# Future-Aware End-to-End Driving: Bidirectional Modeling of Trajectory Planning and Scene Evolution

**Bozhou Zhang**[1,2], **Nan Song**[1,2], **Jingyu Li**[1,2], **Xiatian Zhu**[3]*, **Jiankang Deng**[4], **Li Zhang**[1,2]*

[1]School of Data Science, Fudan University [2]Shanghai Innovation Institute

[3]University of Surrey [4]Imperial College London

https://github.com/LogosRoboticsGroup/SeerDrive

## Abstract

End-to-end autonomous driving methods aim to directly map raw sensor inputs to future driving actions such as planned trajectories, bypassing traditional modular pipelines. While these approaches have shown promise, they often operate under a one-shot paradigm that relies heavily on the current scene context, potentially underestimating the importance of scene dynamics and their temporal evolution. This limitation restricts the model's ability to make informed and adaptive decisions in complex driving scenarios. We propose a new perspective: the future trajectory of an autonomous vehicle is closely intertwined with the evolving dynamics of its environment, and conversely, the vehicle's own future states can influence how the surrounding scene unfolds. Motivated by this bidirectional relationship, we introduce **SeerDrive**, a novel end-to-end framework that jointly models future scene evolution and trajectory planning in a closed-loop manner. Our method first predicts future bird's-eye view (BEV) representations to anticipate the dynamics of the surrounding scene, then leverages this foresight to generate future-context-aware trajectories. Two key components enable this: (1) future-aware planning, which injects predicted BEV features into the trajectory planner, and (2) iterative scene modeling and vehicle planning, which refines both future scene prediction and trajectory generation through collaborative optimization. Extensive experiments on the NAVSIM and nuScenes benchmarks show that SeerDrive significantly outperforms existing state-of-the-art methods.

## 1 Introduction

End-to-end autonomous driving [1] has emerged as a promising paradigm by jointly learning perception [2, 3, 4], prediction [5, 6, 7, 8], and planning [9, 10, 11] in a unified framework. Compared to traditional modular pipelines, this approach simplifies system design and enables planning-oriented optimization through holistic training. Recent advances [12, 13, 14, 15, 16] have demonstrated that directly generating future trajectories from raw sensor inputs can achieve strong performance, highlighting the potential of end-to-end methods for building scalable and efficient driving systems. These methods have been widely evaluated in both open-loop [17] and closed-loop [18] settings, across real-world datasets [19] and simulation platforms [20].

Despite these successes, most existing methods adopt a one-shot paradigm, in which sensor observations, typically from the current time step, are used to directly predict a trajectory several seconds into the future. In this setup, the model must rely heavily on the scene's current situation to infer the ego

*Li Zhang (lizhangfd@fudan.edu.cn) and Xiatian Zhu (xiatian.zhu@surrey.ac.uk) are the corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

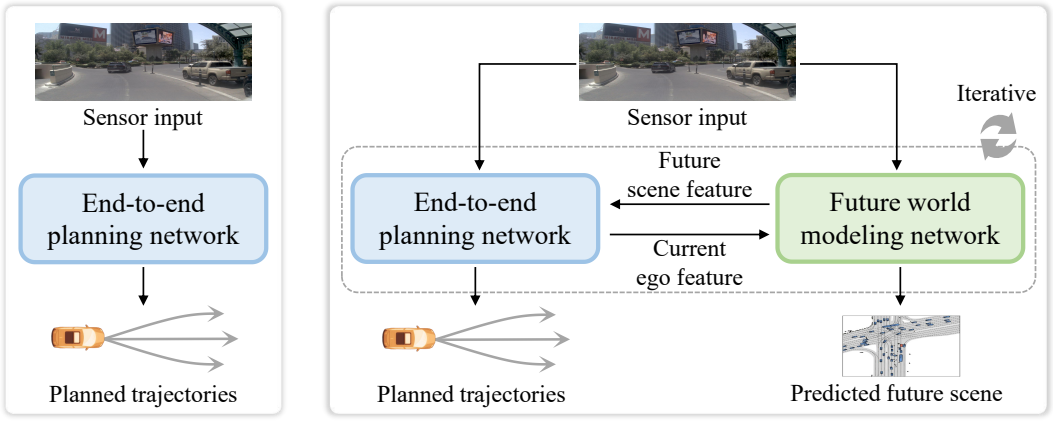



Figure 1: **Paradigm comparison.** (a) Prior end-to-end methods follow a *one-shot* paradigm, directly mapping sensor inputs to planned trajectories based only on the current scene. (b) In contrast, SeerDrive predicts scene evolution with a world model and plans trajectories with a planning model, enabling iterative interaction between the two in a closed-loop process.

vehicle's future motion (Figure 1(a)). While effective in structured environments, this approach tends to *underestimate the importance of how the scene may evolve over time*, a crucial factor in dynamic and interactive driving contexts. Furthermore, *the ego vehicle's own future actions can significantly influence how the surrounding scene unfolds*. These two aspects, the future scene and the agent's future behavior, are inherently coupled and should be modeled together. However, this bidirectional dependency remains underexplored in current end-to-end systems.

