# UNDERSTANDING DRIVER INTENTION: INTENTION-DRIVEN HIERARCHICAL PLANNING FOR ROBUST END-TO-END AUTONOMOUS DRIVING

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

## ABSTRACT

Recent works in imitation-learning-based planning have demonstrated promising scalability in end-to-end autonomous driving, yet they either suffer from mode collapse or directional deviation, failing to generate interpretable multi-modal trajectories that accurately reflect human driving intentions. These issues severely constrain model performance and robustness in complex interactive scenarios, resulting in significant performance degradation in closed-loop evaluations. To address these limitations, we propose Intention Planner, the first intention-driven hierarchical planning framework inspired by the cognitive architecture of human driving behavior. Specifically, our method explicitly models hierarchical driving intention representations from long-term strategic intention to automatic control. With a sophisticatedly designed architecture, these interpretable intentions are introduced as structured conditions to instruct a diffusion-based planner. Our framework also incorporates classifier-free guidance for multi-level behavior generation, which dynamically fuse conscious and unconscious planning systems during inference, enabling the generation of intention-aligned and kinematically feasible trajectories in challenging interactive scenario. Extensive experiments on NAVSIM benchmark demonstrate that Intention Planner outperforms previous end-to-end planners, achieving state-of-the-art performance while maintaining superior robustness in long-tail scenarios.

## 1 INTRODUCTION

Learning-based planning has emerged as a scalable approach to autonomous driving. Recent works such as UniAD (Hu et al., 2023), VAD (Jiang et al., 2023), SparseDrive (Sun et al., 2024), and DiffusionDrive (Liao et al., 2024) represent notable advancements in this direction. Most imitation learning (IL) planners incorporate the past trajectory of ego vehicle and the sensor data as input features to predict the trajectory in the next several seconds. These methods have achieved impressive open-loop performance on standard benchmarks. However, despite these advances, a significant performance gap still persists between open-loop and closed-loop evaluations, posing a major challenge for end-to-end autonomous driving systems.

We argue that a fundamental limitation of existing end-to-end planners lies in the lack of structured hierarchical intention representations. Current approaches typically formulate driving behavior as a monolithic sensor-to-trajectory mapping. Although this formulation provides a unified learning paradigm, it requires the model to infer heterogeneous decision factors with different levels solely from sparse trajectory waypoint supervision. These factors entangled within the learned representations, making it difficult to learning clear semantic correspondence between environmental context and the resulting driving trajectories. Instead of capturing the underlying decision factors that govern a driving behavior, existing planners may exploit accidental correlations between observations and expert demonstrations, leading to causal confusion. In line with prior work (Huang et al., 2025), we argue that causal confusion stems primarily from noise in the expert trajectories that serve as supervision signals in imitation learning frameworks. Learning from such noisy signals weakens the model's ability to capture genuine causal relationships, making them suboptimal supervision targets. When trained solely to reproduce expert trajectories, models may learn shortcuts associated with superficial correlations rather than the genuine causal factors that determine driving behaviors.



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.

Our empirical analysis further reveals that this limitation is further fostered by two common designs in existing planners. First, most recent methods heavily rely on discrete and coarse driving commands (Dauner et al., 2024), which introduce severe semantic ambiguity (e.g., generating straight or left-turn trajectories even when the navigation goal clearly requires a right turn), thereby breaking the causal link between high-level intention and action. Second, directly incorporating ego-vehicle historical states as input introduces additional shortcuts. Planners may simply extrapolate historical motion rather than learning the true causal factors in the environment.

Modeling multiple plausible futures is crucial for capturing behavioral uncertainty in driving, but existing planners typically learn directly from expert trajectories without explicitly modeling the driving intentions associated with different driving behaviors. Consequently, the generated trajectories often lack clear semantic correspondence with human driving intentions, resulting in mode collapse and limited interpretability. Specifically, mode collapse occurs when generated trajectories concentrate around highly similar behaviors, preventing the model from representing diverse future driving decisions. This issue arises because without explicit intention representations, trajectory generators lack sufficient guidance to align with multiple driving intentions. Although recent works (Xing et al., 2025) attempt to mitigate this limitation by incorporating geometric priors (e.g., goal points) as conditioning signals for generator, they mainly constrain the spatial distribution of trajectories and do not provide a hierarchical representation that connects high-level intentions with generated behaviors.

To address this limitation, a promising way is to introduce structured intention representations between perception and action. Decades of cognitive science research (Michon, 1985) demonstrates that human driving is not a monolithic sensor-to-action process, but a sophisticated hierarchical cognitive process that integrates high-level strategic intention, mid-level maneuvering intention, and low-level automatic control, as shown in Figure 1. Human driving decision-making follows a hierarchical cognitive architecture in which a small number of complex and effortful higher-order conscious processes at the apex for strategic and maneuvering judgment. These conscious processes are mainly supported by the dorsolateral prefrontal cortex and the anterior cingulate cortex (Leisman et al., 2014). In contrast, unconscious executive processes primarily mediated by the cerebellum and basal ganglia that handle short-term lateral and longitudinal control such as steering maneuvers and pedal operations. These largely unconscious processes dominate the vast majority of routine driving behavior while higher-order conscious strategic and maneuvering intentions intervene when encountering complex interactions or long-horizon planning requirements.

This hierarchical cognitive structure enables human drivers to adapt efficiently to complex and long-tail scenarios. High-level strategic intentions provide global objectives, maneuvering intentions determine appropriate behaviors under environmental interactions, and low-level control execute feasible vehicle motions. However, current end-to-end autonomous driving paradigms deviate substantially from established principles in cognitive science. Most end-to-end planners (Liao et al., 2024) completely ignore hierarchical driving intentions and model driving behavior as a monolithic sensor-to-trajectory mapping in unconscious process. Some methods (Zheng et al., 2025c) rely solely on generative modeling to learn maneuvering intentions while lacking proper supervision and interpretability. Other methods (Tian et al., 2024) introduce a VLM-assisted fast-slow dual-system to improve reasoning efficiency, but fails to model structured intention representations.

Drawing on the above insights, we propose **Intention Planner**, which is **the first cognitively-inspired intention-driven hierarchical planning framework**. Our framework explicitly models multi-level intention representations and leverages them as structured guidance for trajectory generation. This design improves the semantic alignment between driving intentions and generated trajectories, enabling more interpretable and robust multi-modal planning. Our key contributions are as follows:

- We propose a cognitively-inspired hierarchical intention representation framework for end-to-end trajectory planning, which explicitly models driving intentions across three levels: high-level strategic intention, mid-level maneuvering intention, and low-level automatic control. By introducing structured intention representations, our framework effectively mitigates causal confusion, while preserving end-to-end scalability.

- We design an intention-driven multi-modal trajectory generator that leverages hierarchical intention representations as structured conditions for diffusion-based planning. This design enables the generation of semantically diverse and interpretable trajectories with explicit

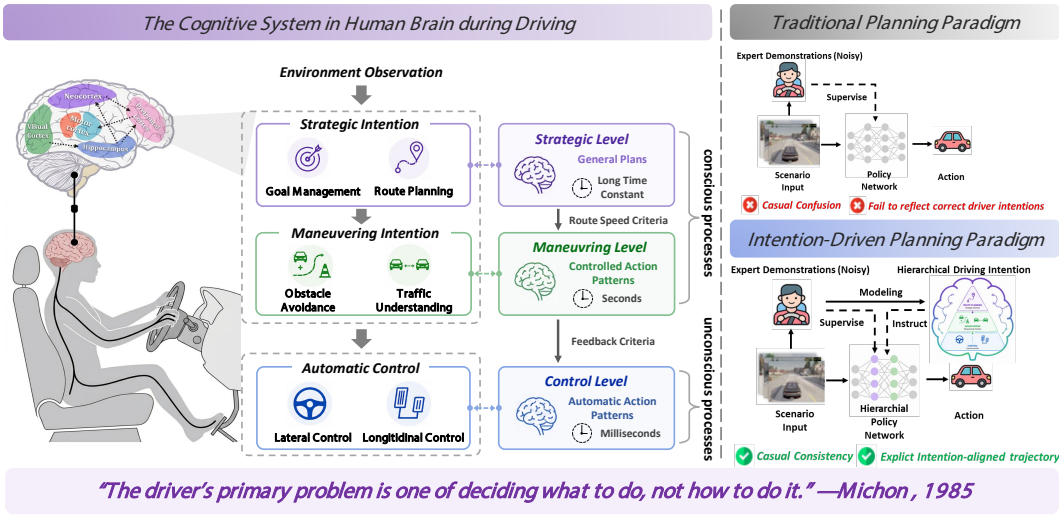



Figure 1: Cognitive inspiration for our Intention-Driven Hierarchical Planning framework. (Left) Human driving cognition system follows a three-level hierarchical architecture: high-level strategic intentions, mid-level maneuvering intentions, and low-level automatic control. (Right) In contrast to the traditional imitation learning paradigm, our cognitively-inspired framework directly models the hierarchical cognitive structure of human driving. By explicitly modeling the driver's multi-level intentions, these interpretable hierarchical intentions serve as explicit conditioning signals that guide the diffusion-based planning decoder.

correspondence between driving intentions and predicted trajectories, alleviating mode collapse and directional deviation.

