# FPD-DRIVE: FUTURE POLICY DISTILLATION FOR END-TO-END AUTONOMOUS DRIVING

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

## ABSTRACT

End-to-end autonomous driving models are typically trained on real-world driving logs, which record only the trajectory actually executed in each scene. This provides limited supervision for learning alternative valid behaviors. Multi-target supervision can mitigate this limitation, but broader trajectory coverage makes safe candidate selection more challenging. We propose FPD-Drive, an end-to-end planner that distills privileged future knowledge into a visual driving policy to improve both multi-trajectory supervision and safety selection. Specifically, we introduce Future-Privileged Multi-Target Distillation to transfer diverse, high-quality trajectory targets from a teacher with access to recorded future observations to the student. These targets extend supervision beyond the single recorded human trajectory. For candidate selection, we further introduce Safety-Aware Score Correction to correct ranking errors between nearby safe and unsafe candidates and raise underestimated safety scores of safe candidates. At deployment, the student requires neither future observations nor teacher inference. Our base model achieves 94.9 PDMS on NAVSIM v1 *navtest* and 55.9 EPDMS on NAVSIM v2 *navhard*. The scaled FPD-Drive-Pro improves these scores to 95.5 PDMS and 58.1 EPDMS, respectively, achieving SOTA performance among the compared methods on both benchmarks. Zero-shot closed-loop evaluation on HUGSIM further yields an HD-Score of 45.1 and RC of 55.3.

## 1 INTRODUCTION

End-to-end autonomous driving aims to learn planning policies directly from sensor observations (Hu et al., 2023; Jiang et al., 2023). Advances in visual representation learning and generative trajectory modeling have enabled planners to produce and evaluate multiple candidate trajectories for a driving scene (Kirby et al., 2026; Liao et al., 2025; Xing et al., 2025). Effective planning requires both generating valid alternatives and reliably selecting a safe trajectory for execution. However, learning these capabilities from human drivers is challenging, because each logged driving clip used for training typically provides only one executed trajectory and its associated visual observations. As a result, it provides limited guidance for learning and evaluating alternative trajectories.

A recorded human trajectory represents one choice among several potentially acceptable driving behaviors. For example, different deceleration timings or speed profiles may be valid in the same scene, although only one is recorded. Closeness to the demonstration therefore cannot fully determine an alternative trajectory's quality. To supervise such alternatives, recent methods use predefined trajectory vocabularies or teachers pretrained on rule-generated targets (Wang et al., 2026c; Ang et al., 2026; Chen et al., 2026a). These approaches broaden supervision, but constructing alternatives that are both diverse and scene-appropriate remains challenging. Predefined vocabularies and handcrafted generation rules can restrict the range of candidate behaviors covered by supervision. And recorded future observations reveal how the scene's structure and evolution can help guide planning alternative trajectories. These motivates a privileged teacher that uses these observations to generate additional trajectory targets.

To benefit from broader trajectory coverage, the planner should also learn to select a safe candidate reliably. Planning metrics provide supervision beyond imitation to help select candidates, but accurate safety assessment remains challenging (Li et al., 2024; Kirby et al., 2026). Geometrically similar trajectories can differ sharply in safety metrics, and a scorer may favor an unsafe candidate



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.

over a nearby safe alternative. Safe candidates may also be overlooked if the scorer underestimates their safety. These two types of scoring errors motivate targeted score correction to distinguish nearby safe and unsafe candidates and raise the scores of underestimated safe alternatives.

We propose FPD-Drive, a future policy distillation framework that enriches trajectory supervision with privileged future knowledge while jointly improving candidate selection. The teacher and student share a coarse-to-fine visual planner that refines trajectory proposals through residual diffusion. During training, the teacher uses recorded future observations to guide trajectory refinement and scoring. We then evaluate its candidates using planning metrics and select high-quality, geometrically diverse trajectory targets. *Future-Privileged Multi-Target Distillation* transfers these targets to the student, extending supervision beyond the recorded demonstration. For candidate selection, *Safety-Aware Score Correction* uses evaluated safety labels for the student's own trajectories to correct scoring errors. It penalizes highly ranked unsafe candidates relative to nearby safe alternatives and increases confidence in underestimated safe candidates.

The deployed student requires neither future observations nor teacher inference. FPD-Drive achieves state-of-the-art performance on NAVSIM v1 and v2. Zero-shot evaluation on HUGSIM further supports its transfer to closed-loop driving. Our contributions are threefold:

- We propose FPD-Drive, which uses recorded future observations to help a teacher generate additional high-quality trajectories for distilling a visual student. Only the student is needed at deployment.
- We introduce *Future-Privileged Multi-Target Distillation*, which transfers diverse, high-quality trajectories from a future-conditioned teacher to the student, extending supervision beyond the single recorded demonstration.
- We develop *Safety-Aware Score Correction* to correct the relative safety scores of unsafe–safe pairs and raise underestimated safety scores for safe trajectories.

## 2 Related Work

**End-to-end autonomous driving.** End-to-end autonomous driving learns planning policies directly from sensor observations. Research on planning representations includes multimodal sensor fusion (Prakash et al., 2021) and integrating perception, prediction, and planning (Hu et al., 2023; Jiang et al., 2023). To account for multiple acceptable driving behaviors, probabilistic and generative planners model distributions over possible trajectories (Jiang et al., 2026; Liao et al., 2025; Xing et al., 2025). ResAD (Zheng et al., 2026) further improves trajectory representations through normalized residual modeling to address imbalanced optimization (Zheng et al., 2026). Beyond generation, metric-based supervision and generalized trajectory evaluation address reliable selection among alternative behaviors (Li et al., 2024; 2025c; Kirby et al., 2026). World models and world-action models (WAMs) introduce predictive information by learning how driving scenes evolve. Visual forecasting (Wang et al., 2024; Gao et al., 2024) and latent prediction (Li et al., 2025a; Wang et al., 2026b) represent future scene dynamics at different levels. Predicted future states support candidate evaluation (Li et al., 2025b; Zheng et al., 2025; Zhong et al., 2026), while joint world–action modeling couples future scene prediction with trajectory generation (Zhang et al., 2025; Liu et al., 2026; Wang et al., 2026d). Vision-language-action (VLA) models draw on pretrained semantic knowledge and language understanding to support planning (Xu et al., 2024; Shao et al., 2024; Hwang et al., 2025). These methods connect semantic scene understanding with driving actions by conditioning trajectory generation on visual-language representations (Zhou et al., 2026b; Li et al., 2026; Wang et al., 2026e). Graph-based visual question answering connects scene understanding with driving decisions (Sima et al., 2025), while reasoning-augmented policies integrate semantic reasoning with action generation (Zhou et al., 2025; NVIDIA et al., 2025). To balance planning performance and inference latency, FPD-Drive adopts a coarse-to-fine visual planner as the common architecture for the teacher and student. It extends DrivoR with a DiT module that refines the regressed trajectories. (Kirby et al., 2026).Furthermore, recorded future observations enrich trajectory supervision through a privileged teacher during training.

**Distillation for autonomous driving.** Knowledge distillation and privileged learning have attracted growing interest in autonomous driving. Teachers with access to ground-truth road structure

