DRIVING-OPD: LEARNING COMPLEMENTARY DRIVING EXPERTS WITH REWARD VARIANCE-AWARE RL AND ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) is increasingly adopted for post-training of Vision-Language-Action (VLA) models in end-to-end autonomous driving. However, existing approaches mainly optimize a single policy, leaving the effect of different RL treatments on driving capabilities underexplored. Under structured driving rewards, we find that GRPO's variance-dependent scaling changes the relative optimization emphasis across rollout groups. Motivated by this, we introduce reward-variance-aware RL, which explores different treatments of variance-dependent scaling to induce complementary driving specializations. Specifically, Dr.GRPO removes standard-deviation normalization and yields a safety-oriented expert, while our Variance-Gated Scaling GRPO (VGS-GRPO) retains controlled variance-dependent scaling and produces a progress-oriented expert from the same base policy and RL data. Building on this finding, we propose Driving-OPD, a three-stage post training framework consisting of supervised fine-tuning (SFT), reward-variance-aware RL for complementary expert learning, and On-Policy Distillation (OPD) for expert integration. We use Win-Margin-Based Scenario Selection and Dual-Teacher Routing to improve multi-expert distillation by providing more effective expert-specific supervision. Without explicit Chain-of-Thought reasoning or additional planning modules, Driving-OPD directly generates trajectories autoregressively and achieves **92.8** PDMS on NAVSIM v1 and **91.4** EPDMS on NAVSIM v2, highlighting the strong post-training potential of autoregressive Driving VLAs.

1 INTRODUCTION

Recent advances in Vision-Language-Action (VLA) models have increasingly extended end-to-end autonomous driving toward unified perception, reasoning, and action generation (Fu et al., 2025; Li et al., 2026a; Xiong et al., 2026; Liu et al., 2025a; Zhou et al., 2026a). Alongside these advances, reinforcement learning (RL) has emerged as an effective post-training strategy for directly optimizing driving-specific objectives beyond imitation learning (Jiang et al., 2025; Xiong et al., 2026; Luo et al., 2025; Yuan et al., 2026b; Xu et al., 2025).

However, existing RL-based post-training approaches often rely on either large-scale optimization over extensive training data (Xiong et al., 2026; Xu et al., 2026b), or carefully curated scenario subsets, sometimes through iterative rollout-based filtering and resampling (Chen et al., 2026). More fundamentally, they typically optimize a single policy under a fixed RL recipe, implicitly assuming that different driving capabilities can improve along the same optimization direction. Yet driving objectives involve heterogeneous signals (Dauner et al., 2024; Cao et al., 2025), ranging from safety-critical failures to fine-grained differences in driving quality, which may respond differently to the same optimization treatment. This raises a broader question: *can different RL treatments uncover complementary driving capabilities rather than forcing them into a single optimization direction?* We study this question in autoregressive Driving VLAs under a simple post-training setting using a small training subset selected through score-based sampling.

To understand how RL emphasizes different driving signals, we examine the contribution of rollout groups in GRPO (Shao et al., 2024) and identify reward variance as a key factor in their rela-

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

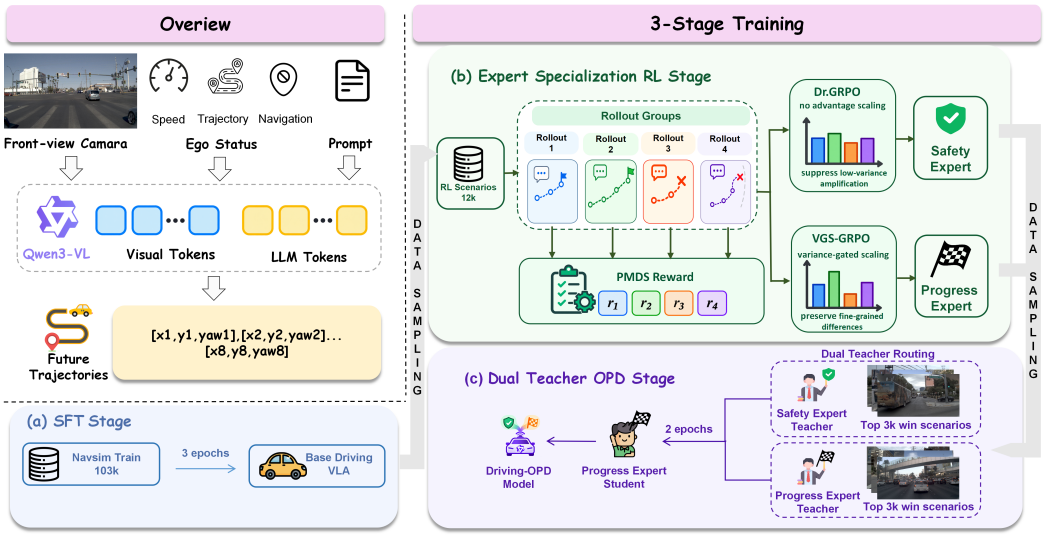

Figure 1: **Overview of Driving-OPD.** Driving-OPD directly generates future trajectories in the autoregressive output space of Qwen3-VL. Our three-stage post-training pipeline consists of (a) SFT for the base policy, (b) Dr.GRPO and VGS-GRPO for learning complementary safety and progress experts, and (c) Dual-Teacher OPD for integrating the two experts into a single policy.

tive optimization emphasis. Under NAVSIM (Dauner et al., 2024)'s PDMS, high-variance groups often contain both successful and safety-critical failed trajectories, whereas low-variance groups mainly reflect finer differences in driving quality. Since GRPO normalizes rewards by the group-wise standard deviation, these groups receive markedly different scaling. We therefore explore *reward-variance-aware RL* through two treatments: Dr.GRPO (Liu et al., 2025b), which removes standard-deviation normalization, and our Variance-Gated Scaling GRPO (VGS-GRPO), which retains controlled variance-dependent scaling. The two treatments induce complementary specializations from the same base policy and RL data: Dr.GRPO yields a safety-oriented expert, while VGS-GRPO produces a progress-oriented expert.

The complementary specializations of the two experts motivate us to integrate their strengths within a single policy. To this end, we propose Driving-OPD, a three-stage *specialize-then-integrate* framework consisting of supervised fine-tuning (SFT), reward-variance-aware RL for complementary expert learning, and On-Policy Distillation (Agarwal et al., 2024; Lu & Lab, 2025) (OPD) for expert integration, as illustrated in Fig. 1. In OPD, the student generates on-policy output tokens, while the expert teacher provides next-token distributions conditioned on the student-generated prefixes, aligning training with the contexts encountered at inference and mitigating the prefix mismatch of off-policy distillation (Ross et al., 2011). To integrate multiple experts, we further introduce Win-Margin-Based Scenario Selection to identify scenarios with clear expert-specific advantages and Dual-Teacher Routing to assign the corresponding safety- or progress-oriented teacher. Together, these designs consolidate complementary driving capabilities into a single autoregressive policy. Without explicit Chain-of-Thought reasoning or additional planning modules, Driving-OPD achieves 92.8 PDMS on NAVSIM v1 (Dauner et al., 2024) and 91.4 EPDMS on NAVSIM v2 (Cao et al., 2025). Our main contributions are summarized as follows:

- We reveal how reward-variance treatment shapes GRPO optimization under structured driving rewards, and develop reward-variance-aware RL to obtain complementary driving experts.
- We propose Driving-OPD, a three-stage *specialize-then-integrate* framework that uses SFT to establish the base policy, reward-variance-aware RL to obtain complementary experts, and OPD to integrate their strengths into a single policy.
- To our knowledge, we are the first to apply OPD to multi-expert integration in Driving VLAs. We further introduce Win-Margin-Based Scenario Selection and Dual-Teacher Routing, enabling Driving-OPD to achieve state-of-the-art performance on NAVSIM.