To address this gap, we draw inspiration from the emerging concept of world models, which offer the ability to learn environment dynamics and simulate future observations. We propose to leverage this capability not only to foresee how the driving scene will change, but also to coordinate it with the ego vehicle's planned actions through mutual interaction.

In this paper, we present **SeerDrive**, a novel end-to-end driving framework that introduces a paradigm shift by explicitly modeling the bidirectional relationship between scene evolution and trajectory planning (Figure 1(b)). To implement this paradigm, we design two key components. *Future-Aware Planning* injects predicted future bird's-eye view (BEV) features into the planner, enabling trajectory generation informed by both current perception and anticipated dynamics. *Iterative Scene Modeling and Vehicle Planning* refines both the predicted scene and the planned trajectory through mutual feedback, supporting adaptive and temporally consistent decision-making. Together, these components enable context-aware planning in complex, dynamic environments.

Our **contributions** are threefold. (I) We propose a new paradigm for end-to-end driving that explicitly captures the bidirectional interaction between scene dynamics and the ego vehicle's future actions, challenging the conventional one-shot planning approach. (II) We instantiate this paradigm through a unified framework, SeerDrive, which jointly models future BEV scene representations and vehicle trajectories through future-aware and iterative interaction mechanisms. (III) We conduct extensive experiments on both the NAVSIM and nuScenes benchmarks, demonstrating that SeerDrive achieves state-of-the-art performance and validates the effectiveness of our proposed design.

## 2 Related work

**End-to-end autonomous driving.** End-to-end autonomous driving [12, 13, 14, 15] has attracted increasing attention, as it contrasts with traditional modular methods by integrating perception [2, 4], prediction [5, 7], and planning [9, 11] into a unified, differentiable framework that enables end-to-end optimization. Early methods [21, 22, 23, 24] often bypass intermediate tasks and directly infer planning trajectories or actions from sensor data, both in open-loop [17] and closed-loop [20] settings. UniAD [12] unifies perception, prediction, and planning into a differentiable framework and leverages a transformer architecture to optimize the entire pipeline in a planning-oriented manner, achieving

strong performance across all tasks. VAD [13] adopts a vectorized representation to improve the efficiency of the end-to-end pipeline. VADv2 [25] introduces a probabilistic planning paradigm with a large vocabulary and demonstrates impressive performance in closed-loop settings. SparseDrive [14] leverages a sparse scene representation to make better use of temporal information. Several other studies [26, 27] explore self-supervised learning to simplify the complex end-to-end pipeline.

Recent efforts have focused increasingly on more challenging end-to-end planning benchmarks [18, 19]. DiffusionDrive [15] proposes a truncated diffusion policy to achieve more accurate and diverse planning. GoalFlow [28] incorporates flow matching and goal-point guidance into end-to-end autonomous driving. DriveTransformer [16] investigates the scaling law within a unified transformer-based architecture. Different from previous methods that focus solely on improving the planning process itself, our approach aims to jointly optimize future world modeling and planning in an adaptive manner to achieve better planning performance.

**World model in autonomous driving.** World models aim to predict future scene dynamics conditioned on the current environment and ego state. Some studies [29, 30, 31] address world modeling by training models to generate realistic videos that are consistent with physical principles. DriveDreamer [32] employs a latent diffusion model guided by 3D boxes, HD maps, and ego states, and introduces an additional decoder for future action prediction. Drive-WM [33] generalizes this setup to multi-camera inputs and explores its application in the end-to-end planning task. Some other methods explore improving the generalization of dynamic world modeling by scaling up both data and model architecture. GAIA-1 [34] models token sequences using an autoregressive transformer conditioned on past states, followed by a diffusion-based decoder for realistic video synthesis. Vista [35] is trained on diverse web-collected driving videos and utilizes latent diffusion to produce high-resolution, long-horizon video outputs.

In contrast to methods that focus on generating complex and realistic images, some approaches generate driving scenes in the bird's-eye view. SLEDGE [36] introduces the first generative simulator designed for agent motion planning. GUMP [37] builds upon generative modeling to capture dynamic traffic interactions, enabling diverse and realistic future scenario simulations. UMGen [38] generates ego-centric, multimodal driving scenes in an autoregressive, frame-by-frame manner. Scenario Dreamer [39] employs a vectorized latent diffusion model that directly operates on structured scene elements, coupled with an autoregressive Transformer for simulating agent behaviors in a data-driven way. Following the simplicity and structured nature of BEV, we likewise adopt it for modeling future scene evolution.

