the only sub-metric on which DriveBeyond falls

Table 4: **Performance on the NAVSIM v2 navtest.**

| Planner | Method | NC↑ | DAC↑ | DDC↑ | TLC↑ | EP↑ | TTC↑ | LK↑ | HC↑ | EC↑ | EPDMS↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *VLA-based Planner* | | | | | | | | | | | |
| ReCogDrive | GRPO | 98.5 | 97.0 | 99.4 | 99.9 | 86.4 | 97.9 | 96.2 | 97.8 | 62.3 | $86.3_{\pm 0.29}$ |
| | Dr.GRPO | 98.5 | 96.7 | 99.2 | 99.9 | 85.8 | 97.9 | 95.4 | 97.7 | 63.9 | $85.9_{\pm 0.32}$ |
| | DAPO | 98.3 | 97.3 | 99.4 | 99.9 | 87.0 | 97.8 | 96.1 | 97.5 | 63.9 | $86.6_{\pm 0.27}$ |
| | NGRPO | 98.4 | 96.9 | 99.4 | 99.8 | 86.8 | 97.7 | 96.4 | 97.9 | 65.7 | $86.7_{\pm 0.26}$ |
| | AVSPO | 98.5 | 97.4 | 99.4 | 99.8 | 86.9 | 97.9 | 96.3 | 97.3 | 63.3 | $86.8_{\pm 0.27}$ |
| | **Ours** | 98.5 | 97.5 | 99.5 | 99.9 | 86.8 | 98.0 | 96.2 | 97.8 | 66.8 | $\mathbf{87.3}_{\pm 0.27}$ |
| AutoVLA | GRPO | 97.5 | 95.3 | 98.8 | 99.9 | 80.4 | 97.6 | 93.2 | 98.2 | 67.8 | $82.4_{\pm 0.15}$ |
| | Dr.GRPO | 98.1 | 95.6 | 98.9 | 99.9 | 78.8 | 98.1 | 92.8 | 98.3 | 70.4 | $83.3_{\pm 0.17}$ |
| | DAPO | 98.1 | 96.2 | 99.3 | 99.9 | 77.5 | 97.8 | 93.5 | 97.7 | 66.2 | $82.6_{\pm 0.15}$ |
| | NGRPO | 98.0 | 95.6 | 99.0 | 99.9 | 79.1 | 97.9 | 93.4 | 98.2 | 67.6 | $83.0_{\pm 0.19}$ |
| | AVSPO | 98.1 | 95.8 | 99.2 | 99.9 | 78.3 | 98.0 | 93.1 | 98.3 | 70.0 | $83.3_{\pm 0.20}$ |
| | **Ours** | 98.6 | 96.6 | 99.3 | 99.9 | 78.7 | 97.8 | 93.5 | 97.1 | 68.1 | $\mathbf{84.1}_{\pm 0.24}$ |
| *WAM-based Planner* | | | | | | | | | | | |
| ExploreVLA | GRPO | 98.7 | 96.3 | 99.6 | 99.8 | 87.1 | 98.2 | 97.8 | 98.3 | 86.9 | $88.9_{\pm 0.13}$ |
| | Dr.GRPO | 98.9 | 97.6 | 99.7 | 99.8 | 85.4 | 98.5 | 97.5 | 98.3 | 86.6 | $90.2_{\pm 0.15}$ |
| | DAPO | 98.9 | 97.4 | 99.6 | 99.8 | 86.7 | 98.4 | 97.6 | 98.3 | 86.8 | $90.0_{\pm 0.13}$ |
| | NGRPO | 98.8 | 96.8 | 99.6 | 99.8 | 86.9 | 98.2 | 97.7 | 98.3 | 86.8 | $89.4_{\pm 0.13}$ |
| | AVSPO | 98.9 | 97.5 | 99.6 | 99.8 | 86.6 | 98.4 | 97.5 | 98.3 | 86.6 | $90.1_{\pm 0.14}$ |
| | **Ours** | 99.1 | 98.3 | 99.7 | 99.9 | 85.5 | 98.7 | 97.3 | 98.4 | 86.7 | $\mathbf{90.6}_{\pm 0.12}$ |

