WIDER COVERAGE, BETTER PLAN: ADMISSIBLE ALTERNATIVES FOR END-TO-END VLA PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In autonomous driving, most scenes admit several routes to proceed, and a planner, often a Vision-Language-Action (VLA) model today, must recognise these options yet commit to one of them. Recent works enable VLA planners to sample extensive trajectories with generative heads, and on challenging benchmarks, the strongest methods pick one of thousands of candidates with a learned scorer. Can the VLA planner generate one or a small set of trajectories, instead of thousands of alternatives, that already contain high-quality options? We argue that supervised fine-tuning (SFT) on an offline dataset, where a single drive trajectory is sampled per scene, collapses the planner onto one mode. As such, we propose to train the planning head of a VLA planner on admissible alternative plans (AAP), additional plans per scene that a rule-based scorer accepts. Applied to Qwen-Drive-1.0 on the NAVSIM v2 navhard benchmark, AAP improves the single plan from 0.315 to 0.373 EPDMS, a gain that holds when both planners drive at equal speed, and turns near-identical samples into distinct options whose best of 16 rises from 0.359 to 0.614. A second stage of reinforcement learning makes the samples more reliable, and a speed-matched control shows that much of its gain comes from more cautious driving. We advocate evaluating planners by the diversity and coverage of their samples, not only by a single benchmark score. Code will be made publicly available.

1 INTRODUCTION

Autonomous driving planners built on vision-language models (VLMs), often called vision-language-action (VLA) models, increasingly decode trajectories with generative planning heads that read the VLM's representation and turn latent features or noise into a route plan (Zhou et al., 2026; Felemban et al., 2026; Li et al., 2026c;b; NVIDIA et al., 2025b). Most are trained by supervised fine-tuning (SFT) on offline dataset, and several add a reinforcement learning stage that aligns the plan with a driving score (Li et al., 2026c; Zhou et al., 2025; 2026).

Advanced benchmarks in autonomous driving, such as NAVSIM (Dauner et al., 2024), score what a plan leads to in a non-reactive simulation, which is cheaper than closed-loop simulation and correlates with the driving goal better than displacement errors as in open-loop simulation, and NAVSIM v2 (Cao et al., 2025) adds pseudo-simulation and navhard, a split of challenging scenarios. On the widly-used and complex navhard split, however, the leading methods do not rely on a single generated plan. A learned scorer that imitates the benchmark's rules rates a vocabulary of thousands of candidate plans, and the planner submits the best-rated one (Li et al., 2024; 2025; 2026d). This leads to our question: Can the VLA planner generate one or a small set of trajectories, instead of thousands of alternatives, that already contain high-quality options?

Diving deep to today's generative heads, such as DiffusionDrive (Liao et al., 2025), we find a vanilla diffusion policy produces samples from different noises that severely overlap. The 16 samples of Qwen-Drive-1.0 are likewise near-copies of one plan (Figure 1 (a)), or noisy variants of it when noise is injected during the flow steps. However, scoring benefits from a vocabulary coving the scene's driving options as much as possible, so plausible options come at least partly from the vocabulary. The near-copy generative pattern of the current methods force them to generate a huge vocabulary to obtain one plausible route option. We argue that a well-trained planner should need far fewer, and a handful of its samples should already contain a good plan.

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: (a) Qwen-Drive-1.0, (b) the AAP-SFT planner and (c) GRPO on AAP-SFT on three illustrative navhard scenes, direct planning. Each camera view shows the 16 samples from different initial noises, one colour per sample, and the three right columns show the same samples of each planner in bird's eye view (BEV), lateral against forward in meters. Each header gives the official EPDMS of the single plan and the mean, best, and worst over the 16 samples, then their 4 s endpoint spread and mean 4 s travel. Qwen-Drive-1.0's samples are near-copies, the AAP-SFT planner's cover distinct options, and GRPO's are consistent and more cautious.

In SFT, the planning head learns to reproduce single trajectory of each scene. Nothing in this objective incentivises the planner to generate samples with a spread over distinct planning options, and VLAs trained this way tend to collapse onto a single mode. Mode collapse further limits driving VLAs trained with GRPO (Shao et al., 2024; DeepSeek-AI, 2025; Liu et al., 2025a; Li et al., 2026c; Zhou et al., 2025; 2026). GRPO learns from a group of samples of the same scene whose rewards vary. Each sample's advantage is its reward minus the group's mean reward, and the update weights each sample by its advantage. When the samples are near-copies, their rewards are nearly equal, every advantage is close to zero, and the update nearly vanishes. We therefore first measure how varied a planner's samples are, and then train its planning head so that samples cover distinct options, which also improves a single plan.

Starting from Qwen-Drive-1.0 (Zhou et al., 2026), a frozen 4B VLM with a 1B flow-matching planning head (its "planning expert"), we fine-tune the head on *admissible alternative plans* (AAP) as a training-data augmentation to obtain an AAP-SFT planner (Figure 2 (a,b)). Alternative plans come from the scene geometry, as lane-centreline offsets driven with jerk-limited speed profiles, and from the rule-based PDM-Closed planner (Dauner et al., 2023)(Figure 4). To keep only admissible alternatives, we score each one offline with EPDMS of NAVSIM v2 (Cao et al., 2025) and keep it only if it has no collision, off-road, wrong-way, red-light, or time-to-collision fault.

With AAP-SFT, the planning head's samples become distinct options (Figure 3 (c)) rather than near-copies of one plan (Figure 3 (a)). Its single plan improves from 0.315 to 0.373 EPDMS, also when both planners drive at the same speed, and the oracle best of its 16 samples rises from 0.359 to 0.614. A second stage of GRPO with EPDMS and diversity rewards (Figure 2 (c)) makes the samples more reliable but spends part of these options, and much of its gain comes from more cautious driving.

Beyond the scores we trace where the variety of its samples comes from, whether initial noise in the flow-matching head or noise during the flow's integration steps (Figure 3), compare the single plan with picks among the samples (Table 3) and with trivial references, stopping in place or replaying the same plan slower (Table 2), and show which driving terms each planner gains or loses (Section 6.3).

