# VARIODRIVE: POST-TRAINING VISION-LANGUAGE MODELS FOR ADAPTIVE AND DIVERSE REASONING IN AUTONOMOUS DRIVING

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

## ABSTRACT

Vision-language-action (VLA) models have shown a promising effectiveness in autonomous driving planning. Existing approaches, however, primarily optimize the final trajectory, providing limited feedback on whether the intermediate reasoning is grounded in the scene and actually supports the plan. Moreover, they typically commit to a single reasoning path per scene, leaving alternative behaviors with different safety, efficiency, and comfort trade-offs unexplored. To address these issues, in this paper, we propose **VarioDrive**, a planning-oriented post-training framework that aligns structured reasoning with trajectory generation and enables adaptive, preference-conditioned planning within a single autoregressive model. Specifically, we first introduce *a time-indexed waypoint tokenization* that factorizes each future pose into time-indexed position and heading tokens, together with a geometry-aware soft supervision that makes token prediction aware of spatial distance, so that trajectories can be directly decoded without an external planner. Then, we organize each response into a scene description, driving-intent reasoning, and a planning answer, and design *stage-decomposed rewards* that separately evaluate visual grounding, intent-trajectory alignment, and preference-dependent trajectory quality during RL post-training. We further develop an *adaptive fast-slow inference* mechanism that generates a single plan for simple scenes while generating and comparing safety-, efficiency-, and comfort-oriented plans for complex scenes before selecting the final trajectory. Experiments on NAVSIM, nuScenes, and Bench2Drive demonstrate a competitive planning performance. On NAVSIM, stage-decomposed rewards improve PDMS by 2.09 points over answer-only RL, and the resulting model reaches 92.42 PDMS with multi-preference selection.

## 1 INTRODUCTION

End-to-end autonomous driving has progressed from modular perception–prediction–planning pipelines toward policies that map sensor observations directly to driving actions (Chen et al., 2024; Chitta et al., 2022; Hu et al., 2023; Jiang et al., 2023; Zheng et al., 2024). Vision-language models extend this approach by bringing scene understanding and language reasoning to driving decisions (Hwang et al., 2024; Renz et al., 2025; Zhou et al., 2025; Fu et al., 2025a; Jiang et al., 2024). Recent approaches use supervised fine-tuning and reinforcement learning (RL) to adapt these models for reasoning and trajectory generation (Wang et al., 2024a; Ma et al., 2024; Wang et al., 2025; Zhou et al., 2025).

However, jointly learning reasoning and trajectory generation does not by itself ensure that reasoning effectively supports planning. Planning-oriented post-training of autoregressive vision-language models involves three closely related considerations. *i) Trajectory representation:* driving requires metric trajectories, whereas high-level language commands do not directly specify waypoint coordinates, and clustered action codes do not explicitly encode coordinate neighborhoods (Jiang et al., 2024; Seff et al., 2023; Zhou et al., 2025). Separate planning modules produce continuous trajectories (Liao et al., 2025; Jiang et al., 2025a), but require an additional reasoning-to-motion interface. *ii) Intermediate reasoning feedback:* outcome-only rewards do not directly assess the correctness of intermediate reasoning. Separate feedback can evaluate whether intermediate outputs are grounded and support the generated plan. *iii) Limited preference exploration:* a scene can admit several plausible trajectories



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.

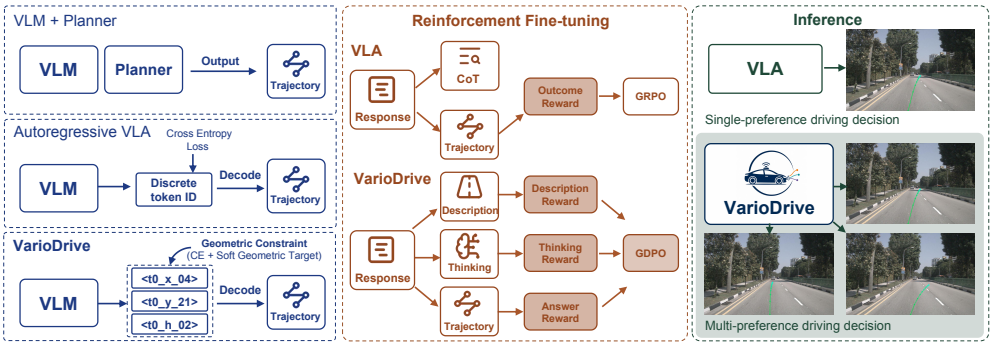



Figure 1: Representative approaches to VLM-based autonomous driving: reasoning coupled to a planning module, autoregressive action-token generation, and VarioDrive's geometric waypoint tokens with structured reward feedback and multi-preference planning.

with different safety, efficiency, and comfort trade-offs. A single reasoning path leaves alternatives unexplored, while same-preference sampling may offer limited behavioral diversity.

We propose **VarioDrive**, a planning-oriented post-training framework for structured reasoning and adaptive, preference-conditioned planning within a single autoregressive model. We organize each response into scene description, driving-intent reasoning, and a planning answer. We use RL post-training with dedicated rewards for visual grounding, intent–trajectory alignment, and preference-dependent trajectory quality, jointly encouraging grounded reasoning and effective planning. The reward dimensions are normalized separately and combined for completion-level policy optimization. Specifically, the ego-centric waypoint tokens encode longitudinal position, lateral position, and heading at each future timestep and decode directly into metric trajectories without an external planner. The geometry-aware soft supervision during supervised fine-tuning encourages spatially accurate waypoint prediction. The preference-dependent trajectory rewards encourage different planning trade-offs during RL. Within the same model, the adaptive fast-slow inference generates a single plan in simple scenes and compares plans prioritizing safety, efficiency, or comfort in complex scenes. Experiments on the NAVSIM, nuScenes, and Bench2Drive benchmarks demonstrate a competitive planning performance. On NAVSIM, stage-decomposed rewards improve PDMS by 2.09 points over answer-only RL, and VarioDrive reaches 92.42 PDMS with multi-preference selection. Ablation studies demonstrate the benefits of geometric supervision and show that multi-preference generation provides better planning candidates at a fixed candidate count. Our contributions are summarized as follows.

- We propose a time-indexed waypoint tokenization that factorizes each future pose into position and heading tokens, together with a geometry-aware soft supervision, which enables spatially accurate trajectory generation within a single autoregressive model without an external planner.
- We design stage-decomposed rewards for RL post-training that separately evaluate scene grounding, intent-trajectory alignment, and preference-dependent trajectory quality, enabling explicit feedback on intermediate reasoning and supporting for the generated plan.
- We develop an adaptive fast-slow inference mechanism that balances the planning quality and computational cost through scene-dependent routing, preference-conditioned generation, and selection within the same model.

## 2 Background and Related Work

Vision-language models connect visual scene understanding and language reasoning to autonomous driving (Sima et al., 2024; Tian et al., 2024; Wang et al., 2024a; Ma et al., 2024). EMMA represents driving inputs and outputs in a unified language space (Hwang et al., 2024), while SimLingo aligns language understanding with closed-loop driving (Renz et al., 2025). We review trajectory representations, post-training methods, and reasoning strategies relevant to this direction.

**Trajectory Representation for Autoregressive Planning.** Driving models connect language outputs to motion through different interfaces (Figure 1). Senna uses language-based planning guidance, while ORION couples reasoning to a generative trajectory planner (Jiang et al., 2024; Fu et al., 2025a). Autoregressive action representations instead allow a model to generate actions through its token interface, as explored in robotics and motion forecasting (Brohan et al., 2022; Zitkovich et al., 2023; Kim et al., 2024; Seff et al., 2023). AutoVLA combines reasoning with discretized physical action tokens in a shared autoregressive sequence (Zhou et al., 2025). VarioDrive follows the autoregressive approach and factorizes each future waypoint into time-indexed position and heading tokens. Geometry-aware supervision accounts for distances between their bin representatives, and the output sequence decodes directly into a metric trajectory.

**Post-Training for Autonomous Driving.** Supervised fine-tuning adapts pretrained models to driving tasks using demonstrations and language annotations (Wang et al., 2024a; Jiang et al., 2024; Zhou et al., 2025). RL further optimizes their outputs using task-specific feedback. AlphaDrive combines supervised fine-tuning with GRPO-based planning rewards (Jiang et al., 2025b), while AutoVLA optimizes planning performance and penalizes unnecessary reasoning (Zhou et al., 2025). AutoDrive-R2 incorporates physics-aware rewards, and MindDrive uses trajectory-level feedback to optimize driving decisions (Yuan et al., 2025; Fu et al., 2025b). Alpamayo-R1 evaluates reasoning quality, reasoning–action consistency, and trajectory quality during RL (Wang et al., 2025). These studies motivate feedback that connects language reasoning to driving outcomes. VarioDrive contributes a post-training framework that combines feedback on scene grounding, intent–trajectory alignment, and preference-dependent trajectory quality with geometric trajectory supervision.

**Structured Reasoning and Adaptive Inference.** Chain-of-thought reasoning expresses decisions through intermediate steps (Wei et al., 2022). In autonomous driving, DriveLM uses graph-structured visual question answering (Sima et al., 2024), while DriveVLM organizes reasoning into scene description, analysis, and planning (Tian et al., 2024). DriveCoT and Reason2Drive also study structured reasoning for driving scenes and decisions (Wang et al., 2024b; Nie et al., 2024). Beyond reasoning structure, AutoVLA adapts between direct trajectory generation and CoT-based planning to control inference cost (Zhou et al., 2025). VarioDrive uses structured reasoning in both modes: Fast inference generates one plan, whereas Slow inference explores and compares plans with different safety, efficiency, and comfort priorities. Additional computation therefore supports alternative driving behaviors through preference-conditioned reasoning and planning.

## 3 Main Framework

As illustrated in Figure 2, VarioDrive builds on Qwen3-VL-4B (Bai et al., 2025) to generate driving reasoning and trajectory tokens with a shared autoregressive decoder. The driving context $m = (\mathcal{I}, \mathcal{S}, \mathcal{N})$ comprises multi-view camera observations, the ego state, and a navigation command. A response $y = (y^{\text{desc}}, y^{\text{thk}}, y^{\text{ans}})$ contains a scene description, driving-intent reasoning, and a planning answer. The answer decodes into ego-centric positions and headings. A learned Fast/Slow decision determines whether to produce a single plan or compare several preference-conditioned plans.