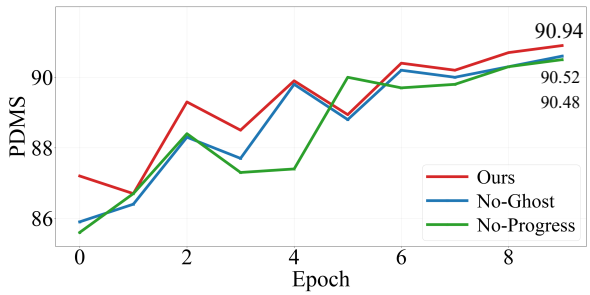



Figure 8: **Component ablation on ReCogDrive.** Reported PDMS on NAVSIM v1 navtest.

clearly below GRPO on navhard (54.5 → 16.1 at S1 for ReCogDrive). EC measures how consistent the plans at consecutive frames are (Cao et al., 2025), not how smooth a single trajectory is, and it is not part of the training reward (equation (19)). The same drop appears for NGRPO (20.9) and AVSPO (17.8), the other two methods that learn from all-zero groups, while GRPO and Dr.GRPO, which ignore such groups, stay at 53–55. On scenes where every rollout fails, learning from these groups makes the policy change its plan between frames instead of keeping a plan that collides; this lowers EC but is what reduces collisions and off-road driving (NC +1.0, DAC +5.1 at S1). On navtest, where such scenes are rare, EC does not drop. Adding a plan-consistency term to the training reward is left for future work.

**Reference points.** For AutoVLA and ExploreVLA, DriveBeyond uses an unnormalized within-group term ($w_a = 10$, Table 2), so Dr. GRPO, which also removes the standard-deviation normalization, is the closest reference point; DriveBeyond improves over it by 0.5 and 0.4 on navtest v1, 0.8 and 0.4 on navtest v2, and 2.7 and 3.2 on navhard. For ReCogDrive, the within-group term is GRPO's $\sigma$-normalized advantage, so its gains over GRPO (0.4, 1.0, and 4.0) isolate the effect of the two added terms.

### C.2 Component Analysis on ReCogDrive

Figure 8 compares the PDMS curves of DriveBeyond, No-Progress, and No-Ghost on NAVSIM v1 navtest. The No-Progress variant sets $w_p = 0$, while No-Ghost removes the ghost advantage term. Removing either module lowers the final PDMS, from 90.94 to 90.48 without the progress term and to 90.52 without the ghost advantage.