The main contributions of this paper lie in (1) admissible alternative plans (AAP), a scorer-filtered augmentation that improves both the single plan and the sample coverage of Qwen-Drive-1.0 (Sections 4 and 6.2), (2) an analysis of what 16 samples of a driving VLA offer and where their variety comes from (Sections 6.1 and 6.3), and (3) a GRPO study showing that reinforcement learning

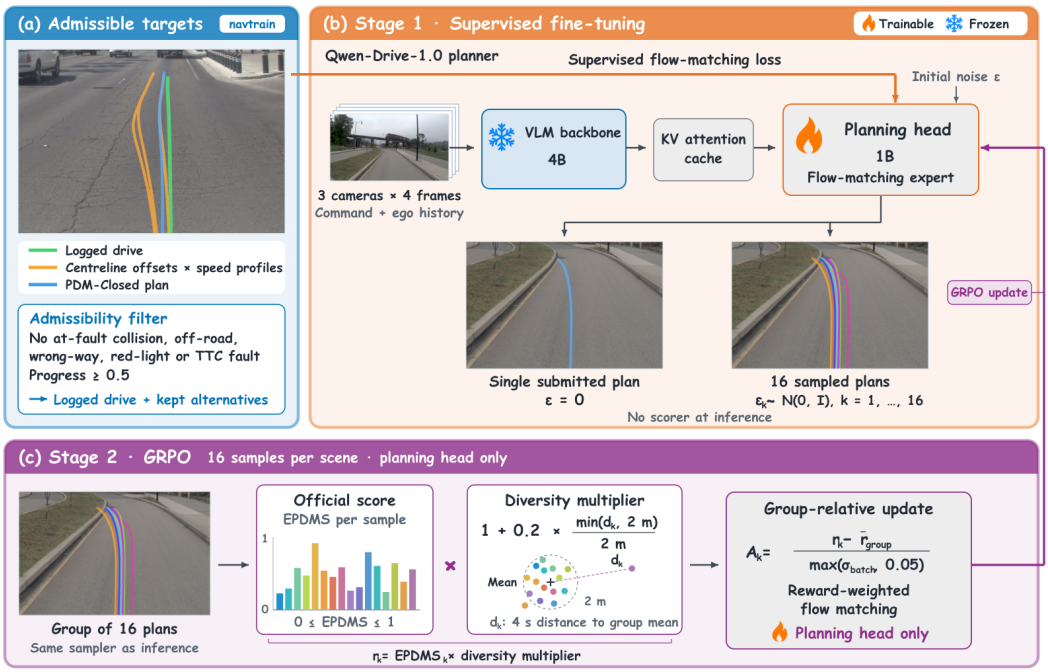


Figure 2: Overview. (a) Admissible alternative plans augment navtrain targets; an alternative is kept if the extended Predictive Driver Model score (EPDMS) finds no collision, off-road, wrong-way, red-light, or time-to-collision fault. (b) The VLM remains frozen, and SFT trains only the planning head; the single plan starts from $\epsilon = 0$ and the 16 samples from $\epsilon_k \sim \mathcal{N}(0, I)$. (c) GRPO groups the 16 samples, rewards each with EPDMS and a diversity bonus, and updates the planning head.

makes the samples more reliable but spends part of their options, and that much of its gain comes from more cautious driving (Section 6.4).

2 Related Work

Vision-language driving planners. End-to-end planners moved from modular stacks (Hu et al., 2023; Jiang et al., 2023) and sensor-fusion imitation (Chitta et al., 2023) to vision-language models that reason in text and emit trajectories as tokens or through dedicated action heads (Hwang et al., 2025; Tian et al., 2024; Felemban et al., 2026; Zhou et al., 2025; Li et al., 2026c;b; NVIDIA et al., 2025b). Qwen-Drive-1.0 (Zhou et al., 2026) pairs a frozen Qwen3.5-4B VLM with a flow-matching planning head that attends to the VLM's cache and is refined by head-only group-relative RL (Shao et al., 2024; Liu et al., 2025a); we use its public weights and inference unchanged.

Generative and multimodal trajectory heads. Anchor-based predictors produce a fixed set of hypotheses (Chai et al., 2019). Diffusion and flow matching (Lipman et al., 2023; Liu et al., 2023) let planners learn a distribution over futures (Chi et al., 2023; Black et al., 2024; Liao et al., 2025; Xing et al., 2025), although each is trained on one demonstration per scene, and DiffusionDrive adds anchors to keep its samples apart. GTRS diffuses waypoint differences (Li et al., 2025), the displacement targets our planning head also predicts, and HAD denoises a few intention anchors and expands them by polar offsets (Yao et al., 2026b). We keep the architecture fixed, change only the training targets, and measure which source of randomness creates samples that differ usefully under the metric.

Privileged references and learned scorers. Privileged planners such as PDM-Closed (Dauner et al., 2023) and the expert of LEAD (Nguyen et al., 2026) plan with ground-truth state and serve as upper references on NAVSIM. Learned scorers such as Hydra-MDP, GTRS, DriveSuprim and ZTRS (Li et al., 2024; 2025; Yao et al., 2026a; Li et al., 2026d) instead select one of many vocabulary trajectories at test time and lead the NAVSIM leaderboards. We add a few admissible alternatives as imitation targets of a generative planning head and evaluate its single plan and its samples without a scorer.

Benchmarks. NAVSIM (Dauner et al., 2024) scores a plan by simulating it for 4 s among the logged agents, which do not react to the ego vehicle. NAVSIM v2 (Cao et al., 2025) also checks what happens after these 4 s. It renders synthetic scenes from states a plan may reach and scores the planner again in them, which it calls pseudo-simulation. It further adds comfort and lane-keeping terms and navhard, a split of difficult scenarios. Closed-loop evaluation in sensor-rendering simulators such as AlpaSim (NVIDIA et al., 2025a) is more realistic but expensive, and a natural next step for testing planners trained with alternatives over longer horizons.

Reinforcement learning for generative planners. Flow-GRPO turns the flow's ordinary differential equation into a stochastic one with the same marginals to obtain transition likelihoods for policy gradients (Liu et al., 2025a). ReCogDrive fine-tunes a diffusion planner on a VLM with group-relative rewards on NAVSIM v1 (Li et al., 2026c). PlannerRFT observes that reinforcement fine-tuning of diffusion planners exploits its rewards poorly when the planner's trajectories are not multimodal, and guides the denoising process toward more promising exploration (Li et al., 2026a). We apply GRPO to the AAP-SFT planner with and without a diversity reward and compare each run with the same planner's plans slowed to equal travel, which separates gains in the reliability of the samples from gains due to cautious driving.

3 PRELIMINARIES

The Qwen-Drive-1.0 planner. Qwen-Drive-1.0 (Zhou et al., 2026) supports multiple tasks, among them end-to-end planning. For planning, a frozen 4B VLM encodes multi-view cameras over several frames, the navigation command, the ego status and the ego history. The VLM can be prompted to first write a driving rationale; we use its direct-planning mode. A 1B flow-matching planning head attends to the VLM's key-value cache and turns noise into a plan of 50 waypoints at 10 Hz.