### 3.1 Geometric Trajectory Representation with Waypoint Tokenizer

We represent an expert trajectory as future ego poses $\tau^{\text{gt}} = ((x_t, y_t, h_t))_{t=0}^{T-1}$, where $x_t$ and $y_t$ are longitudinal and lateral positions and $h_t$ is heading relative to the current ego orientation. The main configuration predicts $T = 8$ poses at $0.5$ s intervals over a $4$ s horizon. Each pose is factorized into three tokens $(r_t^x, r_t^y, r_t^h)$, identifying its future step, component, and discretized value. The expert answer is

$$a = \phi(\tau^{\text{gt}}) = [r_0^x, r_0^y, r_0^h, \ldots, r_{T-1}^x, r_{T-1}^y, r_{T-1}^h], \qquad |a| = 3T. \tag{1}$$

For example, `<t0_x_01><t0_y_04><t0_h_017>` represents the pose at $0.5$ s. Time is part of each token's identity, not an additional token. Each step has its own component vocabularies, sized according to empirical position and heading ranges; this avoids allocating the widest range to every step. Decoding replaces the generated bin indices with metric positions and angular representatives. The resulting interface preserves explicit temporal and geometric structure within standard next-token prediction.

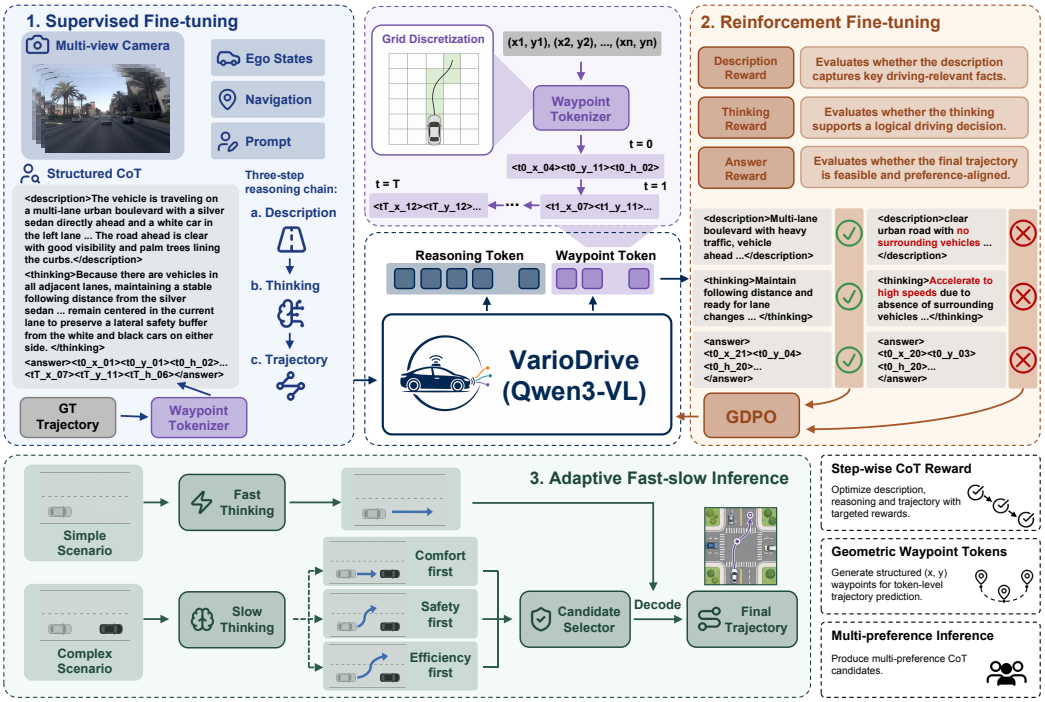



Figure 2: Overview of VarioDrive. Training combines geometry-aware SFT with stage-decomposed RL rewards. Fast inference produces a single plan, while Slow inference generates diverse reasoning and candidate plans under three driving preferences before selecting the final trajectory.

### 3.2 Adaptive Fast-and-Slow Reasoning

VarioDrive learns a Fast/Slow decision during SFT and predicts it from the current driving context. Fast mode generates one complete description–thinking–answer response. Slow mode generates three responses under preferences $f \in \{\text{efficiency}, \text{safety}, \text{comfort}\}$. These preferences direct reasoning toward different trade-offs and produce alternative trajectory proposals.

The model then compares the candidates using the driving context, their reasoning, and their waypoint sequences, and returns the chosen trajectory with a short comparison rationale. Slow mode thus spends additional computation on exploring diverse reasoning and plans before committing to one. Adaptive routing restricts this broader exploration to scenes assigned to Slow mode and uses single-plan generation elsewhere.

## 4 Training

Supervised fine-tuning (SFT) teaches the pose-token representation and structured reasoning format using expert trajectories and synthesized language annotations. Reinforcement fine-tuning (RFT) then improves scene grounding, the usefulness of reasoning for planning, and preference-conditioned planning.

### 4.1 Training Sample Construction for CoT Reasoning and Waypoint Generation

We pair expert trajectories with scene descriptions and driving rationales generated by Qwen3.5-Plus (Alibaba Cloud, 2026). Each training target has three fields, $(c^{\text{desc}}, c^{\text{thk}}, c^{\text{ans}})$, separating scene observations, driving intent, and the final plan. This follows structured reasoning supervision (Wei et al., 2022; Lightman et al., 2023; Wang et al., 2024b; Nie et al., 2024) and makes each field available for a distinct planning-oriented assessment.

*Scene description.* The annotation model generates $c^{\text{desc}}$ from synchronized camera views, summarizing visible road layout, traffic lights, agents, and obstacles relevant to the driving decision.

*Driving-intent reasoning.* Given the description, navigation, ego state, and the expert's demonstrated maneuver, the annotation model explains the driving decision under a specified preference $f \in \{\text{efficiency}, \text{safety}, \text{comfort}\}$. The three preferences emphasize progress, risk avoidance, and smooth motion, respectively, and are balanced during annotation. The demonstrated maneuver is conveyed through a motion summary derived from the expert trajectory.

*Planning answer.* We deterministically encode the expert trajectory as $c^{\text{ans}} = \phi(\tau^{\text{gt}}) = a$, using the $3T$ pose-component tokens in Equation 1.During SFT, the preference changes only the rationale for the same expert target; preference-dependent trajectory trade-offs are learned during RFT.

### 4.2 Training Objective of Supervised Fine-Tuning

Let $z = (z_1, \ldots, z_M)$ be the complete target response. It contains $3T$ pose-component tokens and $L$ language and structural tokens, so $M = L + 3T$. The standard hard-label next-token loss is:

$$\mathcal{L}_{\text{LM}} = -\frac{1}{M} \sum_{j=1}^{M} \log \pi_\theta(z_j \mid m, f, z_{<j}). \tag{2}$$

Hard cross-entropy penalizes all incorrect bins equally, regardless of their geometric distance to the target. We therefore add soft supervision for each future step $t$ and pose component $b \in \{x, y, h\}$. The target distribution $q_{t,b}$ assigns probability to bins near the continuous expert value, and $p_{t,b}$ is the model distribution normalized over the vocabulary for that same step and component. The objective is

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{LM}} + \frac{\lambda_{\text{soft}}}{T} \sum_{t=0}^{T-1} \sum_{b \in \{x,y,h\}} \omega_b D_{\text{KL}}(q_{t,b} \| p_{t,b}). \tag{3}$$

The component weights $\omega_b$ balance position and heading supervision. The auxiliary term encodes geometric neighborhoods, whereas the hard cross-entropy supervises exact tokens and the response structure. Appendix A.8 specifies the binning, soft targets, and component scales.

### 4.3 Stage-Decomposed CoT Rewards

The RL rewards evaluate scene grounding, the alignment between driving-intent reasoning and the generated plan, and trajectory quality. Let $g_{\text{ans}} \in \{0, 1\}$ indicate whether a decoded trajectory passes the shared answer-level checks defined below. The description and reasoning rewards are granted only when this gate is satisfied.

**Scene description reward.** We adopt a VLM-as-a-judge protocol to assess free-form descriptions against the camera observations, since exact text matching cannot capture visual-semantic agreement. The rubric assigns $S_{\text{desc}} = 0$, 1, or 2 to descriptions that are severely inaccurate or incomplete, partially correct, or accurate and sufficiently complete, respectively. The reward is $R_{\text{desc}} = g_{\text{ans}} S_{\text{desc}}$; Appendix A.1.1 gives the rubric and prompt.

**Driving-intent reasoning reward.** Let $\hat{a}$ contain the $3T$ pose-component tokens extracted from the generated answer $y^{\text{ans}}$. We score this fixed sequence with and without the preceding thinking segment. Holding the policy parameters $\bar{\theta}$ fixed during reward evaluation, we compute

$$\ell_+ = \frac{1}{3T} \sum_{u=1}^{3T} \log \pi_{\bar{\theta}}(\hat{a}_u \mid m, f, y^{\text{thk}}, \hat{a}_{<u}),$$
$$\ell_0 = \frac{1}{3T} \sum_{u=1}^{3T} \log \pi_{\bar{\theta}}(\hat{a}_u \mid m, f, \hat{a}_{<u}), \qquad \Delta_{\text{PPL}} = \frac{e^{-\ell_0} - e^{-\ell_+}}{e^{-\ell_0}}. \tag{4}$$

The two passes differ only in the thinking context and score the same generated answer. A positive relative perplexity reduction indicates that the reasoning increases the likelihood of the generated plan. We assign

$$R_{\text{thk}} = g_{\text{ans}} \left( \mathbb{I}[\Delta_{\text{PPL}} \geq \eta_r] + \tfrac{1}{2} \mathbb{I}[0 < \Delta_{\text{PPL}} < \eta_r] \right), \qquad 0 < \eta_r < 1. \tag{5}$$

This provides a likelihood-based alignment signal. The scoring contexts and threshold are detailed in Appendix A.1.2.

**Planning answer reward.** All preferences share the same safety and regularity checks. Let $c_{\text{safe}}$ be the product of the no-at-fault-collision and drivable-area-compliance scores. We define

$$R^f_{\text{ans}} = g_{\text{ans}}\, c_{\text{safe}}\, S_f(\widehat{\tau}, \tau^{\text{gt}}), \qquad \widehat{\tau} = \text{Decode}(y^{\text{ans}}). \tag{6}$$

The gate $g_{\text{ans}}$ requires valid decoding, a sufficiently large $c_{\text{safe}}$, and bounded geometric irregularity. Given this common gate, the preference score $S_f \in [0, 1]$ rewards net progress for efficiency, lateral adherence to the reference path for safety, and smooth, steady motion for comfort. Appendix A.1.3 defines these scores. Preference $f$ is fixed within each rollout group. An auxiliary format reward $R_{\text{form}}$ checks the required fields and waypoint-token structure.