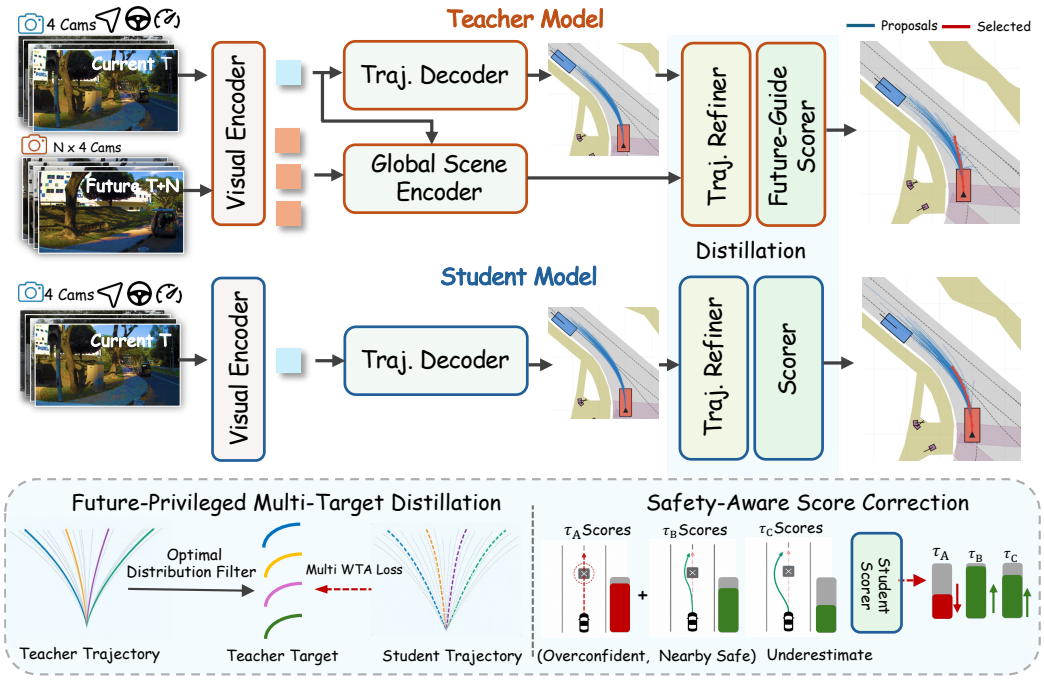



Figure 1: **Overview of FPD-Drive.** The teacher–student framework employs a common coarse-to-fine architecture comprising a visual encoder, a proposal decoder, a diffusion refiner, and a scorer (Sec. 3.1). Recorded future observations additionally guide the teacher's refinement and scoring (Sec. 3.2). **Future-Privileged Multi-Target Distillation** transfers high-quality, geometrically diverse trajectories selected from evaluated teacher candidates to supervise the student's proposal decoder and refiner (Sec. 3.3). **Safety-Aware Score Correction** uses evaluated student candidates to penalize highly ranked unsafe trajectories relative to nearby safe alternatives and increase confidence in underestimated safe candidates (Sec. 3.4). Only the student is used at deployment.

and traffic-participant states provide trajectory targets for sensor-based students (Chen et al., 2020; Zhang et al., 2021; He et al., 2026). Feature distillation also transfers scene and planning representations from these teachers (Zhang et al., 2023; Jia et al., 2023; Yu et al., 2025). Recent VLA methods condition teachers on privileged future information and distill token distributions into student driving policies (Cao et al., 2026; Dou, 2026). To broaden trajectory, Curious-VLA (Chen et al., 2026a) expands training data with trajectories generated by the ReCogDrive-VLA Li et al. (2026). Some concurrent methods explore multi-target trajectory supervision using predefined trajectory vocabularies (Wang et al., 2026c; Sun et al., 2026) or teacher models pretrained on rule-generated trajectories (Ang et al., 2026). Without relying on predefined trajectories or vocabulary, FPD-Drive combines a future-conditioned teacher with multi-target trajectory policy to distill a student. We further introduce Safety-Aware Score Correction to improve safety assessment and candidate ranking.

## 3 Method

**Overview.** Fig. 1 illustrates the overall architecture of FPD-Drive. At planning time $t$, the input $o_t$ consists of four current camera views, the current ego-state, and a navigation command. The planner generates $M = 64$ candidate trajectories and selects one for execution. Each trajectory contains $H = 8$ future poses, represented as $\tau = \{(x_h, y_h, \psi_h)\}_{h=1}^{H}$ in the ego coordinate frame at time $t$. Here, $(x_h, y_h)$ and $\psi_h$ denote position and heading, respectively. Training clips additionally provide a demonstrated trajectory $\tau^{\text{gt}}$ and recorded future camera observations $\mathcal{F}_t$. The future observations are used as privileged teacher inputs and are unavailable to the student at deployment.

### 3.1 COARSE-TO-FINE TRAJECTORY MODELING

We separate trajectory generation into scene-conditioned proposals and residual refinement. As illustrated in Fig. 1, the proposal decoder provides a finite set of starting trajectories, which may cluster within a limited region of trajectory space. A diffusion refiner then models continuous adjustments around these proposals to improve trajectory quality and broaden candidate coverage. This provides more alternatives for selection under different driving preferences. We adopt this coarse-to-fine structure as the common backbone for the teacher and student.

**Trajectory proposals.** Following DrivoR (Kirby et al., 2026), both the teacher and student encode current multi-view images into scene register tokens. We initialize $M$ trajectory tokens by adding the ego-state and navigation embedding $\mathbf{e}$ to learned queries $\{\mathbf{q}_m\}_{m=1}^{M}$. A transformer decoder updates these tokens using the scene tokens $\mathbf{h}$ and predicts the proposals:

$$\mathcal{B} = \{b_m\}_{m=1}^{M} = D\Big(\{\mathbf{q}_m + \mathbf{e}\}_{m=1}^{M}, \mathbf{h}\Big), \qquad b_m \in \mathbb{R}^{H\times 3}. \tag{1}$$

**Residual diffusion.** Drawing on residual trajectory modeling in ResAD (Zheng et al., 2026) and proposal-based diffusion refinement in ChainFlow-VLA (Wang et al., 2026e), we use a Diffusion Transformer (DiT) to predict residual corrections to coarse trajectories. For a proposal $b_m$ and its training target $\tilde{y}_m$, we define the residual $r_m = \tilde{y}_m - \mathrm{sg}(b_m)$, where sg denotes the stop-gradient operator. At a sampled diffusion step $k$, we construct:

$$z_{k,m} = \sqrt{\bar{\alpha}_k}\, r_m + \sqrt{1-\bar{\alpha}_k}\, \epsilon_m, \qquad \epsilon_m \sim \mathcal{N}(0, \mathbf{I}), \tag{2}$$

where $\bar{\alpha}_k$ is the cumulative noise schedule. A DiT predicts the noise as $\hat{\epsilon}_m = \epsilon_\theta(z_{k,m}, k; \mathbf{h}, \mathbf{e})$. Its blocks apply self-attention over trajectory poses and cross-attention to scene tokens. The refined trajectory $\hat{\tau}_m$ is reconstructed from proposal $b_m$ as follows, and will be send to the scorer.

$$\hat{\tau}_m = \mathrm{sg}(b_m) + \frac{z_{k,m} - \sqrt{1-\bar{\alpha}_k}\, \hat{\epsilon}_m}{\sqrt{\bar{\alpha}_k}}. \tag{3}$$

**Candidate scoring.** We use a transformer-based trajectory scorer with self-attention among candidates and cross-attention to scene tokens. Separate MLP prediction heads estimate the five components of the Predictive Driver Model Score (PDMS). The predicted metric components are aggregated into a selection score $\hat{Q}_{\mathbf{w}}(\tau)$ (Dauner et al., 2024). We select the candidate with the highest score for execution. The aggregation weights $\mathbf{w}$ can be adjusted at inference to reflect different driving preferences without retraining (Kirby et al., 2026).

### 3.2 LEARNING A FUTURE-PRIVILEGED TEACHER

The trajectory generator can produce multiple candidate distributions, but a single demonstration provides limited supervision for these alternatives. We therefore train a teacher model using recorded future observations as privileged conditioning. The teacher aggregates observations across multiple future time steps into a global scene representation. This representation guides trajectory refinement to produce additional high-quality target behaviors.

**Future-conditioned planning.** We reuse the visual encoder to extract visual tokens from recorded future images. A trainable global scene encoder $F$ combines these tokens with the current scene tokens $\mathbf{h}$ to form a global scene representation $\mathbf{f}$:

$$\mathbf{f} = F\big(\mathbf{h}, \bar{E}(\mathcal{F}_t)\big) \tag{4}$$

where $\bar{E}$ denotes the visual encoder used for future observations. The global scene encoder initializes queries from $\mathbf{h}$ and combines current and future information through spatial, temporal, and cross-attention. The global scene representation $\mathbf{f}$ is shared across candidates and conditions, both the teacher's refiner and scorer, through cross-attention.

**Teacher training and target generation.** We train the teacher using recorded trajectories and metric supervision. Although each scene provides only a single trajectory, future observations provide additional scene context that helps the teacher generate higher-quality alternatives. After training, we freeze the teacher and sample a bank $\mathcal{T} = \{\tau_j^T\}_{j=1}^{M}$ for each scene. A PDM-closed metric evaluator assigns component labels $m_r(\tau)$ to each candidate. These evaluated labels determine target eligibility, allowing trajectories that differ from the human driver to be retained. We then select high-quality, geometrically distinct candidates to supervise the student.

### 3.3 FUTURE-PRIVILEGED MULTI-TARGET DISTILLATION

Transferring only the teacher's highest-scoring trajectory would retain the single-target limitation. We instead select a small set of high-quality alternatives and assign each to a different student proposal. This separates target quality, target diversity, and supervision assignment.

**Selecting teacher targets.** We retain valid candidates that pass the safety check of PDMS. We initialize the target set with the highest-scoring trajectory and iteratively expand it by adding the candidate that maximizes the minimum average displacement error (ADE) to the selected trajectories. Selected targets must have a pairwise ADE of at least 1 m. The resulting target set $\mathcal{Y} = \{y_j\}_{j=1}^{J}$ contains $J \leq K$ trajectories, where $K = 4$ is the maximum number of targets per scene.