- Extensive experiments on the NAVSIM benchmark and challenging long-tail scenarios demonstrate that our method outperforms existing end-to-end planners, achieving superior planning performance and robustness in complex driving scenarios.

## 2 Related Works

### 2.1 Imitation Learning-based Planning

There are mainly two types of imitation learning-based planning: end-to-end (E2E) planning and mid-to-mid (M2M) planning. E2E methods predict driving trajectories directly from raw sensor data and have advanced rapidly in recent years. UniAD (Hu et al., 2023) first proposes a planning-oriented framework that integrates perception and prediction tasks through query-based designs. VAD (Jiang et al., 2023) introduces a vectorized scene representation that captures instance-level information for agents and static map elements, offering improved computational efficiency and collision avoidance. Very recently, DiffusionDrive (Liao et al., 2024) leverages diffusion models to achieve superior accuracy and efficiency in both planning and prediction. In contrast, M2M planning takes perception outputs or map elements as input. For instance, DiffusionPlanner (Zheng et al., 2025b) benefits significantly from taking the high definition (HD) map and the ground truth trajectories of the other traffic participants as input.

### 2.2 Intention-aware Multi-modal Trajectory Planning

Recent research has increasingly recognized that modeling of driving intentions is essential for generating multi-modal trajectories in end-to-end autonomous driving. Some recent end-to-end planners integrate intention awareness directly into generative planning decoder. Int2Planner (Chen et al., 2025) samples route intention points from the ego vehicle's route path, and generates corresponding

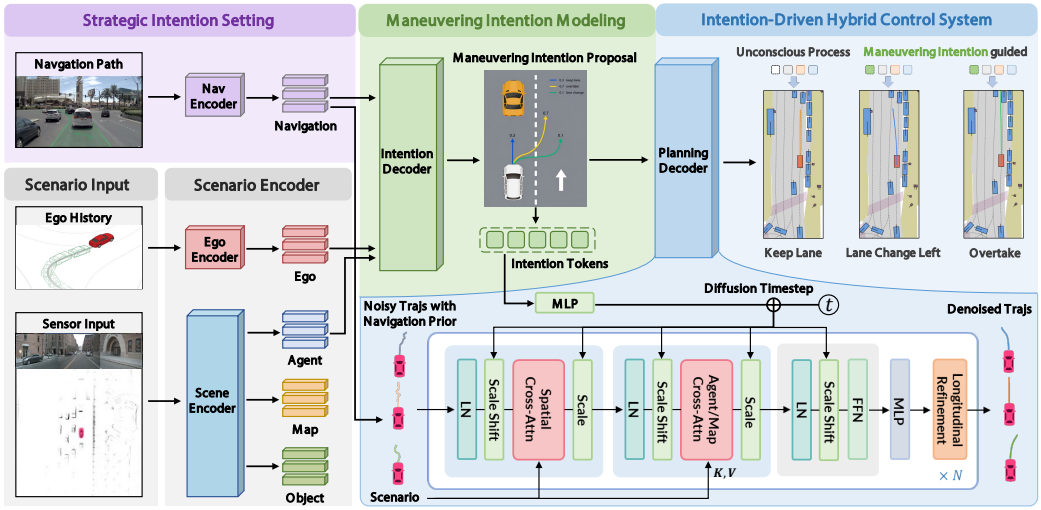



Figure 2: Overall architecture of the Intention Planner framework.

planning trajectories for each intention point, achieving joint optimization of prediction and planning. World4Drive (Zheng et al., 2025a) further introduces an intention-aware physical latent world model. It leverages vision foundation models to extract latent representations containing driving intentions and spatial-semantic priors, then generates and evaluates multi-modal planning trajectories in the latent space. GoalFlow (Xing et al., 2025) focuses on generative modeling and proposes a goalpoint-driven flow matching method for generating multi-modal trajectories, by constructing a goal point vocabulary and designing a scoring mechanism to select optimal goal points. ReAL-AD (Lu et al., 2025) incorporates vision-language models for human-like intention perception and hierarchical reasoning.

# 3 Preliminaries

## 3.1 Problem Formulation

Our research focuses on end-to-end autonomous driving trajectory planning based on diffusion model. The trajectory generation task is formulated as a conditional generative modeling problem. Formally, at each timestep $t$, the ego vehicle receives multi-modal perception inputs, including multi-view camera RGB images $\mathbf{I}_t \in \mathbb{R}^{3\times H\times W}$, LiDAR point clouds $\mathbf{P}_t \in \mathbb{R}^{N_p\times 3}$, and ego state $\mathbf{s}_t \in \mathbb{R}^{d_s}$. The goal is to generate the future trajectory of the ego vehicle $\mathbf{x}_0 \in \mathbb{R}^{N_w\times d_c}$, where $N_w$ denotes the number of future waypoints. We model this task using a conditional diffusion process. Let $\boldsymbol{\kappa}$ denote the encoded scene context extracted by the scenario encoder from the raw inputs $(\mathbf{I}_t, \mathbf{P}_t, \mathbf{s}_t)$. The model $f_\theta$ parameterizes the reverse diffusion process and generates samples from the conditional distribution $p_\theta(\mathbf{x}_0 \mid \boldsymbol{\kappa})$ as

$$\mathbf{x}_0 = f_\theta(\mathbf{I}_t, \mathbf{P}_t, \mathbf{s}_t) = \int_{\mathbf{x}_{1:T}} p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \boldsymbol{\kappa}) \prod_{t=2}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\kappa})\, p(\mathbf{x}_T)\, d\mathbf{x}_{1:T}, \tag{1}$$

where $\mathbf{x}_t$ represents the noisy trajectory at diffusion timestep $t$. This probabilistic formulation naturally captures the multi-modal human driving behavior and serves as the foundation for our method. In Section 4, we further augment the condition $\mathbf{C}$ with hierarchical intention representations.

# 4 Methodology

In this section, we present our **Intention Planner**, as illustrated in Figure 2. Given the environment state and navigation goal, our method first generates a global navigation path that serves as a high-level strategic intention to guide the overall planning process. Building upon this prior, our framework

explicitly models three hierarchical levels of human driving intentions, which mirror the human cognitive architecture as Figure 1. These intentions then act as explicit conditional signals to instruct a diffusion-based planner, enabling the generation of intention-aligned multi-modal future trajectories.

### 4.1 End-to-End Planning Baseline

**Scenario Encoder:** The Scenario Encoder extracts comprehensive multi-modal representations from heterogeneous inputs, providing rich context for downstream intention modeling and trajectory generation. As illustrated in Figure 2, it comprises three specialized encoders operating in parallel. For the scene input, we adopt the vectorized driving scene representation based on the vision backbone of Transfuser (Chitta et al., 2023), a representative end-to-end planner. For the $K$ neighboring agents, we represent it as $\mathbf{F}_{\text{agent}} \in \mathbb{R}^{K \times H_{\text{agent}}}$, where $H_{\text{agent}}$ is the dimension of state information. To encode the navigation path information, we employ sinusoidal embeddings as previous work (Luo et al., 2024), yielding the navigation feature $\mathbf{F}_{\text{nav}} \in \mathbb{R}^{L \times H_{\text{nav}}}$, where $H_{\text{nav}}$ is the feature dimension of each route point.

**Diffusion Decoder:** Our approach builds upon recent diffusion-based end-to-end planning framework (Liao et al., 2024). To improve efficiency while preserving multi-modality, we first generate a set of multi-modal planning anchors $\{\mathbf{a}_k\}_{k=1}^{N_{\text{anchor}}}$ by K-Means from the training set. The diffusion process is then truncated: instead of starting from pure Gaussian noise, it initializes from an anchored Gaussian distribution around these generated planning anchors and adds a small amount of Gaussian noise over a shorter schedule where $T_{\text{trunc}}$ is significantly smaller than $T$. Formally, given a clean trajectory $x^0$ and an anchor $\mathbf{a}_k$, the forward truncated diffusion process is defined as

$$\mathbf{x}^i = \sqrt{\bar{\alpha}_i}\,\mathbf{a}_k + \sqrt{1-\bar{\alpha}_i}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

where $i \in [1, T_{\text{trunc}}]$, $\bar{\alpha}_i = \prod_{s=1}^{i}(1-\beta_s)$, and $\beta_s$ is the variance of the noise added at diffusion step $s$. At inference, the model starts from noisy trajectories sampled around the anchors and progressively denoises them using the efficient DDIM sampler (Song et al., 2020), which requires only a few steps to recover the clean trajectory.