### 4.4 GDPO-Based Policy Optimization

We adopt the reward normalization of Group Reward-Decoupled Normalization Policy Optimization (GDPO) (Liu et al., 2026). For each input-preference pair $(m_i, f_i)$, we sample $G$ completions $\{y_{i,g}\}_{g=1}^{G}$ from $\pi_{\text{old}}$. Each reward dimension $k \in \{\text{desc}, \text{thk}, \text{ans}, \text{form}\}$ is normalized before aggregation:

$$\widetilde{A}^k_{i,g} = \frac{R^k_{i,g} - \mu^k_i}{\sigma^k_i + \epsilon_n}, \qquad \bar{A}_{i,g} = \sum_k w_k \widetilde{A}^k_{i,g}, \qquad A_{i,g} = \frac{\bar{A}_{i,g} - \mu_{\mathcal{B}}}{\sigma_{\mathcal{B}} + \epsilon_n}. \tag{7}$$

The group statistics $\mu^k_i, \sigma^k_i$ use all $G$ completions; $\mu_{\mathcal{B}}, \sigma_{\mathcal{B}}$ are batch statistics after aggregation. The weights $w_k$ balance the normalized reward dimensions. A reward dimension that is constant within a group contributes zero advantage.

For each group, we retain at most two completions with the largest positive final advantages $A_{i,g}$ and apply the clipped policy update to their response tokens. All $G$ completions still determine the normalization statistics. A group with no positive advantage contributes no update. This concentrates optimization on the highest-ranked responses. A positive advantage indicates relative, not absolute, quality. The loss is averaged over selected response tokens with KL regularization toward the SFT reference policy. Rewards are stage-decomposed, whereas advantages and selection are completion-level. Appendix A.1.4 gives the objective and selection mask.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets and metrics.** We train on nuPlan and nuScenes (Karnchanachari et al., 2024; Caesar et al., 2020). Waypoint-only SFT uses 200k image–trajectory pairs (170k nuPlan and 30k nuScenes); structured-CoT SFT uses 30k samples; RFT uses 15k prompts. The same driving scenes may be reused across stages; these counts are not added to obtain a unique-scene total. We predict 8 waypoints over 4 s at 2 Hz. NAVSIM evaluates non-reactive planning through predictive driver model score (PDMS) and its component metrics (Dauner et al., 2024). We additionally evaluate interactive closed-loop driving on Bench2Drive (Jia et al., 2024), and meta-action accuracy and per-class F1 on nuScenes. Displacement errors are in meters. Teacher-forced (TF) ADE/FDE diagnose token prediction using ground-truth prefixes. Autoregressive (AR) ADE/FDE assess complete trajectories generated from the model's own prefixes and include accumulated generation errors; the two protocols are not directly comparable.

**Implementation details.** We use Qwen3-VL-4B, with three epochs of waypoint-only SFT followed by two epochs of structured-CoT SFT. The coefficient ratio of hard next-token cross-entropy to auxiliary geometric soft-target loss is 10:1. RFT updates the full model for one epoch with $G = 8$ completions per group. All training stages use eight NVIDIA H800 GPUs; runtime is measured on one H800 at batch size 8. Appendix A.11 provides the remaining settings.

Table 1: NAVSIM planning results.

| Method | PDMS | NC | DAC | EP | TTC | Comfort |
|---|---|---|---|---|---|---|
| Ego Status MLP | 66.40 | 93.09 | 78.26 | 63.20 | 84.02 | 99.97 |
| TransFuser (Chitta et al., 2022) | 83.88 | 97.78 | 92.63 | 78.88 | 92.89 | 99.98 |
| DRAMA (Yuan et al., 2024) | 86.87 | 98.19 | 95.18 | 81.33 | 94.17 | 100.00 |
| Hydra-MDP (Li et al., 2024) | 91.26 | 99.07 | 98.29 | 85.20 | 96.56 | 100.00 |
| Centaur (Sima et al., 2025) | 92.10 | 99.23 | 98.72 | 85.96 | 97.17 | 99.97 |
| AutoVLA* (Zhou et al., 2025) | 92.12 | 99.14 | 97.08 | 87.55 | 97.12 | 99.98 |
| VarioDrive (SFT) | 83.68 | 97.06 | 92.71 | 76.42 | 92.10 | 99.87 |
| VarioDrive (Fast) | 90.84 | 98.34 | 97.43 | 84.85 | 95.07 | 100.00 |
| VarioDrive (Slow) | 92.42 | 99.26 | 98.21 | 86.73 | 97.95 | 100.00 |

* AutoVLA uses oracle best-of-6 selection. Both methods generate reasoning and tokenized driving actions within a single autoregressive model. NC: no-at-fault collision; DAC: drivable-area compliance; EP: ego progress. TransFuser, DRAMA, Hydra-MDP, and Centaur use camera–LiDAR inputs, whereas VarioDrive and AutoVLA use camera-only inputs.

### 5.2 Main Results

**NAVSIM planning.** Table 1 retains all available planning components. RFT improves PDMS from 83.68 to 90.84, and Slow inference further increases it to 92.42 with the same checkpoint. These results demonstrate the benefits of RL post-training and multi-preference planning.

Table 2: Meta-action planning on nuScenes using a common Qwen3-VL backbone. Acc. is overall accuracy; the seven action columns report per-class F1 (%). The Qwen3-VL baseline uses SFT only; AlphaDrive denotes its training recipe implemented on this backbone.

| Method | Acc. | Straight | Left | Right | Keep | Accel. | Decel. | Stop |
|---|---|---|---|---|---|---|---|---|
| Qwen3-VL (SFT) | 75.16 | 94.82 | 52.58 | 57.68 | 90.13 | 54.54 | 51.10 | 77.31 |
| AlphaDrive | 78.50 | 94.26 | 90.54 | 93.25 | 91.88 | 56.33 | 61.40 | **86.76** |
| VarioDrive | **87.14** | **96.72** | **92.50** | **94.29** | **94.52** | **64.00** | **73.49** | 82.29 |

**Meta-action planning.** We implement the SFT baseline and the AlphaDrive training recipe on the same Qwen3-VL backbone used by VarioDrive. Expert and generated trajectories are mapped to lateral and longitudinal action labels using a common extraction procedure. Table 2 shows that VarioDrive reaches 87.14% accuracy and the highest F1 in most lateral and longitudinal categories. AlphaDrive (Jiang et al., 2025b) retains higher stop F1. This assesses maneuver-level decisions alongside trajectory evaluation.

Table 3: Closed-loop results on Bench2Drive.

| Method | DS | SR | Eff. | Comf. |
|---|---|---|---|---|
| ORION | 77.74 | 54.62 | 151.48 | 17.38 |
| AutoVLA | 78.84 | **57.73** | 146.93 | **39.33** |
| VarioDrive | **79.70** | 55.91 | **155.27** | 19.93 |

DS: Driving Score; SR: Success Rate (%); Eff.: Efficiency; Comf.: Comfortness. ORION (Fu et al., 2025a) and AutoVLA (Zhou et al., 2025) report published results.

Table 4: Adaptive inference on navtest.

| Policy | Slow (%) | PDMS ↑ | Time (s) ↓ |
|---|---|---|---|
| Always Fast | 0.00 | 90.84 | 2.870 |
| Adaptive | 56.22 | 92.13 | 5.300 |
| Always Slow | 100.00 | 92.42 | 7.149 |

**Closed-loop driving.** On Bench2Drive (Table 3), VarioDrive achieves Driving Score 79.70 and Efficiency 155.27, the highest among the three compared methods. AutoVLA has higher Success Rate and Comfortness, indicating room to improve route completion and ride comfort. The external baseline results are published measurements rather than reruns under a common training configuration.

**Efficiency of adaptive fast–slow reasoning.** Adaptive routing invokes Slow mode on 56.22% of navtest scenes and reaches 92.13 PDMS at 5.30 s (Table 4). Relative to Always Slow, it reduces

measured latency by 25.9% while retaining 81.6% of the Fast-to-Slow PDMS gain. The complete-output timings include description, thinking, waypoints, and selection when invoked. The original 0.781 s/3.936 s Fast/Slow timings exclude description and thinking; their ranges are retained in Appendix D. These batch-size-8 measurements characterize the measured quality–latency trade-off; they do not establish single-scene vehicle-deployment latency.

### 5.3 Ablation Studies

Table 5: Waypoint representation. Top: text coordinates versus waypoint tokens after CoT SFT. Bottom: position-and-heading tokenization variants trained with the same waypoint-only SFT budget. The last two rows differ only in whether geometric supervision is used. Teacher-forced (TF) errors are measured in meters using ground-truth token prefixes. See Appendix A.10 for details.

| Representation / setting | PDMS ↑ | TF-ADE ↓ | TF-FDE ↓ |
|---|---|---|---|
| Text coordinates (CoT SFT) | 68.24 | 0.74 | 1.71 |
| Waypoint tokens (CoT SFT) | 83.68 | 0.15 | 0.39 |
| Joint XY + heading ($2T$) | 66.92 | 0.32 | 0.65 |
| Shared X/Y/heading ($3T$) | 67.42 | 0.31 | 0.936 |
| Time-indexed components, no soft loss | 71.93 | 0.24 | 0.57 |
| Time-indexed components + soft loss | **80.96** | **0.17** | **0.43** |

**Waypoint-tokenized planning.** The upper block of Table 5 compares text coordinates with waypoint tokens under structured-CoT SFT: waypoint tokens raise PDMS from 68.24 to 83.68. The lower block compares heading-inclusive tokenization variants under a common backbone, dataset, and waypoint-only SFT budget (details in Appendix A.10). The last two rows share the same representation and differ only in the soft loss: geometry-aware soft supervision raises PDMS from 71.93 to 80.96 and reduces both TF displacement errors.




Figure 3: Training comparison between GDPO and GRPO.

**Effect of GDPO.** We adopt GDPO to normalize heterogeneous reward signals before aggregation (Liu et al., 2026). With the RFT data and setup fixed, the curves show earlier improvement in final trajectory reward, while both methods approach a similar plateau.

**Stage-decomposed reward design.** We ablate the description and thinking rewards on nuScenes, applying the same meta-action extractor to expert and free-running trajectories (Table 6). Accuracy and F1 measure lateral and longitudinal agreement without requiring pointwise coordinate matching.