2 RELATED WORK

Trajectory Generation in Driving VLAs. Recent Driving VLAs mainly follow two trajectory-generation paradigms. One line of work employs dedicated diffusion- or flow-matching-based planners (Xiong et al., 2026; Zhou et al., 2026c; Li et al., 2026b), which take hidden representations from the VLM as conditioning features and decode them into continuous future trajectories. Another directly represents and generates future trajectories within the VLM's autoregressive output space (Lu et al., 2026; Zhou et al., 2026b;a). This allows driving rewards to directly optimize the trajectory-generating VLM policy. Our work investigates how different RL treatments shape such autoregressive driving policies.

RL in Driving VLAs. Existing RL post-training methods for Driving VLAs mainly follow three directions. Some perform RL over large or even full training sets (Xiong et al., 2026; Xu et al., 2026b), while others reduce the RL data through carefully designed, often iterative rollout-based scenario selection (Chen et al., 2026). At the algorithmic level, NoRD (Rawal et al., 2026) identifies difficulty bias in standard GRPO (Shao et al., 2024) and mitigates it with Dr.GRPO (Liu et al., 2025b) by removing reward standard-deviation normalization. While NoRD focuses on recovering suppressed learning signals under a single policy, we further study reward variance as an optimization factor itself. We find that different variance treatments induce distinct and complementary driving capabilities, motivating their specialization and subsequent integration.

On-Policy Distillation. On-Policy Distillation (OPD) has been increasingly explored in Large Language Models (LLMs) for efficient capability transfer and integration (Xu et al., 2026a; Xiao et al., 2026; Yang et al., 2025). Existing methods mainly fall into two categories: one distills a stronger teacher into a smaller student (Agarwal et al., 2024; Zhang et al., 2026), while the other consolidates multiple specialized experts into a single model (Ma et al., 2026; Xu et al., 2026a). In OPD, the student generates on-policy rollouts, while the teacher provides next-token distributions on the resulting student-generated prefixes. Compared with RL methods that rely on multiple rollouts and sparse reward signals, OPD can provide dense token-level supervision from a single student rollout, making capability transfer more data- and compute-efficient. In autonomous driving, existing OPD methods mainly focus on asymmetric single-teacher distillation, either from a larger LLM teacher to a smaller LLM student (Afsharrad et al., 2026) or from a privileged RL teacher to a non-privileged student (Rowe et al., 2026). In contrast, we use OPD to integrate multiple complementary experts derived from the same base Driving VLA.

3 RETHINKING REWARD VARIANCE IN DRIVING GRPO

3.1 REWARD VARIANCE-INDUCED OPTIMIZATION IMBALANCE

We first examine how reward variance affects the group-relative advantage in GRPO. For the i-th sample in a rollout group $\mathcal{G}$, the advantage is computed as

$$A_i^{\mathrm{GRPO}} = \frac{r_i - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}} + \epsilon}, \tag{1}$$

where $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ denote the mean and standard deviation of the group rewards, respectively. The division by $\sigma_{\mathcal{G}}$ assigns much stronger scaling to low-variance groups than to high-variance groups.

Figure 2(a) shows two real rollout groups collected during our RL training. In the low-variance group ($\sigma_{\mathcal{G}} \approx 0.004$), GRPO applies a scaling factor of about $\times 224$, whereas the high-variance group ($\sigma_{\mathcal{G}} \approx 0.424$) is scaled by only about $\times 2.4$. Consequently, very small reward differences in low-variance groups can produce advantage magnitudes comparable to those induced by much larger reward differences in high-variance groups.

This effect is particularly relevant under the structured PDMS reward used in NAVSIM:

$$\mathrm{PDMS} = NC \cdot DAC \cdot \frac{5EP + 5TTC + 2C}{12}, \tag{2}$$

where NC and DAC denote no-at-fault collision and drivable-area compliance, while EP, TTC, and C denote ego progress, time-to-collision, and comfort, respectively. In our rollouts, high-

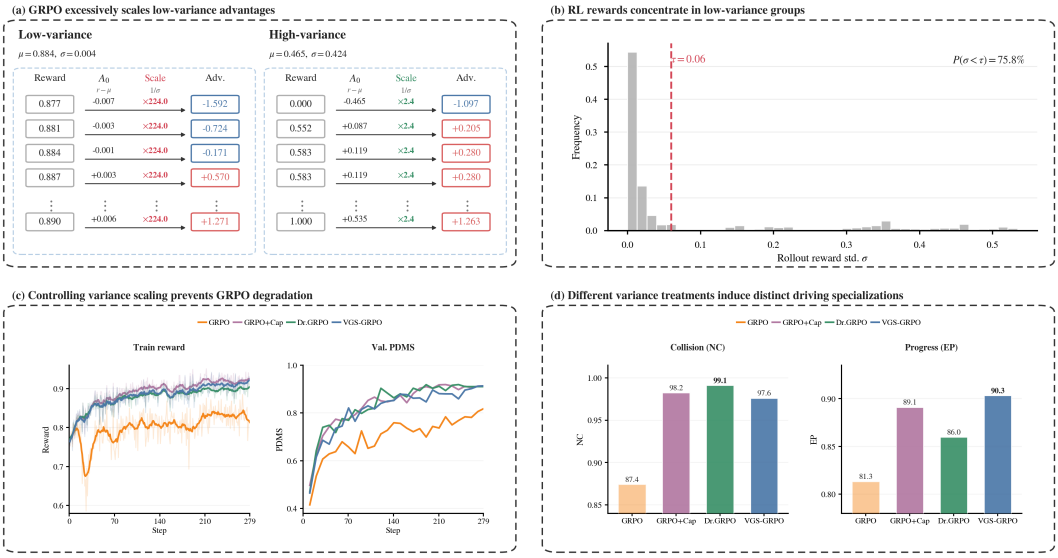

Figure 2: **Reward variance shapes optimization and driving specialization in GRPO.** (a) Standard GRPO applies substantially stronger advantage scaling to low-variance groups than to high-variance groups. (b) Most rollout groups during RL fall into the low-variance regime. (c) Controlling variance-dependent scaling prevents the training degradation of standard GRPO. (d) Different variance treatments induce distinct driving specializations, with Dr.GRPO favoring safety and VGS-GRPO achieving stronger ego progress.

variance groups often contain both successful and safety-critical failed trajectories, whereas low-variance groups mainly reflect finer reward differences among otherwise successful trajectories (see Appendix A for detailed statistics). This creates a variance-dependent imbalance in advantage weighting: small reward differences in low-variance groups receive stronger amplification, whereas reward differences in high-variance groups receive weaker scaling despite being more frequently associated with safety-critical failures. Since low-variance groups also constitute a large fraction of the collected rollouts (Fig. 2(b)), this imbalance is prevalent during RL training.

Under this setting, standard GRPO exhibits clear performance degradation during RL training (Fig. 2(c)). To examine whether this degradation is driven by excessive variance-dependent scaling, we simply limit the maximum scaling factor to $s_{\max} = 8$. This single change is sufficient to prevent the degradation, indicating that excessive scaling is a key contributor to the observed optimization imbalance.

3.2 Reward Variance Shapes Driving Specialization

Preventing training degradation does not necessarily lead to the same driving behavior. NoRD (Rawal et al., 2026) adopts Dr.GRPO, which removes group-wise standard-deviation scaling:

$$A_i^{\text{Dr.GRPO}} = r_i - \mu_{\mathcal{G}}. \tag{3}$$

Consistent with NoRD, we find that Dr.GRPO maintains stable RL training (Fig. 2(c)). Beyond this, Dr.GRPO exhibits a clear safety-oriented specialization, with strong collision avoidance and drivable-area compliance but more conservative improvement in ego progress (Fig. 2(d)). This suggests that reward-variance treatment shapes driving specialization: removing variance-dependent scaling preserves the influence of large reward differences often associated with safety-critical failures, whereas retaining variance sensitivity maintains greater sensitivity to finer reward differences.

4 Methodology

Building on the above analysis, we propose **Variance-Gated Scaling GRPO (VGS-GRPO)**, which retains controlled variance-dependent scaling. Together with Dr.GRPO, the two treatments form our

reward-variance-aware RL strategy and yield complementary progress- and safety-oriented experts from the same base policy and RL data. We instantiate the policy with the native Qwen3-VL architecture and directly generate future trajectories without explicit Chain-of-Thought reasoning or additional planning modules. Based on these experts, we further propose **Driving-OPD**, a three-stage specialize-then-integrate framework (Fig. 1) consisting of supervised fine-tuning, reward-variance-aware RL for expert specialization, and On-Policy Distillation (OPD) for expert integration.