Table 5: **Performance on NAVSIM v2 navhard.**

| Planner | Method | Stage | NC↑ | DAC↑ | DDC↑ | TLC↑ | EP↑ | TTC↑ | LK↑ | HC↑ | EC↑ | S.↑ | EPDMS↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *VLA-based Planner* | | | | | | | | | | | | | |
| ReCogDrive | GRPO | S1 | 96.9 | 89.6 | 98.3 | 99.6 | 82.8 | 95.7 | 96.2 | 97.6 | 54.5 | 71.8 | $29.9_{\pm 0.36}$ |
| | | S2 | 80.0 | 71.9 | 84.3 | 98.2 | 85.9 | 77.4 | 45.9 | 94.1 | 52.1 | 40.7 | |
| | Dr.GRPO | S1 | 96.7 | 89.0 | 98.2 | 99.5 | 82.7 | 95.7 | 94.8 | 97.5 | 53.1 | 74.0 | $29.7_{\pm 0.36}$ |
| | | S2 | 79.9 | 71.7 | 84.7 | 98.1 | 85.9 | 77.4 | 43.4 | 94.5 | 52.6 | 39.0 | |
| | DAPO | S1 | 97.6 | 91.1 | 98.2 | 99.7 | 82.2 | 96.2 | 95.3 | 97.5 | 45.9 | 75.8 | $30.0_{\pm 0.39}$ |
| | | S2 | 79.6 | 72.1 | 85.8 | 97.9 | 84.6 | 75.5 | 43.2 | 96.6 | 48.0 | 41.3 | |
| | NGRPO | S1 | 97.6 | 94.2 | 98.4 | 100.0 | 79.4 | 96.4 | 93.5 | 97.5 | 20.9 | 74.8 | $31.7_{\pm 0.34}$ |
| | | S2 | 86.4 | 77.0 | 85.2 | 98.9 | 78.2 | 83.6 | 48.0 | 96.2 | 19.8 | 43.7 | |
| | AVSPO | S1 | 97.2 | 94.0 | 98.7 | 100.0 | 79.5 | 96.2 | 94.2 | 97.5 | 17.8 | 74.0 | $32.2_{\pm 0.33}$ |
| | | S2 | 86.4 | 77.1 | 85.3 | 98.9 | 78.2 | 83.4 | 47.9 | 96.3 | 20.0 | 43.9 | |
| | **Ours** | S1 | 97.9 | 94.7 | 99.1 | 100.0 | 78.6 | 96.9 | 93.3 | 97.3 | 16.1 | **74.5** | **33.9**$_{\pm 0.36}$ |
| | | S2 | 85.8 | 79.1 | 86.8 | 98.9 | 79.4 | 83.1 | 45.7 | 96.4 | 18.1 | **44.9** | |
| AutoVLA | GRPO | S1 | 99.6 | 90.6 | 100.0 | 99.8 | 42.9 | 99.6 | 89.8 | 74.0 | 16.9 | 61.6 | $34.0_{\pm 0.15}$ |
| | | S2 | 93.0 | 87.3 | 95.4 | 99.4 | 43.4 | 92.0 | 54.8 | 84.2 | 53.0 | 53.0 | |
| | Dr.GRPO | S1 | 99.6 | 94.4 | 100.0 | 99.7 | 42.1 | 99.7 | 90.4 | 72.0 | 18.2 | 63.4 | $34.5_{\pm 0.14}$ |
| | | S2 | 93.5 | 88.1 | 95.5 | 99.2 | 42.7 | 92.4 | 54.6 | 83.3 | 49.1 | 52.6 | |
| | DAPO | S1 | 99.6 | 95.7 | 100.0 | 99.7 | 41.2 | 99.7 | 89.5 | 72.8 | 16.8 | 63.8 | $35.2_{\pm 0.15}$ |
| | | S2 | 93.9 | 89.7 | 95.5 | 99.2 | 40.7 | 92.7 | 54.1 | 83.6 | 48.5 | 53.5 | |
| | NGRPO | S1 | 100.0 | 96.0 | 100.0 | 99.7 | 38.7 | 100.0 | 90.2 | 70.4 | 16.0 | 63.1 | $35.8_{\pm 0.18}$ |
| | | S2 | 95.3 | 89.9 | 95.6 | 99.6 | 35.0 | 93.5 | 54.6 | 83.2 | 49.3 | 53.8 | |
| | AVSPO | S1 | 100.0 | 94.2 | 100.0 | 99.7 | 40.3 | 99.7 | 88.0 | 76.6 | 17.3 | 62.9 | $36.0_{\pm 0.18}$ |
| | | S2 | 95.1 | 90.0 | 95.6 | 99.4 | 35.6 | 93.9 | 54.9 | 85.1 | 51.9 | 54.4 | |
| | **Ours** | S1 | 100.0 | 96.9 | 100.0 | 99.8 | 37.8 | 100.0 | 87.3 | 72.0 | 15.1 | **66.6** | **37.2**$_{\pm 0.17}$ |
| | | S2 | 94.4 | 89.8 | 95.0 | 99.2 | 43.6 | 92.5 | 52.3 | 79.9 | 32.4 | **54.5** | |
| *WAM-based Planner* | | | | | | | | | | | | | |
| ExploreVLA | GRPO | S1 | 96.8 | 87.1 | 99.1 | 99.3 | 84.1 | 95.6 | 95.8 | 97.1 | 74.7 | 75.2 | $31.2_{\pm 0.12}$ |
| | | S2 | 81.1 | 70.1 | 84.7 | 98.6 | 85.3 | 78.1 | 45.9 | 95.3 | 69.2 | 41.0 | |
| | Dr.GRPO | S1 | 97.2 | 90.4 | 99.4 | 99.3 | 83.8 | 96.0 | 96.4 | 97.1 | 75.1 | 78.6 | $32.7_{\pm 0.14}$ |
| | | S2 | 81.3 | 70.4 | 84.6 | 98.5 | 85.2 | 78.3 | 45.3 | 95.2 | 68.7 | 40.9 | |
| | DAPO | S1 | 97.2 | 90.8 | 99.4 | 99.3 | 83.6 | 96.4 | 97.3 | 97.1 | 74.2 | 78.9 | $33.6_{\pm 0.14}$ |
| | | S2 | 81.6 | 71.4 | 84.4 | 98.5 | 85.1 | 78.1 | 45.2 | 95.5 | 69.3 | 41.8 | |
| | NGRPO | S1 | 97.4 | 90.8 | 99.4 | 99.3 | 83.5 | 96.4 | 97.1 | 97.1 | 74.2 | 79.1 | $34.0_{\pm 0.15}$ |
| | | S2 | 82.4 | 71.7 | 84.8 | 98.5 | 85.0 | 78.6 | 45.5 | 95.5 | 69.5 | 42.3 | |
| | AVSPO | S1 | 97.2 | 91.1 | 99.5 | 99.5 | 83.3 | 96.4 | 97.1 | 97.1 | 73.7 | 79.2 | $34.7_{\pm 0.14}$ |
| | | S2 | 82.3 | 72.8 | 85.3 | 98.5 | 84.9 | 78.8 | 45.8 | 95.6 | 69.3 | 43.2 | |
| | **Ours** | S1 | 97.4 | 92.4 | 99.7 | 99.6 | 82.5 | 96.7 | 97.6 | 97.3 | 74.2 | **80.4** | **35.9**$_{\pm 0.10}$ |
| | | S2 | 82.2 | 74.6 | 86.1 | 98.6 | 84.2 | 78.9 | 46.3 | 96.6 | 70.2 | **44.2** | |