**Target assignment and residual construction.** We use optimal bipartite matching to assign the $J$ teacher targets $\mathcal{Y}$ to distinct proposals from the full student set $\mathcal{B}$. The assignment minimizes the total ADE:

$$\pi^* = \arg \min_{\pi \text{ injective}} \sum_{j=1}^{J} d_{xy}(b_{\pi(j)}, y_j). \tag{5}$$

Here, $d_{xy}(a, b)$ is the mean Euclidean distance between corresponding positions, and $\pi^*(j)$ indexes the student proposal matched to teacher target $y_j$. This yields $J$ matched pairs for trajectory supervision.

To further construct noisy residuals for all $M$ candidates, we assign each unmatched proposal its nearest teacher target in ADE while preserving the matched assignments. Specifically, following Eq. equation 2, we add Gaussian noise to the residual between each proposal and its assigned teacher target to form the DiT input. Denoising reconstructs all $M$ candidates, while the following trajectory losses supervise only the $J$ matched pairs:

$$\begin{aligned}
\mathcal{L}_{\text{prop}} &= \frac{1}{J} \sum_{j=1}^{J} d_1(b_{\pi^*(j)}, y_j), \\
\mathcal{L}_{\text{diff}} &= \mathbb{E}_{k,\epsilon} \left[ \frac{1}{HJ} \sum_{j=1}^{J} \| \hat{\epsilon}_{\pi^*(j)} - \epsilon_{\pi^*(j)} \|_F^2 \right], \\
\mathcal{L}_{\text{traj}} &= \mathbb{E}_{k,\epsilon} \left[ \frac{1}{J} \sum_{j=1}^{J} d_1(\hat{\tau}_{\pi^*(j)}, y_j) \right].
\end{aligned} \tag{6}$$

Here, $d_1(a, b)$ denotes the mean per-pose sum of absolute position and heading errors. $\mathbb{E}_{k,\epsilon}$ denotes the expectation $\mathbb{E}$ over the sampled diffusion step $k$ and Gaussian noise $\epsilon$. $\mathcal{L}_{\text{prop}}$ supervises the proposal decoder with an $L_1$ trajectory loss against matched teacher targets. The diffusion refiner is trained with $\mathcal{L}_{\text{diff}}$, a mean squared error loss on predicted noise, and $\mathcal{L}_{\text{traj}}$, an $L_1$ trajectory loss between refined candidates and the same matched targets.

### 3.4 SAFETY-AWARE SCORE CORRECTION

Multi-target distillation provides diverse trajectory supervision, but safe candidate selection also requires accurate safety scores. Highly ranked unsafe candidates can displace safe alternatives, while underestimated safe candidates may be overlooked. In performance evaluation, a zero safety label reduces the overall score to zero, regardless of other metric values. We therefore introduce Safety-Aware Score Correction, using evaluated safety labels for the student's own candidates to correct both errors.

**Score supervision.** The scorer is supervised on all $M$ student trajectories using binary cross-entropy:

$$\mathcal{L}_{\text{metric}} = \frac{1}{M} \sum_{m=1}^{M} \sum_{r \in \mathcal{R}} \text{BCE}(\hat{p}_r(\hat{\tau}_m), \tilde{m}_r(\hat{\tau}_m)). \tag{7}$$

Here, $\hat{p}_r(\tau)$ and $\tilde{m}_r(\tau)$ denote the predicted score and corresponding training label for component $r$, respectively. $\mathcal{R}$ indexes the five metric components. The loss is averaged over the $M$ student trajectories and summed over these components.

**Correcting highly ranked unsafe candidates.** We penalize candidates with high predicted selection scores that fail the evaluated safety checks. We pair each with the nearest eligible alternative in ADE that passes the safety checks within the same scene. Pairs must satisfy geometric proximity constraints; we skip candidates without an eligible alternative. This strategy aims to improve the scorer's ability to distinguish safe and unsafe trajectories near the safety boundary. A pairwise margin loss is applied to the safety logits of each unsafe–safe pair $(\tau^-, \tau^+)$:

$$\mathcal{L}_{\text{neg}} = \sum_{r \in \mathcal{S}} \frac{1}{\max(1, |\mathcal{P}_r|)} \sum_{(\tau^-, \tau^+) \in \mathcal{P}_r} \text{softplus}\big(1 - \ell_r(\tau^+) + \ell_r(\tau^-)\big)\,, \tag{8}$$

where $\mathcal{S} = \{\text{NC}, \text{DAC}\}$ indexes the safety metric components (Appendix C) and $\mathcal{P}_r$ contains pairs whose unsafe candidate fails component $r$. Here, $\ell_r$ denotes the predicted component logit. This loss encourages lower logits for unsafe candidates and higher logits for their safe alternatives on the failed safety components.

**Correcting underestimated safe candidates.** Candidates that pass all safety checks may still receive low predicted safety scores. For these candidates, we consider a safety component underestimated if its predicted probability is below 0.9. An additional positive-label binary cross-entropy loss selectively strengthens supervision on these components:

$$\mathcal{L}_{\text{pos}} = \sum_{r \in \mathcal{S}} \frac{1}{\max(1, |\mathcal{H}_r|)} \sum_{\tau \in \mathcal{H}_r} \text{softplus}(-\ell_r(\tau)). \tag{9}$$

Here, $\mathcal{H}_r$ contains selected safe candidates whose predicted score for component $r$ is underestimated. Both correction losses act on the safety logits, with candidate selection and evaluator labels detached.

### 3.5 Optimization and Deployment

Training proceeds in three stages. First, we train a base coarse-to-fine visual planner using recorded supervision. Second, we initialize the teacher from this base and introduce future observations through the global scene encoder to condition refinement and scoring. Both stages use proposal, diffusion, trajectory reconstruction, and score losses, with demonstrations as trajectory targets and evaluated metrics as score labels. Third, we freeze the teacher and cache its selected targets. The student is initialized from the first-stage base and fine-tuned with multi-target distillation and Safety-Aware Score Correction:

$$\begin{aligned}\mathcal{L}_{\text{student}} = \lambda_{\text{prop}}\mathcal{L}_{\text{prop}} + \lambda_{\text{diff}}\mathcal{L}_{\text{diff}} + \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} \\ +\lambda_{\text{metric}}\mathcal{L}_{\text{metric}} + \lambda_{\text{neg}}\mathcal{L}_{\text{neg}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}}.\end{aligned} \tag{10}$$

We use future observations, teacher, and metric evaluator only during training or target preparation.

## 4 Experiments

### 4.1 Experimental Setup

**Benchmarks and metrics.** We evaluate on the `navtest` split of NAVSIM v1 (Dauner et al., 2024) and report the Predictive Driver Model Score (PDMS) and its five components. On the `navhard` split of NAVSIM v2 (Cao et al., 2025), we follow the official two-stage evaluation protocol and report the combined Extended PDMS (EPDMS), stage-wise scores, and component metrics. The metrics of NAVSIM v1 and v2 are detailed in Appendix C. We additionally conduct a zero-shot closed-loop evaluation on HUGSIM (Zhou et al., 2026a) across Easy, Medium, Hard, and Extreme difficulty levels. And we report the averages value of route completion (RC) and HD-Score for each level. The Appendix A provides implementation details for training, inference, and dataset.