**Joint world modeling and planning.** Several works [27, 40, 41, 42, 43] explore joint modeling and collaborative optimization of world models and planning models, employing various strategies. In the field of autonomous driving, some works extend the world model by introducing an action token. OccWorld [44] employs an occupancy-based representation and jointly generates future occupancy and ego actions in an autoregressive manner. Drive-OccWorld [45] further extends this pipeline to operate directly from images, enabling 4D occupancy forecasting and planning. In contrast, some works start from the end-to-end autonomous driving pipeline and aim to enhance planning through world modeling. Existing methods either leverage world modeling as an additional supervision signal during training [26, 27], or use it to assist in selecting the best trajectory among multiple candidates [43]. Our approach conducts an in-depth exploration of world modeling and end-to-end planning design, enabling deep interaction between the two processes and significantly improving planning performance.

## 3 Methodology

### 3.1 Preliminary

**Task formulation.** End-to-end autonomous driving maps sensor inputs (e.g., camera and LiDAR) to future ego trajectories, often using multi-modal outputs to capture diverse possible futures. Auxiliary tasks like detection, map segmentation, and agent motion prediction are commonly integrated to enhance scene understanding and support safer planning. World models in autonomous driving aim to predict future scene evolution based on current observations, enabling agents to simulate and evaluate future outcomes. They offer structured representations of the environment, which help improve planning accuracy and decision reliability.

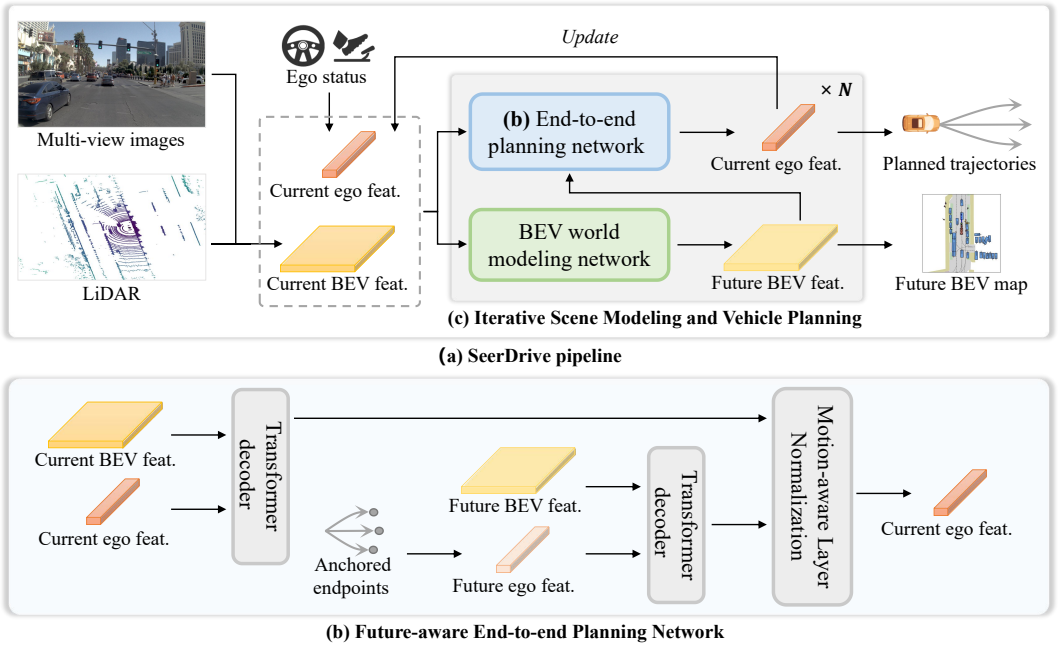



Figure 2: Overview of **SeerDrive**. (a) Multi-view images and LiDAR inputs are encoded to obtain the current BEV feature, while ego status is encoded to produce the current ego feature. These are then used to predict the future BEV feature. All three types of features are subsequently used for trajectory planning. (b) The end-to-end planning network generates future trajectories based on the current ego feature, current BEV feature, and future BEV feature. (c) The BEV world modeling network and the end-to-end planning network operate iteratively, updating the current ego feature to progressively improve planning performance.

**Framework overview.** As illustrated in Figure 2, the SeerDrive framework consists of two key components. (a) shows the overall pipeline, where multi-view images and LiDAR inputs are fused to obtain the current BEV feature, and the ego status is encoded as the current ego feature. These features are fed into the BEV world modeling and end-to-end planning networks, which operate iteratively to predict the future BEV map and generate planned trajectories. (b) details the end-to-end planning network, which incorporates the future BEV feature—produced by the world modeling network—to enhance trajectory planning. (c) depicts the iterative interaction between world modeling and planning, enabling gradual refinement and improved planning performance.