### 4.2 Leveraging Navigation Information for Planning

To explicitly model the driver's high-level strategic intention and mitigate the causal confusion prevalent in existing planners, we leverage the navigation path in place of the coarse driving command (Dauner et al., 2024) commonly adopted in prior works. These discrete driving commands provide only a coarse and semantically ambiguous abstraction of the driver's intention. They lack continuous geometric guidance and long-horizon context, frequently resulting in directional deviations (detailed schematic diagrams and additional experimental results are provided in Appendix A.1 and A.3).

We generate the navigation path from the opensource standard-definition (SD) map and the ego vehicle's Global Positioning System (GPS) trajectories using our proposed toolchain (detailed generation process can be found in Appendix A.2). The navigation path is represented as a sequence of topologically connected road segments $\{n_1, n_2, \ldots, n_m\} \in \mathcal{N}$. Building upon this representation, we leverage the navigation path as a high-level strategic intention prior to condition downstream trajectory generation. For anchor-based planning framework, we formulate a navigation path-conditioned probabilistic distribution over candidate planning anchors. The key innovation lies in leveraging prior knowledge of navigation path to filter the candidate planning actions, which establishes a hierarchical probability as:

$$p(a \mid \boldsymbol{\kappa}) = \underbrace{p(n \mid \boldsymbol{\kappa})}_{\text{Navigation prior}} \cdot \underbrace{p(a' \mid n, \boldsymbol{\kappa})}_{\text{Conditioned policy}}, \tag{3}$$

where $n \in \mathcal{N}$ denotes the navigation path prior, $\boldsymbol{\kappa}$ is the scene context of driving environment, and $a' \in \mathcal{A}' \subset \mathcal{A}$ represents the action subset which filtered by navigation path. Specifically, the navigation-aligned subvocabulary is obtained by selecting the top-$N_{\text{anchor}}$ anchors follows:

$$\mathcal{A}' = \operatorname{top}_{\substack{N_{\text{anchor}} \\ a_i \in \mathcal{A}}} \left( \frac{1}{K} \sum_{k=1}^{K} \max_{m} \Big[ \alpha \cdot \frac{\sigma_d}{\|\mathbf{p}_k - \mathbf{s}_m\|} + (1-\alpha) \cdot \frac{\langle \mathbf{v}_k, \mathbf{u}_m \rangle}{\|\mathbf{v}_k\| \|\mathbf{u}_m\|} \Big] \right), \tag{4}$$

where $\mathbf{p}_k$ denotes point coordinates of $a_i$, $\mathbf{s}_m$ is navigation path sampling point coordinates, $\mathbf{v}_k$ is direction vector of $a_i$, $\mathbf{u}_m$ is navigation path direction vector, $\frac{\sigma_d}{\|\mathbf{p}_k - \mathbf{s}_m\|}$ is spatial similarity, $\frac{\langle \mathbf{v}_k, \mathbf{u}_m \rangle}{\|\mathbf{v}_k\| \|\mathbf{u}_m\|}$ indicates directional similarity by cosine similarity, $\alpha$ is weighting coefficient for spatial proximity and $\sigma_d$ is scaling coefficient for spatial proximity.

This plug-in design enables seamless integration with existing anchor-based planners while injecting strong high-level guidance. By explicitly aligning anchors with the strategic intention, it significantly reduces directional deviations without compromising the generative capacity of diffusion models.

### 4.3 Intention-Driven Hierarchical Disentangled Planning

Building upon the high-level strategic intention provided by the navigation path, we further explicitly model the human cognitive hierarchy by designing our planning decoder as a hybrid of an unconscious automatic system (control level) and a conscious intention-driven system (maneuvering level).

**Unconscious Automatic Planning System:** The unconscious system models the largely automatic, millisecond-level control that dominates routine driving behavior, faithfully mirroring the cerebellum-mediated automatic control in the human cognitive hierarchy. Specifically, it performs inertial extrapolation from the ego vehicle's current state to generate a deterministic prior reference trajectory:

$$\mathbf{x}_{t+\Delta t}^{\text{inertial}} = \mathbf{x}_t + \mathbf{v}_t \cdot \Delta t, \tag{5}$$

where $\mathbf{x}_t = (x_t, y_t)$ and $\mathbf{v}_t = (v_{x,t}, v_{y,t})$ denote the ego-vehicle position and velocity vector, respectively, and $\Delta t$ is the time interval for each future pose. This inertial reference trajectory serves as the default anchor prior for the diffusion decoder in routine scenarios (i.e., when high-level intention guidance is not required). By providing a physically plausible baseline, it significantly reduces the learning burden and allows the model to focus on learning context-driven deviations.

**Conscious Intention-Driven Planning System:** The conscious system models the high-level maneuvering intentions when facing challenging interactive scenarios (e.g., obstacle avoidance, lane changes, and traffic interactions), mirroring the higher-order maneuvering-level conscious processes in human cognitive hierarchy. The lateral maneuvering intention is explicitly predicted by a dedicated intention decoder. Given the navigation features $\mathbf{F}_{\text{nav}} \in \mathbb{R}^{L \times H_{\text{nav}}}$, ego history features $\mathbf{F}_{\text{ego}} \in \mathbb{R}^{T_h \times H_{\text{ego}}}$, and the scene context $\boldsymbol{\kappa}$ produced by the scenario encoder, the decoder first applies linear projections to map all these features into a shared embedding space of dimension $D_{\text{int}}$ and concatenates them:

$$\mathbf{F}_{\text{concat}} = [\text{Proj}_{\text{nav}}(\mathbf{F}_{\text{nav}});\ \text{Proj}_{\text{ego}}(\mathbf{F}_{\text{ego}});\ \text{Proj}_{\kappa}(\boldsymbol{\kappa})]. \tag{6}$$

Then, we employ stacked self-attention blocks over $\mathbf{F}_{\text{concat}}$ to capture interactions among navigation, ego history, and scene contexts:

$$\mathbf{Q}_{\text{int}} = \text{SelfAttn}(\mathbf{F}_{\text{concat}}, \mathbf{F}_{\text{concat}}, \mathbf{F}_{\text{concat}}), \tag{7}$$

where $\mathbf{Q}_{\text{int}} = \{\mathbf{q}_i\}_{i=1}^{K}$ denotes the set of $K$ intention queries. These queries are subsequently passed through a set of parallel MLP prediction heads to obtain the corresponding intention probability:

$$p_i = \text{Softmax}\big(\text{MLP}_{\text{head}}(\mathbf{q}_i)\big), \quad i = 1, \ldots, K. \tag{8}$$

We subsequently derive the one-hot encoded intention category from the probability distribution and feed it through a lightweight encoder $E_{\text{int}}(\cdot)$ to produce the final explicit intention embeddings $\mathbf{c}_{\text{int}} = E_{\text{int}}(\mathbf{y}_i) \in \mathbb{R}^{D_{\text{int}}}$, which serve as maneuvering-level conditional guidance for the planning decoder. They are injected into the planning decoder via adaptive LayerNorm (AdaLN) (Peebles & Xie, 2023) modules through successive scale-shift operations. Specifically, given intermediate hidden features $\mathbf{h}$, the AdaLN operation is formulated as:

$$\text{AdaLN}(\mathbf{h}, \mathbf{c}_{\text{int}}) = \gamma(\mathbf{c}_{\text{int}}) \odot \frac{\mathbf{h} - \mu}{\sigma} + \beta(\mathbf{c}_{\text{int}}), \tag{9}$$

where $\gamma(\cdot)$ and $\beta(\cdot)$ are the scale and shift parameters generated from $\mathbf{c}_{\text{int}}$ via an MLP, and $\mu, \sigma$ are the mean and standard deviation computed by Layer Normalization.

A dedicated longitudinal refinement head then further refines this lateral drivepath $\hat{\mathbf{x}}_0 \in \mathbb{R}^{H_w \times 2}$ generated by diffusion decoder. Implemented as an MLP, this head takes the lateral waypoints as input and predicts 1D longitudinal displacements $\mathbf{d}_t \in \mathbb{R}$ for each future timestep. The lateral branch

produces a geometrically coherent drivepath under maneuvering intention, while the longitudinal head specializes in velocity adjustments.

**Unified Hierarchical Planning System:** The unconscious and conscious systems are unified within a single diffusion decoder through classifier-free guidance (CFG) (Ho & Salimans, 2022), enabling the model to learn both the planning behavior with intention conditions and the behavior without intention conditions. During training, the conscious intention is randomly dropped with probability $p_{\text{drop}} = 0.2$. At inference, the guided noise prediction is computed as

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, \mathbf{c}_{\text{int}}, \boldsymbol{\kappa}) = (1 - w)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \emptyset, \boldsymbol{\kappa}) + w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}_{\text{int}}, \boldsymbol{\kappa}), \tag{10}$$

where $w$ is the guidance scale that amplifies the difference between the conditional and unconditional distributions, enabling effective scaling of high-level intention guidance while gracefully supporting unconditional trajectory generation. This dynamic fusion ensures that the output trajectories are strongly guided by both the conscious decoupled maneuvering intention and the unconscious physical prior while improving robustness when intention predictions are uncertain.