### 4.2 Experimental Results

**Quantitative results.** We introduce two model variants, FPD-Drive and FPD-Drive-Pro. We train the base model on NAVSIM `trainval` (106K) and SimScale (134K), following (Kirby et al., 2026; He et al., 2026). At the same training-data scale, our base model achieves state-of-the-art performance among the compared methods (Table 1 and Table 2). We attribute these gains to multiple

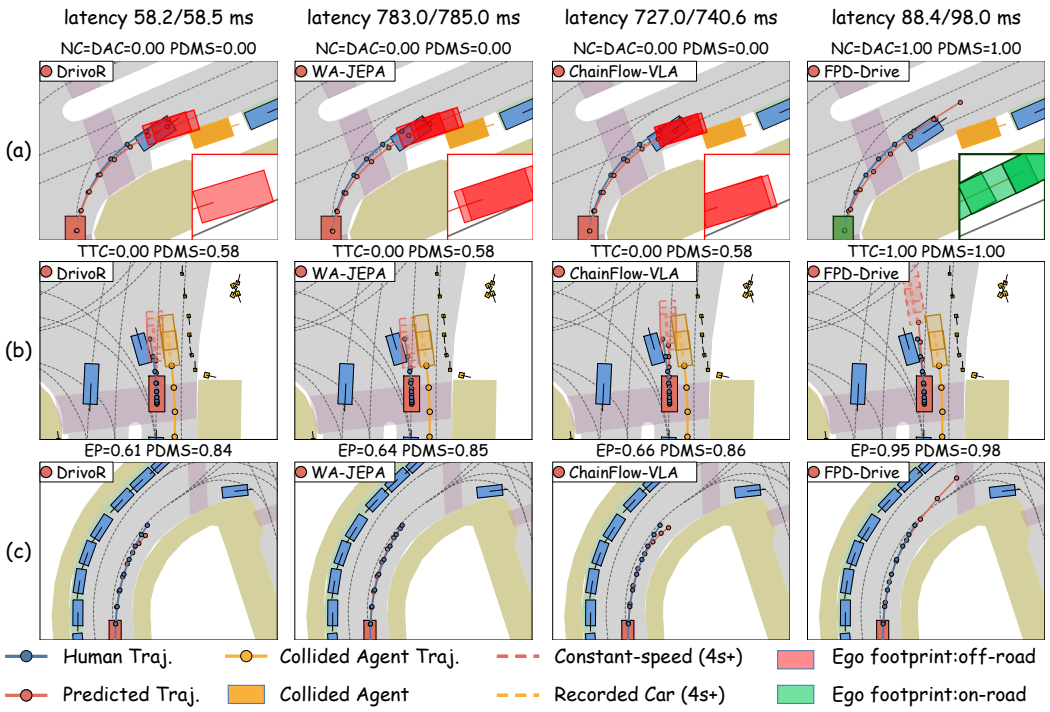



Figure 2: **Qualitative comparison on NAVSIM v1 `navtest`.** We show the metrics for each scene, with PDMS and inference latency as mean/p99 in milliseconds.

Table 1: **Comparison on the NAVSIM v1 `navtest`**. Best overall PDMS is in **bold**, second-best is underlined.

| Entry | Venue | NAVSIM v1 | | | | | PDMS↑ |
|---|---|---|---|---|---|---|---|
| | | NC↑ | DAC↑ | EP↑ | TTC↑ | C↑ | |
| Human Driver | – | 100 | 100 | 87.5 | 100 | 99.9 | 94.8 |
| MeanFuser | CVPR 26 | 98.6 | 97.0 | 82.8 | 95.0 | 100.0 | 89.0 |
| ReCogDrive | ICLR 26 | 97.9 | 97.3 | 87.3 | 94.9 | 100.0 | 90.8 |
| WA-JEPA | arXiv 26.08 | 99.4 | 98.2 | 85.3 | 97.2 | 100.0 | 91.6 |
| DriveFine | ECCV 26 | 98.8 | 99.2 | 86.9 | 96.2 | 100.0 | 91.8 |
| DriveSuprim | AAAI 26 | 98.6 | 98.6 | 91.3 | 95.5 | 100.0 | 93.5 |
| CLOVER | arXiv 26.05 | 98.6 | 99.1 | 92.8 | 95.8 | 100.0 | 94.5 |
| Drive-JEPA | arXiv 26.01 | 99.1 | 98.2 | 91.4 | 96.2 | 99.9 | 93.7 |
| DrivoR | CVPR 26 | 99.0 | 98.9 | 90.0 | 96.7 | 100.0 | 93.7 |
| DrivoR-Scale | CVPR 26 | 99.1 | 99.2 | 91.6 | 96.9 | 100.0 | 94.6 |
| ChainFlow-VLA | arXiv 26.05 | 99.1 | 99.3 | 92.0 | 97.0 | 100.0 | 94.8 |
| DriveZero | arXiv 26.09 | 99.0 | 99.2 | 93.1 | 96.0 | 100.0 | 94.8 |
| DriveZero-Scale | arXiv 26.09 | 99.2 | 99.4 | 92.5 | 97.3 | 99.9 | 95.3 |
| **FPD-Drive** | – | 99.2 | 99.5 | 91.3 | 97.5 | 99.9 | 94.9 |
| **FPD-Drive-Pro** | – | 99.3 | 99.3 | 93.1 | 97.4 | 100.0 | **95.5** |

trajectory supervision and more reliable candidate selection. The future-privileged teacher provides high-quality scene-adaptive alternatives, extending supervision beyond the recorded human trajectory. Safety-Aware Score Correction improves safety assessment, helping the student select safe trajectories more reliably. As shown in Fig. 2, our method achieves higher success rates in complex scenarios with modest computational overhead. Moreover, zero-shot closed-loop results on HUGSIM (Table 3) further support our model's generalization to closed-loop driving. Motivated

Table 2: **Comparison on the NAVSIM v2 `navhard`**. Best combined EPDMS is in **bold**, second-best is underlined.

| Entry | Venue | Stage | NAVSIM v2 | | | | | | | | | EPDMS↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | NC↑ | DAC↑ | DDC↑ | TLC↑ | EP↑ | TTC↑ | LK↑ | HC↑ | EC↑ | |
| RAP | ICLR 26 | 1 | 97.1 | 94.4 | 98.8 | 99.8 | 83.9 | 96.9 | 94.7 | 96.4 | 66.2 | 39.6 |
| | | 2 | 83.2 | 83.9 | 87.4 | 98.0 | 86.9 | 80.4 | 52.3 | 95.2 | 52.4 | |
| ZTRS | ECCV 26 | 1 | 98.9 | 97.6 | 100.0 | 100.0 | 66.7 | 98.9 | 96.2 | 96.7 | 44.0 | 48.1 |
| | | 2 | 91.1 | 90.4 | 95.8 | 99.0 | 63.6 | 89.8 | 60.4 | 97.6 | 66.1 | |
| GuideFlow | CVPR 26 | 1 | 99.6 | 98.0 | 99.4 | 99.3 | 79.7 | 99.3 | 94.9 | 97.1 | 58.2 | 51.5 |
| | | 2 | 91.4 | 89.5 | 95.2 | 98.9 | 77.5 | 89.6 | 52.6 | 93.6 | 51.0 | |
| SimScale | CVPR 26 | 1 | 99.6 | 99.1 | 99.9 | 100.0 | 69.6 | 99.6 | 95.8 | 95.6 | 28.4 | 53.3 |
| | | 2 | 94.5 | 94.2 | 95.8 | 99.2 | 75.8 | 92.8 | 60.1 | 96.1 | 43.2 | |
| DrivoR | CVPR 26 | 1 | 98.9 | 97.3 | 99.2 | 99.6 | 77.7 | 99.1 | 95.3 | 97.6 | 68.4 | 52.3 |
| | | 2 | 92.3 | 92.2 | 97.0 | 99.0 | 72.1 | 90.3 | 56.3 | 97.1 | 38.8 | |
| DrivoR-Scale | CVPR 26 | 1 | 99.1 | 98.2 | 99.3 | 99.8 | 75.4 | 98.7 | 94.9 | 97.6 | 70.2 | 54.6 |
| | | 2 | 92.3 | 91.6 | 97.3 | 99.1 | 75.7 | 90.6 | 56.1 | 98.4 | 44.7 | |
| DriveZero | arXiv 26.09 | 1 | 98.7 | 98.9 | 99.9 | 99.8 | 79.5 | 98.7 | 95.8 | 97.6 | 51.6 | 51.5 |
| | | 2 | 90.5 | 89.2 | 94.6 | 98.6 | 81.2 | 88.3 | 57.2 | 94.5 | 41.3 | |
| DriveZero-Scale | arXiv 26.09 | 1 | 98.7 | 99.3 | 100.0 | 99.3 | 76.7 | 98.4 | 95.3 | 96.9 | 44.4 | 56.8 |
| | | 2 | 94.4 | 95.4 | 97.7 | 98.4 | 76.1 | 92.5 | 60.0 | 95.7 | 44.0 | |
| **FPD-Drive** | – | 1 | 99.3 | 98.4 | 99.7 | 99.6 | 78.0 | 99.1 | 96.0 | 97.3 | 50.7 | 55.9 |
| | | 2 | 94.2 | 93.5 | 95.9 | 99.3 | 79.8 | 92.1 | 59.8 | 94.8 | 41.4 | |
| **FPD-Drive-Pro** | – | 1 | 99.8 | 98.0 | 99.9 | 99.8 | 77.0 | 99.3 | 96.2 | 97.3 | 59.6 | **58.1** |
| | | 2 | 95.4 | 92.8 | 95.1 | 99.3 | 80.5 | 93.5 | 59.0 | 96.8 | 47.1 | |