With answer and format rewards fixed, adding description reward improves left/right F1 from 60.59/69.68 to 91.58/93.14. Separately adding thinking reward raises accuracy to 85.92%; using both reaches 87.14%. The two middle rows are alternative reward settings. The NAVSIM control reaches 88.75 PDMS with answer-only RFT and 90.84 with full stage-reward RFT, a 2.09-point gain (Table 10).

Table 6: Intermediate-reward ablation on nuScenes. Answer and format rewards remain enabled; the middle rows separately add description or thinking reward. Acc. is meta-action accuracy; action columns report per-class F1 (%).

| Reward setting | Acc. | Straight | Left | Right | Keep | Accel. | Decel. | Stop |
|---|---|---|---|---|---|---|---|---|
| Answer + format | 80.95 | 96.34 | 60.59 | 69.68 | 93.24 | 50.65 | 54.12 | 71.26 |
| + description | 82.20 | 95.24 | 91.58 | 93.14 | 93.83 | 58.02 | 67.10 | 81.23 |
| + thinking | 85.92 | 96.04 | 92.05 | 92.57 | **94.74** | 55.29 | 64.00 | 79.58 |
| + both | **87.14** | **96.72** | **92.50** | **94.29** | 94.52 | **64.00** | **73.49** | **82.29** |

**Planning relevance of intermediate context.** At a fixed checkpoint, retaining, removing, or replacing the generated description–thinking prefix yields 90.84, 74.40, and 33.38 PDMS, respectively; only the waypoint answer is regenerated. A separate navigation intervention changes the reasoning consistently in 79/80 turn scenes after replacing the command with `go straight`. These tests show context dependence and command sensitivity. Removing a trained prefix also changes the input distribution, while navigation directly conditions planning; neither test alone establishes causal faithfulness (Appendix B.2).

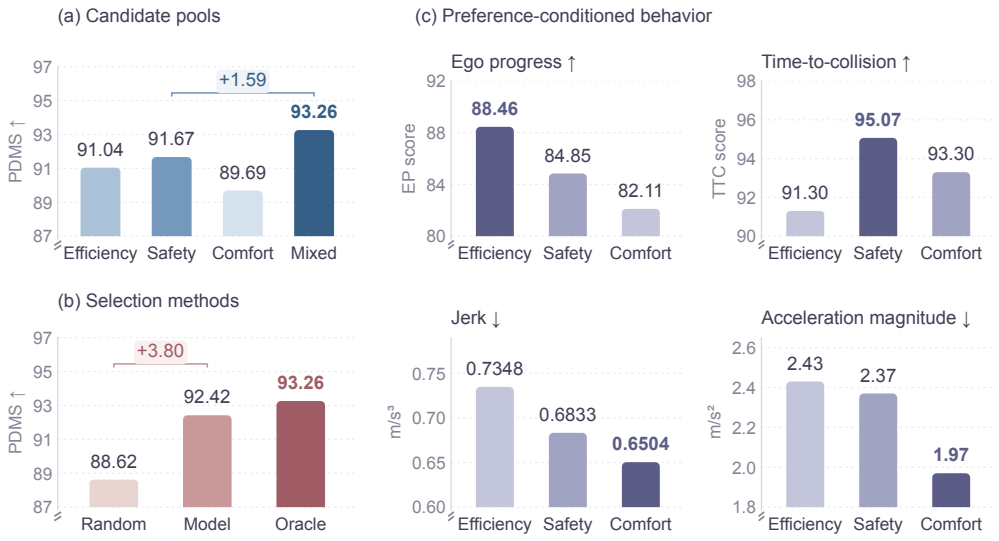



Figure 4: Preference-conditioned planning and candidate selection. (a) Oracle selection from three-candidate pools; Mixed contains one candidate per preference. (b) Selectors on the same mixed pool. (c) Preference-conditioned behavior: single-candidate NAVSIM scores (top) and dynamics of complete trajectories (bottom). Darker bars indicate better results within each chart.

**Preference-conditioned behavior.** Figure 4(c) shows the greatest progress for efficiency, highest TTC for safety, and lowest jerk and acceleration magnitude for comfort. Mean pairwise trajectory distance is 1.08 m; Appendix C gives complete preference diagnostics.

**Candidate generation and selection.** With three candidates per pool, the mixed-preference oracle reaches 93.26 PDMS versus 91.67 for the strongest same-preference oracle (Figure 4(a)). On the same mixed pool, model selection reaches 92.42 versus 88.62 for random selection, leaving a 0.84-point oracle gap (b). Candidate count is matched; model selection adds computation.

## 6 Conclusion

We presented VarioDrive, a planning-oriented post-training framework for vision-language models in autonomous driving. In VarioDrive, geometry-aware waypoint supervision and stage-decomposed rewards connect structured reasoning to trajectory generation, while adaptive fast–slow inference explores preference-conditioned alternatives in complex scenes. Experiments on three benchmarks demonstrate its effectiveness.

**Limitations.** The predefined trajectory grid does not adapt online or enforce the vehicle dynamics. The reward and model-selection errors remain inevitable, and the batch timings do not establish a real-time deployment. Thus, our future work will focus on studying the adaptive discretization and broader closed-loop evaluation under the execution delays and long-tail scenarios.

## AI USE STATEMENT

Generative models provide the structured reasoning annotations, routing and selection supervision, and description-reward judgments as described in the method. ChatGPT also assisted the manuscript revision, including language editing, notation checks, LaTeX preparation, and presentation of existing experimental results. This assistance did not involve new model training or benchmark evaluations. The authors are responsible for verifying the final manuscript and its reported results.

## ETHICS STATEMENT

Autonomous driving is safety-critical, and benchmark performance does not certify safe public-road operation. Generated reasoning and learned candidate selection can fail under unfamiliar or partially observed conditions; practical deployment requires independent safety validation and reliable fallback control. Use and redistribution of the driving datasets, synthetic annotations, and model artifacts must respect their applicable licenses and privacy requirements.

## REPRODUCIBILITY STATEMENT

The main text specifies the trajectory representation, training objectives, inference procedure, and evaluation protocols. The technical appendices provide reward definitions, data-synthesis prompts, and training settings. Appendices B–D give the supplementary experimental results, intervention procedures, and timing scopes. The supplementary LaTeX project includes the tabulated aggregate results and the data used to prepare the additional result visualizations.

## REFERENCES

Alibaba Cloud. Qwen3.5-Plus. Alibaba Cloud Model Studio Documentation, 2026. URL https://www.alibabacloud.com/help/en/model-studio/models. Accessed: 2026-05-07.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

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

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

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. *Advances in Neural Information Processing Systems*, 37:28706–28719, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24823–24834, 2025a.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Hongwei Xie, Bing Wang, Guang Chen, Dingkang Liang, and Xiang Bai. Minddrive: A vision-language-action model for autonomous driving via online reinforcement learning. *arXiv preprint arXiv:2512.13636*, 2025b.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17853–17862, 2023.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: Towards Multi-Ability Benchmarking of Closed-Loop End-To-End Autonomous Driving. *arXiv preprint arXiv:2406.03877*, 2024.

Anqing Jiang, Yu Gao, Zhigang Sun, Yiru Wang, Jijun Wang, Jinhao Chai, Qian Cao, Yuweng Heng, Hao Jiang, Yunda Dong, Zongzheng Zhang, Xianda Guo, Hao Sun, and Hao Zhao. Diffvla: Vision-language guided diffusion planning for autonomous driving. *arXiv preprint arXiv:2505.19381*, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

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.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving. In *European Conference on Computer Vision*, pp. 403–420. Springer, 2024.

Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*, pp. 292–308. Springer, 2024.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11993–12003, 2025.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8579–8590, 2023.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pp. 256–274. Springer, 2024.

Chonghao Sima, Kashyap Chitta, Zhiding Yu, Shiyi Lan, Ping Luo, Andreas Geiger, Hongyang Li, and Jose M Alvarez. Centaur: Robust end-to-end autonomous driving with test-time training. *arXiv preprint arXiv:2503.11650*, 2025.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv preprint arXiv:2405.01533*, 1(2):3, 2024a.

Tianqi Wang, Enze Xie, Ruihang Chu, Zhenguo Li, and Ping Luo. Drivecot: Integrating chain-of-thought reasoning with end-to-end driving. *arXiv preprint arXiv:2403.16996*, 2024b.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024.

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. *arXiv preprint arXiv:2509.01944*, 2025.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 87–104. Springer, 2024.

Zewei Zhou, Tianhui Cai, Seth Z 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. *arXiv preprint arXiv:2506.13757*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

# A Technical Details and Supplementary Material

## A.1 Reward Design

The two process rewards are conditioned on answer-level acceptance. The binary indicator $g_{\text{ans}}$ is distinct from the continuous preference-conditioned answer score: a trajectory that passes the shared checks can receive a nonzero description or reasoning reward even when its preference score is below one. Appendix A.1.3 specifies the checks.

### A.1.1 Scene Description Reward

We use an LLM-as-Judge protocol with a vision-language model, providing camera observations and the generated description. This permits direct assessment of free-form scene claims against visual evidence, which cannot be reduced to exact text matching. The judge assigns $S_{\text{desc}} \in \{0, 1, 2\}$ for severely inaccurate or incomplete, substantially correct but imperfect, and accurate and sufficiently complete descriptions, respectively. The rubric focuses on road layout, traffic controls, nearby agents, obstacles, and visibility. We apply the answer gate to this raw rubric score:

$$R_{\text{desc}} = g_{\text{ans}}\, S_{\text{desc}}. \tag{8}$$

**Description-judge prompt.**

```
You are an expert evaluator for autonomous driving scene understanding.
You will be given driving camera observations and a predicted scene
description (PRED).
Score how well PRED is grounded in the observations, focusing on factual
correctness and completeness of driving-critical evidence such as road
layout, traffic lights, nearby vehicles, pedestrians, obstacles,
weather, and illumination.

# Scoring rubric (0-2):
2 Complete and Accurate: PRED is basically complete and accurately
describes the visible driving scene. It covers the critical objects and
conditions needed for planning, with no meaningful hallucinations.

1 Partially Correct but Imperfect: PRED captures the main scene context,
but contains some hallucinations, factual inaccuracies, or incomplete
descriptions of driving-critical details.

0 Severely Hallucinated or Incomplete: PRED is severely hallucinated,
largely inconsistent with the observations, or too incomplete to support
safe driving planning.

# PRED:
{description}

# Output Format:
Return only the integer score (0, 1, or 2). Do not include any
explanation, reasoning, or additional text.
```

### A.1.2 Intent Reasoning Reward

Let $\hat{a} = (\hat{a}_1, \ldots, \hat{a}_{3T})$ be the trajectory tokens in the model's generated answer, covering longitudinal position, lateral position, and heading. We score this fixed sequence twice, using the same policy parameters $\bar{\theta}$ without gradients. The two contexts contain the same driving input and preference; the first additionally contains the generated thinking segment:

$$\ell_+ = \frac{1}{3T} \sum_{u=1}^{3T} \log \pi_{\bar{\theta}}(\hat{a}_u \mid m, f, y^{\text{thk}}, \hat{a}_{<u}), \tag{9}$$

$$\ell_0 = \frac{1}{3T} \sum_{u=1}^{3T} \log \pi_{\bar{\theta}}(\hat{a}_u \mid m, f, \hat{a}_{<u}). \tag{10}$$

The preceding answer tokens $\hat{a}_{<u}$ are identical in both passes. Neither context contains the generated description; the thinking context includes its boundary tags. We average only over the trajectory tokens, excluding the prompt, reasoning, and boundary tags. The answer is not regenerated when the thinking segment is removed, and no expert answer is substituted in this comparison.