# D THEORETICAL ANALYSIS

This appendix states the properties of the DriveBeyond advantages that are used in the main text: the reward threshold at which a progress-calibrated advantage becomes positive (Section D.2, used in Section 4.3), the expected first-order update and its variance under a fixed reference (Section D.3, used in Section 4.3), and the expected policy-gradient contribution of the ghost term on tied groups (Section D.4, used in Section 4.4).

## D.1 SETUP

Fix a scene $s$ and its $k$-th visit, and abbreviate $A_i = A_{s,i}^{(k)}$, $\mu = \mu_s^{(k)}$, $\sigma = \sigma_s^{(k)}$ (equation (1)), and $b = \max\{\mu_s^{\mathrm{IL}}, \mu_s^{(k-1)}\}$. The memory is read before it is updated. Thus $b$ and REF are fixed when the current group is evaluated. Assume $G \geq 2$, $\tau \geq 0$, $w_a > 0$, and $w_p \geq 0$. Equations (4) and (5)

then read

$$A_i = w_a(R_i - \mu) + \begin{cases} w_p(\mu - b), & \sigma > \tau, \\ \dfrac{\mu - \mathrm{REF}}{G+1}, & \sigma \le \tau. \end{cases} \tag{20}$$

Here $w_a$ is shared within a realized group and may depend on its reward spread. When the centered component is normalized by $\sigma$, it is evaluated as $(R_i - \mu)/\sigma$ for $\sigma > 0$ and defined as zero at an exact tie. For the algebra below, this is equivalent to assigning any finite positive $w_a$ at $\sigma = 0$.

### D.2 Calibration by the Historical Reference

The Progress Module changes the reward threshold at which an advantage becomes positive. This threshold gives a precise interpretation of the progress term in equation (4) of Section 4.3.

**Proposition D.1** (Effective reward threshold)**.** *For a group with $\sigma > \tau$, define*

$$B = \mu + \frac{w_p}{w_a}(b - \mu). \tag{21}$$

*Then $A_i = w_a(R_i - B)$ and $A_i > 0$ if and only if $R_i > B$. If $w_p > 0$ and $b > \mu$, then $B > \mu$ and*

$$\{i : A_i > 0\} \subseteq \{i : R_i > \mu\}. \tag{22}$$

*If $b < \mu$, the inclusion reverses. For $0 \le w_p \le w_a$, $B$ lies between $\mu$ and $b$; at $w_p = w_a$, $B = b$.*

*Proof.* Rearranging the Progress advantage gives

$$A_i = w_a R_i - w_a\mu + w_p\mu - w_p b = w_a\Big(R_i - \mu - \frac{w_p}{w_a}(b - \mu)\Big).$$

Because $w_a > 0$, the sign of $A_i$ is the sign of $R_i - B$. Moreover, $B - \mu = \frac{w_p}{w_a}(b - \mu)$, which establishes the set inclusions. The final statement follows because $B$ is then a convex combination of $\mu$ and $b$. □

For $w_p > 0$, when the current mean is below the historical reference, the threshold rises by $\frac{w_p}{w_a}(b - \mu)$. A trajectory can therefore remain a relative winner while losing its positive advantage. When the current mean exceeds the reference, the threshold decreases. For $0 < w_p < w_a$, the calibration is partial: a reward below $b$ can still have positive advantage if it exceeds $B$. These statements concern the comparison level only; they do not bound the return of the next policy.