Sampling and training the head. The head generates a plan x from noise ϵ along the path $x_t = t\,x + (1-t)\,\epsilon$. At each of ten steps $t = 0, 0.1, \ldots, 0.9$ it predicts the clean plan $\hat{x} = f_\theta(x_t, t, c)$ from the VLM cache c and moves towards it,

$$x_{t+0.1} = x_t + 0.1\,\frac{\hat{x} - x_t}{1-t}, \qquad (1)$$



Figure 3: Where sample variety comes from (one navhard scene, 16 samples). (a, c) 16 initial noises. (b, d) Step noise on the last three steps. AAP-SFT predicts per-step displacements summed into positions, so its step noise is smoothed out (d) instead of giving jitter (b).

so the last step returns $\hat{x}$. Samples differ either in the *initial noise* $\epsilon_k \sim \mathcal{N}(0, I)$ or in the *step noise* that perturbs x_t on the last three steps (Figure 3), and we take the plan from $\epsilon = 0$ as the single plan. Qwen-Drive-1.0 trains the head with the VLM frozen, first on logged drives to predict the clean plan (SFT), then with GRPO, whose rollouts add step noise on the last three steps and whose reward includes the NAVSIM v1 PDMS on navtrain.

The NAVSIM v2 navhard score. The extended PDM score (EPDMS) of NAVSIM v2 (Cao et al., 2025) scores what a plan leads to, not only how well it imitates a logged drive. It multiplies four hard terms (no at-fault collision, drivable area, driving direction and traffic light) with a weighted mean of soft terms (progress, time to collision, lane keeping and two comfort terms). The navhard split holds 450 real scenes and 5,462 synthetic *second-stage scenes* rendered from states a planner may reach four seconds later, and averages over 225 *evaluation groups*, each a real scene, its previous frame and their second-stage scenes, weighted by how close each starts to the plan's 4 s endpoint. Two consequences shape our analysis. Stopping satisfies every hard term and loses only progress, so caution alone scores well, and a plan's endpoint decides which second-stage scenes count, so samples that end in different places open headroom that a single plan cannot reach.

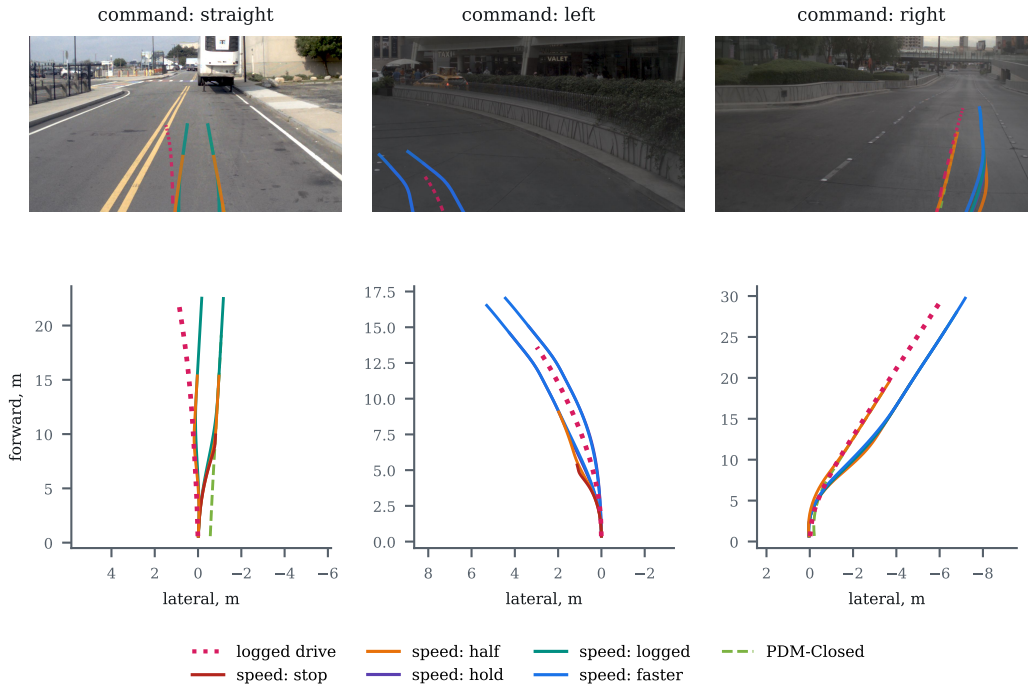


Figure 4: Admissible alternatives for three navtrain scenes, one per navigation command, each with six kept alternatives, on the front camera (top) and in BEV (bottom). The logged drive (red, dotted) is the only target without augmentation. The kept centreline alternatives are solid lines coloured by speed profile (legend), and the PDM-Closed plan is green and dashed.

4 Admissible Alternative Plans

Why the planning head needs more than the logged drive. The near-copies of Qwen-Drive-1.0 leave little headroom, since the best of its 16 samples, chosen in every evaluation group with the ground truth, reaches only 0.359 against 0.315 for its single plan. Step noise raises the best of 16 to 0.422, but only by making the plan jagged (Figure 3 (b)). Yet most scenes admit several plans the benchmark accepts, slower, faster, shifted within the lane or stopping, and we make those plans explicit training targets (Section 5).

Candidate plans. For each training scene, a generator without any learned model proposes candidates from the scene geometry (Figure 4). It offsets the lane centreline of the scene's metric cache by -1, 0 or $+1$ m, blended in over 2.5 s, and drives it with one of five jerk-limited speed profiles (stop, half the mean logged speed, hold the current speed, the logged speed profile, or 1.2 times the current speed, capped at 2 m/s above the logged maximum). The cached plan of the rule-based PDM-Closed planner (Dauner et al., 2023), extended from 4 to 5 s at constant speed and yaw rate, is one more candidate.

Admissibility. A candidate is admissible, and kept, only if it stays within NAVSIM's comfort bounds and the official scorer, with its human-penalty filter off so that the logged drive cannot excuse a fault, finds no at-fault collision, drivable-area, driving-direction, traffic-light or time-to-collision fault and a progress score of at least one half, so that it is safe and still makes progress. Candidates within 0.5 m of the logged drive or of a kept candidate are dropped as near-duplicates, and at most six are kept per scene, those with the highest EPDMS. The kept candidates are the admissible alternative plans (AAP) of the scene, and together with the logged drive they form its training targets.

5 Training the Planning Head

Stage 1, fine-tuning on admissible alternatives (AAP-SFT). We fine-tune only the planning head of Qwen-Drive-1.0, with the VLM frozen and the architecture, inputs and inference unchanged. For

a scene with VLM cache c, logged drive $x^{\log}$ and admissible alternatives $\mathcal{A}$, each training example draws its target y, which is $x^{\log}$ with probability $1-p$ and otherwise a uniformly chosen plan of $\mathcal{A}$ (the logged drive if $\mathcal{A}$ is empty), together with its own noise $\epsilon \sim \mathcal{N}(0, I)$ and flow time t drawn uniformly on $[0, 0.9]$, the interval the sampler of Eq. 1 visits. The head minimises

$$\mathcal{L}_{\mathrm{SFT}} = \mathbb{E}\left\| f_\theta(x_t, t, c) - y \right\|^2, \qquad x_t = t\, y + (1-t)\, \epsilon. \tag{2}$$

The head predicts the step-to-step displacements of the plan's positions, which are summed into waypoints, with headings kept absolute, and targets and loss are taken in this normalised representation, which keeps step noise smooth (Figure 3 (d)).