4.1 Problem Definition

Given camera observations I_{cam}, ego states S_{ego}, and navigation information L_{nav}, autonomous driving aims to predict a future ego trajectory

$$\mathbf{T} = \Phi(I_{\text{cam}}, S_{\text{ego}}, L_{\text{nav}}), \tag{4}$$

where $\mathbf{T} = (\mathbf{w}_1, \ldots, \mathbf{w}_H)$ denotes a sequence of H future waypoints. We represent the ego states S_{ego} and navigation information L_{nav} as text in the input prompt and instantiate Φ with the native Qwen3-VL architecture. Let $\mathcal{O} = \{I_{\text{cam}}, S_{\text{ego}}, L_{\text{nav}}\}$ denote the model input. Trajectory prediction is then formulated as autoregressive generation in the VLM output space:

$$p_\theta(\mathbf{T} \mid \mathcal{O}) = \prod_{t=1}^{H} p_\theta(\mathbf{w}_t \mid \mathcal{O}, \mathbf{w}_{<t}). \tag{5}$$

4.2 VGS-GRPO

Building on our finding that reward-variance treatment shapes the capabilities acquired during RL, we propose Variance-Gated Scaling GRPO (VGS-GRPO), which retains controlled variance-dependent scaling to preserve sensitivity to fine-grained reward differences. This provides a complementary variance treatment to Dr.GRPO and yields a progress-oriented specialization.

For each prompt q, we sample K completions with rewards $\{r_{q,i}\}_{i=1}^{K}$ and compute

$$\mu_q = \frac{1}{K}\sum_{i=1}^{K} r_{q,i}, \qquad \sigma_q = \mathrm{Std}\left(\{r_{q,i}\}_{i=1}^{K}\right). \tag{6}$$

We then define a variance-gating coefficient

$$\beta_q = \frac{1}{1 + \exp\left(k(\sigma_q - \tau)\right)}, \tag{7}$$

and use it to control the group-wise scaling factor:

$$s_q = \min\left(\frac{1}{(1-\beta_q) + \beta_q\sigma_q + \epsilon}, s_{\max}\right). \tag{8}$$

The resulting advantage is

$$A_{q,i}^{\text{VGS-GRPO}} = s_q\left(r_{q,i} - \mu_q\right). \tag{9}$$

We set $\tau = 0.06$, which separates the low-variance regime containing approximately 75.8% of the rollout groups in our training distribution (Fig. 2(b)), and use $k = 40$ to control the transition between the two regimes. For high-variance groups, β_q approaches zero and the scaling approaches that of Dr.GRPO, preserving their corrective signals. For low-variance groups, a larger β_q retains additional scaling for fine-grained reward differences, while $s_{\max} = 8$ prevents the extreme amplification of standard GRPO.

As shown in Fig. 2(d), different variance treatments lead to distinct driving specializations. Dr.GRPO achieves the strongest collision safety, whereas VGS-GRPO reaches the highest ego progress. We therefore use the Dr.GRPO policy as the safety expert and the VGS-GRPO policy as the progress expert for subsequent expert integration.

4.3 On-Policy Distillation of Complementary Experts

We use On-Policy Distillation (OPD) to integrate the RL experts into a single policy. Both experts originate from the same base Driving VLA and share the same architecture and autoregressive output space, allowing their token-level distributions to directly supervise the student on its own rollouts.

Let π_θ denote the student policy and π_ϕ the teacher policy. OPD can use different distribution-matching objectives, including forward KL, reverse KL, and Jensen–Shannon divergence (JSD). Following (Yuan et al., 2026a), we adopt the symmetric JSD with equal weighting:

$$m_t = \frac{1}{2}\left(\pi_\theta^t + \pi_\phi^t\right), \qquad D_{\mathrm{JSD}}^t = \frac{1}{2} D_{\mathrm{KL}}\left(\pi_\theta^t \| m_t\right) + \frac{1}{2} D_{\mathrm{KL}}\left(\pi_\phi^t \| m_t\right), \tag{10}$$

where π_θ^t and π_ϕ^t denote the student and teacher next-token distributions conditioned on the same student-generated prefix $(x, y_{<t})$. The distillation objective is

$$\mathcal{L}_{\mathrm{OPD}} = \mathbb{E}_{x,\, y\sim\pi_\theta(\cdot|x)}\left[\frac{1}{|y|}\sum_{t=1}^{|y|} D_{\mathrm{JSD}}^t\right]. \tag{11}$$

We find that the progress expert performs better on most training scenarios, while the safety expert has clear advantages on a smaller set of safety-critical cases (see Appendix C). We therefore initialize the student from the progress expert and aim to transfer the complementary capability of the safety expert into this policy.

To identify the scenarios that best represent the strengths of each expert, we introduce **Win-Margin-Based Scenario Selection**. For each training scenario x, we evaluate the progress and safety experts and obtain their driving scores $R_P(x)$ and $R_S(x)$. We define

$$\Delta(x) = R_P(x) - R_S(x), \tag{12}$$

where the sign of $\Delta(x)$ determines the better-performing expert and $|\Delta(x)|$ measures its winning margin. We divide the scenarios into progress-winning and safety-winning subsets, rank each subset by the corresponding winning margin, and select the top 3K scenarios from each, resulting in a 6K-scenario distillation set.

Distilling only from the safety expert can improve safety but may degrade the progress capability already present in the initialized student. We therefore introduce **Dual-Teacher Routing**: safety-winning scenarios are supervised by the safety expert, while progress-winning scenarios are supervised by the progress expert. In this way, the safety expert transfers its complementary capability without sacrificing the progress strength of the initialized policy.

5 Experiments

5.1 Experimental Setup

Benchmarks and Evaluation Metrics. We conduct experiments on NAVSIM v1 (Dauner et al., 2024) and NAVSIM v2 (Cao et al., 2025). NAVSIM combines real-world sensor data with simulation-based planning metrics, enabling scalable evaluation of autonomous driving policies beyond conventional open-loop trajectory errors. For NAVSIM v1, we train on the `navtrain` split and evaluate on `navtest`, reporting PDMS as the primary metric. For NAVSIM v2, we evaluate on `navtest` and report EPDMS, which extends PDMS with additional safety, rule-compliance, and comfort metrics. Detailed definitions of the evaluation metrics are provided in Appendix F.

Implementation Details. We use Qwen3-VL-4B as the base model and adopt a three-stage training pipeline consisting of SFT, RL, and OPD. In the SFT stage, the model is trained for 3 epochs on 103K `navtrain` expert trajectories. For RL, we first run inference with the SFT model on the training set and construct a 12K-sample RL subset through stratified sampling, which includes all identified bad cases together with samples stratified by ego-progress intervals. Starting from the same SFT checkpoint, we separately apply VGS-GRPO and Dr.GRPO for 3 epochs to obtain the progress expert and the safety expert. Following the scenario-selection strategy described in Sec. 4.3, we then select 3K scenarios for each expert, resulting in a 6K-sample distillation set. Finally, the progress expert is used to initialize the student, which is further trained with dual-teacher OPD for 2 epochs.

Table 1: **Performance comparison on NAVSIM v1 `navtest`.** We compare Driving-OPD with representative methods across major driving paradigms.