### D.3 Policy-Gradient Interpretation of Reference Calibration

The preceding results hold directly for the advantage computation. To study its interaction with likelihood gradients, we now fix one scene and condition on the stored memory. Let $o_1, \ldots, o_G$ be independent trajectories from a differentiable distribution $\pi_\theta(\cdot \mid s)$ with common support, and let $R(o) \in [0, 1]$ have no explicit dependence on $\theta$. All expectations below are over these $G$ trajectories. Define

$$z_i = \nabla_\theta \log \pi_\theta(o_i \mid s), \qquad m = \mathbb{E}[R], \qquad v = \nabla_\theta m = \mathbb{E}[Rz]. \tag{23}$$

Assume differentiation can pass through the expectations below and $\mathbb{E}\|z\|^2 < \infty$. Then $\mathbb{E}[z] = 0$. In this subsection, $w_a > 0$ and $w_p \ge 0$ are constants independent of the current group and of $\theta$, and $b$, REF, and $\tau$ are held fixed. The estimator uses the exact trajectory score and an average over the $G$ real trajectories, before clipping and regularization.

The following result evaluates the Progress formula on an unconditioned group, i.e., without conditioning on the event $\sigma > \tau$, which isolates the role of the reference.

**Proposition D.2** (Mean update and reference-dependent variance)**.** *Define*

$$\widehat{g}_b = \frac{1}{G}\sum_{i=1}^{G}\left[w_a(R_i - \mu) + w_p(\mu - b)\right] z_i, \qquad \kappa = w_a\left(1 - \frac{1}{G}\right) + \frac{w_p}{G}. \tag{24}$$

*Then $\mathbb{E}[\widehat{g}_b] = \kappa v$. If $w_p > 0$ and $q := \mathbb{E}\|z\|^2 > 0$, define $h := \mathbb{E}[R\|z\|^2]$ and*

$$b^* = m + \frac{\kappa}{w_p}\left(\frac{h}{q} - m\right). \tag{25}$$

*The total gradient variance satisfies*

$$\operatorname{tr}\operatorname{Cov}(\widehat{g}_b) = \operatorname{tr}\operatorname{Cov}(\widehat{g}_{b^*}) + \frac{w_p^2 q}{G}(b-b^*)^2. \tag{26}$$

*Proof.* Write $\overline{z} = G^{-1}\sum_i z_i$ and $\overline{Rz} = G^{-1}\sum_i R_i z_i$. The estimator becomes

$$\widehat{g}_b = w_a\overline{Rz} + (w_p - w_a)\mu\overline{z} - w_p b\overline{z}.$$

Independence and $\mathbb{E}[z] = 0$ imply $\mathbb{E}[\overline{Rz}] = v$, $\mathbb{E}[\mu\overline{z}] = v/G$, and $\mathbb{E}[\overline{z}] = 0$. Hence $\mathbb{E}[\widehat{g}_b] = [w_a + (w_p - w_a)/G]v = \kappa v$.

For the second statement, set $T = \widehat{g}_0$. Expanding the sums and eliminating terms with an independent zero-mean score gives

$$\mathbb{E}\|\overline{z}\|^2 = \frac{q}{G}, \qquad \mathbb{E}[\overline{Rz}^{\mathsf{T}}\overline{z}] = \frac{h}{G},$$
$$\mathbb{E}[\mu\|\overline{z}\|^2] = \frac{h + (G-1)mq}{G^2}, \qquad D := \mathbb{E}[T^{\mathsf{T}}\overline{z}] = \frac{\kappa h}{G} + \frac{(w_p - w_a)(G-1)mq}{G^2}.$$

Since the mean is independent of $b$, the variance is

$$\operatorname{tr}\operatorname{Cov}(\widehat{g}_b) = \mathbb{E}\|T\|^2 - \|\kappa v\|^2 - 2w_p b D + \frac{w_p^2 q}{G}b^2.$$

This quadratic is minimized at $GD/(w_p q) = m + (\kappa/w_p)(h/q - m)$. Completing the square proves equation (26). □

Thus the stored reference $b$ does not alter the mean update; its distance from $b^*$ determines the trace of the covariance, a control-variate interpretation of the reference (Greensmith et al., 2004). If $R$ and $\|z\|^2$ are uncorrelated, then $b^* = m$, and with a reference constrained to $[0, 1]$ the variance-minimizing value is the projection of $b^*$ onto that interval. The first statement uses only $\mathbb{E}[\overline{z}] = 0$ and the fact that $b$ is fixed when the group is sampled, so it also holds when $w_a$ varies with the group, as for the $\sigma$-normalized within-group term of ReCogDrive; the factor $\kappa$ and the variance identity assume constant weights. Neither statement ranks the candidate references of Section 5.3, since $b^*$ is unknown.