Stage 2, GRPO with a diversity reward. Starting from the AAP-SFT planner, we train the planning head alone with GRPO (Shao et al., 2024), keeping the VLM frozen. For each scene, a group of G rollouts $x^{(1)}, \ldots, x^{(G)}$ differs only in its initial noise, and each rollout is rewarded with the official EPDMS on navtrain, without its two-frame comfort term, which needs the previous frame's plan, multiplied by a diversity bonus,

$$r_k = \mathrm{EPDMS}\big(x^{(k)}\big)\left(1 + \lambda\, \frac{\min(d_k, D)}{D}\right), \tag{3}$$

where d_k is the distance of the rollout's 4 s endpoint from the group's mean endpoint. Setting $\lambda = 0$ gives the EPDMS reward alone. The advantage is $A_k = (r_k - \bar{r})/\max(\sigma, 0.05)$, with $\bar{r}$ the group mean and σ the standard deviation of all rewards in the batch, floored since near-tied groups would otherwise be over-weighted (Liu et al., 2025b). The update, which we call reward-weighted flow matching, applies the loss of Eq. 2 with each rollout as its own target, weighted by its positive advantage,

$$\mathcal{L}_{\mathrm{RL}} = \mathbb{E}\Big[\frac{1}{G}\sum_{k=1}^{G} \max(A_k, 0)\left\| f_\theta\big(x_t^{(k)}, t, c\big) - x^{(k)} \right\|^2\Big], \tag{4}$$

so rollouts below the group mean are not used. Section 6.4 also compares it with step-noise exploration as in Flow-GRPO (Liu et al., 2025a).

6 Experiments

We compare planners first by the single plan they submit (Table 2) and then by their 16 samples (Table 3), explain why the augmentation works, and then study reinforcement learning on top of it. Appendix A adds all settings, the inference cost, results over training seeds, ablations of the alternatives and more qualitative examples.

6.1 Experimental Setup

Data. We train on 15,000 command-balanced navtrain scenes. At least one admissible alternative survives in 91.9% of them (4.5 on average), with a median EPDMS of 0.936 and an endpoint spread of 2.55 m. Every navhard result uses all 5,912 scenes, the official two-stage scorer and direct planning, and navtest PDMS is reported as a secondary, one-stage benchmark. Our reproduction of Qwen-Drive-1.0's evaluation matches every published navtest row within 0.3 PDMS.

Implementation details. Stage 1 uses $p = 1/2$ and runs 2,000 steps, about one epoch. Each step takes 8 scenes, runs the frozen VLM once per scene and trains the head on 4 examples per scene, with AdamW at a learning rate of 10^{-5}, 50 warm-up steps, cosine decay and gradient clipping at 1, on one GPU. Predicting waypoints instead of displacements gives almost the same single-plan score (0.365 against 0.373). Stage 2 draws 8 scenes per step with $G = 16$ rollouts each, uses $\lambda = 0.2$ and $D = 2$ m, and evaluates Eq. 4 with two draws of t and ϵ per rollout. It runs 300 steps of AdamW at a constant learning rate of $2 \cdot 10^{-6}$ with gradient clipping at 1 and no KL term, and its first two steps only check that the rewards vary.

Measures. Next to the single plan a planner submits, we score 16 samples from 16 initial noises with the measures of Table 1. EPDMS evaluates a single plan, and these measures add which driving a planner imitates and which plans its samples cover.

References and controls. Besides published planners, we evaluate two naive planners, stopping in place and holding the current velocity, and two controls. Logged-drive fine-tuning trains the head as in Stage 1 on the logged drive alone ($p = 0$), and slowed plans replay a planner's own plans more slowly along unchanged paths.

Table 1: The six measures used throughout. The first four are official EPDMS on the full navhard split, and the last two describe how the samples drive.

Measure	Definition	What it tells
EPDMS	EPDMS of the submitted single plan ($\epsilon = 0$ for a flow-matching head)	The deployable score, with no sampling and no scorer
Mean of 16	Mean EPDMS over all 16 samples	The quality of the samples as a whole
Oracle best of 16	EPDMS when the best of the 16 samples is chosen in every evaluation group with the ground truth	The headroom within the samples, privileged and not deployable
Worst of 16	EPDMS when the worst of the 16 samples is chosen in every evaluation group	The floor, whether the planner ever produces unusable plans
Spread	Median over scenes of the mean distance of the 4 s endpoints from their centre	Whether the samples are distinct options or copies
Travel	Median over scenes of the mean 4 s path length of the 16 samples	Whether the samples drive or stop, and how cautiously

6.2 Single-Plan Evaluation

Table 2 lists every planner by the one plan it submits. Qwen-Drive-1.0 (SFT) scores 0.315, close to stopping in place (0.299), which satisfies every hard term of EPDMS. Its reasoning mode and its RL head do not raise this, so all our numbers are direct planning. The AAP-SFT planner's single plan scores 0.373 against 0.315 for Qwen-Drive-1.0 (0.351 on average over five training seeds), and it scores higher in more scenes than it scores lower, so the gain is not carried by a few scenes. Logged-drive fine-tuning, the same training on the logged drive alone, gives 0.286, so the alternatives, not the extra training, carry the gain. With one plan and no scorer, the AAP-SFT planner exceeds every published planner that generates one plan. It also exceeds two planners that pick among their own candidates with a learned scorer, MindDrive (Sun et al., 2025) and HAD-L (Yao et al., 2026b), and stays below those that select one of thousands of vocabulary trajectories with a learned scorer, GTRS (Li et al., 2025), ZTRS (Li et al., 2026d) and DriveSuprim (Yao et al., 2026a), or rescore perturbations of a 7B VLA plan, HiST-VLA (Wang et al., 2026). Our number therefore measures the planner's own driving, the plan it predicts directly, rather than the outcome of an external selection over a large candidate set, which can obscure how well the model itself drives.

Table 2: Planners by the one plan they submit: navhard two-stage EPDMS (0–1, all 5,912 scenes) and navtest PDMS ($\times 100$). Published rows are as reported by their papers and differ in candidate budget and metric version. Bold: the best value of each block. $\hookrightarrow$: derived from the row above. †Before NAVSIM's September 2025 metric fix (ours include it). ‡Evaluated on an earlier navhard release (244 real and 4,164 synthetic scenes) rather than the 5,912 scenes used by the other rows. ¶navhard as reported by ZTRS. *AAP-SFT's own plans replayed slower, a control for caution, not a planner.