# 5 Experiments

## 5.1 Evaluation Setups

We extensively evaluate our method on the NAVSIM (Dauner et al., 2024) dataset. The original NAVSIM dataset consists of over a thousand clips captured with eight cameras and one LiDAR, with planning annotations provided at a frequency of 2 Hz. Navigation paths are additionally generated for each frame using our novel toolchain. The NAVSIM leaderboard employs non-reactive simulation environments and closed-loop evaluation metrics, providing a fair and reliable assessment of planning performance. To address the inherent data imbalance, we conduct a thorough analysis and screening of the NAVSIM dataset. Specifically, we select the long-tail scenarios (i.e., rare and challenging cases) for further evaluation (detailed information in Appendix A.4). This targeted validation on difficult, less-represented situations enables a more rigorous assessment of our Intention Planner's effectiveness and robustness in complex real-world autonomous driving conditions.

## 5.2 Implementation Details

We adopt the same perception modules and ResNet-34 (He et al., 2016) backbone as Transfuser for fair comparison. We stack two cascaded diffusion decoder layers and apply a truncated diffusion policy with 20 clustered anchors. The training diffusion schedule is truncated at 50 out of 1000 steps to diffuse the anchors, while during inference we keep 2 denoising steps. The training and inference recipe directly follows DiffusionDrive (Liao et al., 2024). Specifically, we use three cropped and downscaled forward-facing camera images, concatenated into a single $1024 \times 256$ image, together with a rasterized BEV LiDAR as input. Our model is trained from scratch on the navtrain split for 100 epochs using the AdamW optimizer on 4 NVIDIA A100 GPUs with a total batch size of 512 and a learning rate of $6 \times 10^{-4}$.

## 5.3 Comparison with State-of-the-Art Methods

As shown in Table 1, our proposed Intention Planner achieves superior performance on the NAVSIMv1 navtest split, outperforming previous leading baselines across nearly all metrics. It attains a strong overall PDMS score of 91.5. These results demonstrate clear advantages in safety, comfort, and efficiency. Table 2 further highlights the superior robustness of our method on long-tail scenarios. While most existing approaches suffer from significant performance degradation in these challenging cases, Intention Planner not only maintains strong robustness but also achieves additional performance gains, attaining an even higher PDMS score of 92.1. We also compare Intention Planner with VLM/VLA-based planners on the navtest benchmark, as shown in Table 4. Intention Planner achieves the highest PDMS score while maintaining substantially higher inference efficiency of 22.59 Hz (44.27 ms per frame).

Table 1: Performance on the NAVSIMv1 navtest benchmark with closed-loop metrics. "C" denotes camera, and "L" denotes LiDAR. Best results are marked in bold. For fair comparison, all methods use the official scores from versions with the same ResNet-34 backbone.

| Method | Input | Backbone | NC↑ | DAC↑ | TTC↑ | Comf.↑ | EP↑ | PDMS↑ |
|---|---|---|---|---|---|---|---|---|
| VADv2 (Jiang et al., 2023) | C & L | ResNet-34 | 97.2 | 89.1 | 91.6 | **100** | 76.0 | 80.9 |
| UniAD (Hu et al., 2023) | C & L | ResNet-34 | 97.8 | 91.9 | 92.9 | **100** | 78.8 | 83.4 |
| PARA-Drive (Weng et al., 2024) | C & L | ResNet-34 | 97.9 | 92.4 | 93.0 | 99.8 | 79.3 | 84.0 |
| TransFuser (Chitta et al., 2023) | C & L | ResNet-34 | 97.7 | 92.8 | 92.8 | **100** | 79.2 | 84.0 |
| GoalFlow (Xing et al., 2025) | C & L | ResNet-34 | 98.3 | 93.8 | 94.3 | **100** | 79.8 | 85.7 |
| Hydra-MDP (Li et al., 2024) | C & L | ResNet-34 | 98.3 | 96.0 | 94.6 | **100** | 78.7 | 86.5 |
| DiffusionDrive (Liao et al., 2024) | C & L | ResNet-34 | 98.2 | 96.2 | 94.7 | **100** | 82.2 | 88.1 |
| WoTE (Li et al., 2025) | C & L | ResNet-34 | 98.5 | 96.8 | 94.9 | 99.9 | 81.9 | 88.3 |
| DIVER (Song et al., 2025) | C & L | ResNet-34 | 98.5 | 96.5 | 94.9 | **100** | 82.6 | 88.3 |
| DriveSuprim (Yao et al., 2026) | C & L | ResNet-34 | 97.8 | 97.3 | 93.6 | **100** | 86.7 | 89.9 |
| DiffusionDrive v2 (Zou et al., 2025) | C & L | ResNet-34 | 98.3 | 97.9 | 94.8 | 99.9 | **87.5** | 91.2 |
| **Intention Planner (Ours)** | C & L | ResNet-34 | **98.7** | **98.3** | **95.3** | **100** | 86.9 | **91.5** |

Table 2: Performance of different methods on the long-tail scenarios of NAVSIMv1 navtest split. "C" denotes camera, and "L" denotes LiDAR. Best results are marked in bold.

| Method | Input | Backbone | NC↑ | DAC↑ | TTC↑ | Comf.↑ | EP↑ | PDMS↑ |
|---|---|---|---|---|---|---|---|---|
| DiffusionDrive (Liao et al., 2024) | C & L | ResNet-34 | 98.4 | 95.7 | **95.5** | 99.9 | 80.5 | 87.3 |
| GoalFlow (Xing et al., 2025) | C & L | V2-99 | 98.2 | 92.9 | 83.2 | 99.9 | 83.2 | 87.3 |
| DiffusionDrive v2 (Zou et al., 2025) | C & L | ResNet-34 | 97.8 | 96.6 | 95.0 | 99.9 | 85.2 | 89.5 |
| **Intention Planner (Ours)** | C & L | ResNet-34 | **99.0** | **98.0** | **95.5** | **100** | **88.6** | **92.1** |

### 5.4 Qualitative Comparison

As shown in Figure 3, we compare trajectory predictions by different methods across representative challenging urban scenarios, including right lane changes with static vehicle nudging and complex multi-lane intersections. DiffusionDrive frequently generates unsafe trajectories that drive out of the drivable area or exhibit significant directional deviation. Meanwhile, DiffusionDrive v2 suffers from mode collapse in several scenarios, producing nearly identical predictions across different modes. In contrast, our Intention Planner generates clear, human-like style trajectories with explicit intention modeling. It accurately executes specific driving intentions, such as right lane change, left lane change, overtaking static vehicles, nudging, and lane keeping, demonstrating superior planning quality and robustness in complex real-world driving situations.

### 5.5 Ablation Study

To validate the effectiveness of each key component in Intention Planner, we conduct a systematic ablation study on the NAVSIMv1 navtest split as shown in Table 3 . Following the incremental design in our methodology, we start from a diffusion-based baseline and progressively add our proposed modules. The main metric is the overall planning decision-making score (PDMS). These results clearly demonstrate that each proposed component contributes positively to the performance.

Table 3: Ablation study of key components in Intention Planner on the NAVSIMv1 navtest benchmark. PDMS scores are reported for the full test split.

| Components | PDMS↑ |
|---|---|
| Base Model | 88.1 |
| + Longitudinal Refinement Module (Sec. 4.3) | 88.3 |
| + Navigation Path Fusion as Strategic Intention (Sec. 4.2) | 89.4 |
| + Maneuvering Intention Instruction (Sec. 4.3) | 91.3 |
| + Hybrid System via Classifier-Free Guidance (Sec. 4.3) | 91.5 |

Table 4: Comparison with VLM/VLA-based planners on the NAVSIMv1 navtest benchmark.

| Method | NC↑ | DAC↑ | TTC↑ | Comf.↑ | EP↑ | PDMS↑ | Frequency(Hz)↑ |
|---|---|---|---|---|---|---|---|
| VGGDrive (Wang et al., 2026) | 98.6 | 96.3 | 95.6 | **100.0** | 82.9 | 88.8 | 0.96 |
| AutoVLA (Zhou et al., 2025) | 98.4 | 95.6 | **98.0** | 99.9 | 81.9 | 89.1 | 0.93 |
| DriveVLA-W0 (Li et al., 2026b) | 98.7 | **99.1** | 95.3 | 99.3 | 83.3 | 90.2 | 13.46 |
| WAM-Flow (Xu et al., 2026) | **99.2** | 98.3 | 97.0 | 99.7 | 82.3 | 90.3 | 11.11 |
| ReCogDrive (Xiong et al., 2026) | 97.9 | 97.3 | 94.9 | **100.0** | **87.3** | 90.8 | 13.33 |
| SGDrive (Li et al., 2026a) | 98.6 | 97.8 | 96.2 | **100.0** | 85.8 | 91.1 | 1.10 |
| **Intention Planner (Ours)** | 98.7 | 98.3 | 95.3 | **100.0** | 86.9 | **91.5** | **22.59** |