Perplexity and its relative reduction are

$$\mathrm{PPL}_+ = e^{-\ell_+}, \quad \mathrm{PPL}_0 = e^{-\ell_0}, \quad \Delta_{\mathrm{PPL}} = \frac{\mathrm{PPL}_0 - \mathrm{PPL}_+}{\mathrm{PPL}_0} = 1 - e^{\ell_0 - \ell_+}. \tag{11}$$

A positive reduction means that the generated intent reasoning increases the conditional likelihood of the actual planning answer. We use three reward levels:

$$R_{\mathrm{thk}} = g_{\mathrm{ans}} \begin{cases} 0, & \Delta_{\mathrm{PPL}} \le 0, \\ \frac{1}{2}, & 0 < \Delta_{\mathrm{PPL}} < \eta_r, \\ 1, & \Delta_{\mathrm{PPL}} \ge \eta_r, \end{cases} \qquad \eta_r = \tfrac{2}{3}. \tag{12}$$

The high-reward boundary $\eta_r = 2/3$ corresponds to $\ell_+ - \ell_0 = \log 3$, since $1 - e^{-\log 3} = 2/3$. A zero reduction receives zero reward. This is a likelihood-based alignment measure for the generated decision, rather than a statement-by-statement semantic verification of the reasoning.

### A.1.3 Waypoint Answer Reward

The answer reward combines shared acceptance checks with a preference-specific score. The reward uses all $H = T = 8$ predicted poses over 4 s. For the definitions below, let $p_1, \ldots, p_H$ denote their XY positions and $q_1, \ldots, q_H$ the expert reference positions, indexed from one to leave $p_0 = q_0 = (0, 0)$ for the current ego position.

**Shared checks.** Let $s_{\mathrm{NC}}, s_{\mathrm{DAC}} \in [0, 1]$ be the no-at-fault-collision and drivable-area-compliance scores, and define $c_{\mathrm{safe}} = s_{\mathrm{NC}} s_{\mathrm{DAC}}$. We also measure geometric irregularity by smoothing the coordinates and taking the RMS second difference:

$$\bar{p}_t = \frac{1}{|\mathcal{W}_t|} \sum_{j \in \mathcal{W}_t} p_j, \qquad \mathcal{W}_t = \{t-1, t, t+1\} \cap \{0, \ldots, H\}, \tag{13}$$

$$J_2 = \sqrt{\frac{1}{H-1} \sum_{t=1}^{H-1} \|\bar{p}_{t+1} - 2\bar{p}_t + \bar{p}_{t-1}\|_2^2}. \tag{14}$$

This quantity measures coordinate regularity at the fixed sampling interval; it is not the time-normalized physical jerk used in the trajectory diagnostics. The binary acceptance indicator and answer reward are

$$g_{\mathrm{ans}} = \mathbb{I}[\text{valid decoding}]\, \mathbb{I}[c_{\mathrm{safe}} \ge c_{\min}]\, \mathbb{I}[J_2 < \eta_J], \tag{15}$$

$$R^f_{\mathrm{ans}} = g_{\mathrm{ans}}\, c_{\mathrm{safe}}\, S_f, \qquad c_{\min} = 0.25, \quad \eta_J = 1.74. \tag{16}$$

Acceptance is a binary decision, whereas $c_{\mathrm{safe}}$ retains graded benchmark scores. For example, a partial NC score of $0.5$ may pass when the other checks are satisfied; passing does not mean every benchmark component equals one. The same $g_{\mathrm{ans}}$ gates description and reasoning rewards before GDPO normalization.

**Efficiency preference.** Let $d = q_H/(\|q_H\|_2 + \epsilon)$ be the regularized reference direction, $l = p_H^\top d$ the endpoint's longitudinal reach, and $e_\perp = \|p_H - ld\|_2$ its lateral drift. The score rewards net progress along the reference direction:

$$S_{\mathrm{eff}} = \mathrm{clip}_{[0,1]} \left( \frac{l - \lambda_\perp e_\perp}{\rho_{\max}(q_H^\top d + \epsilon)} \right), \qquad \lambda_\perp = 1, \quad \rho_{\max} = 1.2. \tag{17}$$

This score subtracts lateral drift from longitudinal progress and is anchored to the expert direction. It differs from the NAVSIM ego-progress metric.

**Safety preference.** Let $u_t$ be the unit tangent estimated from adjacent expert waypoints and $n_t$ its perpendicular unit normal. For $e_t = p_t - q_t$, the lateral-dominant displacement and its score are

$$D_w = \frac{1}{H}\sum_{t=1}^{H}\sqrt{(n_t^\top e_t)^2 + (u_t^\top e_t/7)^2}, \qquad S_{\text{safe}} = e^{-D_w/0.8}. \tag{18}$$

The longitudinal error is downweighted relative to lateral error. This favors adherence to the reference path while allowing longitudinal variation; it does not require interpreting the expert trajectory as the geometric lane center.

**Comfort preference.** Define step lengths $v_t = \|p_t - p_{t-1}\|_2$ and their standard deviation $\sigma_v$. The comfort score combines coordinate smoothness and steady step lengths:

$$S_{\text{comf}} = \exp\left[-\frac{1}{2}\left(\frac{J_2}{\eta_J} + \frac{\sigma_v}{\zeta}\right)\right], \qquad \zeta = 0.5. \tag{19}$$

The step lengths are measured in meters per sampling step. This expression is the geometric mean of two exponential scores, so either irregular motion or variable step lengths reduces the reward. Each preference uses its corresponding $S_f$ while retaining the same acceptance checks and $c_{\text{safe}}$ factor.

### A.1.4 Reward Aggregation and Selected Policy Updates

An auxiliary format reward checks the description–thinking–answer fields and a valid waypoint answer. GDPO first computes the group statistics for each reward dimension using all $G$ completions of the same input–preference pair. It then combines the standardized dimensions and whitens the aggregate over the batch, as in Equation 7. A constant dimension contributes zero. Raw process rewards gated to zero may still have a negative centered value when other completions receive positive process rewards.

Selection uses the final completion-level advantage $A_{i,g}$ after this normalization. Define

$$\mathcal{P}_i = \{g : A_{i,g} > 0\}, \qquad M_{i,g} = \mathbb{I}[g \in \mathcal{S}_i], \tag{20}$$

$$\mathcal{S}_i = \text{TopIndices}_{\min(2,|\mathcal{P}_i|)}\big(\{A_{i,g}\}_{g\in\mathcal{P}_i}\big). \tag{21}$$

Here TopIndices returns the indices of the largest values. An empty $\mathcal{P}_i$ yields an empty selected set. We keep one completion if only one has positive advantage, and two otherwise. All original completions still determine the group statistics; selection is not followed by re-centering within the retained subset.

**Selected-response objective.** With $H_{i,g,u} = (m_i, f_i, y_{i,g,<u})$, define

$$\rho_{i,g,u}(\theta) = \frac{\pi_\theta(y_{i,g,u} \mid H_{i,g,u})}{\pi_{\text{old}}(y_{i,g,u} \mid H_{i,g,u})}. \tag{22}$$

The selected-response policy surrogate is

$$\begin{aligned}\mathcal{J}_{\text{clip}}(\theta) &= \frac{1}{Z_{\mathcal{B}}}\sum_{i,g} M_{i,g}\sum_{u=1}^{|y_{i,g}|}\psi(\rho_{i,g,u}, A_{i,g}),\\ \psi(\rho, A) &= \min\{\rho A, \text{clip}(\rho, 1-\epsilon_c, 1+\epsilon_c)A\},\end{aligned} \tag{23}$$

where $Z_{\mathcal{B}} = \max(1, \sum_{i,g} M_{i,g}|y_{i,g}|)$ counts selected response tokens, excluding prompts and padding. Advantages and selection masks are held fixed during differentiation. The rewards are stage-decomposed, while the selection and advantage are completion-level. Here $\epsilon_n$ stabilizes normalization and $\epsilon_c$ controls clipping. We retain KL regularization to the SFT reference policy over the selected responses.

The mask applies to every generated token of a selected response, including its description, reasoning, answer, and structural delimiters. Prompt and padding tokens are excluded. Equation 23 averages over the retained response tokens, so the denominator is $\sum_{i,g} M_{i,g}|y_{i,g}|$, with a lower bound of one for an empty selection. If the entire batch has no positive advantage, the optimizer update is skipped. The reward-derived advantages and selection masks are stop-gradient quantities. Reference-policy KL regularization uses the same selected-response mask.

### A.2 Data Synthesis Prompts and Construction Details

This section describes the structured data synthesis pipeline. The prompt templates below are presented with horizon and token-format placeholders instantiated for the eight-waypoint, four-second configuration. The goal of the synthesis stage is to construct planning-oriented supervision that jointly contains grounded scene descriptions, trajectory-aligned reasoning, and deterministic waypoint-token answers. For each nuPlan sample, we use the synchronized camera observations, ego-vehicle state, navigation command, historical trajectory, and expert future trajectory to synthesize a structured response in the following format:

```
<description>observable scene evidence</description>
<thinking>planning-relevant reasoning</thinking>
<answer>time-indexed position and heading tokens</answer>
```

The three fields are designed to be functionally separated: the description captures observable visual evidence, the thinking field explains why the target motion is appropriate, and the answer field contains only ego-centric waypoint tokens that represent the full future trajectory.