### 3.2 Feature encoding

As illustrated in Figure 2 (a), multiple types of features are encoded. Given multi-view images $\mathcal{I}$, and LiDAR features $\mathcal{P}$, the encoder transforms these multi-modal sensor inputs into a current BEV feature map $F_{\text{bev}}^{\text{curr}} \in \mathbb{R}^{H \times W \times C}$, where $H$ and $W$ are the spatial dimensions of the BEV feature, and $C$ is the number of feature channels. Following prior works [15, 43, 46], we adopt TransFuser [21] to obtain a unified BEV representation. Subsequently, a lightweight BEV decoder is employed to generate the current BEV semantic map $\mathcal{B}_{\text{curr}}$ for supervision. Following prior works [14, 15, 43], the anchored multi-modal trajectories $\mathcal{T}$ and ego status $\mathcal{E}$ are encoded with a simple multi-layer perceptron (MLP) encoder to produce the multi-modal ego feature $F_{\text{ego}}^{\text{curr}} \in \mathbb{R}^{M \times C}$, where $M$ denotes the number of trajectory modes. The process is shown below:

$$\begin{aligned} F_{\text{bev}}^{\text{curr}} &= \text{TransFuser}(\mathcal{I}, \mathcal{P}), \\ F_{\text{ego}}^{\text{curr}} &= \text{EgoEncoder}(\mathcal{T}, \mathcal{E}), \\ \mathcal{B}_{\text{curr}} &= \text{BEVDecoder}(F_{\text{bev}}^{\text{curr}}). \end{aligned} \tag{1}$$

### 3.3 Future BEV world modeling

Given the current BEV feature and ego feature obtained above, a BEV world model is employed to predict the future BEV representation. Instead of modeling future images, which is both challenging and computationally intensive, we follow recent works [27, 36, 43] that model structured and simplified BEV representations as a more efficient alternative. The current BEV feature $F_{\text{bev}}^{\text{curr}}$ is first flattened along the spatial dimensions and then repeated across the modality dimension, resulting in an updated $F_{\text{bev}}^{\text{curr}} \in \mathbb{R}^{M \times HW \times C}$. It is then concatenated with the current ego feature $F_{\text{ego}}^{\text{curr}}$ to form the scene feature $F_{\text{scene}}^{\text{curr}} \in \mathbb{R}^{M \times (HW+1) \times C}$. The BEV world model, implemented as a Transformer encoder, produces the future scene feature $F_{\text{scene}}^{\text{fut}} \in \mathbb{R}^{M \times (HW+1) \times C}$. From this, the future BEV feature $F_{\text{bev}}^{\text{fut}}$ is extracted. A lightweight BEV decoder is then applied to generate the future BEV semantic map $\mathcal{B}_{\text{fut}}$ for supervision. The overall process is illustrated below:

$$\begin{aligned} F_{\text{scene}}^{\text{fut}} &= \text{BEVWorldModel}(F_{\text{scene}}^{\text{curr}}), \\ \mathcal{B}_{\text{fut}} &= \text{BEVDecoder}(F_{\text{bev}}^{\text{fut}}). \end{aligned} \tag{2}$$

### 3.4 Future-aware end-to-end planning

After obtaining the current BEV feature, current ego feature, and future BEV feature, the end-to-end planning network jointly reasons over the present scene and its future evolution to generate the planned trajectories. However, enabling the planning network to simultaneously consider both the current and future BEV features is non-trivial. Directly interacting the ego feature with both can lead to entangled representations, causing confusion in ego planning and degrading performance. To address this, we adopt a decoupled strategy, where the ego feature interacts with the current and future BEV features independently, and the results are subsequently fused for final trajectory planning.

As illustrated in Figure 2 (b), the current ego feature $F_{\text{ego}}^{\text{curr}}$ interacts with the current BEV feature $F_{\text{bev}}^{\text{curr}}$ through a Transformer decoder. The updated ego feature is then passed through an MLP decoder to generate the planned trajectories $\mathcal{T}_{\text{a}}$ for supervision. Since the future BEV feature represents the future scene and corresponds to the future ego state, we initialize the future ego feature $F_{\text{ego}}^{\text{fut}}$ using the endpoints of the anchored trajectories. It then interacts with the future BEV feature $F_{\text{bev}}^{\text{fut}}$ through a Transformer decoder. Similarly, an MLP decoder is applied to produce the planned trajectories $\mathcal{T}_{\text{b}}$ for supervision. The process is shown below:

$$\begin{aligned} F_{\text{ego}}^{\text{curr}} &= \text{TransformerDecoder}(F_{\text{ego}}^{\text{curr}}, F_{\text{bev}}^{\text{curr}}), \\ F_{\text{ego}}^{\text{fut}} &= \text{TransformerDecoder}(F_{\text{ego}}^{\text{fut}}, F_{\text{bev}}^{\text{fut}}), \\ \mathcal{T}_{\text{a}} &= \text{EgoDecoder}(F_{\text{ego}}^{\text{curr}}), \\ \mathcal{T}_{\text{b}} &= \text{EgoDecoder}(F_{\text{ego}}^{\text{fut}}). \end{aligned} \tag{3}$$