Planner	how the plan is produced	navhard EPDMS	navtest PDMS
Privileged reference			
Privileged expert (Nguyen et al., 2026)†	privileged expert, ground-truth state, one plan	0.513	94.5
Published: many candidates, the pick of a learned scorer			
HiST-VLA (Wang et al., 2026)†	7B VLA plan perturbed into candidates (count not stated), learned scorer	**0.509**	–
MindDrive (Sun et al., 2025)†	N k-means anchors refined into N candidates (N not stated), learned scorer	0.305	88.9
HAD-L (Yao et al., 2026b)	20 anchors, top 2 expanded to 50 candidates, learned scorer	0.323	89.9
DriveSuprim (Yao et al., 2026a)¶‡	8,192 vocabulary, learned scorer, 256 kept for fine scoring	0.447	**93.5**
ZTRS (Li et al., 2026d)‡	16,384 vocabulary, learned scorer trained by RL, no demonstrations	0.462	–
GTRS-Dense (Li et al., 2025)†	8,192 vocabulary, learned scorer	0.453	–
GTRS-E, ensemble of six (Li et al., 2025)†	8,192 vocabulary + 100 diffusion proposals, six learned scorers	0.494	–
Published: one generated plan			
Latent TransFuser (LTF) (Chitta et al., 2023; Nguyen et al., 2026)†	regression, one plan	0.231	83.8
GTRS diffusion generator alone (Li et al., 2025)†	100 diffusion proposals, no scorer	0.256	–
LTFv6 + LEAD (Nguyen et al., 2026)†	TransFuser v6 regression, one plan	0.314	**86.4**
HiST-E2E (Wang et al., 2026)†	VLA commands into a DiffusionDrive planner, one plan	**0.342**	–
Naive planners (our evaluation)			
Constant velocity	hold the current velocity vector for the horizon	0.119	24.0
Stop in place	brake and hold the current position	**0.299**	**61.8**
Qwen-Drive-1.0 and our fine-tuned planning heads (our evaluation)			
Qwen-Drive-1.0 (SFT) (Zhou et al., 2026)	4B VLM + 1B flow-matching head, one plan	0.315	**87.9**
$\hookrightarrow$ Logged-drive fine-tuning (ours, control)	head fine-tuned on the logged drive alone	0.286	86.1
AAP-SFT (ours)	the same, head fine-tuned on logged drive + admissible alternatives	0.373	86.5
$\hookrightarrow$ GRPO on AAP-SFT (ours)	+ GRPO with EPDMS reward only	0.358	76.7
$\hookrightarrow$ GRPO on AAP-SFT (ours)	+ GRPO with EPDMS and diversity rewards	0.375	80.0
$\hookrightarrow$ AAP-SFT slowed plans (ours)*	the same plans replayed slower, no training	**0.416**	80.0

Planners under slowed execution. Because stopping satisfies every hard term of EPDMS, a lower driving speed alone can raise the navhard score. To separate this effect from the effect of the alternatives, we execute the plans of both planners along their unchanged paths at a reduced speed, the same fraction of each plan's own speed, without retraining. At 0.87 of the original speed, the AAP-SFT planner scores 0.399 against 0.338 for Qwen-Drive-1.0, and at 0.70 it scores 0.422 against 0.378. The advantage therefore persists at matched speed. On navtest, a one-stage benchmark that weights progress more heavily, the AAP-SFT planner scores slightly below Qwen-Drive-1.0 (86.5 against 87.9 PDMS).

6.3 Diversity and Coverage of the Samples

Sample diversity and coverage. Table 3 measures the 16 samples of each planner. Qwen-Drive-1.0's samples lie within 0.11 m of their centre (Figure 1 (a)), and choosing the best of them in every scene reaches only 0.359. The AAP-SFT planner's samples spread 2.47 m (Figure 1 (b) and Figure 6 in the appendix), and the best of 16 reaches 0.614. This selection uses the ground truth and is not deployable. For reference, the privileged expert reported by LEAD (Nguyen et al., 2026), which like PDM-Closed (Dauner et al., 2023) plans with ground-truth state, reaches 0.513 (Table 2). It shows that 16 samples of a 1B planning head already contain much better plans, where learned scorers search thousands of vocabulary trajectories. The gap grows with the number of samples (Figure 5). Logged-drive fine-tuning gives near-copies (0.20 m) and little headroom (0.364), so the options come from the alternatives, not from fine-tuning itself.

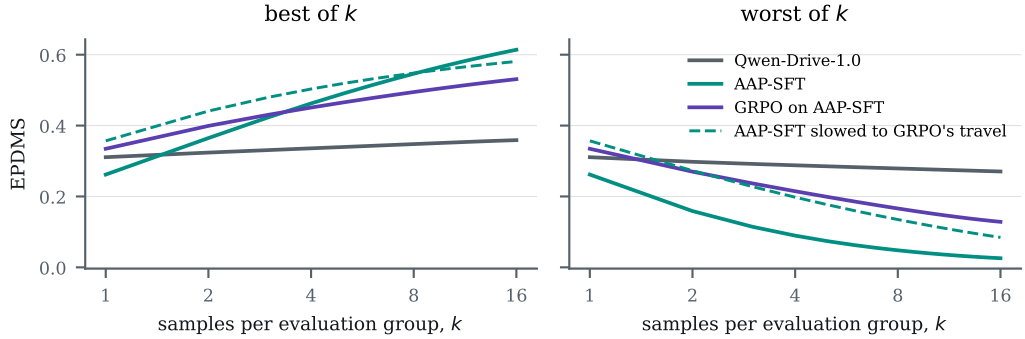


Figure 5: Best of k (left) and worst of k (right) samples, chosen per evaluation group with the ground truth, as the exact expectation over random k-subsets of the 16 samples. Dashed: the AAP-SFT planner's plans slowed to the travel of GRPO on AAP-SFT, a control for caution.

Initial noise versus step noise. A planner's samples can differ through the initial noise of the flow or through noise injected during its integration steps (Figure 3). For Qwen-Drive-1.0, step noise on the last three steps spreads the samples by 1.76 m and raises the best of 16 to 0.422, but the plans become jagged, and their 4 s path length grows to 40.0 m against 16.3 m under initial noise. For the AAP-SFT planner the opposite holds. Step noise barely moves its plans (0.25 m, best of 16 0.385), because the summed displacements smooth it out, while initial noise yields 2.47 m and 0.614. The alternatives therefore teach the head distinct options, and the initial noise selects among them.

What the alternatives change in driving. The per-term scores of the two single plans show where the gain comes from. On the second-stage scenes, which start where a plan ends, the AAP-SFT planner collides less (no at-fault collision 0.886 against 0.814), leaves the drivable area less (0.796 against 0.741) and keeps a safer time to collision (0.862 against 0.775). It pays with progress, on real scenes (0.778 against 0.823) and on second-stage scenes (0.740 against 0.831), and with consistency between consecutive frames (two-frame comfort 0.591 against 0.709). The augmentation thus trades progress and temporal consistency for safety on the scenes a plan leads to.