Method Family	Method	Venue	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	**PDMS↑**
Autoregressive VLA	AutoVLA (Zhou et al., 2026b)	NeurIPS'25	98.4	95.6	**98.0**	99.9	81.9	89.1
	Curious-VLA (Chen et al., 2026)	CVPR'26 F	98.4	96.9	97.9	98.1	88.5	90.3
	DriveTeach-VLA (Yang et al., 2026b)	ECCV'26	98.5	96.9	97.9	98.2	88.5	90.4
	DriveMA-4B (Zheng et al., 2026)	arXiv'26	98.7	97.8	95.5	99.9	86.7	91.2
World Model / WAM	WoTE (Li et al., 2025)	ICCV'25	98.5	96.8	94.9	99.9	81.9	88.3
	DriveVLA-W0 (Li et al., 2026a)	ICLR'26	98.7	**99.1**	95.3	99.3	83.3	90.2
	DriveWorld-VLA (Liu et al., 2026)	arXiv'26	**99.1**	98.2	96.1	**100.0**	85.9	91.3
	Discrete-WAM (Yao et al., 2026)	arXiv'26	98.8	98.4	95.3	**100.0**	88.7	92.2
Diffusion / Flow Matching	DiffusionDrive (Liao et al., 2025)	CVPR'25	98.2	96.2	94.7	**100.0**	82.2	88.1
	ReCogDrive (Xiong et al., 2026)	ICLR'26	97.9	97.3	94.9	**100.0**	87.3	90.8
	DriveLaW (Xia et al., 2026)	CVPR'26	99.0	97.1	96.7	**100.0**	81.3	89.1
	DiffusionDriveV2 (Zou et al., 2025)	arXiv'25	98.3	97.9	94.8	99.9	87.5	91.2
Ours	Progress Expert	–	97.6	97.3	91.5	**100.0**	**90.3**	90.6
	Safety Expert	–	**99.1**	98.5	96.6	99.9	86.0	91.8
	Driving-OPD	–	99.0	98.3	96.5	**100.0**	88.8	**92.8**

Table 2: **Performance comparison on NAVSIM v2 `navtest`.** EPDMS* denotes results computed with the original NAVSIM v2 evaluation code before the human-behavior filtering fix, while EPDMS denotes results under the corrected protocol.

Method Family	Method	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS* ↑	**EPDMS↑**
Autoregressive VLA	Curious-VLA (Chen et al., 2026)	98.4	96.9	99.2	**99.8**	88.5	97.9	96.9	98.1	81.5	–	85.3
	ELF-VLA (Luo et al., 2026)	98.9	98.1	99.4	**99.8**	88.5	98.4	96.9	**98.3**	87.2	–	87.1
World Model / WAM	Epona (Zhang et al., 2025)	97.1	95.7	99.3	99.7	88.6	96.3	97.0	98.0	67.8	–	85.1
	DriveVLA-W0 (Li et al., 2026a)	98.5	**99.1**	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1	–
	DreamerAD (Yang et al., 2026a)	98.0	97.2	99.5	**99.8**	87.8	97.4	**97.5**	**98.3**	72.4	–	87.7
	Discrete-WAM (Yao et al., 2026)	98.5	98.2	**99.7**	**99.8**	90.5	97.9	97.2	**98.3**	78.1	87.0	90.4
Diffusion / Flow Matching	DiffusionDrive (Liao et al., 2025)	98.2	95.9	99.4	**99.8**	87.5	97.3	96.8	**98.3**	87.7	–	84.5
	ReCogDrive (Xiong et al., 2026)	98.3	95.2	99.5	**99.8**	87.1	97.5	96.6	**98.3**	86.5	83.6	–
	WAM-Flow (Xu et al., 2026b)	98.5	94.5	99.5	**99.8**	86.9	96.8	97.4	97.6	73.9	84.7	–
	DiffusionDriveV2 (Zou et al., 2025)	97.7	96.6	99.2	**99.8**	88.9	97.2	96.0	97.8	**91.0**	85.5	87.5
Ours	**Driving-OPD**	**99.0**	98.3	99.2	99.7	**91.3**	**98.8**	96.9	98.2	83.2	**87.8**	**91.4**

5.2 Main Results

Results on NAVSIM. Tabs. 1 and 2 report the results on NAVSIM v1 and NAVSIM v2, respectively. We compare Driving-OPD with representative methods across major driving paradigms, including world-model-based methods, diffusion- or flow-matching-based trajectory planners, and autoregressive Driving VLAs. The two RL experts retain complementary strengths on NAVSIM v1. Integrating them through OPD further improves the overall PDMS to 92.8, surpassing either expert individually. Driving-OPD therefore establishes a new state of the art among purely autoregressive Driving VLAs while remaining competitive with recent world-model-based and diffusion-based approaches. Fig. 3 visualizes the behaviors of the two experts and the resulting Driving-OPD policy.

On NAVSIM v2, Driving-OPD further achieves 91.4 EPDMS on the `navtest` split under the corrected evaluation protocol. Although our RL training still uses the NAVSIM v1-style PDMS as the reward and is not specifically adapted to the v2 EPDMS formulation, the resulting policy maintains strong performance. This suggests that the learned policy remains robust to changes in the evaluation metric beyond the reward used during RL training.

5.3 Ablation Experiments

We conduct ablation studies on both the RL and OPD stages. All reported ablation results are evaluated on the full NAVSIM v1 `navtest` split.

Table 3: **Ablation on reward-variance-aware RL.** All RL variants are initialized from the same SFT checkpoint and trained with the same RL data and number of optimization steps.

Method	PDMS↑	NC↑	DAC↑	EP↑	TTC↑	Comf.↑
SFT	88.8	98.9	96.1	82.4	96.2	**100.0**
GRPO	75.9	87.4	93.8	81.3	73.1	99.9
GRPO + Cap	**91.8**	98.3	98.0	89.1	94.5	**100.0**
Dr.GRPO	**91.8**	**99.1**	**98.5**	86.0	**96.6**	99.9
VGS-GRPO	90.6	97.6	97.3	**90.3**	91.5	**100.0**

Table 4: **Ablation on On-Policy Distillation.** We ablate the divergence objective, win-margin-based data selection, and dual-teacher routing.

Divergence	Win-Margin	Dual-T.	PDMS↑	NC↑	DAC↑	EP↑	TTC↑	Comf.↑
JSD	✓	✓	**92.8**	99.0	98.3	88.8	96.5	**100.0**
Forward KL	✓	✓	**92.8**	98.9	98.3	89.5	95.8	99.9
Reverse KL	✓	✓	92.6	99.1	98.3	88.4	96.5	**100.0**
JSD	×	✓	92.5	98.8	98.0	**89.6**	95.6	**100.0**
JSD	✓	×	92.0	**99.2**	**98.6**	86.0	**96.9**	99.9

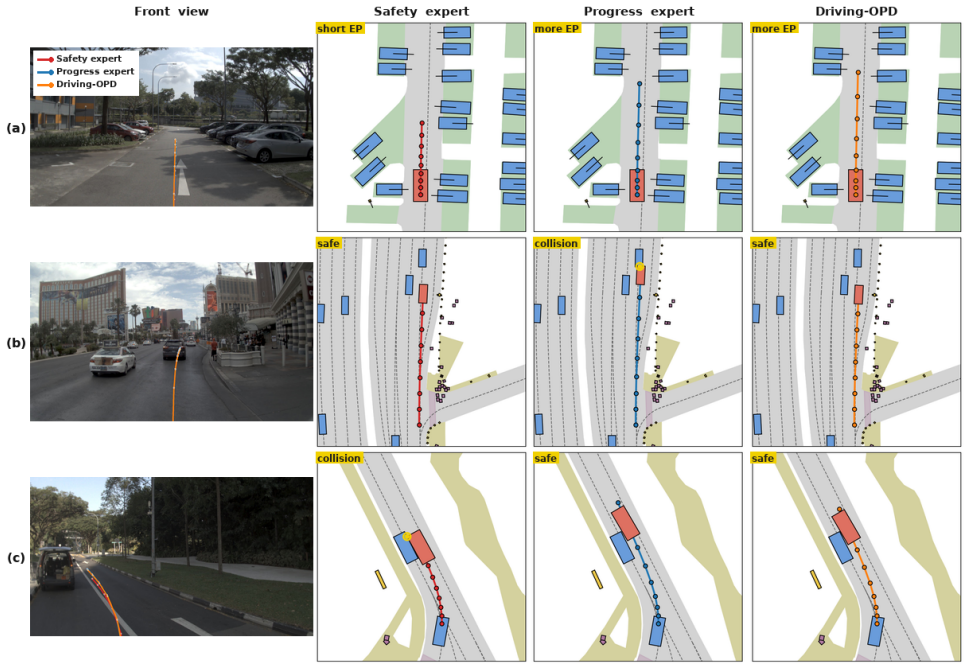

Figure 3: **Qualitative comparison of the two RL experts and Driving-OPD.** (a) The progress expert achieves larger ego progress. (b) The safety expert avoids a collision made by the progress expert. (c) The progress expert avoids a collision missed by the safety expert. Driving-OPD integrates the complementary strengths of both experts.