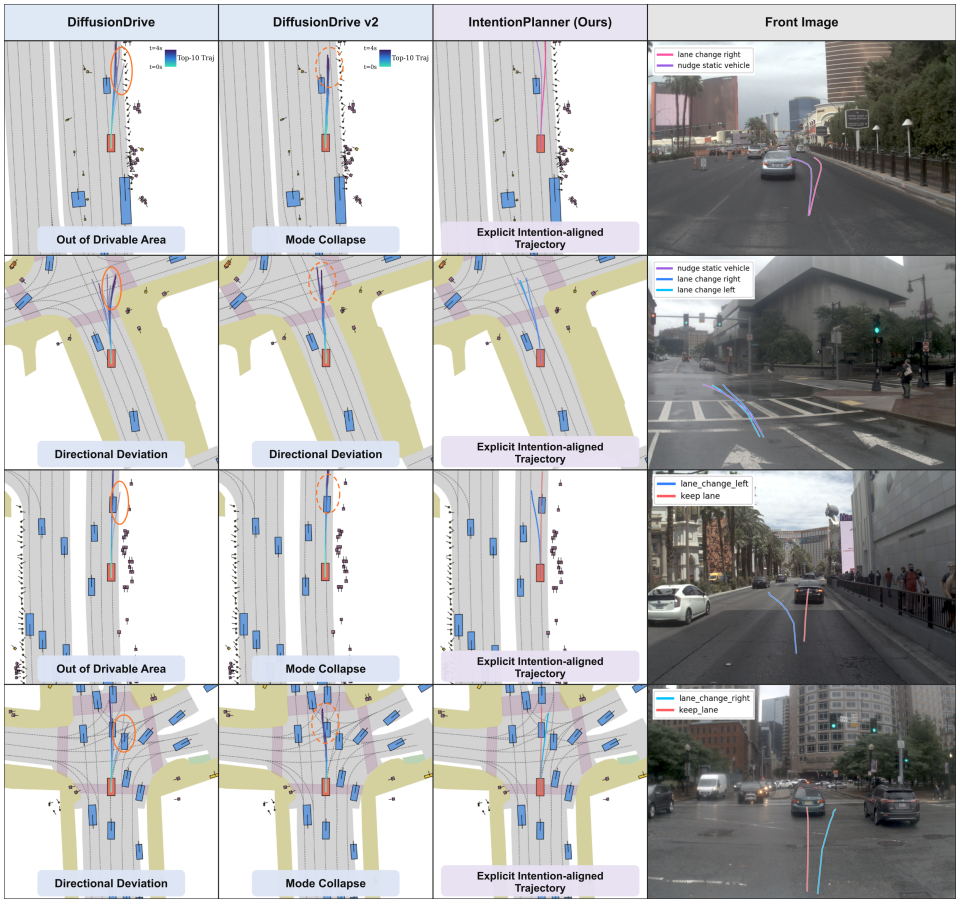



Figure 3: Qualitative comparison of trajectory generation. DiffusionDrive generates unsafe trajectories out of the drivable area or exhibiting directional deviation. DiffusionDrive v2 suffers from severe mode collapse. In contrast, our Intention Planner produces explicit intention-aware trajectories that clearly convey specific driving intentions (e.g., lane changes, overtaking, nudging and lane keeping).

## 6 Conclusion

In this paper, we presented Intention Planner, a cognitively-inspired hierarchical planning framework for end-to-end autonomous driving. By explicitly modeling hierarchical intention representations, our method effectively mitigates causal confusion and multi-modal deficiencies prevalent in existing planners. Extensive experiments on the NAVSIM benchmark demonstrate that Intention Planner substantially outperforming prior baselines in both standard and long-tail scenarios. Qualitative results further highlight our framework's ability to produce intention-aligned behaviors where previous methods suffer from unsafe deviations or severe mode collapse.

### AI USE STATEMENT

In this work, we used generative AI tools for language refinement, grammar checking, and improving the clarity and readability of the manuscript. We have not used generative AI tools for generating research ideas, designing the proposed method, or producing scientific claims and conclusions, and these tasks are not applicable to this work. Additionally, we used generative AI tools for assisting with minor LaTeX formatting and writing organization during manuscript preparation. We have reviewed all AI-assisted work. Specifically, all AI-assisted revisions were manually checked by the authors for technical correctness, consistency with the proposed method, and accuracy of the presented claims. Any AI-assisted suggestions related to implementation details or formatting were verified before being incorporated into the final manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

X. Chen et al. Int2planner: An intention-based multi-modal motion planner for integrated prediction and planning. *arXiv preprint arXiv:2501.12799*, 2025.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13867–13882, 2023.

Daniel Dauner et al. Navsim: Data-driven non-reactive autonomous vehicle simulation. In *NeurIPS Datasets and Benchmarks Track*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16053–16062, 2023.

Yi Huang, Zhan Qu, Lihui Jiang, Bingbing Liu, and Hongbo Zhang. Prioritizing perception-guided self-supervision: A new paradigm for causal modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2511.08214*, 2025.

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. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8275–8285, 2023.

Gerry Leisman, Orit Braun-Benjamin, and Robert Melillo. Cognitive-motor interactions of the basal ganglia in development and diseases. *Frontiers in Systems Neuroscience*, 8:16, 2014. doi: 10.3389/fnsys.2014.00016.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4032–4042, June 2026a.

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. *arXiv preprint arXiv:2504.01941*, 2025.

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

Z. Li et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. *arXiv preprint arXiv:2411.15139*, 2024.

Yuhang Lu, Jiadong Tu, et al. Real-ad: Towards human-like reasoning in end-to-end autonomous driving. *arXiv preprint arXiv:2507.12499*, 2025.

Katie Z Luo, Xinshuo Weng, Yan Wang, Shuang Wu, Jie Li, Kilian Q Weinberger, Yue Wang, and Marco Pavone. Augmenting lane perception and topology understanding with standard definition navigation maps. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4029–4035. IEEE, 2024.

John A. Michon. A critical view of driver behavior models: What do we know, what should we do? In Leonard Evans and Richard C. Schwing (eds.), *Human Behavior and Traffic Safety*, pp. 485–524. Plenum Press, New York, 1985. doi: 10.1007/978-1-4613-2173-6_19.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Diver: Reinforced diffusion breaks imitation bottlenecks in end-to-end autonomous driving. *arXiv preprint arXiv:2507.04049*, 2025.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024.

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

Jie Wang, Guang Li, Zhijian Huang, Chenxu Dang, Hangjun Ye, Yahong Han, and Long Chen. Vg-gdrive: Empowering vision-language models with cross-view geometric grounding for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10954–10964, June 2026.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Z. Xing et al. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. *arXiv preprint arXiv:2503.05689*, 2025.

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026.

Yifang Xu, Jiahao Cui, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Feipeng Cai, Neng Zhang, Yaoyi Li, Jia Cai, et al. Wam-flow: Parallel coarse-to-fine motion planning via discrete flow matching for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24918–24928, 2026.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11910–11918, 2026.

Y. Zheng et al. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. *arXiv preprint arXiv:2507.00603*, 2025a.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. *arXiv preprint arXiv:2501.15564*, 2025b.

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. *arXiv preprint arXiv:2510.08562*, 2025c.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38:27920–27956, 2025.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

# A Appendix

## A.1 Investigation: Missing of Strategic Intention leads to Causal Misidentification

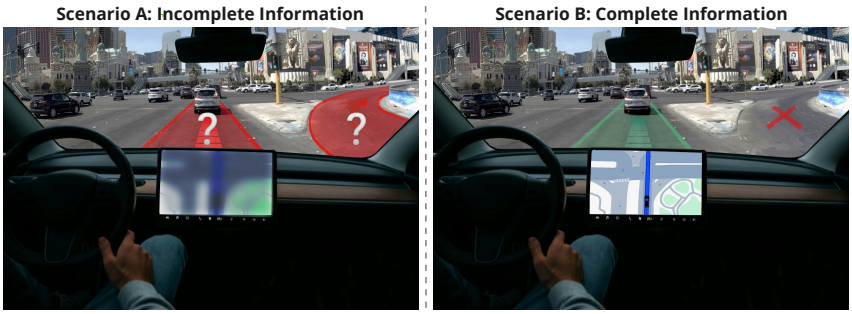



Figure 4: Causal misidentification due to the missing key information, navigation path. The vehicle is in front of an intersection, which can go forward and turn right. But based on the driver's intention (known from ground truth), the planner should learn to go forward. Scenario A relies on the traffic light or other traffic participant behaviors to decide where to go. Scenario B instead correctly attends to the root cause, which is navigation path as shown on the in-car display.