Table 3: **Zero-shot closed-loop evaluation on HUGSIM.** Best average scores are in **bold**, second-best are underlined. E/M/H/X denote Easy/Medium/Hard/Extreme. FPD-Drive reports preliminary results on 345 scenarios. **Visualizations are shown in the supplementary video.**

| Method | Venue | RC↑ | | | | | HD-Score↑ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | E | M | H | X | Avg. | E | M | H | X | Avg. |
| VAD | ICCV 23 | 51.3 | 31.1 | 25.3 | 26.5 | 31.4 | 36.3 | 9.5 | 8.0 | 11.5 | 13.4 |
| LTF | TPAMI 22 | 67.8 | 35.1 | 26.2 | 40.5 | 38.9 | 58.9 | 18.0 | 9.8 | 25.9 | 23.7 |
| UniAD | CVPR 23 | 78.4 | 60.5 | 33.6 | 17.8 | 45.9 | 64.9 | 45.8 | 20.6 | 6.6 | 32.7 |
| DrivoR | CVPR 26 | 80.9 | 50.5 | 33.8 | 47.1 | 49.8 | 73.3 | 34.6 | 18.8 | 32.5 | 35.7 |
| GigaPixel | arXiv 26.06 | 78.6 | 60.7 | 32.7 | 35.4 | 50.1 | 67.4 | 51.9 | 19.1 | 21.6 | 38.5 |
| DriveZero | arXiv 26.09 | 85.5 | 63.7 | 31.7 | 43.3 | 53.9 | 78.5 | 47.2 | 16.0 | 28.7 | 39.4 |
| **FPD-Drive** | - | 86.2 | 62.8 | 38.2 | 43.3 | **55.3** | 78.5 | 57.0 | 25.2 | 28.1 | **45.1** |

by the benefits of data augmentation in DriveZero-Scale (He et al., 2026), we further train FPD-Drive-Pro with additional generated data (158.8K, details in Appendix B) to evaluate the benefits of expanding the training distribution.

**Qualitative analysis.** Fig. 2 compares the planned trajectory of the open-source method. In row (a), every baseline leaves the road and hits a vehicle parked off the road; FPD-Drive stays on the road and passes it. In row (b), every baseline is on course to collide with the vehicle. The dashed paths show where each car would be if it kept its current speed. For the baselines, these land on the vehicle on the right; FPD-Drive would pass quickly. In row (c), no method fails any metric, and the difference is progress: the baselines stop partway along the human route; FPD-Drive follows it to the end. Safety does not cost progress. Additional visualizations are presented in Appendix D.

**Ablation studies.** We conduct all ablations using our base FPD-Drive. Component ablations in Table 4 examine the teacher and student designs. For each teacher, the two PDMS values corre-

Table 4: **Ablations of FPD-Drive's architecture and distillation components on NAVSIM v1.** Teacher PDMS is reported as learned-scorer selection / PDM-Closed evaluator selection.

| Model | Traj. Dec. | Ref. DiT | Global Scene Enc. | Future Token | Scorer | Distillation | | PDMS |
|---|---|---|---|---|---|---|---|---|
| | | | | | | FPMT | SASC | |
| Teacher | ✓ | ✗ | ✗ | ✗ | ✓ | – | – | 94.0 / 96.7 |
| | ✓ | ✗ | ✗ | ✓ | ✓ | – | – | 94.8 / 98.8 |
| | ✓ | ✓ | ✗ | ✓ | ✓ | – | – | 95.1 / 99.2 |
| | ✓ | ✓ | ✓ | ✓ | ✗ | – | – | – / 98.8 |
| | ✓ | ✓ | ✓ | ✓ | ✓ | – | – | 95.6 / 99.6 |
| Student | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | 94.0 |
| | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | 94.1 |
| | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | 94.7 |
| | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | 94.2 |
| | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | 94.9 |

Table 5: **Ablations of FPD-Drive's teacher-target supervision on NAVSIM v1.**

**(a) Multi-target distillation**

| Target $K$ | $\bar{J}$ | PDMS↑ |
|---|---|---|
| 1 | 1.0 | 94.6 |
| 2 | 1.9 | 94.7 |
| **4 (default)** | 3.1 | 94.9 |
| 6 | 3.3 | 94.9 |

**(b) Supervised trajectory selection and distillation policy**

| Selection | Distillation | Stu. PDMS↑ |
|---|---|---|
| Teacher Scorer | **w/o** Fut. (Self-Disl.) | 94.1 |
| PDM-Closed | **w/o** Fut. (Self-Disl.) | 94.3 |
| Teacher Scorer | **w/** Fut. (FPD) | 94.6 |
| PDM-Closed | **w/** Fut. (FPD) | 94.9 |

spond to selection by the learned scorer and the PDM-Closed evaluator, respectively. PDM-Closed selects higher-quality trajectories under the evaluation metrics, supporting its use for teacher-target selection. However, we still train the scorer jointly with the teacher because gradients from metric prediction help improve scene representations for trajectory generation. Future tokens and the global scene encoder improve the teacher's trajectory prediction, while residual DiT refinement benefits both the teacher and student, supporting our backbone design. The student performs best when Future-Privileged Multi-Target Distillation and Safety-Aware Score Correction are combined, suggesting complementary contributions to trajectory generation and selection.

The ablation on multi-target distillation in Table 5(a) examines the effect of target budget. Here, $K$ denotes the maximum target budget per scene, and $\bar{J}$ denotes the average number of targets retained after quality and diversity filtering. Using multiple targets improves student performance, with diminishing gains at larger target budgets. The ablation on supervised trajectory selection in Table 5(b) shows that selecting higher-quality targets with PDM-Closed improves student performance compared with learned-scorer selection. Under both selection criteria, distillation from a future-privileged teacher outperforms self-distillation without future information. Additional ablations on future-token injection locations, token-level distillation policy, distillation target sources and number of DiT denoising steps are provided in Appendix E.

## 5 Conclusion

We presented FPD-Drive, a framework that uses recorded futures to expand what a driving policy can learn from each demonstration. A future-informed teacher provides multiple alternatives, which are evaluated and filtered into trajectory targets for the student. Targeted score correction helps the student distinguish unsafe candidates from nearby safe alternatives and recognize safe options it would otherwise undervalue. FPD-Drive achieves state-of-the-art performance on NAVSIM v1 and v2, while zero-shot closed-loop evaluation on HUGSIM further demonstrates its generalization to closed-loop driving. Limitations and future directions are discussed in Appendix F.

## STATEMENT

**AI Use Statement** An LLM-based writing assistant was used solely to assist with drafting, editing, and polishing the manuscript based on author-provided content. It was not used to develop the research methodology, conduct experiments, or generate numerical results. All scientific content and findings were provided and verified by the authors, who take full responsibility for the final manuscript.

**Reproducibility Statement** To support reproducibility, we describe the model architecture, training objectives, and teacher–student optimization procedure in Sec. 3, and specify the benchmarks and evaluation protocols in Sec. 4.1. Training schedules, hyperparameters, and inference settings are provided in Appendix A, with architectural configurations in Appendix D. Appendix B describes the training-data composition and generation pipeline, while Appendix C defines the evaluation metrics and safety checks. Additional ablations are presented in Appendix E.

## REFERENCES

Sining Ang, Yuguang Yang, Canyu Chen, and Yan Wang. CLOVER: Closed-Loop Value Estimation and Ranking for End-to-End Autonomous Driving Planning. *arXiv preprint arXiv:2605.15120*, 2026.

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.

Jiajun Cao, Xiaoan Zhang, Xiaobao Wei, Liyuqiu Huang, Zijian Wang, Hanzhen Zhang, Zhengyu Jia, Wei Mao, Hao Wang, Xianming Liu, Shuchang Zhou, Yang Wang, and Shanghang Zhang. EvoDriveVLA: Evolving Driving VLA Models via Collaborative Perception-Planning Distillation. *arXiv preprint arXiv:2603.09465*, 2026.

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*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4709–4722. PMLR, 2025.

Canyu Chen, Yuguang Yang, Zhewen Tan, Yizhi Wang, Ruiyi Zhan, Haiyan Liu, Xuanyao Mao, Jason Bao, Xinyue Tang, Linlin Yang, Bingchuan Sun, Yan Wang, and Baochang Zhang. Devil is in Narrow Policy: Unleashing Exploration in Driving VLA Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Findings Track*, 2026a.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by Cheating. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 66–75. PMLR, 2020.

Li-Heng Chen, Ke Cheng, Yahui Liu, Lei Shi, Shi-Sheng Huang, and Hongbo Fu. VistaGEN: Consistent Driving Video Generation with Fine-Grained Control Using Multiview Visual-Language Reasoning. *arXiv preprint arXiv:2603.28353*, 2026b.