Ablation on RL. Tab. 3 compares different RL strategies under the same training data and optimization budget. Standard GRPO suffers from substantial performance degradation, whereas simply bounding its scaling factor recovers the PDMS from 75.9 to 91.8. Dr.GRPO and VGS-GRPO also maintain strong overall performance, achieving PDMS scores of 91.8 and 90.6, respectively, while showing different strengths across driving metrics. These results demonstrate that controlling variance-dependent scaling is important for stable RL optimization, and that different variance treatments lead to different policy behaviors.

Ablation on On-Policy Distillation. Tab. 4 evaluates the main design choices in our OPD stage. For the distillation objective, JSD and forward KL achieve the same final PDMS of 92.8, while reverse KL performs slightly worse at 92.6. We adopt JSD as it maintains relatively strong performance in the later stage of distillation (Fig. 4(a)). For data selection, Win-Margin-Based Data Selection improves PDMS from 92.5 to 92.8 compared with random sampling from the corresponding expert-winning subsets, showing the benefit of prioritizing scenarios with larger expert advantages. Finally, using only the safety teacher improves safety-related metrics but substantially reduces EP

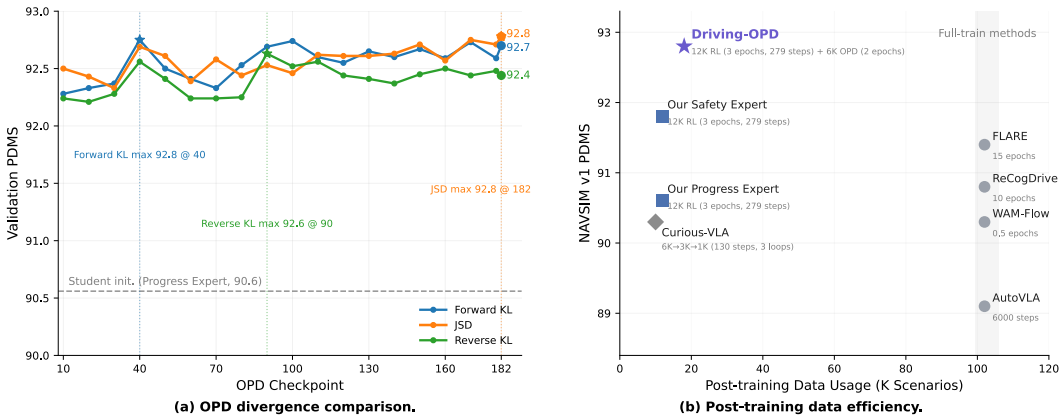

Figure 4: **Additional analyses of Driving-OPD.**

to 86.0. Dual-Teacher Routing retains supervision from the progress expert while introducing the complementary safety supervision, leading to the best overall PDMS of 92.8.

5.4 Progress Expert Selection for OPD

Although GRPO+Cap achieves a higher standalone PDMS than VGS-GRPO (91.8 vs. 90.6), VGS-GRPO exhibits a stronger progress specialization, reaching an EP of 90.3 compared with 89.1 for GRPO+Cap. To directly evaluate their suitability as the progress expert, we replace VGS-GRPO with GRPO+Cap and apply the same Win-Margin-Based Scenario Selection and Dual-Teacher Routing strategy. The resulting policy achieves 92.4 PDMS, lower than the 92.8 obtained with VGS-GRPO. This confirms that stronger complementary specialization, rather than higher standalone PDMS, is more beneficial for expert integration.

5.5 Data Efficiency

We further compare with representative methods whose RL training recipes are publicly available (Chen et al., 2026; Zhou et al., 2026b; Xie et al., 2026; Xiong et al., 2026; Xu et al., 2026b), as shown in Fig. 4(b). Unlike Curious-VLA, which repeatedly performs policy-dependent offline rollouts and diversity-aware scenario resampling over three ADAS rounds (6K→3K→1K), both of our RL experts are trained on the same fixed 12K subset obtained through simple one-time score-based filtering. Despite this simpler data pipeline, Driving-OPD achieves strong performance with substantially fewer unique post-training scenarios than full-data RL methods.

6 Conclusion

In this work, we presented Driving-OPD, a post-training framework for autoregressive Driving VLAs. We showed that variance-dependent scaling in standard GRPO can cause substantial optimization imbalance, while different reward-variance treatments lead to distinct driving specializations. Based on this observation, Dr.GRPO and our VGS-GRPO learn complementary safety- and progress-oriented experts, which are further integrated through On-Policy Distillation. Extensive experiments on NAVSIM v1 and v2 demonstrate strong driving performance. Notably, Driving-OPD directly generates trajectories autoregressively in the VLM output space, without explicit Chain-of-Thought reasoning or additional planning modules. We hope this work encourages further exploration of specialized RL post-training and expert integration for Driving VLAs.

AI use statement

Generative AI tools were used to assist with language editing, including improving the clarity, concision, and organization of the manuscript; literature search and organization; and code implemen-

tation and debugging. All AI-assisted outputs, including text and code, were carefully reviewed and verified by the authors. The authors take full responsibility for the content of the final manuscript and the correctness of the implementation.

Reproducibility Statement

Our experiments are based on the publicly available Qwen3-VL model and the public NAVSIM dataset. The implementation details are described in Implementation Details paragraph, while additional information on the RL data subset and scenario selection is provided in Appendix B. Detailed training settings and hyperparameters are provided in Appendix E. Together, these sections document the settings used in our experiments and are intended to facilitate reproduction of the reported results.

References

Amirhossein Afsharrad, Amirhesam Abedsoltan, Ahmadreza Moradipari, and Sanjay Lall. On-policy distillation of language models for autonomous vehicle motion planning. *arXiv preprint arXiv:2604.07944*, 2026.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Conference on Robot Learning (CoRL)*, 2025.

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

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

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

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*, 2025.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27137–27146. IEEE, 2025.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 7890–7911, 2026a.

Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, et al. Unidrivevla: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026b.

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

Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, Long Chen, et al. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving. *arXiv preprint arXiv:2602.06521*, 2026.

Ruixun Liu, Lingyu Kong, Derun Li, and Hang Zhao. Occvla: Vision-language-action model with implicit 3d occupancy supervision. *arXiv preprint arXiv:2509.05578*, 2025a.

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*, 2025b.

Jinghui Lu, Jiayi Guan, Zhijian Huang, Jinlong Li, Guang Li, Lingdong Kong, Yingyan Li, Han Wang, Shaoqing Xu, Yuechen Luo, et al. Xiaomi onevl: One-step latent reasoning and planning with vision-language explanation. *arXiv preprint arXiv:2604.18486*, 2026.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, et al. Adathinkdrive: Adaptive thinking via reinforcement learning for autonomous driving. *arXiv preprint arXiv:2509.13769*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24833–24842, 2026.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

Ishaan Rawal, Shubh Gupta, Yihan Hu, and Wei Zhan. Nord: A data-efficient vision-language-action model that drives without reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Luke Rowe, Roger Girgis, Rodrigue de Schaetzen, Daphne Cornelisse, Alaap Grandhi, Felix Heide, Eugene Vinitsky, Christopher Pal, and Liam Paull. Scaling self-play for end-to-end driving. *arXiv preprint arXiv:2606.19641*, 2026.

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.

Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, et al. Drivelaw: Unifying planning and video generation in a latent driving world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39701–39712, 2026.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Chengen Xie, Chonghao Sima, Tianyu Li, Bin Sun, Junjie Wu, Zhihui Hao, and Hongyang Li. Flare: Learning future-aware latent representations from vision-language models for autonomous driving. *arXiv preprint arXiv:2601.05611*, 2026.