Clipping further changes which sampled terms contribute to the update. For a fixed advantage and away from clipping boundaries, equation (14) gives

$$\nabla_\theta \ell_{\text{clip}}(\rho(\theta), A) = \begin{cases} A\nabla_\theta\rho(\theta), & A > 0,\ \rho < 1 + \epsilon_{\text{clip}}, \\ A\nabla_\theta\rho(\theta), & A < 0,\ \rho > 1 - \epsilon_{\text{clip}}, \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

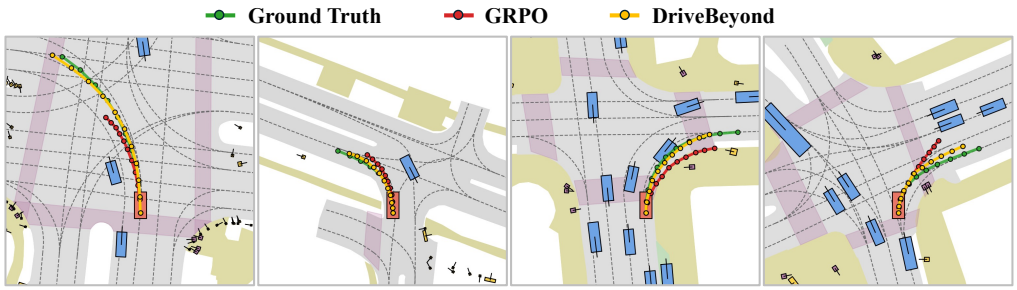
This follows by selecting the smaller of the two linear expressions for each sign of $A$. A common advantage offset can change both a sample's weight and, when its sign changes, its active clipping region. At the sampling policy, $\rho = 1$ and clipping is inactive. If $\rho$ is the exact trajectory likelihood ratio, its gradient there equals $z_i$, connecting the score-function calculation to the initial reward-surrogate gradient.

The statements of this section concern the expected update at $\theta = \theta_{\text{old}}$; later PPO optimization steps and planner-specific aggregation or regularization require their corresponding update equations, and no convergence rate or return improvement follows from them.

### D.4 Expected Update from the Ghost Term on Tied Groups

Under the score-function model of Section D.3, for any event $\mathcal{E}$ that is determined by the sampled group, the likelihood-ratio identity gives

$$\mathbb{E}\Big[\mathbf{1}_{\mathcal{E}}\sum_{i=1}^{G} z_i\Big] = \nabla_\theta \Pr_\theta(\mathcal{E}). \tag{28}$$




(A) PDMS: 0.80, 1.00 (B) PDMS: 0.73, 0.88 (C) PDMS: 0.00, 0.95 (D) PDMS: 0.44, 0.88

Figure 9: **Qualitative comparisons during evaluation.** Trajectories with GRPO (red) and DriveBeyond (yellow), with the ground truth in green. Numbers below each scene give the PDMS of GRPO and DriveBeyond, respectively.

Take $\tau = 0$ and let $\mathcal{E}_r$ be the event that all $G$ rewards equal $r$, which has probability $\Pr_\theta(R{=}r)^G$. By equation (5), every trajectory in such a group receives the same advantage $(r - \mathrm{REF})/(G+1)$, so the group's contribution to the advantage-weighted score $G^{-1}\sum_i A_i z_i$ is $\frac{r-\mathrm{REF}}{G+1}\,\mathbf{1}_{\mathcal{E}_r}\,G^{-1}\sum_i z_i$, whose expectation is

$$\frac{1}{G}\,\frac{r-\mathrm{REF}}{G+1}\,\nabla_\theta \Pr_\theta(R{=}r)^G. \tag{29}$$

For $0 < \mathrm{REF} < 1$ this contribution is nonzero even though the within-group term vanishes. Taken on its own, it is an ascent direction for $-\Pr_\theta(R{=}0)^G$ when $r = 0$ and for $\Pr_\theta(R{=}1)^G$ when $r = 1$, i.e., it pushes the policy away from uniformly failing and toward uniformly succeeding on the scene whenever the corresponding probability gradient is nonzero. It is only one component of the expected update: the within-group terms and the other tied events act on the same parameters, so the net change of $\Pr_\theta(R{=}r)^G$ along the full update is not determined by this component alone, and no such monotonicity is claimed.