### A.3 Scene Description Prompt

The scene description is generated by Qwen3.5-Plus from synchronized camera views. We instruct the annotation model to report only visually grounded facts and to avoid planning intent, safety judgment, or future action speculation. The description prompt does not supply the expert future trajectory, keeping this field focused on visual observations. The four-view annotation prompt below and the policy input packaging in Appendix A.9 serve different stages of data preparation.

```
You are an expert driving scene observer.

You are given four synchronized camera views of the same driving moment:
- front view
- left view
- right view
- back view

Write a concise scene description in at most 2 sentences using only what
is directly visible in these four views.

Describe only observable facts:
- visible vehicles, pedestrians, cyclists, traffic lights, cones,
barriers, lane markings, parked cars, and road layout;
- rough relative positions such as ahead, ahead-left, left side, right
lane, behind, or along the curb;
- visible motion cues such as moving, slowing, stopped, parked,
crossing, or blocking;
- road surface, weather, lighting, or visibility only when they are
visibly apparent.

Hard constraints:
- Only describe what is seen.
- Mention only the 3-5 most decision-relevant observations for the next
4 seconds.
- Do not describe what the ego vehicle should do.
- Do not mention plans, intentions, safety judgments, comfort judgments,
or navigation reasoning.
- Do not infer hidden hazards, unseen traffic, or future actions.
- Exclude building names, architectural style, business names, and
background landmarks unless they directly constrain drivable space.
- Exclude parked objects or distant scene details unless they directly
affect nearby free space or motion.
- If something is uncertain, use cautious wording such as "appears" or
"seems".
- Keep the text under 70 words, specific, compact, and
evidence-grounded.
```

```
Output only the description text without extra tags.
```

### A.4 REASONING PROMPT

After obtaining the description, Qwen3.5-Plus generates a driving rationale conditioned on the description, ego speed, route command, and a summary of the expert's demonstrated driving decision derived from its trajectory. A preference instruction specifies whether the rationale should emphasize efficiency, safety, or comfort. The annotation model explains the demonstrated decision; the expert trajectory remains the supervision target for the planning answer.

```
You are an expert autonomous driving decision analyst.

Your task is to generate a concise, high-value chain of thought for
training a smaller driving model.
You must explain why the target plan summary is appropriate using only
the provided scene description, route intent, and target motion summary.

INPUT CONTEXT:
- Scene Description: "{description_text}"
- Ego Status: Speed {ego_speed_rounded} m/s
- Navigation Command: "{nav_command}"
- Target Plan Summary: "{target_plan_summary_text}"
- Reasoning Focus: "{focus}"

INSTRUCTIONS:
1. Explain why the target plan summary is appropriate now.
2. Use only the most decision-relevant observations from the scene
description.
3. Mention only the 1-2 most important motion decisions in the target
plan summary.
4. Tie the reasoning to both the visible scene and the navigation
command.
5. Make the reasoning action-oriented: explain why this motion pattern
fits the current scene.
6. Reflect the specified focus through explicit priority and trade-off
logic.
7. Keep the reasoning concise, direct, and useful for policy learning.
8. Use 2-3 short sentences and keep the reasoning under 90 words.
9. Output a single compact paragraph without bullet points or tags.

QUALITY CONSTRAINTS:
- Do not simply restate the scene description.
- Do not give generic driving advice.
- Do not mention irrelevant background details.
- Do not explain multiple possible actions; justify the given target
plan.
- Do not justify every phase separately unless the scene clearly
requires it.
- Do not mention waypoint token indices, discretized coordinates, or
hidden trajectory labels.
- Do not justify the plan using hypothetical route changes, overtaking
opportunities, hidden hazards, or unobserved future events unless they
are explicitly supported by the scene description.
- Prefer concrete causal wording such as "because", "so", "therefore",
or "which means".

FOCUS-SPECIFIC RULES:
{focus_instruction}

Output only the reasoning text.
```

### A.5 PREFERENCE-CONDITIONED REASONING

To support diverse but valid driving behaviors, each synthesized reasoning sample is assigned one of three balanced reasoning focuses: efficiency, safety, or comfort. The focus changes the rationale while leaving the expert waypoint target unchanged. SFT therefore teaches preference-conditioned explanations and a shared output format. RL subsequently encourages preference-dependent motion through the trajectory scores in Appendix A.1.3.

Table 7: Preference instructions used during reasoning synthesis.

| **Focus** | **Instruction** |
|---|---|
| Efficiency | Prioritize timely and assertive progress while remaining consistent with the visible scene and route. Favor maintaining useful speed, preserving momentum, reducing unnecessary hesitation, and using available free space or lane continuity. If slowing, yielding, or stopping is required, explain why it is necessary rather than preferred. |
| Safety | Prioritize risk reduction, safety margins, and anticipation of possible conflicts. Focus on nearby hazards, uncertain agent behavior, limited clearance, crossing risk, or traffic-control constraints. Prefer conservative interpretation when the scene contains ambiguity, occlusion, vulnerable road users, or unstable traffic. |
| Comfort | Prioritize smoothness, stability, and low-disturbance motion while staying consistent with the scene and route. Focus on avoiding abrupt acceleration, harsh braking, sudden steering, or unnecessary speed oscillation. Explain how the target action supports smooth longitudinal and lateral control. |

### A.6 CANDIDATE-COMPARISON SUPERVISION

We also use Qwen3.5-Plus to annotate comparisons among preference-conditioned candidates for Slow inference. For each complex training scene, we construct a candidate-selection sample that contains the visual scene context, ego state, navigation command, and multiple preference-conditioned waypoint-token plans. We then use Qwen3.5-Plus to generate a concise checking rationale and the selected best candidate. This supervision teaches the same model to compare its own candidate plans and return a candidate index.

Each comparison sample contains $K$ candidate plans:

$$\mathcal{C} = \{(f_k, y_k^{\text{ans}}, \hat{\tau}_k)\}_{k=1}^{K}, \tag{24}$$

where $f_k$ denotes the preference condition, $y_k^{\text{ans}}$ is the waypoint-token sequence, and $\hat{\tau}_k$ is the decoded waypoint trajectory, which can also be represented by a compact motion summary. The annotation model produces a checking rationale $y^{\text{chk}}$ and a selected candidate index $k^\star$, which are formatted as:

```
<check>concise comparison and justification</check>
<selection>selected_candidate_id</selection>
```

The annotation prompt for the comparison is:

```
You are an expert autonomous driving plan evaluator.

Your task is to compare multiple candidate future plans for the same
driving scene and select the best one for training a smaller driving
model. You must judge the candidates using only the provided scene
description, ego status, navigation command, and candidate waypoint
plans.

INPUT CONTEXT:
- Scene Description: "{description_text}"
- Ego Status: Speed {ego_speed_rounded} m/s
- Navigation Command: "{nav_command}"

CANDIDATE PLANS:
```

```
{candidate_plan_list}

Each candidate contains:
- Candidate ID
- Preference Focus
- Waypoint Tokens for the next 4 seconds
- Decoded Motion Summary

INSTRUCTIONS:
1. Compare all candidates with respect to visible safety constraints,
navigation consistency, progress, and motion smoothness.
2. Prefer plans that avoid conflicts with visible vehicles, pedestrians,
cyclists, traffic lights, lane boundaries, and nearby obstacles.
3. Reject candidates that are inconsistent with the navigation command,
contain implausible motion, or show unnecessary risk.
4. When multiple candidates are safe, choose the one with better route
progress and smoother motion.
5. Use only evidence from the scene description and the candidate plans.
6. Keep the checking rationale concise, specific, and action-oriented.
7. Output 2-4 short sentences in the <check> field.
8. Output exactly one candidate ID in the <selection> field.

QUALITY CONSTRAINTS:
- Do not invent hidden hazards or unobserved future events.
- Do not simply restate every candidate.
- Do not mention low-level token indices unless needed for comparison.
- Do not select a candidate only because of its preference label.
- The selected candidate must be justified by safety, route consistency,
progress, or smoothness.
- Do not output any extra text outside the required XML blocks.

Output format:
<check>concise comparison and justification here</check>
<selection>candidate_id_here</selection>
```

#### A.6.1 Learned Routing and Inference

During data synthesis, Qwen3.5-Plus labels scenes as simple or complex. SFT teaches the model to predict Fast or Slow from the driving context. Fast mode generates one complete structured response. Slow mode generates three responses, one for each preference, and then compares the candidates in an additional model pass. The selected waypoint sequence becomes the final plan. Oracle candidate scores are used only for post-hoc evaluation; they are unavailable during model selection.

### A.7 Training Query Format

The synthesized samples are converted into an instruction-following format for supervised fine-tuning. The user-side prompt specifies the ego speed, navigation command, optional preference focus, and the required XML output schema:

```
You are an expert driving assistant. Your current speed is {speed} m/s,
the navigation command is '{nav_command}', {preference_instruction}
based on the understanding of the driving scene and the navigation
information,
what is your plan for the next four seconds? Output the scene
description in
<description> </description>, the planning reasoning process in
<thinking> </thinking>, and the final planning answer in <answer>
</answer>
tags, respectively. Keep the scene description concise and factual,
within
2 short sentences and 70 words. Keep the planning reasoning process
concise
and action-oriented, within 2-3 short sentences and 90 words. The answer
must
```

```
contain only waypoint tokens that describe the full future trajectory.
Do not
output any extra text outside these tags. For example, a correct answer
format
is like:
'<description>scene description here</description>
<thinking>reasoning process here</thinking>
<answer><traj_start><t0_x_10><t0_y_06><t0_h_17>...<t7_x_15><t7_y_18><t7_
h_90><traj_end></answer>'.
```

The corresponding system prompt enforces strict structural compliance:

```
You are an expert driving assistant. Output exactly three XML blocks in
this order:
<description>...</description>
<thinking>...</thinking>
<answer>...</answer>.
Do not output any extra text outside these three blocks.
```

### A.8 Waypoint Token Construction

The main configuration predicts $T = 8$ ego poses at future times $(t + 1) \times 0.5\,\text{s}$ for $t \in \{0, \ldots, 7\}$. Each pose contains longitudinal position $x_t$, lateral position $y_t$, and heading $h_t$ relative to the current ego orientation. Positions are in meters and heading is in radians. The answer contains $3T = 24$ pose-component tokens; trajectory delimiters and XML tags are excluded from this count.