To incorporate the future ego feature into the current ego feature, we adopt motion-aware layer normalization (MLN) [4] to obtain a future-aware ego representation. This representation is then passed through an MLP decoder to generate the final planned trajectories $\mathcal{T}_{\text{final}}$:

$$\begin{aligned} F_{\text{ego}}^{\text{curr}} &= \text{MLN}(F_{\text{ego}}^{\text{curr}}, F_{\text{ego}}^{\text{fut}}), \\ \mathcal{T}_{\text{final}} &= \text{EgoDecoder}(F_{\text{ego}}^{\text{curr}}). \end{aligned} \tag{4}$$

### 3.5 Iterative scene modeling and vehicle planning

As illustrated in Figure 2 (c), the BEV world modeling network and the end-to-end planning network operate in an iterative manner to progressively improve planning performance. This is motivated by the mutual dependency between future scene evolution and ego trajectories—future traffic dynamics influence the ego's motion plans, while the ego's planned actions, in turn, shape the future scene. In the end-to-end planning network, the future BEV feature $F_{\text{bev}}^{\text{fut}}$ serves as a reference for generating the planned trajectories. Meanwhile, the refined ego feature $F_{\text{ego}}^{\text{curr}}$ is fed back into the BEV world modeling network to produce an updated future BEV feature. This iterative process is repeated $N$ times, yielding $N$ pairs of predicted future semantic maps and ego trajectories, denoted as $(\mathcal{B}_{\text{fut}}^{(1)}, \mathcal{T}_{\text{a}}^{(1)}, \mathcal{T}_{\text{b}}^{(1)}, \mathcal{T}_{\text{final}}^{(1)}), \ldots, (\mathcal{B}_{\text{fut}}^{(N)}, \mathcal{T}_{\text{a}}^{(N)}, \mathcal{T}_{\text{b}}^{(N)}, \mathcal{T}_{\text{final}}^{(N)})$, which are all supervised during training.

### 3.6 End-to-end learning

The model is trained in an end-to-end manner with a loss function comprising two components: the BEV semantic map loss $\mathcal{L}_{\text{map}}$ and the planned trajectory loss $\mathcal{L}_{\text{traj}}$. The semantic map loss includes the current BEV map loss $\mathcal{L}_{\text{map}}^{\text{curr}}$ and the future BEV map losses from $N$ iterations, denoted as $\mathcal{L}_{\text{map}}^{\text{fut (1)}}, \ldots, \mathcal{L}_{\text{map}}^{\text{fut (N)}}$. The trajectory loss consists of $N$ sets of trajectory planning losses, each containing three terms as described in the end-to-end planning network: $(\mathcal{L}_{\text{traj}}^{\text{a (1)}}, \mathcal{L}_{\text{traj}}^{\text{b (1)}}, \mathcal{L}_{\text{traj}}^{\text{final (1)}}), \ldots, (\mathcal{L}_{\text{traj}}^{\text{a (N)}}, \mathcal{L}_{\text{traj}}^{\text{b (N)}}, \mathcal{L}_{\text{traj}}^{\text{final (N)}})$. The overall loss function for end-to-end training is as follows:

$$
\begin{aligned}
\mathcal{L}_{\text{map}} &= \lambda_1 \mathcal{L}_{\text{map}}^{\text{curr}} + \lambda_2 (\mathcal{L}_{\text{map}}^{\text{fut (1)}} + \cdots + \mathcal{L}_{\text{map}}^{\text{fut (N)}}), \\
\mathcal{L}_{\text{traj}} &= \lambda_3 ((\mathcal{L}_{\text{traj}}^{\text{a (1)}} + \mathcal{L}_{\text{traj}}^{\text{b (1)}} + \mathcal{L}_{\text{traj}}^{\text{final (1)}}) + \cdots + (\mathcal{L}_{\text{traj}}^{\text{a (N)}} + \mathcal{L}_{\text{traj}}^{\text{b (N)}} + \mathcal{L}_{\text{traj}}^{\text{final (N)}})), \\
\mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{map}} + \mathcal{L}_{\text{traj}}.
\end{aligned}
\tag{5}
$$

where $\lambda_1$, $\lambda_2$, and $\lambda_3$ are the balancing factors.