6.4 Second-Stage Reinforcement Learning

Single plan and mean sample. Table 3 reports GRPO on the AAP-SFT planner (Stage 2, Section 5) with the EPDMS reward alone and with the diversity reward. Because slower plans score higher on navhard (Section 6.2), each run is compared with the AAP-SFT planner's own plans slowed to the

Table 3: Analysis of the single plan and the 16 samples each planner predicts, on the full navhard split with direct planning. Columns follow Table 1; the mean of 16 is reported as mean ± std, where std is the standard deviation of the 16 samples' scores. GRPO rows are Stage 2 of Section 4 with the reward shown. *The AAP-SFT planner's own plans, slowed without training to the travel of the GRPO run above, a control for how much of GRPO's gain is caution.

Planner	EPDMS, single plan	EPDMS, mean ± std (16 samples)	EPDMS, oracle best of 16	EPDMS, worst of 16	spread (m)	travel (m)
Qwen-Drive-1.0 (SFT)	0.315	0.311 ± **0.004**	0.359	**0.270**	0.11	16.3
Logged-drive fine-tuning (ours, control)	0.286	0.283 ± 0.011	0.364	0.215	0.20	17.0
AAP-SFT (ours)	0.373	0.262 ± 0.061	**0.614**	0.026	2.47	15.6
↪ GRPO, EPDMS reward (ours)	0.358	0.326 ± 0.033	0.504	0.136	1.14	8.8
slowed AAP-SFT plans*	0.392	**0.369** ± 0.047	0.543	0.140	0.95	8.1
↪ GRPO, EPDMS + diversity reward (ours)	0.375	0.335 ± 0.026	0.531	0.128	1.10	10.1
slowed AAP-SFT plans*	**0.416**	0.357 ± 0.059	0.581	0.085	1.23	9.5

same travel. Neither run improves the single plan (0.358 and 0.375 against 0.373). Both raise the mean of 16 (0.326 and 0.335 against 0.262), but the slowed plans of the AAP-SFT planner reach comparable or higher values without training (0.369 and 0.357). The increase in the mean is therefore explained by shorter plans (8.8 and 10.1 m in 4 s against 15.6 m, Figure 1 (c)).

Reliability of the samples. At matched travel, GRPO with the diversity reward yields more consistent samples than slowing the AAP-SFT planner. The standard deviation of its 16 samples' scores is 0.026 against 0.059 for the slowed plans. Its worst of 16 is 0.128 against 0.085, and its worst of k stays above the slowed plans from four samples on (Figure 5). With the EPDMS reward alone, the standard deviation also decreases (0.033 against 0.047) but the worst of 16 does not improve (0.136 against 0.140). Qwen-Drive-1.0 has the highest worst of 16 (0.270) because its samples are near-copies of one plan. Samples that cover distinct options also include options that fail.

Coverage of the samples. GRPO reduces the headroom among the samples. The best of 16 falls from 0.614 for the AAP-SFT planner to 0.504 with the EPDMS reward and 0.531 with the diversity reward, below the slowed plans at the same travel (0.543 and 0.581), so part of this loss is not explained by shorter plans. The spread shrinks from 2.47 m to 1.14 and 1.10 m, close to that of the slowed plans (0.95 and 1.23 m), so the smaller spread largely follows from the shorter plans. The diversity reward does not widen the spread when the rollouts differ only in their initial noise, but it does when exploration uses step noise, as in Flow-GRPO (Liu et al., 2025a), where the spread grows from 1.14 to 2.13 m and the best of 16 from 0.572 to 0.581.

7 Conclusion and Limitations

We fine-tune the planning head of Qwen-Drive-1.0 on admissible alternative plans generated from scene geometry and filtered with the EPDMS scorer, while the VLM, inputs, and inference remain unchanged. On navhard, the single plan scores 0.373 against 0.315, and the advantage persists when both planners execute their plans at the same reduced speed. The 16 samples of the fine-tuned head become distinct options rather than near-copies, and their best of 16 rises from 0.359 to 0.614. A second stage of GRPO with a diversity reward makes the samples more reliable than slower driving alone does, at the cost of part of these options. Reporting the single plan together with the diversity, coverage and travel of the samples shows what driving a planner has learned, and we encourage the community to analyse planners along these lines rather than by benchmark scores alone.

Limitations and future work. We use the direct-planning mode of Qwen-Drive-1.0 and do not study its driving rationale, whose sampling adds a further source of diversity and coverage that must stay aligned with the plan that follows. The alternatives are admitted by a short-horizon check, the 4 s EPDMS on navtrain, which does not account for errors that compound over longer horizons, and longer-horizon or closed-loop criteria may yield better targets. The diverse samples of the AAP-SFT planner give GRPO the variation within each group that it needs, yet our second stage does not yet turn this diversity into a significantly better single plan or reach the headroom of the best of 16. This likely requires rewards that distinguish scenes where caution is warranted from scenes where progress is, and training data that samples such corner cases more often. Selecting among the samples without the ground truth could therefore raise a planner's score, and since navtrain labels reward progress while navhard's second-stage scenes reward plans that remain safe, a selector trained on signals that correlate with two-stage or closed-loop outcomes is a natural next step.

AI USE STATEMENT

An AI coding assistant (Claude) was used under the authors' direction and review to write evaluation, training and analysis code, to run and monitor experiments, to help design experiments, to search for and check related work, and to draft and polish parts of the text. Every reported number is generated from the official scorer's output files by scripts in our code, which we will release upon acceptance; no number was typed or edited by hand. The authors take full responsibility for the content.

REPRODUCIBILITY STATEMENT

We use the public Qwen-Drive-1.0 weights and inference unchanged and the official NAVSIM v1.1 and v2 scorers. Upon acceptance we will release code containing the construction of the navhard and navtrain input files from NAVSIM's scene loader; the sampling code for initial and step noise with its correctness checks; a wrapper around the official two-stage scorer that records every evaluation group's score without changing the scorer; the alternative-plan generator; the fine-tuning (AAP-SFT) and GRPO code; and the scripts that generate every table, figure and number in the paper from the result files. The seeds of the initial and step noise are fixed per sample and stored with every prediction.

ETHICS STATEMENT

The work evaluates an open driving model on a public benchmark and involves no human subjects or new data collection. Offline benchmark scores do not establish on-road safety, and the headroom we report is an upper bound chosen with the benchmark's ground truth; it should not be read as the performance of a deployable system.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 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 *Proceedings of The 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4709–4722. PMLR, 2025. arXiv:2506.04218.

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 100 of *Proceedings of Machine Learning Research*, pp. 86–99. PMLR, 2019.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023. arXiv:2303.04137.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE TPAMI*, 45(11):12878–12895, 2023. doi: 10.1109/TPAMI.2022.3200245.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Proceedings of The 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1268–1281. PMLR, 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, pp. 28706–28719, 2024. doi: 10.52202/079017-0902.