**Time-dependent vocabularies.** For each step $t$ and component $b \in \{x, y, h\}$, define a token vocabulary $\mathcal{V}_{t,b} = \{u_{t,b,k}\}_{k=0}^{K_{t,b}-1}$ with ordered scalar representatives $v_{t,k}^b$. The discretization uses empirical ranges that vary with the prediction step. In the main configuration, X and Y spacings are $0.5\,\text{m}$ and $0.25\,\text{m}$, respectively; heading spacing is approximately one degree. The component vocabulary has $\sum_t \sum_b K_{t,b} = 2179$ tokens, plus two trajectory delimiters. This construction avoids repeating the widest component range at every prediction step.

Table 8: Numbers of longitudinal, lateral, and heading bins at each future step in the main pose-token configuration.

| Future time (s) | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 |
|---|---|---|---|---|---|---|---|---|
| $K_{t,x}$ | 27 | 44 | 61 | 79 | 96 | 114 | 131 | 148 |
| $K_{t,y}$ | 13 | 20 | 33 | 49 | 69 | 93 | 119 | 147 |
| $K_{t,h}$ | 35 | 62 | 87 | 111 | 134 | 154 | 171 | 182 |

**Quantization and decoding.** Write $\xi_t^x = x_t$, $\xi_t^y = y_t$, and $\xi_t^h = h_t$ for the expert pose components. Quantization selects the nearest representative:

$$Q_t^b(\xi) = \operatorname*{arg\,min}_{0 \le k < K_{t,b}} |\xi - v_{t,k}^b|, \qquad r_t^b = u_{t,b,Q_t^b(\xi_t^b)}. \tag{25}$$

Values outside the represented range map to its nearest endpoint. For positions, representatives follow $v_{t,k}^b = b_{t,\min} + k\Delta_b$. For heading, $v_{t,k}^h = h_{t,\min} + k\Delta_{t,h}$ with $\Delta_{t,h} = (h_{t,\max} - h_{t,\min})/K_{t,h}$. Heading is quantized on this bounded interval; its endpoints are not treated as adjacent. Given a generated index $\widehat{k}_t^b$, decoding returns $\widehat{\xi}_t^b = v_{t,\widehat{k}_t^b}^b$. Heading is therefore predicted explicitly rather than recovered from the XY path.

The answer uses the order

$$a = [r_0^x, r_0^y, r_0^h, \ldots, r_7^x, r_7^y, r_7^h], \qquad |a| = 24. \tag{26}$$

Generated answers use the same format. The timestep is part of the token identity and requires no extra token.

**Geometric soft supervision.** Let $\mathcal{N}_{t,b}$ contain the nearest $\min(5, K_{t,b})$ representatives to the unquantized expert component. The target is

$$q_{t,b}(k) = \frac{\mathbb{I}[k \in \mathcal{N}_{t,b}] \exp(-|\xi_t^b - v_{t,k}^b|/s_b)}{\sum_{j \in \mathcal{N}_{t,b}} \exp(-|\xi_t^b - v_{t,j}^b|/s_b)}. \tag{27}$$

The scales are $s_x = 1.0\,\text{m}$, $s_y = 0.5\,\text{m}$, and $s_h = 0.1\,\text{rad}$. For heading, the distance is the absolute angular difference within the selected interval, not a periodic distance. Let $j(t, b)$ be the position of the corresponding component token in the target sequence $z$. The model distribution is

$$p_{t,b}(k) = \frac{\pi_\theta(u_{t,b,k} \mid m, f, z_{<j(t,b)})}{\sum_{v \in \mathcal{V}_{t,b}} \pi_\theta(v \mid m, f, z_{<j(t,b)})}. \tag{28}$$

Thus, the target has mass only on neighboring bins, whereas the model distribution is normalized over all bins for the same step and component. We use component weights $\omega_x = 1.3$, $\omega_y = 1$, and $\omega_h = 1$, with $\lambda_{\text{soft}} = 0.1$ for a hard-to-soft loss coefficient ratio of 10:1. The auxiliary term in Equation 3 averages over future steps and sums the weighted component losses. Its implementation as soft cross-entropy has the same parameter gradients as $D_{\text{KL}}(q_{t,b} \| p_{t,b})$, since target entropy is independent of the model parameters.

### A.9 Training Sample Packaging

The synthesized description and reasoning are paired with the visual observations, ego state, navigation command, and the expert pose-token target. The system instruction specifies the number of poses and prediction horizon:

```
You are an autonomous driving policy model. Use the visual observations,
vehicle state, and planning conditions to predict 8 future ego poses
over the next 4 seconds. Each pose contains a longitudinal position, a
lateral position, and an ego-relative heading, represented by three
time-indexed tokens in that order. In <answer>, output only the
trajectory token sequence.
```

The structured target contains all three fields:

```
<description>scene description</description>
<thinking>preference-conditioned rationale</thinking>
<answer><traj_start>
<t0_x_i0><t0_y_j0><t0_h_k0>...<t7_x_i7><t7_y_j7><t7_h_k7><traj_end></ans
wer>
```

The indices above are schematic. The description and reasoning are generated during annotation; the pose-token answer is constructed directly from the expert trajectory. Waypoint-only SFT uses the trajectory target before the reasoning fields are introduced.

### A.10 Representation Variants

The waypoint-only SFT comparison in Table 5 retains heading in every variant and uses $T = 8$ future poses. The joint-2D variant emits one joint XY token and one heading token per pose, giving $2T$ component tokens, with X/Y grid spacings of $1.0/0.5\,\text{m}$. The shared-component variant emits separate X, Y, and heading tokens, giving $3T$ tokens, with X/Y spacings of $0.5/0.25\,\text{m}$. Both variants share their token namespaces across timesteps and use a global coordinate envelope.

The time-indexed variants emit one token for each step and pose component, also giving $3T$ tokens. They retain X/Y spacings of $0.5/0.25\,\text{m}$ and use the step-dependent position and heading supports described above. Consequently, joint-to-shared comparisons change both factorization and resolution, while shared-to-time-indexed comparisons change temporal namespaces and represented supports. They assess representation designs under a common training setting. The no-soft and full time-indexed variants hold the representation fixed and differ only in auxiliary soft supervision, providing the direct soft-loss ablation. The text-coordinate comparison in the table's upper block uses the separate structured-CoT SFT setting.

### A.11 Training Settings

The waypoint-only SFT stage uses 200k image–trajectory pairs, comprising 170k from nuPlan and 30k from nuScenes. Structured-CoT SFT uses 30k samples, and RFT uses 15k prompts. The same scenes may appear in more than one stage, so these counts are not a unique-scene total. All three stages update the full model. Table 9 summarizes the reported training configuration.

Table 9: Training configuration. An RFT group contains eight completions of the same preference-conditioned prompt.

| Setting | Waypoint SFT | Structured-CoT SFT | RFT |
|---|---|---|---|
| Samples/prompts | 200k | 30k | 15k prompts |
| Epochs | 3 | 2 | 1 |
| Learning rate | $5 \times 10^{-6}$ | $10^{-5}$ | $2 \times 10^{-5}$ |
| Effective batch size | 8 | 8 | 32 |
| Trainable parameters | Full model | Full model | Full model |
| Hardware | 8× H800 | 8× H800 | 8× H800 |
| Hard:soft loss coefficient ratio | 10:1 | 10:1 | — |
| Rollout group size | — | — | 8 |
| Sampling temperature | — | — | 0.8 |
| Sampling top-$p$ | — | — | 0.4 |
| Reference KL coefficient | — | — | 0.04 |

The policy receives four historical frames at 2 Hz from the front, front-left, and front-right cameras, together with ego state and navigation. It predicts eight future waypoints over 4 s at 2 Hz. Camera-only refers to environmental observations supplied to the policy: the drivable region used for training reward evaluation is not an additional policy input. The three-field query above is used for structured-CoT training; waypoint-only SFT supervises the answer tokens before the reasoning fields are introduced.

### A.12 Meta-action evaluation

For the nuScenes decision-level evaluation, the same trajectory-to-action procedure is applied to the expert trajectory and the free-running predicted trajectory. It assigns a lateral label from straight/left/right and a longitudinal label from keep speed/accelerate/decelerate/stop. The reported accuracy and per-class F1 compare these labels. This representation tests agreement in maneuver category without requiring every predicted waypoint to coincide with the expert, while geometric errors can still remain within a category. Displacement, NAVSIM, and closed-loop metrics provide the complementary trajectory-level evaluations.

The SFT baseline and AlphaDrive training recipe use the same Qwen3-VL backbone as VarioDrive in Table 2. In the reward ablation (Table 6), answer and format rewards are always enabled; description and thinking rewards are added separately or together. These intermediate rewards evaluate output components and are aggregated into the completion-level policy advantage.

## B Training stages and intermediate-context interventions

### B.1 Contribution of supervision and reward design

Table 10: Training-stage comparisons and subsequent Slow inference. The two RFT rows compare alternative reward objectives; the last row changes inference using the full-RFT checkpoint.

| Configuration | PDMS | EP | TTC | NC | DAC | Comfort |
|---|---|---|---|---|---|---|
| Waypoint-only SFT | 80.96 | 78.18 | 86.18 | 95.16 | 93.84 | 99.82 |
| Structured-CoT SFT | 83.68 | 76.42 | 92.10 | 97.06 | 92.71 | 99.87 |
| Answer-only RFT | 88.75 | 85.26 | 93.46 | 98.19 | 96.14 | 100.00 |
| Stage-reward RFT | 90.84 | 84.85 | 95.07 | 98.34 | 97.43 | 100.00 |
| + Slow selection (inference) | 92.42 | 86.73 | 97.95 | 99.26 | 98.21 | 100.00 |

Table 10 separates the effects of structured supervision, reward design, and inference. Structured-CoT SFT increases PDMS from 80.96 to 83.68 and improves TTC from 86.18 to 92.10. Answer-only RFT reaches 88.75 PDMS; adding the description and thinking rewards raises PDMS to 90.84, with further improvements in TTC, NC, and DAC and a small decrease in EP. The two RFT rows compare alternative reward objectives. The final row applies multi-preference proposal and selection by the same model to the full-RFT checkpoint, reaching 92.42 PDMS; it is an inference change rather than another training stage.