Let us imagine how a human drives to a new location: Firstly, the driver sets a destination goal. Then, the navigation system generates a navigation path consisting of an ordered sequence of roads, to guide the journey to the new location. The navigation path provides an explicit, high-level instruction for the vehicle's intended trajectory. Building on this insight and integrating it with prior analyses of IL, we identify a significant shortcoming in existing research: the absence of explicit modeling of navigation paths leads to causal misidentification in planning tasks. Figure 4 shows how the problem happens in the IL-based planners. The IL-planner learns to go straight when the vehicle is allowed to go straight and turn right at the same time. By using IL, when the model cannot distinguish the cause and effect of actions in the environment, causal confusion often occurs during the learning process. This is because the model fails to find the root cause of these actions: the navigation path as shown on the in-car display. Some works also illustrate it as a "copycat" problem in the context of IL planning. Therefore, we believe the mentioned issue is because existing frameworks predominantly focus on leveraging live perception and prediction data while ignoring higher-level driving intent. Although these systems are excellent at understanding the immediate environment, they lack a robust mechanism to encode long-term navigation paths or constraints into the planning process.

## A.2 Data Preparation: How to generate Navigation Path for Strategic Prior

In this section, we introduce the whole process of navigation path generation, including preprocessing the open-source navigation (or SD) map and vehicle's GPS trajectories, recovering the navigation path, and cropping in a region of interest (ROI) for each single frame, as shown in Figure 5.

### A.2.1 Preprocessing

Given a scene of driving raw data from any autonomous driving dataset, we firstly extract the GPS trajectory points $\mathbf{P} = \{p_n \mid n = 1, \ldots, N\}$. Each point $p_n$ needs to be aligned with the map coordinate system. To achieve this, we transform $p_n = (x_n^{\text{global}}, y_n^{\text{global}})$ from the global coordinate system to the point $p_n^{\text{map}} = (x_n^{\text{map}}, y_n^{\text{map}})$ in map coordinate system, such as the whole genome sequencing 84 (WGS84), using a homogeneous transform defined as

$$\mathbf{T}_{\text{global2map}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \quad \begin{bmatrix} x_n^{\text{map}} \\ y_n^{\text{map}} \\ 1 \end{bmatrix} = \mathbf{T}_{\text{global2map}} \begin{bmatrix} x_n^{\text{global}} \\ y_n^{\text{global}} \\ 1 \end{bmatrix}.$$

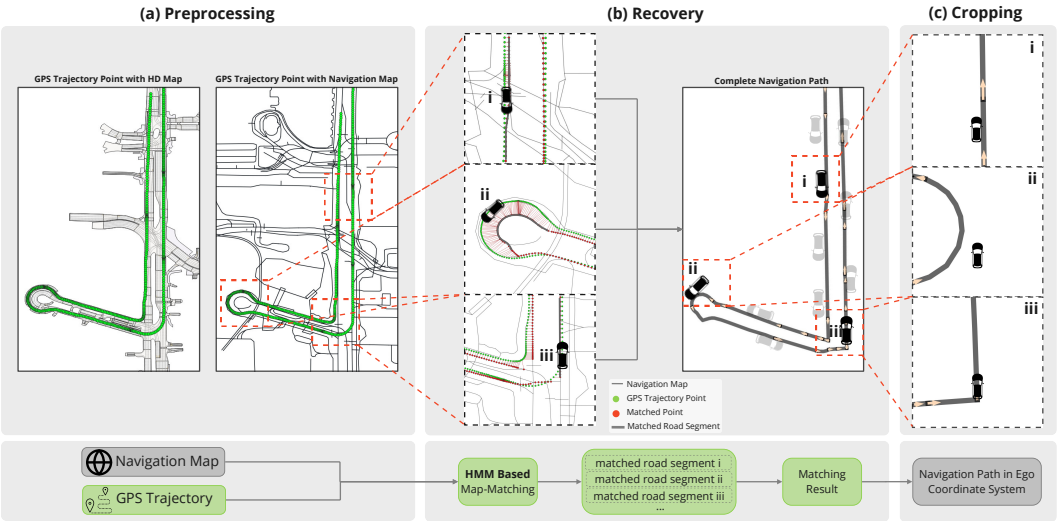



Figure 5: Overall architecture of generating the navigation path. The vehicle drives from **i** to **ii** to **iii**. **(a)** shows the input of the navigation path recovery module: trajectory points $\{p_n^{\text{map}} \mid n = 1, \ldots, N\}$ (as green dots) and the corresponding navigation map $G_{BBox}(V, E)$. On the left side of **(a)**, we also render the HD map as a reference. **(b)** shows the process of recovering a navigation path $\mathcal{N}$ by HMM. **(c)** illustrates the corresponding navigation paths in the ego coordinate system $\mathcal{N}_i^{\text{ego}}, \mathcal{N}_{ii}^{\text{ego}}, \mathcal{N}_{iii}^{\text{ego}}$ at different positions as **i**, **ii**, and **iii**. Please zoom in for more details.

After transforming all GPS trajectory points, we obtain the set of transformed coordinates $\{p_1^{\text{map}}, \ldots, p_N^{\text{map}}\}$. Further, we calculate the minimum bounding box of the trajectory as:

$$\text{BBox} = \left[\min_i(x_i^{\text{map}}), \min_i(y_i^{\text{map}}), \max_i(x_i^{\text{map}}), \max_i(y_i^{\text{map}})\right],$$

where $(x_i^{\text{map}}, y_i^{\text{map}})$ are the 2D coordinates of each transformed point $p_i^{\text{map}}$. Using this bounding box, we retrieve the corresponding navigation map data via the OSM API, as shown in Figure 5 **(a)**. As a result, we construct a road network as a directed graph $G_{BBox}(V, E)$, where $V$ represents the set of road intersections and $E$ denotes the set of road segments within the bounding box. Each road segment $e_n \in E$ is associated with a set of attributes, including '$RoadID$', '$RoadType$', '$LaneNumber$', and '$RoadLength$'. Moreover, each segment is represented as a sequence of '$SamplingPoint$' as $\mathbf{S} = \{s_n \mid n = 1, \ldots, N\}$, where $s_n$ corresponds to a discrete spatial coordinate along the road segment.

### A.2.2 Navigation Path Recovery

A navigation path $\mathcal{N}$ can be defined as a sequence of topologically connected road segments $\{e_i, e_j, \ldots, e_m\} \in E$. To achieve that, we employ a Hidden Markov Model (HMM)-based algorithm, which assumes that the observed trajectory is generated from a sequence of hidden states. The set of observed variables is defined as $\mathbf{X} = \{x_n \mid n = 1, \ldots, N\}$, where each $x_n$ represents the spatial information of $p_n^{\text{map}}$. Similarly, the hidden variables is given by $\mathbf{Y} = \{y_n \mid n = 1, \ldots, N\}$, where each $y_n$ corresponds to the road segment location associated with $p_n^{\text{map}}$.

The observation probability of each candidate segment is first calculated based on the HMM framework. Subsequently, the matching probability for each candidate segment is determined. The HMM algorithm then matches the observed trajectory to the most probable road segment, as a *"matched road segment"*.

### A.2.3 Ego-Coordinate Alignment and ROI Cropping

After recovering the complete navigation path $\mathcal{N}$ of the whole trajectory, we transform it into the ego coordinate system and crop it to fit the ROI in each frame. The transformed navigation path of timestamp $n$ in the ego coordinate system is represented as $\mathcal{N}_n^{\text{ego}}$. As illustrated in Figure 5 **(c)**, this process ensures that the navigation path $\mathcal{N}_n^{\text{ego}}$ is ready for further processing and analysis.




Figure 6: Qualitative comparison of original DiffusionDrive (Liao et al., 2024) and our navigation-path-enhanced DiffusionDrive on challenging scenes of NAVSIM navtest split. These figures compare the impact of navigation paths on anchors across different scenarios, illustrating the differences between planning with and without navigation guidance. In each row, the first column shows the front camera image with the ground truth as the reference (green lane). The second column shows the original fixed 20 anchors as dotted lines by K-Means from DiffusionDrive (Liao et al., 2024). The third column shows the top 20 trajectories from the official release version of DiffusionDrive. The fourth and fifth columns show our anchors and trajectories. The generated navigation path is shown as gray line. Ego vehicle is marked in red; other vehicle agents are in blue; the ground truth is shown as thick orange-yellow line. We also render the HD map as reference, which is not the input of our framework.

### A.3 More Experiments about Navigation Path Prior

To further demonstrate the effectiveness, generalization ability, and robustness of the navigation path as a high-level strategic prior (introduced in Sec. 4.2 and Appendix A.1), we conduct additional experiments on both anchor-based and non-anchor-based baselines. These studies complement the main ablation results in Table 3 by directly comparing the navigation path against discrete navigation commands and evaluating performance when the navigation path is unavailable during inference.