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*, volume 37, pp. 28706–28719, 2024.

Hao Dou. FIRE-VLA: Failure-Informed Self-Evolution for Vision-Language-Action Models in Autonomous Driving. *arXiv preprint arXiv:2608.13395*, 2026.

Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. MagicDrive-V2: High-Resolution Long Video Generation for Autonomous Driving with Adaptive Control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28135–28144, 2025.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A Generalizable Driving World Model with High Fidelity and Versatile Controllability. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91560–91596, 2024.

Hao He, Chengcheng Hu, Zirun Su, Heng Zhang, Haisong Liu, Jinke Li, Haochen Tian, Zhenwei Shen, Hongyang Li, Zhichao Li, Yunchen Yang, Bochao Huang, Siyu Zhang, Kuangye Chen, Xiongjie Zhang, Wentao Dai, Hengchen Dai, Siyuan Liu, Zehao Huang, and Naiyan Wang. DriveZero: End-to-End Driving Beyond Human Demonstrations. *arXiv preprint arXiv:2609.06055*, 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.

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*, 2025.

Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. DriveAdapter: Breaking the Coupling Barrier of Perception and Planning in End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7953–7963, 2023.

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. 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-End Vectorized Autonomous Driving via Probabilistic Planning. In *International Conference on Learning Representations*, 2026.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, Anh-Quan Cao, Nermin Samet, Tuan-Hung VU, and Matthieu Cord. Driving on Registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32058–32069, 2026.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing End-to-End Autonomous Driving with Latent World Model. In *International Conference on Learning Representations*, 2025a.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-End Driving with Online Trajectory Evaluation via BEV World Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27137–27146, 2025b.

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*, 2026.

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*, 2025c.

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.

Shuai Liu, Hechangle Gong, Hao Jiang, Runlin He, Junxiang Zhan, Kai Huang, Sheng Yang, and Shaoqing Ren. MM-Future: Multi-Mode Joint World-Action Modeling for Autonomous Driving. *arXiv preprint arXiv:2609.20377*, 2026.

NVIDIA, Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, Liang Feng, Greg Heinrich, Jack Huang, Peter Karkus, Boyi Li, Pinyi Li, Tsung-Yi Lin, Dongran Liu, Ming-Yu Liu, Langechuan Liu, Zhijian Liu, Jason Lu, Yunxiang Mao, Pavlo Molchanov, Lindsey Pavao, Zhenghao Peng, Mike Ranzinger, Ed Schmerling, Shida Shen, Yunfei Shi, Sarah Tariq, Ran Tian, Tilman Wekel, Xinshuo Weng, Tianjun Xiao, Eric Yang, Xiaodong Yang, Yurong You, Xiaohui Zeng, Wenyuan Zhang, Boris Ivanovic, and Marco Pavone. Alpamayo-R1: Bridging Reasoning and Action Prediction for Generalizable Autonomous Driving in the Long Tail. *arXiv preprint arXiv:2511.00088*, 2025.

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7077–7087, 2021.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L. Waslander, Yu Liu, and Hongsheng Li. LMDrive: Closed-Loop End-to-End Driving with Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15120–15130, 2024.

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 *Computer Vision – ECCV 2024*, volume 15110 of *Lecture Notes in Computer Science*, pp. 256–274. Springer Nature Switzerland, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. SparseDriveV2: Scoring is All You Need for End-to-End Autonomous Driving. In *Computer Vision – ECCV 2026*, volume 17049 of *Lecture Notes in Computer Science*, pp. 446–463. Springer Nature Switzerland, 2026.

Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, Hangjun Ye, Long Chen, and Hongyang Li. SimScale: Learning to Drive via Real-World Simulation at Scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36365–36374, 2026.

Junli Wang, Yinan Zheng, Xueyi Liu, Zebin Xing, Pengfei Li, Kun Ma, Hangjun Ye, Guang Chen, Guang Li, Long Chen, Zhongpu Xia, and Qichao Zhang. MeanFuser: Fast One-Step Multi-Modal Trajectory Generation and Adaptive Reconstruction via MeanFlow for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17884–17893, 2026a.

Linbo Wang, Yupeng Zheng, Qiang Chen, Shiwei Li, Yichen Zhang, Zebin Xing, Qichao Zhang, Xiang Li, Deheng Qian, Pengxuan Yang, Yihang Dong, Ce Hao, Xiaoqing Ye, Junyu Han, Yifeng Pan, and Dongbin Zhao. Latent-WAM: Latent World Action Modeling for End-to-End Autonomous Driving. *arXiv preprint arXiv:2603.24581*, 2026b.

Linhan Wang, Zichong Yang, Chen Bai, Guoxiang Zhang, Xiaotong Liu, Xiaoyin Zheng, Xiao-Xiao Long, Chang-Tien Lu, and Cheng Lu. Drive-JEPA: Video JEPA Meets Multimodal Trajectory Distillation for End-to-End Driving. *arXiv preprint arXiv:2601.22032*, 2026c.

Xinlin Wang, Yujiao Xiang, Yuheng Zhou, Jingqi Wang, Minqing Huang, Jiajie Huang, Dongxu Wei, Tingguang Zhou, Xiyang Wang, Gong Chen, Zhi Xu, Feiyang Tan, Hangning Zhou, and Mu Yang. WA-JEPA: Rethinking the Video JEPA Paradigm for World-Action Modeling in Autonomous Driving. *arXiv preprint arXiv:2608.20974*, 2026d.

Xiyang Wang, Xinlin Wang, Tingguang Zhou, Gong Chen, Xingtai Gui, Zhi Xu, Xiaolei Wu, Feiyang Tan, Hangning Zhou, and Mu Yang. ChainFlow-VLA: Causal Flow Planning with Vision-Language Models. *arXiv preprint arXiv:2605.23270*, 2026e.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the Future: Multiview Visual Forecasting and Planning with World Model for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14749–14759, 2024.

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.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K. Wong, Zhenguo Li, and Hengshuang Zhao. DriveGPT4: Interpretable End-to-End Autonomous Driving Via Large Language Model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024.

Rui Yu, Xianghang Zhang, Runkai Zhao, Huaicheng Yan, and Meng Wang. DistillDrive: End-to-End Multi-Mode Autonomous Driving Distillation by Isomorphic Hetero-Source Planning Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26188–26197, 2025.

Jimuyang Zhang, Zanming Huang, and Eshed Ohn-Bar. Coaching a Teachable Student. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7805–7815, 2023.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive Diffusion World Model for Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27220–27230, 2025.

Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-End Urban Driving by Imitating a Reinforcement Learning Coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15222–15232, 2021.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, XianPeng Lang, and Dongbin Zhao. World4Drive: End-to-End Autonomous Driving via Intention-aware Physical Latent World Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28632–28642, 2025.

Zhiyu Zheng, Shaoyu Chen, Haoran Yin, Xinbang Zhang, Jialv Zou, Xinggang Wang, Qian Zhang, and Lefei Zhang. ResAD: Normalized Residual Trajectory Modeling for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3729–3739, 2026.

Ruiguo Zhong, Benshan Ma, Xiaolong Chen, Lang Zhang, Mingyue Feng, Yaonong Wang, Pei Liu, and Jun Ma. DA-WAM: Decision-Aligned Future Latents for Driving World Models. *arXiv preprint arXiv:2608.19085*, 2026.

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. HUGSIM: A Real-Time, Photo-Realistic and Closed-Loop Simulator for Autonomous Driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4):4673–4691, 2026a.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. OpenDriveVLA: Towards End-to-end Autonomous Driving with Large Vision Language Action Model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(16):13782–13790, 2026b.

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*, volume 38, pp. 27920–27956, 2025.

# APPENDIX

## A TRAINING AND INFERENCE DETAILS

Training consists of three stages. In Stage 1, we train the coarse-to-fine planner described in Sec. 3.1, which serves as the common backbone for the teacher and student, for 30 epochs, starting with a learning rate of $2 \times 10^{-4}$. In Stage 2, we initialize the teacher from the Stage 1 model and train it for 10 epochs with privileged future observations, starting with a learning rate of $2 \times 10^{-4}$. In Stage 3, we freeze the teacher and train the student, initialized from the Stage 1 model, using Future-Privileged Multi-Target Distillation and Safety-Aware Score Correction for 10 epochs, starting with a learning rate of $2 \times 10^{-5}$. We conducted all training on 8×H800 GPUs with a batch size of 16, using the AdamW optimizer and a cosine learning rate decay strategy with linear warmup. We adopt gradient clipping with global norm 1.0, and the training precision is *BF16*. Table 6 details the model configuration.