## 4 Experiments

### 4.1 Datasets and metrics

**Datasets.** We conduct experiments on two large-scale real-world autonomous driving datasets: NAVSIM [19] and nuScenes [17]. NAVSIM, built upon nuPlan [47], is designed for non-reactive simulation in complex scenarios with dynamic intention changes. It contains 1,192 training/validation (navtrain) and 136 testing (navtest) scenarios, with 8-camera and LiDAR data at 2 Hz. nuScenes includes 1,000 scenes with 6-camera and LiDAR data at 2 Hz. We follow the standard 700/150 train/validation split and evaluate planning in an open-loop setting.

**Evaluation metrics.** For the NAVSIM dataset, we use the PDM Score (PDMS), which comprises multiple sub-metrics: No At-Fault Collisions (NC), Drivable Area Compliance (DAC), Time-to-Collision (TTC), Comfort (Comf.), and Ego Progress (EP). For the nuScenes dataset, we follow the VAD [13] setting and report the L2 Displacement Error and Collision Rate.

### 4.2 Implementation details

**Model settings.** The model for the NAVSIM dataset uses both images and LiDAR as input, whereas the nuScenes model relies solely on images. For backbone networks, ResNet34 is employed for NAVSIM and ResNet50 for nuScenes. Regarding the number of camera views, 3 views are used in NAVSIM and 6 in nuScenes. The input image resolution is 1024×256 following TransFuser for NAVSIM, and 640×360 for nuScenes. The number of trajectory modes is set to 256 for NAVSIM and 6 for nuScenes. For NAVSIM, the planning horizon is 4 seconds with 8 future steps, and for nuScenes, it is 3 seconds with 6 steps. In both settings, we predict the future BEV semantic map corresponding to the final planning step.

**Training settings.** The model is trained on 8 NVIDIA GeForce RTX 3090 GPUs. For NAVSIM, the batch size is 16 per GPU, with 30 training epochs and a total training time of around 5 hours. For nuScenes, the batch size is 1 per GPU, with 12 epochs and a training time of about 12 hours. The learning rate is set to $2 \times 10^{-4}$ for NAVSIM and $1 \times 10^{-4}$ for nuScenes, both optimized using AdamW [48]. The loss balancing factors are set to $\lambda_1 = 10$, $\lambda_2 = 0.1$, and $\lambda_3 = 1$ for NAVSIM, and all set to 1 for nuScenes.

### 4.3 Comparison with state of the art

As shown in Table 1, we compare SeerDrive with several state-of-the-art methods on the NAVSIM dataset. Our approach achieves the highest PDM Score of 88.9. Under the same ResNet34 [49] backbone, SeerDrive outperforms recent methods, including the trajectory refinement model Hydra-NeXt [50], the online trajectory evaluation method WoTE [43], and the truncated diffusion policy method DiffusionDrive [15], demonstrating the effectiveness of our iterative world modeling and

Table 1: Performance comparison of planning on the NAVSIM *navtest* split with closed-loop metrics. "C & L" represents the use of both camera and LiDAR as sensor inputs. ResNet34 [49] is employed as the backbone for BEV feature extraction, following TransFuser [21]. The best and second best results are highlighted in **bold** and underline, respectively. † denotes the use of V2-99 [51] as the image backbone.

| Method | Input | NC ↑ | DAC ↑ | TTC ↑ | Comf. ↑ | EP ↑ | PDMS ↑ |
|---|---|---|---|---|---|---|---|
| VADv2-$\mathcal{V}_{8192}$ [25] | C & L | 97.2 | 89.1 | 91.6 | **100** | 76.0 | 80.9 |
| Hydra-MDP-$\mathcal{V}_{8192}$ [46] | C & L | 97.9 | 91.7 | 92.9 | **100** | 77.6 | 83.0 |
| UniAD [12] | Camera | 97.8 | 91.9 | 92.9 | **100** | 78.8 | 83.4 |
| LTF [21] | Camera | 97.4 | 92.8 | 92.4 | **100** | 79.0 | 83.8 |
| PARA-Drive [55] | Camera | 97.9 | 92.4 | 93.0 | 99.8 | 79.3 | 84.0 |
| TransFuser [21] | C & L | 97.7 | 92.8 | 92.8 | **100** | 79.2 | 84.0 |
| LAW [26] | Camera | 96.4 | 95.4 | 88.7 | 99.9 | 81.7 | 84.6 |
| DRAMA [56] | C & L | 98.0 | 93.1 | 94.8 | **100** | 80.1 | 85.5 |
| Hydra-MDP++ [57] | Camera | 97.6 | 96.0 | 93.1 | **100** | 80.4 | 86.6 |
| DiffusionDrive [15] | C & L | 98.2 | 96.2 | 94.7 | **100** | 82.2 | 88.1 |
| WoTE [43] | C & L | **98.5** | 96.8 | **94.9** | 99.9 | 81.9 | 88.3 |
| Hydra-NeXt [50] | C & L | 98.1 | **97.7** | 94.6 | **100** | 81.8 | 88.6 |
| SeerDrive | C & L | 98.4 | 97.0 | **94.9** | 99.9 | **83.2** | **88.9** |
| GoalFlow† [28] | C & L | 98.4 | 98.3 | 94.6 | 100 | 85.0 | 90.3 |
| SeerDrive† | C & L | 98.8 | 98.6 | 95.8 | 100 | 84.2 | 90.7 |