DeepSeek-AI. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Abdulwahab Felemban, Nussair Hroub, Jian Ding, Eslam Abdelrahman, Xiaoqian Shen, Abduallah Mohamed, and Mohamed Elhoseiny. iMotion-LLM: Instruction-conditioned trajectory generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2710–2720, 2026.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862, 2023. doi: 10.1109/CVPR52729.2023.01712.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, Yin Zhou, James Guo, Dragomir Anguelov, and Mingxing Tan. EMMA: End-to-end multimodal model for autonomous driving. *Transactions on Machine Learning Research (TMLR)*, 2025. arXiv:2410.23262.

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 (ICCV)*, pp. 8306–8316. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00766.

Hongchen Li, Tianyu Li, Jiazhi Yang, Mingyang Shang, Gaoqiang Wu, Caojun Wang, Haochen Tian, Zengrong Lin, Zhihui Hao, XianPeng Lang, Jia Hu, and Hongyang Li. PlannerRFT: Reinforcing diffusion planners through closed-loop and sample-efficient fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24929–24938, 2026a.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. DriveVLA-W0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 7890–7911, 2026b.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. ReCogDrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2026c. arXiv:2506.08052.

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

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M. Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025.

Zhenxin Li, Nadine Chang, Wenhao Yao, Xinglong Sun, Zi Wang, Maying Shen, Jingde Chen, Jingyu Song, Kailin Li, Zuxuan Wu, Shiyi Lan, and Jose M. Alvarez. Zero-human demonstration end-to-end autonomous driving with trajectory scorer. In *Computer Vision – ECCV 2026*, pp. 91–110. Springer Nature Switzerland, 2026d. doi: 10.1007/978-3-032-37132-4_6.

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 (CVPR)*, pp. 12037–12047, 2025. doi: 10.1109/CVPR52734.2025.01124.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.02747.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, pp. 45650–45685, 2025a. doi: 10.52202/085713-1362.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2209.03003.

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. In *Conference on Language Modeling (COLM)*, 2025b.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. LEAD: Minimizing learner-expert asymmetry in end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39775–39785, 2026.

NVIDIA, Yulong Cao, Riccardo de Lutio, Sanja Fidler, Guillermo Garcia Cobo, Zan Gojcic, Maximilian Igl, Boris Ivanovic, Peter Karkus, Janick Martinez Esturo, Marco Pavone, Aaron Smith, Ellie Tanimura, Michal Tyszkiewicz, Michael Watson, Qi Wu, and Le Zhang. AlpaSim: A modular, lightweight, and data-driven research simulator for autonomous driving. https://github.com/NVlabs/alpasim, 2025a.

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

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

Bin Sun, Yaoguang Cao, Yan Wang, Rui Wang, Jiachen Shang, Xiejie Feng, Jiayi Lu, Jia Shi, Shichun Yang, Xiaoyu Yan, and Ziying Song. MindDrive: An all-in-one framework bridging world models and vision-language model for end-to-end autonomous driving. *arXiv preprint arXiv:2512.04441*, 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. In *Proceedings of The 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4698–4726. PMLR, 2024.

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

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. GoalFlow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1602–1611, 2025. doi: 10.1109/CVPR52734.2025.00157.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M. Alvarez, and Zuxuan Wu. DriveSuprim: Towards precise trajectory selection for end-to-end planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(14):11910–11918, 2026a. doi: 10.1609/aaai.v40i14.38178.

Wenhao Yao, Xinglong Sun, Zhenxin Li, Shiyi Lan, Zi Wang, Jose M. Alvarez, and Zuxuan Wu. HAD: Combining hierarchical diffusion with metric-decoupled RL for end-to-end driving. *arXiv preprint arXiv:2604.03581*, 2026b.

Xin Zhou, Zongchuang Zhao, Zhibo Yang, Mingsheng Li, Humen Zhong, Shuai Bai, Du Chu, Ruizhe Chen, Zhaohai Li, Jun Tang, Qiuyue Wang, Mingkun Yang, Jiazhao Zhang, Dayiheng Liu, Dingkang Liang, and Xiang Bai. Qwen-Drive-1.0: An initial step towards a vision-language foundation model for autonomous driving. *arXiv preprint arXiv:2609.00111*, 2026.

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. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, pp. 27920–27956, 2025.

A SUPPLEMENTARY MATERIAL

All results use the full navhard split (5,912 scenes), the official two-stage scorer and direct planning. Single plan: $\epsilon = 0$. Mean and oracle best of 16: 16 initial noises, best chosen per evaluation group with the ground truth.

A.1 IMPLEMENTATION DETAILS

Table 4 lists every setting of the alternative generator (Section 4) and of both training stages (Section 5), as used in the runs of the main text.

Table 4: Settings of the alternative generator and of both training stages.

Setting	Value
Training scenes	15,000 command-balanced navtrain scenes
Lateral offsets of the lane centreline	-1, 0, $+1$ m, blended in over 2.5 s (quintic)
Speed profiles	stop, $0.5\times$ mean logged speed, hold, logged, $\min(1.2v_0, v_{\max}^{\log}+2\,\mathrm{m/s})$
Speed controller	acceleration in $[-3, 1.5]$ m/s^2, |jerk| ≤ 2 m/s^3, 50 poses at 10 Hz
Extra candidate	cached PDM-Closed plan, extended from 4 to 5 s
Filters	comfort bounds; no collision, drivable-area, direction, traffic-light, TTC fault; progress ≥ 0.5
Deduplication, cap	drop within 0.5 m ADE of the logged drive or a kept plan; at most 6, highest EPDMS
Stage 1	$p = 0.5$, 2,000 steps, 8 scenes $\times$ 4 examples, AdamW, lr 10^{-5}, 50 warm-up, cosine to 0.1, clip 1
Flow time, noise	$t \sim \mathcal{U}[0, 0.9]$, $\epsilon \sim \mathcal{N}(0, I)$, displacement targets
Stage 2	8 scenes $\times$ $G = 16$ rollouts, $\lambda = 0.2$, $D = 2$ m, 2 flow draws per rollout
Optimisation	300 steps, AdamW, constant lr $2 \cdot 10^{-6}$, clip 1, no KL, 2 gate steps without update
Hardware	one H100 (80 GB) per run

A.2 INFERENCE COST

Table 5 times one scene at a time on one GPU after warm-up. AAP-SFT keeps the architecture of Qwen-Drive-1.0 and therefore its cost, and 16 samples add only planning-head time because they share one VLM prefill.