Following the training-data setup of DrivoR (Kirby et al., 2026), we train our base model, FPD-Drive, on approximately 106K NAVSIM `trainval` scenes and 134K SimScale scenes. NAVSIM (Dauner et al., 2024) provides real-world driving logs derived from nuPlan, including multi-view images and human driving trajectories; SimScale (Tian et al., 2026) augments recorded scenes through reactive simulation, providing rendered multi-view observations and pseudo-expert trajectories. We further train FPD-Drive-Pro with additional controllably generated driving scenarios to broaden behavioral coverage, with generation details and data composition in Sec. B and Table 7, and the resulting scaling trends in Fig. 3.

At inference, the model takes four camera views from the current frame, the current ego state, and a navigation command. The refiner starts from Gaussian residual noise and runs 8 DDIM steps (Song et al., 2021) before adding the residual to each of the 64 trajectory proposals. The scorer then selects one refined trajectory for execution. The output is an $8 \times 3$ sequence of future ego poses $(x_h, y_h, \psi_h)$ at 0.5-second intervals, covering the next 4 seconds. All poses are expressed in the current ego coordinate frame, with positions $(x_h, y_h)$ in meters and heading $\psi_h$ in radians.

Table 6: **Model configuration.**

| Component | Student configuration |
|---|---|
| Visual token dim | $4 \times 16 \times 256$ |
| Visual backbone | DINOv2 (ViT-S/14 vision transformer), frozen |
| Visual backbone adaptation | LoRA, rank 32 |
| Cameras | Front, left, right, and rear |
| Input resolution | $1148 \times 672$ |
| Input history | Current frame only |
| Ego status | 11-dim current-frame token |
| Continuous action | Ego-relative $(x, y, \text{yaw})$ poses |
| Prediction horizon | 8 poses at 2 Hz (4 s) |
| Trajectory proposals | 64 |
| Trajectory tokens dim | $8 \times 256$ |
| Attention heads | 8 |
| Proposal decoder | 4 layers |
| Scoring decoder | 4 layers, 5 MLP heads |
| Residual refiner | 12-layer DiT, 8 heads |
| Refinement process | DDIM, 8 denoise steps |
| Model parameters | 123.3M |

## B DATA SCALING

Common driving patterns dominate the scenario distribution, leaving uncommon traffic conditions and safety-critical events underrepresented. This imbalance limits the diversity of future behaviors and responses that planners can learn from recorded demonstrations.

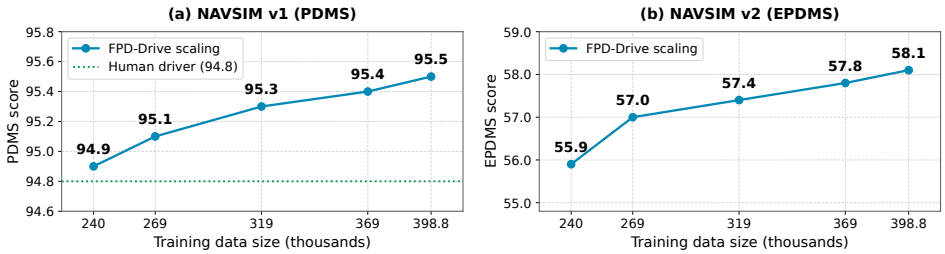



Figure 3: **Training-data scaling of FPD-Drive.**

We combined approximately 106K NAVSIM scenes and 134K SimScale scenes as the original training data. To expand coverage of underrepresented turning behaviors, we allocated approximately 60.6K generated scenes to left commands, 96.4K to right commands, and 1.8K to straight commands.

Fig. 4 illustrates our data generation pipeline. We used controllable driving video generation (Chen et al., 2026b; Gao et al., 2025), conditioned on text prompts, bounding-box layouts, road maps, and camera poses. Manually specified layouts and object identity descriptions control the objects, spatial relationships, and interactions in each scenario which can be used to calculate metric labels. A vision-language model evaluator Bai et al. (2025) assesses the generated videos before their inclusion in the augmentation pool. Table 7 summarizes the marginal quotas applied within this pool. These generated samples provide complementary supervision intended to broaden the training distribution and support reasoning about diverse future outcomes. The generated data was used only for **FPD-Drive-Pro**, while **FPD-Drive** used only the public datasets.

Table 7: **Data allocation used for the 158.8K generated scenes.**

| *Navigation command group* | | | *Speed group* | | |
|---|---|---|---|---|---|
| Condition | Original data (%) | Generated data (K) | Condition | Original data (%) | Generated data (K) |
| Left | 25.36 | 60.6 | 2–5 m/s | 32.40 | 55.9 |
| Right | 11.46 | 96.4 | 5–10 m/s | 35.89 | 46.4 |
| Straight | 63.18 | 1.8 | Other speeds | 31.71 | 56.5 |

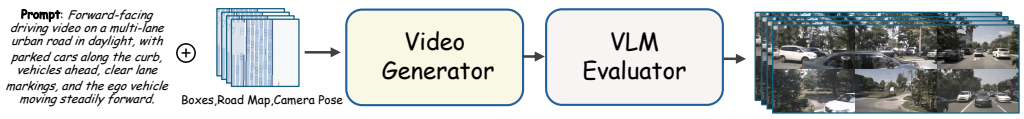



Figure 4: **Overview of the data generation pipeline.**

## C NAVSIM Evaluation Metrics

NAVSIM evaluates planning in a pseudo-closed-loop setting. An LQR controller tracks the predicted 4 s trajectory using a kinematic bicycle model at 10 Hz. The resulting ego states are evaluated against the map, route, and surrounding traffic. Each sub-metric takes a value in $[0, 1]$ and is reported as a percentage. The composite score combines multiplicative factors with a fixed weighted average of the remaining metrics. NAVSIM v1 and v2 differ in both their metric sets and scenario organization.

NAVSIM v1 (`navtest`) uses two multiplicative factors: no at-fault collisions (NC) and drivable-area compliance (DAC). NC considers only collisions for which the ego vehicle is responsible, while DAC checks whether the vehicle footprint remains within the drivable area. The weighted

component includes ego progress (EP), time-to-collision within bounds (TTC), and comfort (C). EP measures progress along the centerline relative to the PDM-Closed reference. TTC checks whether the time to collision with any other vehicle remains above a fixed threshold. C checks whether the simulated trajectory satisfies bounds on acceleration, jerk, and yaw rate. The composite score is:

$$\mathrm{PDMS} = \mathrm{NC} \cdot \mathrm{DAC} \times \frac{5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{C}}{12}. \tag{11}$$

NAVSIM v2 (`navhard_two_stage`) adds two multiplicative factors: driving-direction compliance (DDC) and traffic-light compliance (TLC). DDC penalizes travel against the lane direction within a 1 s window. Its score is 1 for wrong-way travel below 2 m, 0.5 for travel between 2 m and 6 m, and 0 for travel beyond 6 m. TLC checks whether the ego vehicle crosses a stop line at a red light.

The weighted component additionally includes lane keeping (LK), which checks lateral deviation from the centerline using a 0.5 m threshold over a 2 s window. The original comfort metric is replaced by history comfort (HC) and extended comfort (EC). HC evaluates the simulated trajectory together with the recorded past to assess comfort at their junction. EC concatenates trajectories predicted at two adjacent frames of the same scene and applies the comfort bounds to the combined sequence. This penalizes inconsistencies between successive planning steps. The extended composite score is:

$$\mathrm{EPDMS} = \mathrm{NC} \cdot \mathrm{DAC} \cdot \mathrm{DDC} \cdot \mathrm{TLC} \times \frac{5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{LK} + 2\,\mathrm{HC} + 2\,\mathrm{EC}}{16}. \tag{12}$$

NAVSIM v2 also changes how it organizes scenarios and aggregates scores. Its 5,912 scenarios form 225 groups, each evaluated from two start frames separated by 0.5 s. The first stage evaluates the original recorded frames. The second stage evaluates synthetic continuations in which surrounding traffic responds to the planned ego behavior. For each start frame, scores are averaged separately within each stage, and the two stage averages are multiplied. The resulting scores for the two start frames are then averaged to obtain the group score. The final EPDMS is the mean across all groups.

## D Experiment Details

**Qualitative analysis.** Fig. 5 shows additional examples of planned trajectories from open-source methods on single H800 GPU. Row (a) is an outright crash: every baseline drives into the car ahead, while FPD-Drive keeps clear of it. Row (b) is about road boundaries: at a corner, every baseline cuts the curve and leaves the drivable area, which zeroes DAC; FPD-Drive stays on the road. Row (c) repeats the TTC pattern of the main paper in a new scene — the dashed constant-speed paths of the baselines again land on the car ahead; that of FPD-Drive stays well clear of it. Row (d) shows the progress side: with every metric satisfied by every method, FPD-Drive travels 37 m while the baselines travel 25–26 m, close to the recorded human route of 25 m.