## E Qualitative Results

Figure 9 compares trajectories trained with GRPO and with DriveBeyond on NAVSIM v1 navtest. In (A), GRPO produces a feasible trajectory but makes less progress, with a PDMS of 0.80, whereas DriveBeyond advances further along the intended path and reaches 1.00. In (B), neither trajectory exhibits apparent unsafe behavior, but DriveBeyond stays closer to the ground truth, while GRPO makes less progress and drifts laterally near its endpoint, scoring 0.73 against 0.88. The differences are more pronounced in (C) and (D). In (C), GRPO's trajectory extends into a non-drivable area, which sets its PDMS to 0.00, while DriveBeyond remains within the drivable area and achieves 0.95. In (D), GRPO deviates substantially from the ground truth near the end of the turn and scores 0.44, whereas DriveBeyond follows the turn closely and achieves 0.88.

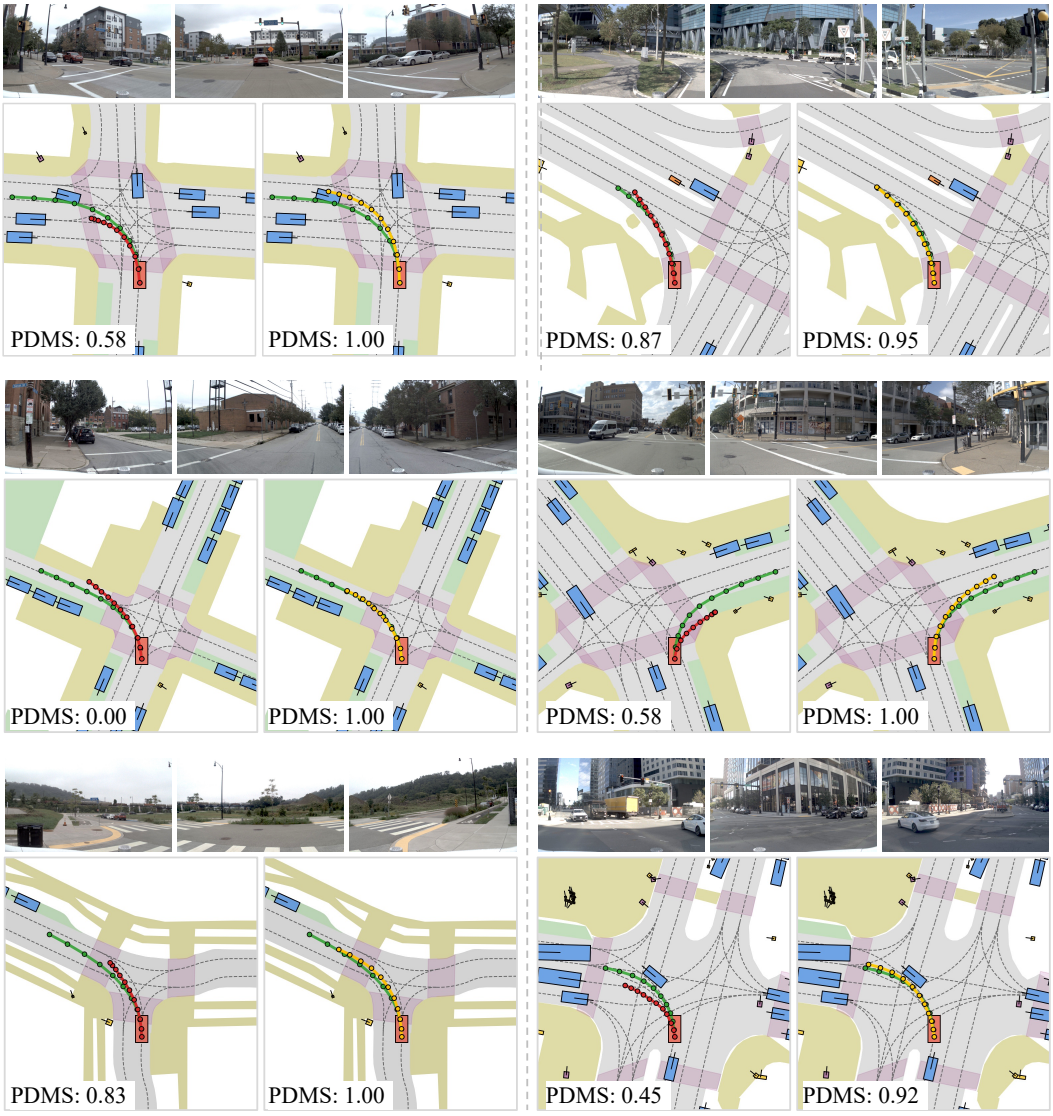



Figure 10: **Qualitative trajectory comparisons.** Ground truth (green), GRPO (red), and DriveBeyond (yellow).