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 *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Mingwang Xu, Jiahao Cui, Feipeng Cai, Hanlin Shang, Zhihao Zhu, Shan Luan, Yifang Xu, Neng Zhang, Yaoyi Li, Jia Cai, et al. Wam-diff: A masked diffusion vla framework with moe and online reinforcement learning for autonomous driving. *arXiv preprint arXiv:2512.11872*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24918–24928, 2026b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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*, 2026a.

Yuguang Yang, Canyu Chen, Zhewen Tan, Yizhi Wang, Zichao Feng, Chunyang Liu, Kehua Sheng, Juan Zhang, Linlin Yang, Baochang Zhang, et al. Teaching vision-language-action models what to see and where to look. In *European Conference on Computer Vision*, pp. 317–335. Springer, 2026b.

Ziyang Yao, Haochen Liu, Yuncheng Jiang, Zeyu Zhu, Zibin Guo, Jingru Wang, Tianle Liu, Jianwei Cui, Kuiyuan Yang, Hongwei Xie, et al. Discrete-wam: Unified discrete vision-action token editing for world-policy learning. *arXiv preprint arXiv:2606.05645*, 2026.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-opd: Learning to see fine details for multimodal llms via on-policy self-distillation. *arXiv preprint arXiv:2605.18740*, 2026a.

Zhenlong Yuan, Chengxuan Qian, Jing Tang, Rui Chen, Zijian Song, Lei Sun, Xiangxiang Chu, Yujun Cai, Dapeng Zhang, and Shuo Li. Autodrive-r^2: Incentivizing reasoning and self-reflection capacity for vla model in autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 97934–97955, 2026b.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler II, Qian Qian, Gregory D Lyng, Sanjit Singh Batra, and Robert E Tillman. Fast and effective on-policy distillation from reasoning prefixes. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25553–25569, 2026.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, et al. Epona: Autoregressive diffusion world model for autonomous driving. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27220–27230. IEEE, 2025.

Weicheng Zheng, Yixin Huang, Qiao Sun, Derun Li, and Hang Zhao. Drivema: Driving vision-language-action models with verifiable meta-actions. *arXiv preprint arXiv:2605.31271*, 2026.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13782–13790, 2026a.

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. *Advances in Neural Information Processing Systems*, 38: 27920–27956, 2026b.

Zewei Zhou, Ruining Yang, 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*, 2026c.

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

A ANALYSIS OF REWARD VARIANCE AND ADVANTAGE MAGNITUDE

To further examine how reward variance affects the advantage magnitude induced by different advantage formulations, we analyze 35,712 rollout groups collected during RL training and divide them into five bins according to the group reward standard deviation σ. For a rollout group g, let

$$\Delta_{g,i} = r_{g,i} - \mu_g, \tag{13}$$

where $r_{g,i}$ is the reward of the i-th rollout and μ_g is the group mean reward. Under an advantage formulation m, the resulting advantage can be written as

$$A_{g,i}^{(m)} = s_g^{(m)} \Delta_{g,i}, \tag{14}$$

where $s_g^{(m)}$ denotes the corresponding group-wise scale factor. To compare how different formulations behave across rollout groups with different reward variances, we analyze the absolute advantage magnitude $|A|$.

Fig. 5 summarizes how different reward-variance treatments reshape advantage magnitudes across variance regimes.

(a) Group distribution and hard-failure rates. All-zero groups have the smallest possible reward variance, but they do not produce a learning signal because every rollout receives the same reward. We therefore exclude these groups from the following statistics. Panel (a) reports the distribution of the remaining rollout groups together with two failure statistics: the fraction of rollouts with PDMS$= 0$ and the fraction of groups containing at least one PDMS$= 0$ rollout. Both statistics increase substantially in the high-variance regime. Since PDMS$= 0$ corresponds to a hard failure caused by either NC or DAC, high reward variance is more strongly associated with groups containing safety-critical failures. In contrast, lower-variance groups mainly reflect finer reward differences among otherwise successful rollouts.

(b) Mean absolute advantage. We first measure the typical absolute advantage magnitude within each variance bin. Let $\mathcal{G}_b$ denote the set of rollout groups assigned to bin b. For each group, we average the absolute advantage over its K rollouts and then average over all groups in the bin:

$$\overline{|A|}_b^{(m)} = \frac{1}{|\mathcal{G}_b|} \sum_{g \in \mathcal{G}_b} \frac{1}{K} \sum_{i=1}^{K} \left| A_{g,i}^{(m)} \right|. \tag{15}$$

This quantity characterizes the typical per-rollout advantage magnitude in a variance regime and is independent of how many groups fall into that bin. Standard GRPO produces large absolute advantages even for low-variance groups, making their magnitudes comparable to those of

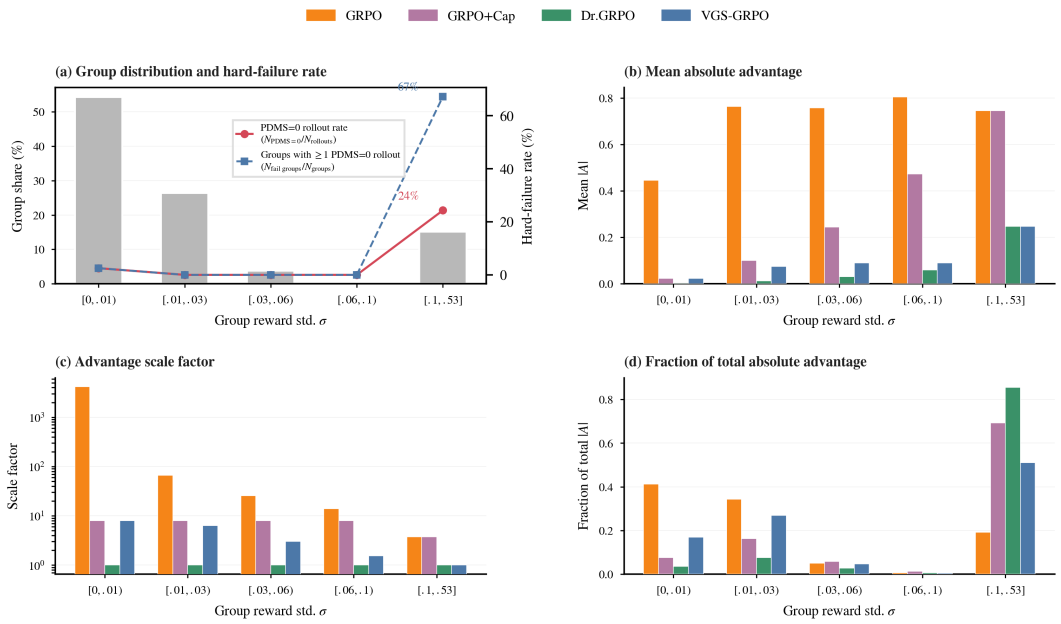

Figure 5: **Reward variance and advantage magnitude.** (a) Group distribution and hard-failure rates across reward-variance bins. High-variance groups are substantially more likely to contain PDMS$=0$ rollouts. (b) Mean absolute advantage under different advantage formulations. (c) Mean scale factor applied to $r-\mu$. (d) Fraction of the total absolute advantage contributed by each variance bin.

high-variance groups despite much smaller underlying reward differences. Dr.GRPO removes this variance-dependent amplification and therefore keeps low-variance advantages substantially smaller. GRPO+Cap and VGS-GRPO retain controlled amplification in the low-variance regime.

(c) Advantage scale factor. Panel (c) isolates the scaling mechanism by directly measuring how strongly the centered reward $\Delta_{g,i}$ is multiplied. We report the average group-wise scale factor within each variance bin:

$$\bar{s}_b^{(m)} = \frac{1}{|\mathcal{G}_b|} \sum_{g \in \mathcal{G}_b} s_g^{(m)}. \tag{16}$$

Standard GRPO applies extremely large scaling when σ is small because of its $1/\sigma$ normalization, reaching an average scale of 4138 in the lowest-variance bin. The scale then decreases rapidly as σ increases. GRPO+Cap bounds this amplification with a maximum scale of 8, whereas Dr.GRPO removes variance-dependent scaling entirely and uses a constant scale of 1. VGS-GRPO also bounds the maximum scale at 8, but gradually reduces the scaling as reward variance increases. It therefore preserves greater sensitivity to fine-grained low-variance reward differences without the extreme amplification of standard GRPO.