Table 8: **Ablation study on future-token injection in the privileged teacher.**

| Setting | Traj. Dec. | Ref. DiT | Scorer | PDMS↑ |
|---|---|---|---|---|
| Causal baseline (No Future Token) | × | × | × | 94.13 |
| Decoder only | ✓ | × | × | 94.26 |
| DiT only | × | ✓ | × | 95.29 |
| Scorer only | × | × | ✓ | 94.79 |
| Decoder + Scorer | ✓ | × | ✓ | 94.89 |
| Decoder + DiT | ✓ | ✓ | × | 95.37 |
| Decoder + DiT + Scorer | ✓ | ✓ | ✓ | 95.41 |
| **Ref. DiT + Scorer (Ours Teacher)** | × | ✓ | ✓ | **95.57** |

## E More Ablation Studies

**Future-token injection in the teacher.** We investigate where to introduce future tokens in the privileged teacher. Starting from a causal teacher without future access, we evaluate three possi-

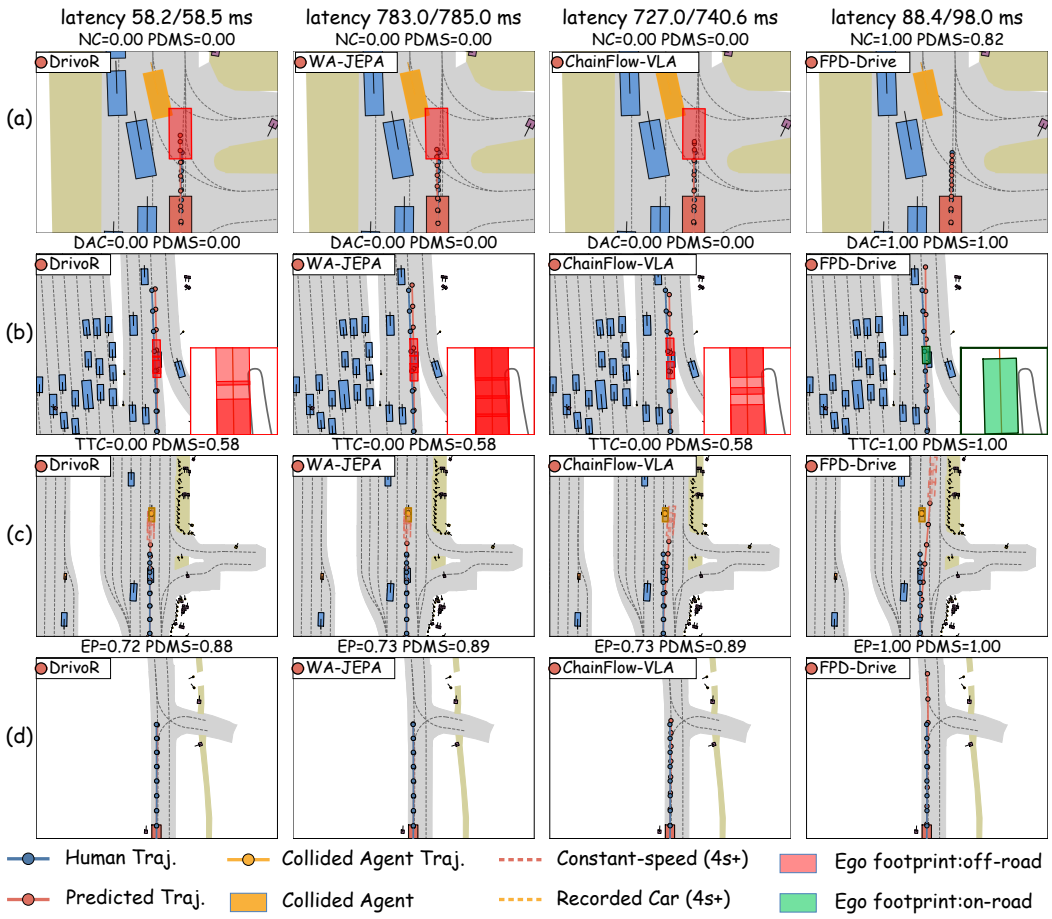



Figure 5: **Additional NAVSIM `navtest` scenarios.**

ble injection locations: the trajectory decoder, residual DiT refiner, and scorer. Future conditioning improves the teacher across different configurations. Interestingly, injecting future tokens into all modules does not provide further benefits. The Ref. DiT + Scorer configuration achieves the best performance, suggesting that future information is most valuable for trajectory refinement and candidate evaluation rather than initial trajectory generation. The quantitative results are shown in Table 8.

**Token-level versus multi-target distillation.** Inspired by privileged policy distillation in EvoDriveVLA (Cao et al., 2026), we compare token-level distillation with our Future-Privileged Multi-Target Distillation. We adapt their token-distribution matching idea to our diffusion planner by applying a token-level KL divergence between teacher and student distributions during DiT denoising. The teacher has access to recorded future observations, whereas the student uses only current observations. The token-level variant underperforms multi-target distillation, as summarized in Table 9. This suggests that matching token distributions during denoising is less effective in our setting than explicitly supervising multiple complete trajectories selected for planning quality and geometric diversity. Future-Privileged Multi-Target Distillation assigns these alternatives to distinct student proposals, directly encouraging coverage of multiple valid behaviors in the same scene.

**Ablation of distillation target sources.** We ablate the candidate-trajectory source for multi-target supervision. Inspired by CLOVER Ang et al. (2026), we construct 64 predefined trajectory vocabularies and compare them with 64 trajectories generated by our future-privileged teacher for each scene. For top-$K$ supervision, both sources use the same target budget $K = 4$, PDM-Closed evalu-

ation, and safety and diversity criteria described in Sec. 3.3. Table 10 shows that teacher-generated candidates provide more effective supervision, supporting the value of learning scene-adaptive alternatives beyond manually specified motion patterns.

Table 9: **Token-level versus multi-target distillation on NAVSIM v1.**

| Distillation strategy | PDMS↑ |
|---|---|
| EvoDrive-VLA | 85.3 |
| Ours w/ Token-level KL | 94.5 |
| FPD-Drive | 94.9 |

Table 10: **Ablation of distillation target sources on NAVSIM v1.** Both variants supervise our student using up to $K = 4$ targets selected from 64 candidate trajectories per scene.

| Candidate source | Candidates | Stu. PDMS↑ |
|---|---|---|
| Predefined trajectory vocabularies | 64 | 94.2 |
| Future-privileged teacher (ours) | 64 | **94.9** |

**Components of Safety-Aware Score Correction.** We ablate the two regularization terms $\mathcal{L}_{\text{neg}}$ and $\mathcal{L}_{\text{pos}}$ in equation 8 and equation 9. We report PDMS and the percentage of unsafe cases in the test set, where a case is unsafe if the selected trajectory has a zero value of NC or DAC. As shown in Table 11, combining the two terms yields the lowest unsafe-case rate.

Table 11: **Ablation of the Components of SASC.**

| $\mathcal{L}_{\text{neg}}$ | $\mathcal{L}_{\text{pos}}$ | Unsafe cases (%)↓ | PDMS↑ |
|---|---|---|---|
| ✗ | ✗ | 1.91 | 94.7 |
| ✓ | ✗ | 1.77 | 94.8 |
| ✗ | ✓ | 1.89 | 94.7 |
| ✓ | ✓ | **1.35** | **94.9** |

**Number of DiT denoising steps.** We study FPD-Drive 's sensitivity to the number of DDIM denoising steps (Song et al., 2021), using 2, 4, 6, 8, and 12 steps with the same trained model. This ablation examines how much iterative refinement is needed for effective planning and whether additional denoising steps justify their inference cost. Table 12 summarizes the settings.

Table 12: **Ablation of DiT denoising steps on NAVSIM v1.**

| Denoising steps | 2 | 4 | 6 | 8 | 12 |
|---|---|---|---|---|---|
| PDMS↑ | 94.7 | 94.7 | 94.8 | **94.9** | 94.9 |

## F LIMITATIONS

FPD-Drive still incurs inference overhead from iterative DiT sampling. Future work will explore single-step denoising and refinement, motivated by MeanFuser (Wang et al., 2026a) 's efficient one-step trajectory generation. Further evaluation is needed to assess robustness during longer-horizon interactions. We have not evaluated the model under extreme weather conditions, such as heavy rain and snow, or deployed it on real vehicles, so we leave these settings for future validation.