#### A.3.1 Plug-in Integration with Anchor-based Baselines.

For anchor-based methods, we adopt VADv2 (Jiang et al., 2023) and DiffusionDrive (Liao et al., 2024) as strong baselines and evaluate on the NAVSIM dataset. We keep their original planner modules and all other experimental setups unchanged. Our navigation-path-guided anchor filter is integrated purely as a lightweight pre-processing module without any parameter tuning or retraining of the planner.

Specifically, we retain the same anchor vocabulary size as the baselines: 4096 or 8192 anchors for VADv2 and 20 anchors for DiffusionDrive (as visualized in Fig. 6). By fusing the continuous geometric and directional information from the navigation path, the selected anchors become significantly

| Planning Method | Anchor | PDMS↑ |
|---|---|---|
| UniAD (Hu et al., 2023) | None | 83.4 |
| Hydra-MDP (Li et al., 2024) | None | 83.0 |
| Transfuser (Chitta et al., 2023) | None | 84.0 |
| VADv2 (Jiang et al., 2023) | 4096 | 80.2 |
| VADv2 (Jiang et al., 2023) + Command | 4096 | **81.0** (+1.0) |
| VADv2 (Jiang et al., 2023) + Path | 4096 | **83.1** (+3.6) |
| VADv2 (Jiang et al., 2023) | 8192 | 80.9 |
| VADv2 (Jiang et al., 2023) + Command | 8192 | **81.4** (+0.6) |
| VADv2 (Jiang et al., 2023) + Path | 8192 | **83.3** (+3.0) |
| DiffusionDrive (Liao et al., 2024) | 20 | 88.1 |
| DiffusionDrive (Liao et al., 2024) + Command | 20 | **88.7** (+0.7) |
| DiffusionDrive (Liao et al., 2024) + Path | 20 | **89.4** (+1.5) |

Table 5: Comparison of navigation path prior versus navigation command-based anchor filtering on the NAVSIM navtest split (closed-loop PDMS). We integrate our navigation-path-guided anchor filter as a lightweight plug-in module into strong anchor-based baselines without retraining the planner.

more reasonable and representative, leading to reduced directional deviations and improved planning quality with negligible computational overhead.

### A.3.2 Comparison with Navigation Command-based Anchor Filtering.

A natural alternative for anchor filtering is to use coarse discrete navigation commands (e.g., *left*, *right*, *forward*, *unknown*). To validate the superiority of our continuous navigation path prior, we implement a competitive navigation-command-based anchor filter as a strong baseline. Specifically, we classify the orientation vectors between the starting and ending points of each anchor into four categories according to angular intervals: *left* ($[-135°, -30°)$), *forward* ($[-30°, 30°)$), *right* ($[30°, 135°)$), and *unknown* for the remaining angles. All other experimental settings (anchor number, training schedule, etc.) remain identical.

As shown in Table 5, the navigation-command-based approach yields only marginal improvements over the vanilla baseline, while our navigation-path-based fusion achieves substantially superior performance across all PDMS metrics (NC, DAC, TTC, Comf., EP). This result confirms that the geometrically rich, long-horizon, and semantically precise nature of the navigation path provides much stronger strategic guidance than discrete commands, effectively mitigating causal confusion and directional deviations.

## A.4 Data Preparation: Mining Long-Tail Scenarios and Recognizing Driver Intentions

To systematically identify challenging long-tail scenarios and accurately capture the true intentions of expert drivers from raw trajectories, we develop a dedicated data mining toolchain specifically tailored for the NAVSIM dataset. This toolchain introduces a three-level hierarchical atomic labeling framework that enables fine-grained annotation of ego-vehicle behavior by decomposing complex driving sequences into atomic trajectory tags, quality evaluations, and semantic scenario classifications.

**Level 1 (Trajectory Tagging)** defines a comprehensive set of atomic labels that describe fundamental motion and environmental attributes of each trajectory frame. These tags include *Turn Type*, *Lane Change*, *Acceleration Type*, *Velocity Type*, *Jerk Type*, *Intersection Stage*, *Traffic Light* status, *Lane Positioning*, *Current Lane Curvature*, *Overtake or Follow Target*, etc. Most tags are automatically extracted from the provided planning annotations and generated navigation paths, with several advanced tags planned for future extension.

**Level 2 (Driving Behavior Evaluation)** performs rule-based quality assessment by combining Level 1 tags to evaluate critical aspects of driving comfort, safety, and stability. For instance, aggressive acceleration combined with high jerk is labeled as *Poor* Acceleration Comfort; unsafe lane

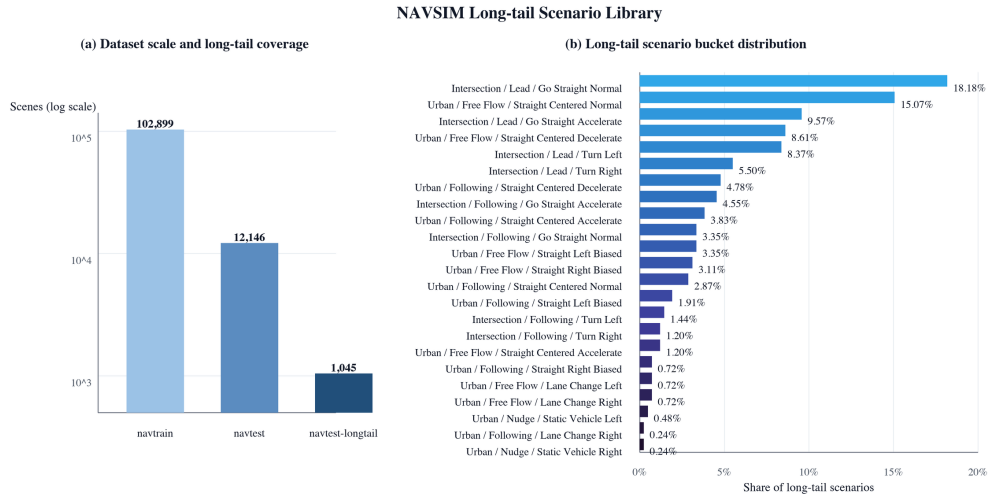



Figure 7: Statistical information of NAVSIM long-tail scenario library. (a) Dataset scale comparison among the original NAVSIM training set, evaluation set, and the constructed long-tail navtest subset. (b) Distribution of long-tail scenarios across different scenario buckets, where each category is defined by the combination of road structure, interaction type, and ego-vehicle maneuver.

positioning on narrow urban roads is flagged as *Risky* Narrow Road Handling; and proper deceleration responses to cut-in events are recognized as *Good* Cut-in Response. This level provides quantitative indicators of behavioral quality and highlights unsafe or uncomfortable maneuvers.

**Level 3 (Driving Behavior Classification)** organizes the tagged trajectories into a hierarchical semantic scenario taxonomy. Primary scenarios include *Intersection* and *Urban Road*, with secondary and third-level categories such as "Green Light Straight Following" (INT-S-GF), "Urban Straight Following Left-biased" (URB-F-SL-L), and "Urban Overtake Moving Vehicle Bypass" (URB-O-DY). Each category is defined by precise logical combinations of Level 1 tags, enabling automatic and interpretable scenario labeling.

By applying this toolchain and the three-level framework to the NAVSIM trajectories, we mine long-tail scenarios—i.e., rare and difficult cases that constitute only a small fraction of the dataset (e.g., specific intersection maneuvers with poor quality labels, biased lane-keeping in curves, or complex overtake/nudge behaviors), as shown in Figure 7. These extracted trajectories are curated into a dedicated NAVSIM long-tail scenario library, which serves as a focused benchmark for evaluating the robustness and generalization capability of Intention Planner. This targeted data preparation strategy directly addresses the severe class imbalance in autonomous driving datasets and provides a rigorous testbed for validating true driver intention recognition and safe planning in challenging real-world conditions.

### A.5 Maneuver Intention Taxonomy and Supervision

We represent each maneuver intention as a joint lateral–longitudinal label $y = (y^{\text{lat}}, y^{\text{lon}})$. The lateral taxonomy consists of 11 categories: lane keeping, left turn, right turn, left-biased lane keeping, right-biased lane keeping, lane change left, lane change right, overtaking left, overtaking right, nudging left, and nudging right. The longitudinal taxonomy consists of three categories: constant speed, acceleration, and deceleration. Their product defines a vocabulary of 33 interpretable maneuver intentions.

During training, each expert trajectory is automatically assigned a maneuver label using the trajectory-tagging pipeline described in Appendix A.4. The resulting ground-truth intention labels are used only as supervisory targets for the intention decoder and are never provided as model inputs. At inference time, neither the expert future trajectory nor its ground-truth intention label is available. Instead, the intention decoder predicts a probability distribution over all 33 maneuver classes using only the

Table 6: Top-$K$ accuracy of the predicted maneuver intention distribution on the navtest split.

| Metric | Top-1 | Top-3 | Top-5 | Top-7 |
|---|---|---|---|---|
| Overall Acc | 87.12% | 91.19% | 95.74% | 97.64% |

navigation, ego-state, and scene features available to the planner. The Top-$K$ predicted intention hypotheses are subsequently embedded and used to condition the trajectory generator. We use $K = 7$ in all experiments.