(d) Fraction of total absolute advantage. Finally, Panel (d) measures the share of the total absolute advantage assigned to each variance regime. For each bin, we sum the absolute advantages of all rollouts in that bin and normalize by the corresponding sum over all rollout groups:

$$C_b^{(m)} = \frac{\sum_{g \in \mathcal{G}_b} \sum_{i=1}^{K} \left| A_{g,i}^{(m)} \right|}{\sum_{g} \sum_{i=1}^{K} \left| A_{g,i}^{(m)} \right|}. \tag{17}$$

Unlike Panel (b), this quantity depends on both the typical advantage magnitude and the number of groups in each variance regime. It therefore captures how the aggregate absolute-advantage magnitude is distributed across the training data.

Table 5: **Details of the 12K RL training subset.**

Category / EP interval	Candidates	Selected	Sampling rate
Bad cases	2,270	2,270	100.0%
Good, EP< 0.5	1,033	0	0.0%
Good, $[0.5, 0.6)$	4,132	1,946	47.1%
Good, $[0.6, 0.7)$	8,958	1,946	21.7%
Good, $[0.7, 0.8)$	15,789	1,946	12.3%
Good, $[0.8, 0.9)$	19,001	1,946	10.2%
Good, $[0.9, 1.0]$	51,900	1,946	3.7%
Total	103,083	12,000	–

Standard GRPO assigns 75.4% of the total absolute advantage to groups with $\sigma < 0.03$, concentrating most of the aggregate absolute advantage in the low-variance regime. In contrast, Dr.GRPO assigns 85.6% to the high-variance regime with $\sigma \geq 0.1$, where hard failures occur much more frequently. GRPO+Cap also shifts a larger share of the total absolute advantage toward high-variance groups, with 69.2% of the total absolute advantage in this regime. VGS-GRPO produces a more balanced allocation: 51.1% lies in the high-variance regime, while 43.8% is retained in the two lowest-variance bins. These distinct distributions of absolute advantage are consistent with the safety-oriented specialization of Dr.GRPO and the stronger progress-oriented specialization of VGS-GRPO observed in the main experiments.

B RL Training Subset Construction

We construct the 12K RL training subset from the full NAVSIM training set using the SFT model itself. Specifically, we perform deterministic inference ($T = 0$) with the SFT model on all 103,083 training samples and evaluate the predicted trajectories using the official PDMS metrics. Based on these results, we retain all identified bad cases and perform progress-stratified sampling over the remaining successful cases. The sampling is conducted with a fixed random seed of 42. The detailed subset composition is summarized in Tab. 5.

We define a sample as a bad case if trajectory parsing fails, or if either NC$= 0$ or DAC$= 0$. This results in 2,270 bad cases, all of which are retained in the RL subset. Among the remaining samples with NC> 0 and DAC> 0, we use the ego-progress (EP) score obtained from the SFT prediction for stratified sampling. Samples with EP< 0.5 are excluded, while those with EP≥ 0.5 are divided into five intervals: $[0.5, 0.6)$, $[0.6, 0.7)$, $[0.7, 0.8)$, $[0.8, 0.9)$, and $[0.9, 1.0]$. We uniformly sample 1,946 examples from each interval, resulting in 9,730 successful samples. Together with the 2,270 retained bad cases, this produces the final 12,000-sample RL training subset.

This sampling strategy preserves all identified hard-failure cases while preventing the RL subset from being dominated by high-progress trajectories. Lower-EP successful samples are therefore retained at substantially higher rates than high-EP samples, yielding greater representation of medium- and low-progress cases in the final RL subset.

C Complementarity Analysis of Driving Experts

We further examine the sample-level complementarity between the progress expert and the safety expert on both the NAVSIM training and test sets. For each scenario, we compare the official PDMS of the trajectories generated by the two experts. A scenario is counted as a win for an expert when its PDMS is strictly higher; otherwise, it is counted as a tie. The resulting statistics are summarized in Tab. 6.

The progress expert achieves a higher PDMS on the majority of non-tied scenarios on both splits, while the safety expert remains preferable on a smaller but non-negligible subset. We therefore use the progress expert as the student initialization in OPD and introduce the safety expert as the complementary teacher.

Table 6: **Sample-level complementarity between the progress expert (PE) and safety (SE) expert.**

	Sample-level comparison			PDMS		
Split	PE wins	SE wins	Ties	SE	PE	Oracle
NAVSIM train	55.1%	5.1%	39.8%	94.81	95.15	97.49
NAVSIM test	57.8%	8.6%	33.5%	91.75	90.56	94.98

Table 7: **Input and output configuration.**

Item	Setting
Camera	Current `CAM_F0` only
History	4 ego poses at 2 Hz, $t - 1.5$ s to t
Prediction	8 waypoints at 2 Hz, $t + 0.5$ s to $t + 4.0$ s
Waypoint	$[x, y, \text{yaw}]$ in the current ego frame
Coordinates	x: forward, y: left, yaw in radians
Precision	Two decimal places
Navigation	left / straight / right / unknown
Output	`<trajectory>...</trajectory>`, without CoT

To quantify the remaining complementarity, we further report an oracle that selects the higher-PDMS trajectory from the two experts for each scenario. The oracle reaches 97.49 PDMS on NAVSIM train and 94.98 on NAVSIM test, substantially exceeding either individual expert. This gap indicates that the two experts retain complementary strengths at the scenario level, motivating their integration through OPD.

D Input and Output Representation

Our main SFT→RL→OPD pipeline uses a single current front-view image (`CAM_F0`), together with textual ego-state, navigation, and trajectory-history information. No multi-view or temporal image input is used. The input and output configuration is summarized in Tab. 7.

The historical poses and future waypoints are expressed in the current ego coordinate frame. The prompt additionally provides the current longitudinal and lateral velocity, longitudinal acceleration, and yaw rate. The model directly generates the eight future waypoints without explicit Chain-of-Thought reasoning or intermediate predictions.

D.1 Prompt Template

The following prompt template is used throughout the main SFT→RL→OPD pipeline.

System

You are a driving assistant that generates safe and accurate actions.

User

The image is the front view at the current time.
Command: {left / straight / right / unknown}.
Current velocity:
[longitudinal v_lon={v_lon:.2f} m/s, lateral lat_v={lat_v:.2f} m/s].
Current acceleration:
[longitudinal a_lon={a_lon:.2f} m/s^2, yaw rate yaw_rate={yaw_rate:.2f} deg/s].
Historical trajectory [x,y,yaw]:
[

```
[{x1:.2f},{y1:.2f},{yaw1:.2f}],
[{x2:.2f},{y2:.2f},{yaw2:.2f}],
[{x3:.2f},{y3:.2f},{yaw3:.2f}],
[{x4:.2f},{y4:.2f},{yaw4:.2f}]
]
```

Using the driving scene, ego state, and history, predict the future trajectory.
Output the complete 8-point future trajectory in chronological order ($t + 0.5$s, $t + 1.0$s, ..., $t + 4.0$s) inside `<trajectory></trajectory>`.
Each future point must be [x,y,yaw] in the current ego frame: x is longitudinal position in meters with positive direction forward; y is lateral position in meters with positive direction left; yaw is heading in radians.
If the navigation command conflicts with the actual road structure in the image, follow the road structure. Keep numerical precision to two decimal places. Directly predict the future trajectory without chain-of-thought. Do not produce any text outside the `<trajectory></trajectory>` block.

Assistant

```
<trajectory>
[[x,y,yaw],...,[x,y,yaw]]
</trajectory>
```

E Training Details

We use Qwen3-VL-4B for all three training stages: SFT, RL, and OPD. The main training settings are summarized in Tab. 8. The vision encoder, multimodal merger, and language model are jointly optimized with full-parameter training in all stages. SFT is trained on 16 GPUs for 3 epochs with a learning rate of 2×10^{-5} and a global batch size of 128. We use a cosine learning-rate schedule with a warmup ratio of 0.03.