Table 5: Inference cost per scene of the AAP-SFT planner in milliseconds, median over 100 navhard scenes after 10 warm-up scenes, batch 1, direct planning, bf16 with FlashAttention-2 on one NVIDIA H100 80GB HBM3, with a 4.54B VLM and a 1.04B planning head and a peak GPU memory of 13.9 GB. Qwen-Drive-1.0 has the same architecture, and its GPU time is 418 ms for the single plan and 654 ms for 16 samples. The 16 samples share one VLM prefill. GPU time is the VLM prefill plus the planning head, and CPU preprocessing can overlap with it.

Stage	Time (ms)
CPU input preprocessing, 12 images and prompt	404
VLM prefill, key-value cache	164
Planning head, 10 flow steps, 1 sample	256
Planning head, 10 flow steps, 16 samples batched	495
GPU time, single plan	420
GPU time, 16 samples	659

A.3 Training Seeds

Table 6 retrains AAP-SFT with four further seeds and everything else unchanged. The oracle best of 16 is stable across seeds, while the single plan varies more.

Table 6: AAP-SFT trained with five seeds, identical otherwise. Qwen-Drive-1.0: single plan 0.315, mean of 16 0.311, best of 16 0.359.

Planner	single plan	mean of 16	oracle best of 16
AAP-SFT, training seed 0 (main text)	0.373	0.262	0.614
AAP-SFT, training seed 1	0.356	0.262	0.613
AAP-SFT, training seed 2	0.314	0.244	0.591
AAP-SFT, training seed 3	0.350	0.260	0.610
AAP-SFT, training seed 4	0.360	0.244	0.597

A.4 Ablations of the Alternatives

Each ablation changes one setting of AAP-SFT and keeps the rest. Table 7 varies the fraction of training targets that are alternatives, and Table 8 which alternatives are used.

Table 7: Fraction p of training examples whose target is an alternative. Everything else as AAP-SFT.

Fraction of alternatives	single plan	mean of 16	oracle best of 16
$p = 0$ (logged-drive fine-tuning)	0.286	0.283	0.364
$p = 0.25$	0.304	0.261	0.557
$p = 0.5$ (AAP-SFT)	0.373	0.262	0.614
$p = 0.75$	0.387	0.259	0.634
$p = 1$	0.351	0.253	0.615

Table 8: Which alternatives are used, with $p = 0.5$. Predicting waypoints instead of displacements gives a single plan of 0.365.

Training targets	single plan	mean of 16	oracle best of 16
All candidate families (AAP-SFT)	0.373	0.262	0.614
Lateral offsets only	0.370	0.266	0.630
Speed profiles only	0.342	0.272	0.595
Without the PDM-Closed plan	0.355	0.263	0.612
PDM-Closed plan only	0.284	0.250	0.502
Without the rule filter	0.347	0.233	0.616
One alternative per scene, highest EPDMS	0.317	0.239	0.555
One alternative per scene, random	0.373	0.264	0.617
Qwen-Drive-1.0's published loss	0.325	0.240	0.634

A.5 Qualitative Examples

Figure 6 shows five more navhard scenes as in Figure 1, and Figure 7 the two noise sources of Figure 3 with their top-down views.

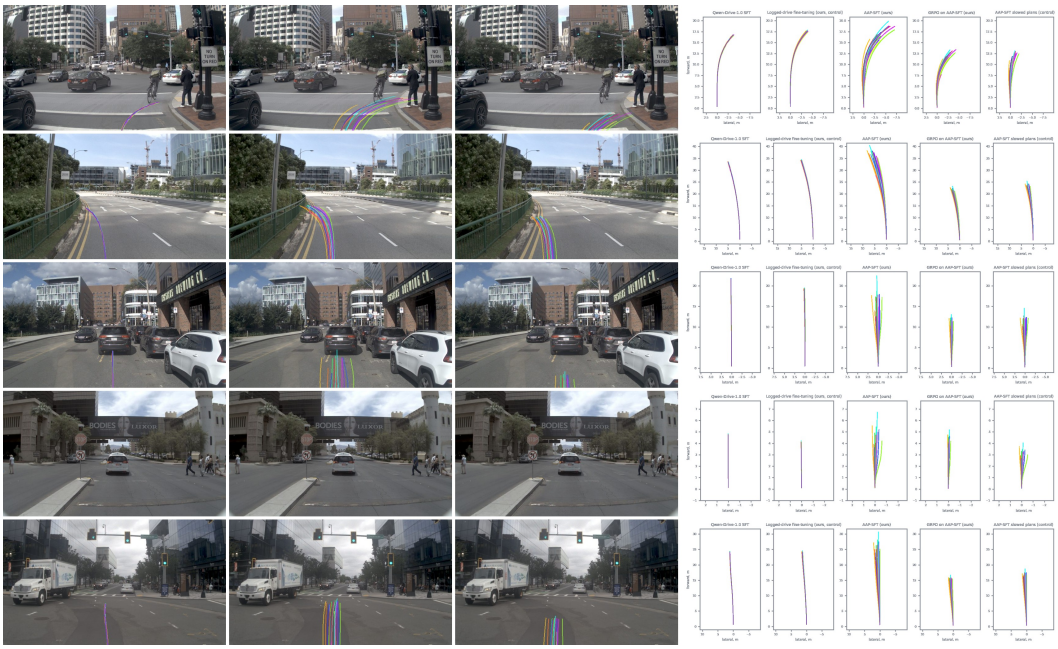


Figure 6: Five more navhard scenes, 16 samples from different initial noises. Columns: Qwen-Drive-1.0, AAP-SFT and GRPO on AAP-SFT on the front camera, then the samples in BEV for Qwen-Drive-1.0, logged-drive fine-tuning, AAP-SFT, GRPO on AAP-SFT and AAP-SFT slowed to GRPO's travel.

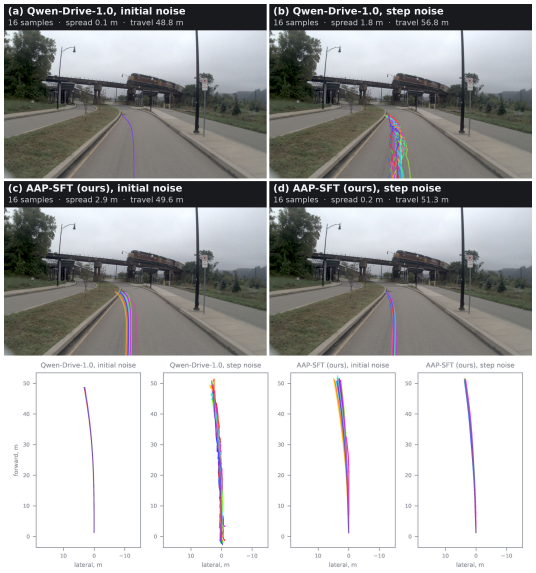


Figure 7: Initial noise against step noise on the last three steps, for Qwen-Drive-1.0 and AAP-SFT, one navhard scene, 16 samples.