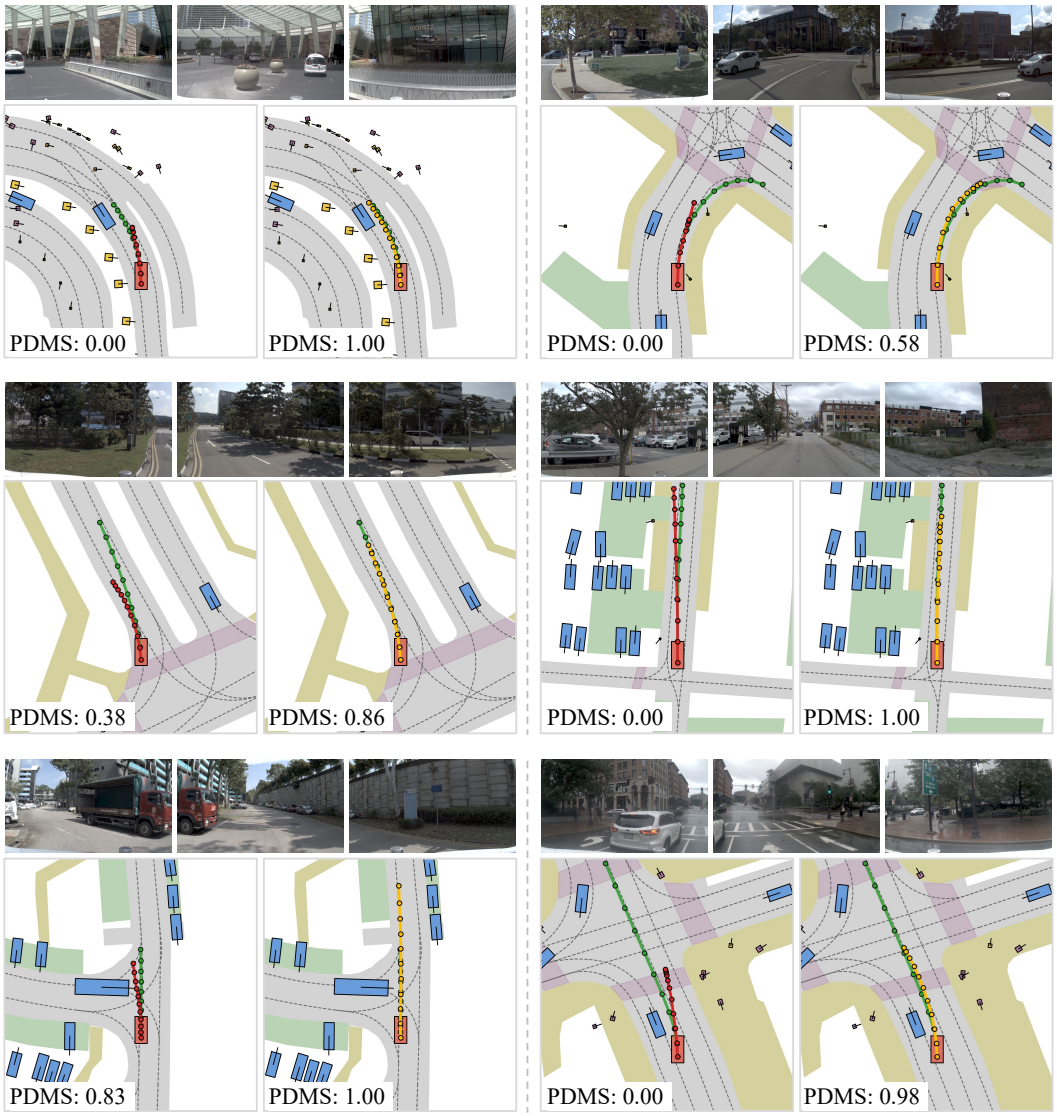



Figure 11: **Qualitative trajectory comparisons.** Ground truth (green), GRPO (red), and DriveBeyond (yellow).