For RL, both the safety and progress experts are initialized from the SFT checkpoint and trained under the same optimization setting. We use 32 GPUs for 3 epochs with a learning rate of 1×10^{-6}. Each optimization step samples 128 prompts and generates $K = 8$ rollouts per prompt, resulting in 1,024 completions per step. The per-device batch size is 1 and the gradient accumulation factor is 32. We use a constant learning rate without warmup, with a KL coefficient of 0.01 and a clipping range of $[0.2, 0.3]$.

For OPD, the progress expert is used to initialize the student. The student is trained for 2 epochs with a learning rate of 2×10^{-6} and a global batch size of 64. The OPD stage uses 16 GPUs in total. Eight GPUs on one node are used to train the student, while the other eight GPUs serve the two frozen teachers, with four GPUs assigned to each teacher to return the teacher distributions required for distillation. We use one on-policy sample per prompt and optimize the JSD-based distillation objective with $\beta = 0.5$.

F Evaluation Metrics

NAVSIM v2 (Cao et al., 2025) extends PDMS to the Extended Predictive Driver Model Score (EPDMS) by incorporating additional safety, rule-compliance, and comfort criteria. In addition to no-at-fault collision (NC) and drivable-area compliance (DAC), EPDMS introduces driving-direction compliance (DDC) and traffic-light compliance (TLC) as multiplicative terms. Its weighted terms include time-to-collision (TTC), ego progress (EP), lane keeping (LK), history comfort (HC), and extended comfort (EC), with weights 5, 5, 2, 2, and 2, respectively. The overall score is computed as

$$\text{EPDMS} = \prod_{m \in \{\text{NC,DAC,DDC,TLC}\}} \tilde{m} \cdot \frac{5\tilde{\text{TTC}} + 5\tilde{\text{EP}} + 2\tilde{\text{LK}} + 2\tilde{\text{HC}} + 2\tilde{\text{EC}}}{16}, \tag{18}$$

Table 8: **Training details for SFT, RL, and OPD.**

Stage	Epochs	Learning rate	Global batch	GPUs
SFT	3	2×10^{-5}	128	16
RL	3	1×10^{-6}	128 prompts $\times$ 8	32
OPD	2	2×10^{-6}	64	16

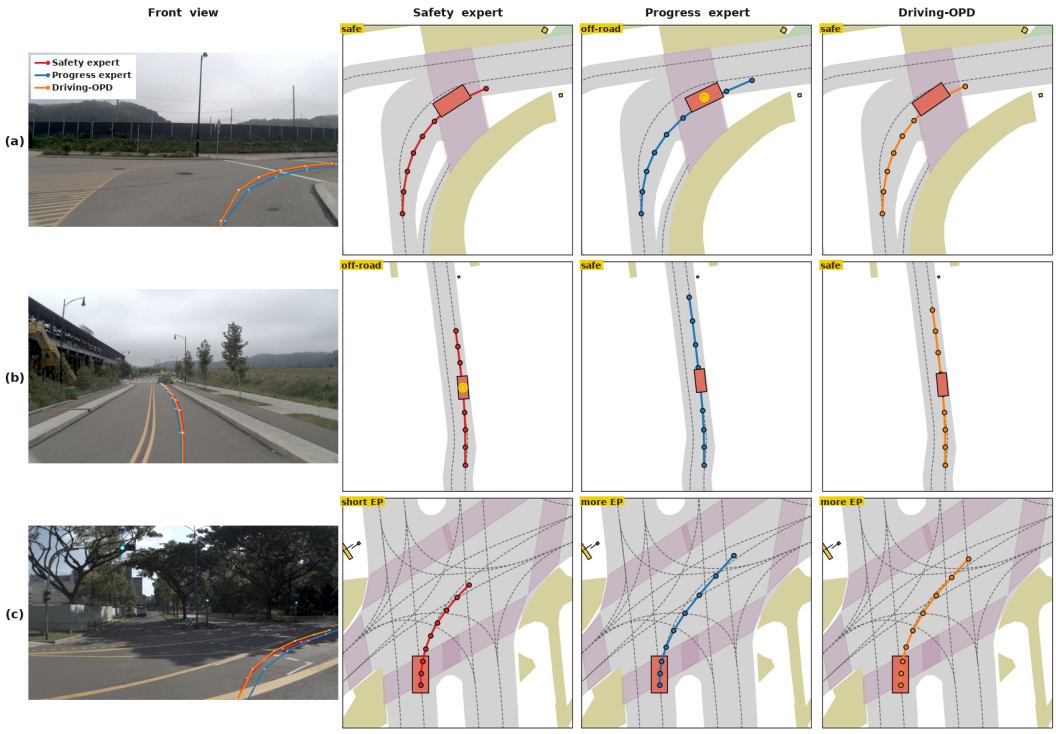

Figure 6: **Successful cases of Driving-OPD(1).**

where each metric is filtered according to the corresponding human trajectory:

$$\tilde{m} = \begin{cases} 1, & m(\text{human}) = 0, \\ m(\text{agent}), & \text{otherwise.} \end{cases} \tag{19}$$

This filtering avoids penalizing the planner for violations that are also present in the corresponding human trajectory.

G Additional Qualitative Results

We provide additional qualitative results of Driving-OPD. Figs. 6 and 7 show successful cases, while Fig. 8 shows failure cases.

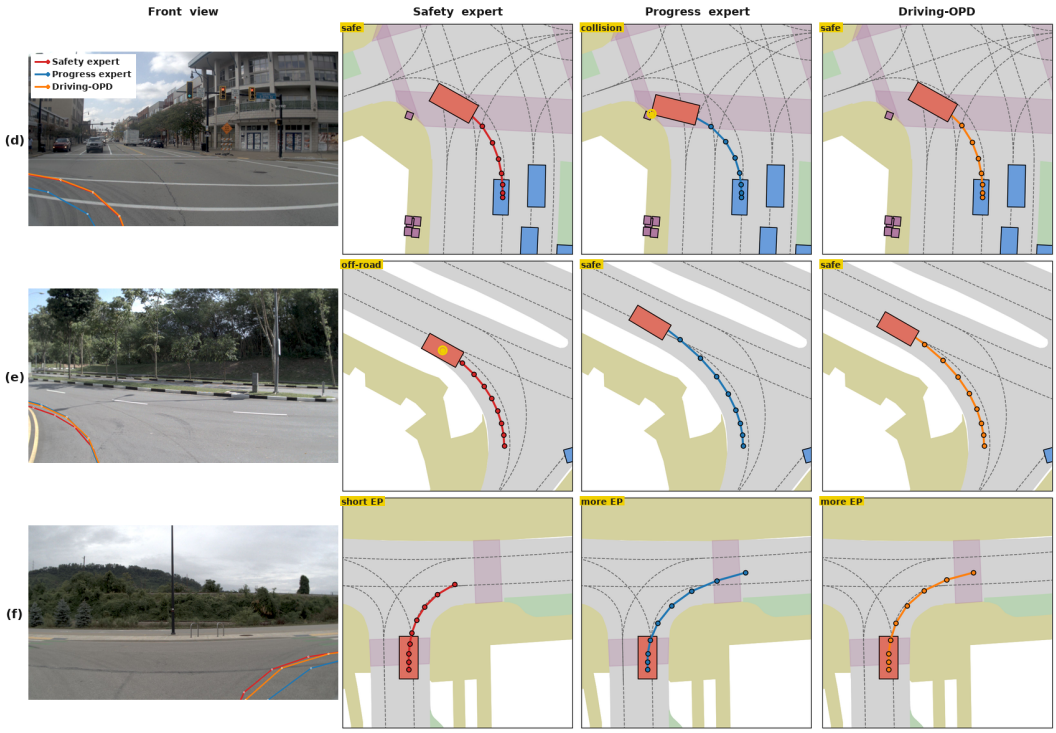

Figure 7: **Successful cases of Driving-OPD(2).**



Figure 8: **Failure cases of Driving-OPD.**