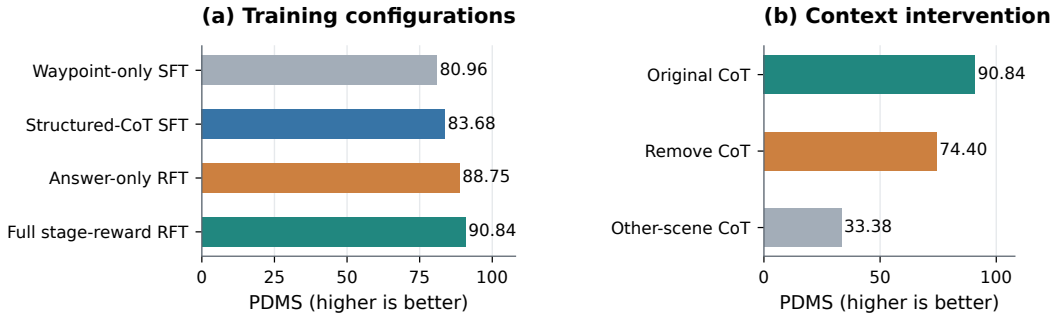



Figure 5: Training feedback and intermediate-context dependence. (a) Full stage-reward RFT improves over answer-only RFT by 2.09 PDMS; the two objectives are alternative training configurations. (b) At a fixed checkpoint, the generated description–thinking prefix is retained, removed, or replaced before regenerating the waypoint answer. Tables 10 and 11 give the complete results.

### B.2 Description–thinking prefix intervention

Table 11: Description–thinking prefix interventions at a fixed checkpoint. Only the waypoint answer is regenerated. Autoregressive (AR) displacement errors use free-running generation and are in meters.

| Context | PDMS ↑ | AR-ADE ↓ | AR-FDE ↓ |
|---|---|---|---|
| Original generated CoT | 90.84 | 1.19 | 2.51 |
| Remove CoT | 74.40 | 2.68 | 6.29 |
| CoT from another scene | 33.38 | 6.99 | 14.45 |

We first generate a complete response, then regenerate only its waypoint answer under three prefix conditions: retain the generated description and thinking, remove both fields, or replace both with the corresponding fields from another scene. The checkpoint, observations, and navigation input are fixed. Removing the prefix reduces PDMS from 90.84 to 74.40, while using another scene's prefix reduces it to 33.38. The corresponding AR-ADE/FDE values increase from 1.19/2.51 to 2.68/6.29 and 6.99/14.45. These results show that the policy uses the supplied intermediate context and is sensitive to its scene relevance. Prefix removal also changes the input distribution of a CoT-trained model, so it is distinct from training a waypoint-only model.

Here autoregressive (AR) ADE/FDE are measured after free-running answer generation and include accumulated decoding errors. Teacher-forced (TF) ADE/FDE in the representation comparison instead condition each trajectory-token prediction on ground-truth preceding tokens. TF evaluation isolates token prediction from accumulated prefix errors; AR evaluation measures the complete generated trajectory. The two protocols are reported separately.

### B.3 Route-command intervention

In a complementary test, we replace left/right-turn commands with `go straight` in 80 samples while keeping the observations, ego state, and decoding settings fixed. The reasoning changes consistently with the revised command in 79/80 cases. Mean endpoint lateral magnitude decreases from 5.925 m to 4.797 m, and mean absolute endpoint yaw decreases from 0.620 rad to 0.474 rad.

Table 12: Route-command intervention in 80 turn samples. The columns report means over samples, before and after replacing the route command with go straight while fixing the observations.

| Command | Endpoint lateral magnitude (m) | Absolute endpoint yaw (rad) |
|---|---|---|
| Original turn command | 5.925 | 0.620 |
| Counterfactual go straight | 4.797 | 0.474 |

The mean absolute paired changes are 2.328 m and 0.295 rad, respectively; these measure per-sample changes and therefore differ from the changes in aggregate magnitudes. Together with prefix replacement, this test supports the planning relevance of the structured context. Since navigation also directly conditions trajectory generation, command sensitivity alone does not establish that every trajectory change is mediated by the generated reasoning.

# C Candidate selection and preference-conditioned trajectories

## C.1 Candidate pools and in-model selection

Table 13: Preference-conditioned candidate pools and selection. Same-preference best-of-3 oracle rows are independent controls. Random, model, and oracle selection use the same three mixed-preference candidates. All scores are higher-is-better.

| Candidate pool / selector | PDMS | EP | TTC | NC | DAC | Comfort |
|---|---|---|---|---|---|---|
| Efficiency (single) | 88.79 | 88.46 | 91.30 | 97.12 | 96.94 | 99.84 |
| Efficiency, Oracle | 91.04 | 89.39 | 92.10 | 98.26 | 97.17 | 100.00 |
| Safety (single) | 90.84 | 84.85 | 95.07 | 98.34 | 97.43 | 100.00 |
| Safety, Oracle | 91.67 | 84.94 | 96.35 | 98.62 | 98.05 | 100.00 |
| Comfort (single) | 87.16 | 82.11 | 93.30 | 97.55 | 96.54 | 100.00 |
| Comfort, Oracle | 89.69 | 84.90 | 96.18 | 97.68 | 97.18 | 100.00 |
| Multi-preference, Random | 88.62 | 85.25 | 93.65 | 97.37 | 97.17 | 99.94 |
| Multi-preference, Model selection | 92.42 | 86.73 | 97.95 | 99.26 | 98.21 | 100.00 |
| Multi-preference, Oracle | 93.26 | 89.67 | 98.84 | 99.75 | 98.67 | 100.00 |

Each same-preference oracle row in Table 13 is an independent control with three candidates sampled under one preference. The multi-preference pool contains exactly three candidates: one each for efficiency, safety, and comfort. Random selection, selection by the same model, and oracle selection use this same pool. The oracle chooses the highest-scoring candidate using post-hoc benchmark scores; it provides an upper bound for the given pool, and these scores are unavailable to the model during selection.

The strongest same-preference oracle reaches 91.67 PDMS, whereas the multi-preference oracle reaches 93.26. Thus, preference conditioning improves the observed candidate pool at a fixed candidate count. On the multi-preference pool, selection by the same model improves over random selection by 3.80 PDMS, reaching 92.42 and leaving a 0.84-point gap to the oracle. This comparison matches the number of candidates rather than total computation, because selecting among candidates requires another pass through the shared model. The model chooses the oracle candidate in 63.20% of scenes. No rule-based fallback is used in these experiments.

Figure 4(a,b) summarizes the candidate-pool and selector comparisons. Table 4 and Appendix D give adaptive-inference latency and its measurement scope.

## C.2 Trajectory diversity and dynamic quantities

The three preference-conditioned candidates have a mean pairwise trajectory distance of 1.08 m, a median of 0.90 m, and a 90th percentile of 1.90 m. Their RMS endpoint dispersion is

$$D_{\text{end}} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\sigma_{x,i}^2 + \sigma_{y,i}^2\right)} = 1.51\,\text{m}, \tag{29}$$

Table 14: Preference diagnostics from complete generated trajectories without smoothing. Curvature is descriptive; the other dynamic quantities and autoregressive (AR) displacement errors (m) are lower-is-better.

| Preference | Jerk ($m/s^3$) | Lat. acc. RMS ($m/s^2$) | Acc. mag. ($m/s^2$) | Curvature (1/m) | AR-ADE | AR-FDE |
|---|---|---|---|---|---|---|
| Efficiency | 0.7348 | 1.3851 | 2.43 | 0.1390 | 1.15 | 2.21 |
| Safety | 0.6833 | 1.2657 | 2.37 | 0.1556 | 0.79 | 1.51 |
| Comfort | 0.6504 | 1.3237 | 1.97 | 0.2181 | 1.16 | 2.24 |

where $M$ is the number of scenes and $\sigma_{x,i}$ and $\sigma_{y,i}$ are the endpoint-coordinate standard deviations across candidates in scene $i$. This quantity measures within-scene candidate diversity, rather than error relative to an expert trajectory.

Table 14 reports dynamic quantities computed from complete generated trajectories without smoothing. Efficiency yields the highest ego progress, safety the lowest lateral acceleration and displacement errors, and comfort the lowest jerk and acceleration magnitude. Figure 4(c) visualizes the progress, TTC, jerk, and acceleration comparisons. Mean absolute curvature is a descriptive statistic: the required curvature depends on the route, so a smaller value is not universally preferable.

The preference-dynamics and prefix-intervention experiments are separate diagnostics. Their AR-ADE/FDE values are therefore retained in their respective tables rather than combined into one comparison: for example, the safety row in Table 14 reports 0.79/1.51, whereas the original-prefix row in Table 11 reports 1.19/2.51. Aligning their sample sets, checkpoints, and decoding settings is necessary before comparing these error magnitudes directly.

## D Inference latency and measurement scope

Table 15: Latency scopes on one H800, batch size 8. Partial-stage measurements omit description and thinking. Times are seconds.

| Scope | Fast | Slow |
|---|---|---|
| Waypoint-only generation (+ selection for Slow) | 0.781 | 3.936 |
| Complete structured output | 2.870 | 7.149 |
| Structured candidate generation | — | 4.469 |
| Selection | — | 2.680 |

Table 16: Original partial-stage timing ranges, retained separately from complete-output latency.

| Mode | Minimum (s) | Maximum (s) | Mean (s) |
|---|---|---|---|
| Fast | 0.741 | 0.813 | 0.781 |
| Slow | 3.334 | 4.952 | 3.936 |

Table 15 distinguishes waypoint-only latency from complete structured-output latency. The original Fast measurement, 0.781 s, covers waypoint generation without description or thinking. The original Slow measurement, 3.936 s, includes approximately 1.22 s for batched waypoint-only candidate generation, 2.68 s for selection, and minor overhead. Including description and thinking increases the complete Fast and Slow times to 2.870 s and 7.149 s; the latter comprises 4.469 s for structured candidate generation and 2.680 s for selection. Table 16 preserves the minimum, maximum, and mean of the partial-stage measurements.

All reported timings use one H800 at batch size 8. Adaptive routing sends 56.22% of navtest scenes to Slow mode and reaches 92.13 PDMS at a measured mean latency of 5.30 s. Its latency reduction

relative to Always Slow and retained Fast-to-Slow PDMS gain are

$$1 - \frac{5.30}{7.149} \simeq 25.9\%, \tag{30}$$

$$\frac{92.13 - 90.84}{92.42 - 90.84} \simeq 81.6\%. \tag{31}$$

The Adaptive latency is measured directly rather than inferred as a weighted average of the two fixed modes. These measurements characterize a quality–latency trade-off. They measure complete-output generation at batch size 8, rather than single-scene execution on a vehicle; no vehicle-control deadline follows from these measurements.