Table 2: Performance comparison of planning on the nuScenes *validation* split. ResNet50 [49] is used as the backbone for all methods, except for UniAD [12], which adopts ResNet101. "w/o bev" indicates without future BEV injection, and "w/o iter" indicates without iterative world modeling and planning.

| Method | L2 ($m$) ↓ | | | | Col. Rate (%) ↓ | | | |
|---|---|---|---|---|---|---|---|---|
| | 1$s$ | 2$s$ | 3$s$ | Avg. | 1$s$ | 2$s$ | 3$s$ | Avg. |
| ST-P3 [22] | 1.33 | 2.11 | 2.90 | 2.11 | 0.23 | 0.62 | 1.27 | 0.71 |
| BEV-Planner [54] | 0.28 | 0.42 | **0.68** | 0.46 | 0.04 | 0.37 | 1.07 | 0.49 |
| VAD-Tiny [13] | 0.46 | 0.76 | 1.12 | 0.78 | 0.21 | 0.35 | 0.58 | 0.38 |
| LAW [26] | 0.26 | 0.57 | 1.01 | 0.61 | 0.14 | 0.21 | 0.54 | 0.30 |
| PARA-Drive [55] | 0.25 | 0.46 | 0.74 | 0.48 | 0.14 | 0.23 | 0.39 | 0.25 |
| VAD-Base [13] | 0.41 | 0.70 | 1.05 | 0.72 | 0.07 | 0.17 | 0.41 | 0.22 |
| GenAD [58] | 0.28 | 0.49 | 0.78 | 0.52 | 0.08 | 0.14 | 0.34 | 0.19 |
| UniAD [12] | 0.44 | 0.67 | 0.96 | 0.69 | 0.04 | 0.08 | 0.23 | 0.12 |
| BridgeAD [53] | 0.29 | 0.57 | 0.92 | 0.59 | 0.01 | **0.05** | 0.22 | 0.09 |
| MomAD [52] | 0.31 | 0.57 | 0.91 | 0.60 | 0.01 | **0.05** | 0.22 | 0.09 |
| SparseDrive [14] | 0.29 | 0.58 | 0.96 | 0.61 | 0.01 | **0.05** | 0.18 | 0.08 |
| SeerDrive $_{\text{w/o bev}}$ | 0.22 | 0.49 | 0.73 | 0.48 | 0.02 | **0.05** | 0.16 | 0.08 |
| SeerDrive $_{\text{w/o iter}}$ | 0.28 | 0.47 | 0.81 | 0.52 | 0.02 | 0.07 | 0.20 | 0.10 |
| SeerDrive | **0.20** | **0.39** | 0.69 | **0.43** | **0.00** | **0.05** | **0.14** | **0.06** |

planning strategy. Furthermore, when replacing the ResNet-34 backbone with V2-99 [51], as done in GoalFlow [28], our method achieves better performance with significantly lower computational cost while supporting end-to-end training.

As shown in Table 2, we further evaluate our method on the nuScenes dataset. It achieves notable performance gains compared with recent state-of-the-art methods, including SparseDrive [14], MomAD [52], and BridgeAD [53]. Although the nuScenes dataset contains relatively simple scenarios and imperfect evaluation metrics [54], our method still achieves clear improvements, demonstrating the effectiveness of the two main designs.

### 4.4 Ablation study

**Effects of components.** As shown in Table 3, we conduct an ablation study on the key components of our method, including Future-Aware Planning and Iterative Scene Modeling and Vehicle Planning. In the first row, both modules are removed, so the planned trajectories rely only on the current BEV feature and no iterative process is applied. This leads to a drop in PDMS from 88.9 to 87.1. In the second row, we remove the future BEV injection for planning, which also results in decreased performance, showing the importance of future BEV features for planning. In the third row, we remove the iterative process, and performance drops as well, indicating its role in refining the results. The last row presents the full SeerDrive model with both modules included, achieving the highest performance.