The taxonomy does not contain a separate "no-intention" class. Routine driving behavior is represented by explicit combinations such as lane keeping with constant speed. In our ablation study, the intention-free variant refers to removing the maneuver-intention condition from the trajectory generator rather than introducing an additional null-intention category.

### A.6 Intention Consistency Analysis

To verify whether the predicted maneuver intention representations accurately reflect human driving behaviors, we conduct an intention consistency analysis on the NAVSIM navtest split, which contains 12,146 driving scenes.

First, we evaluate whether the final selected trajectory is aligned with the predicted maneuvering intention. Specifically, we extract the maneuver intention represented by the selected trajectory and compare it with the ground-truth driving intention. The results show that in 87.02% of evaluated scenes, the selected trajectory exhibits consistent maneuver semantics with the expert intention, demonstrating that the predicted intentions establish strong correspondence between semantic decisions and generated trajectories.

Furthermore, we analyze the predicted intention distribution during inference. Instead of committing to a single deterministic intention, our intention decoder predicts a probability distribution over 33 maneuver categories and retains the Top-$K$ high-probability intentions as semantic candidates for downstream trajectory generation. In our experiments, we use Top-7 intention hypotheses to preserve potential behavioral uncertainty. The Top-$K$ intention prediction accuracy is summarized in Tab. 6.

The results show that the ground-truth intention is covered by the Top-7 predictions in 97.64% of evaluated scenes, indicating that the predicted intention distribution accurately captures the uncertainty of expert driving decisions.

Importantly, preserving multiple intention hypotheses is essential because many driving scenarios admit several plausible maneuvers before the final decision is made. Rather than forcing all candidate trajectories to share a single intention representation, our framework explicitly maintains uncertainty at the semantic decision level and propagates it to the downstream trajectory generator. This design enables the planner to produce diverse trajectory candidates while maintaining clear semantic correspondence between generated behaviors and underlying driving intentions.

### A.7 More Visualization of Generated Trajectories

We conducted an extensive comparative visualization of the trajectory generation capabilities of our Intention Planner, DiffusionDrive (Liao et al., 2024), and DiffusionDrive v2 (Zou et al., 2025) across diverse driving scenarios, as shown in Figure 8 and Figure 9.




Figure 8: Comparative visualization on navtest dataset.

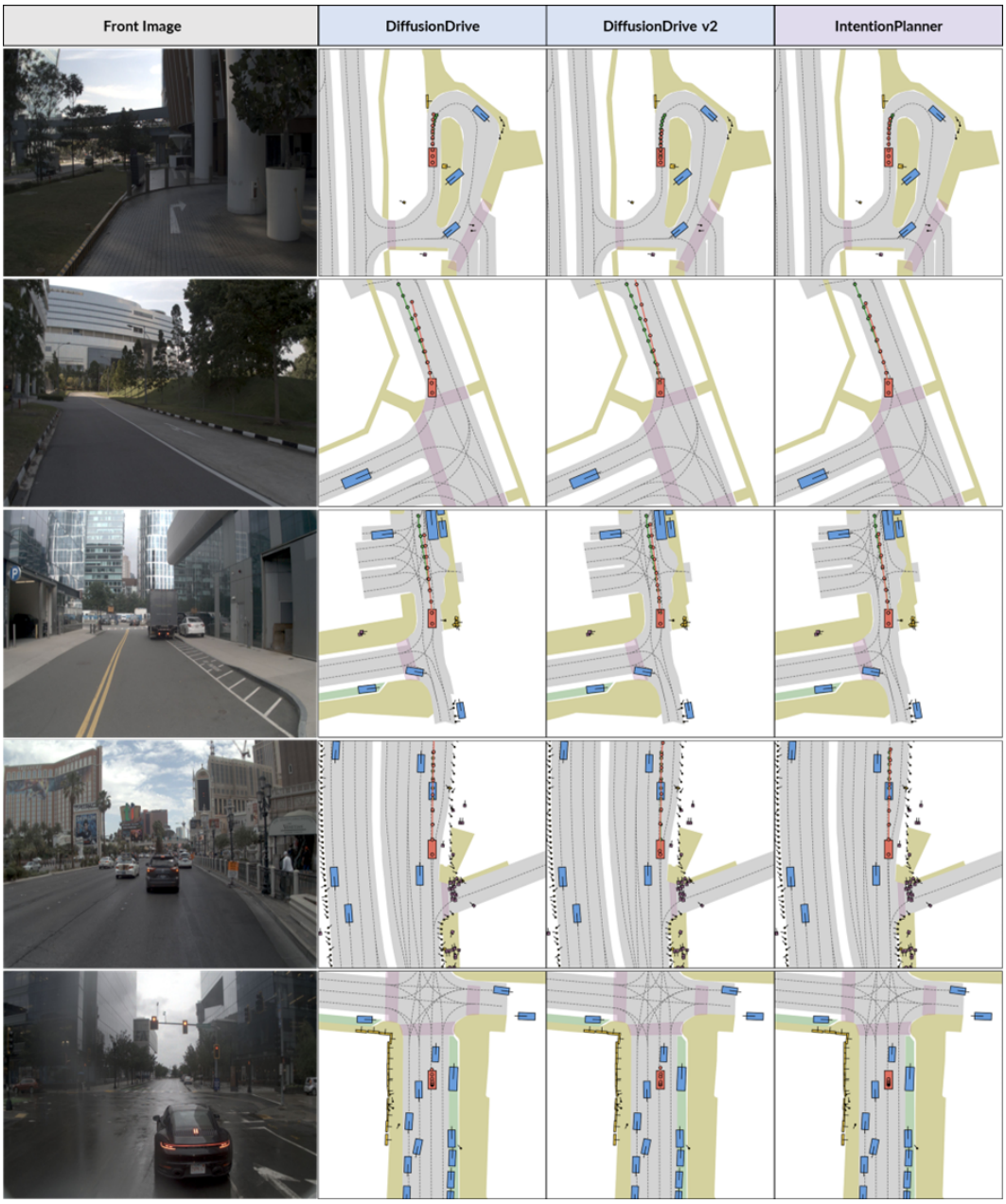



Figure 9: Comparative visualization on navtest dataset.

### A.8 Granular Inference-time Breakdown

We also conduct a detailed inference-time profiling analysis for DiffusionDrive Liao et al. (2024), DiffusionDriveV2 Zou et al. (2025), and our Intention Planner under identical settings on a single NVIDIA A100 GPU. The overall latency comparison is summarized in Tab. 7.

To further identify the computational bottlenecks, we decompose the end-to-end latency into major pipeline components, including perception encoding, scene context decoding, diffusion generation, trajectory selection, and post-processing. The detailed breakdown is presented in Tab. 8.

The module-wise analysis shows that the primary additional latency of our model compared with the original DiffusionDrive originates from the coarse-to-fine trajectory selector. Specifically, this module introduces approximately 14.693 ms of additional latency, accounting for 33.19% of the total inference

Table 7: Overall inference latency comparison on a single NVIDIA A100 GPU under identical settings.

| **Method** | **Latency (ms)↓** | **Frequency (Hz)↑** |
|---|---|---|
| DiffusionDrive (Liao et al., 2024) | 25.600 | 39.06 |
| DiffusionDriveV2 (Zou et al., 2025) | 40.415 | 24.74 |
| **Intention Planner (Ours)** | 44.267 | 22.59 |

Table 8: Granular inference-time latency breakdown of different pipeline components.

| **Component** | **DiffusionDrive** | **DiffusionDriveV2** | **Intention Planner (Ours)** |
|---|---|---|---|
| Perception Encoding and Multimodal Fusion | 15.749 | 15.718 | 15.688 |
| Driving Scene and Context Decoding | 2.060 | 2.025 | 2.047 |
| Diffusion Decoder | 6.061 | 5.199 | 8.084 |
| Coarse-to-fine Trajectory Selector | – | 14.836 | 15.637 |
| Trajectory Post-processing | 1.730 | 2.637 | 2.811 |
| **Total Latency** | **25.600** | **40.415** | **44.267** |

time. Importantly, this selector is not introduced specifically for hierarchical intention representation learning, as a comparable trajectory selection module is also adopted in DiffusionDriveV2.

When excluding the trajectory selector, the remaining pipeline of our model achieves an inference latency of approximately 29.574 ms (33.81 Hz), introducing only 3.974 ms additional overhead compared with DiffusionDrive. Furthermore, all intention-related modules together introduce only 2.023 ms marginal latency, accounting for merely 4.57% of the total inference time. These results demonstrate that the proposed hierarchical intention representation framework introduces limited computational overhead and is not the primary bottleneck of the overall inference pipeline.