Table 3: Ablation study on the key components of our model. "Iter. S&V" refers to Iterative Scene Modeling and Vehicle Planning.

| Future-aware plan | Iter. S&V | NC ↑ | DAC ↑ | TTC ↑ | Comf. ↑ | EP ↑ | PDMS ↑ |
|---|---|---|---|---|---|---|---|
| | | 98.3 | 95.6 | 94.5 | **100** | 81.1 | 87.1 |
| | ✓ | 98.2 | 96.6 | 94.3 | **100** | 82.0 | 87.9 |
| ✓ | | 98.2 | 96.5 | 94.4 | **100** | 82.5 | 88.1 |
| ✓ | ✓ | **98.4** | **97.0** | **94.9** | 99.9 | **83.2** | **88.9** |

**Effects of Future-Aware Planning.** As shown in Table 4, we conduct an ablation study on the design of the future-aware end-to-end planning. The first two rows analyze different strategies for incorporating the future BEV feature. In the first row, future BEV injection is removed entirely. In the second row, the decoupled strategy is omitted, and the network learns jointly from current and future BEV features. Both settings lead to notable performance degradation, highlighting the effectiveness of our proposed design. In the third and fourth rows, we analyze how the current and future ego features are combined. In the third row, the Motion-aware Layer Normalization (MLN) is removed, and the two features are concatenated and passed through an MLP for dimension alignment. In the fourth row, MLN is also removed, and the features are directly added. Both variations lead to a notable drop in performance, demonstrating the effectiveness of MLN in producing a future-aware ego feature.

Table 4: Ablation study on the design of future-aware end-to-end planning.

| Type | NC ↑ | DAC ↑ | TTC ↑ | Comf. ↑ | EP ↑ | PDMS ↑ |
|---|---|---|---|---|---|---|
| w/o Future BEV | 98.2 | 96.6 | 94.3 | **100** | 82.0 | 87.9 |
| w/o Decouple | 98.0 | 96.0 | 94.0 | 99.5 | 81.6 | 87.3 |
| MLN2Cat | 98.3 | 96.6 | 94.7 | **100** | 82.4 | 88.3 |
| MLN2Add | 98.2 | **97.0** | 94.1 | **100** | 82.9 | 88.5 |
| SeerDrive | **98.4** | **97.0** | **94.9** | 99.9 | **83.2** | **88.9** |

**Effects of the number of iterations.** As shown in Table 5, a moderate number of iterations for scene modeling and vehicle planning achieves a good balance between efficiency and performance. We observe that using two iterations yields the best trade-off.

Table 5: Ablation study on the number of iterations for scene modeling and vehicle planning.

| Number | NC ↑ | DAC ↑ | TTC ↑ | Comf. ↑ | EP ↑ | PDMS ↑ |
|---|---|---|---|---|---|---|
| 1 | **98.4** | 96.6 | 94.5 | **100** | 82.2 | 88.1 |
| 2 | **98.4** | 97.0 | **94.9** | 99.9 | **83.2** | **88.9** |
| 3 | **98.4** | **97.2** | **94.9** | **100** | 82.5 | 88.7 |

**Prediction steps of future BEV.** We predict the future BEV only at the final planning step, as the last frame provides the most informative reference for determining the trajectory direction. This design yields an effective and efficient planning reference. As shown in Table 6, we further analyze this choice by predicting a sequence of intermediate BEV representations and concatenating them as planner inputs. However, incorporating intermediate predictions brings no significant performance gains while increasing model complexity. Therefore, predicting only the final future BEV at the last planning step proves to be the more effective and efficient strategy.

Table 6: Ablation study on the number of prediction steps for future BEV.

| Prediction steps | NC ↑ | DAC ↑ | TTC ↑ | Comf. ↑ | EP ↑ | PDMS ↑ |
|---|---|---|---|---|---|---|
| 1s-2s-3s-4s | 98.3 | 97.0 | 94.6 | **100** | 83.0 | 88.8 |
| 2s-4s | **98.4** | **97.2** | 94.8 | **100** | 83.1 | **88.9** |
| 4s (Origin) | **98.4** | 97.0 | **94.9** | 99.9 | **83.2** | **88.9** |

**The performance of $\mathcal{T}_{\text{a}}$, $\mathcal{T}_{\text{b}}$, and $\mathcal{T}_{\text{final}}$.** As shown in Table 7, we conduct an ablation study to separately evaluate the performance of $\mathcal{T}_{\text{a}}$, $\mathcal{T}_{\text{b}}$, and $\mathcal{T}_{\text{final}}$. The results show that $\mathcal{T}_{\text{final}}$ significantly outperforms both $\mathcal{T}_{\text{a}}$ and $\mathcal{T}_{\text{b}}$, indicating that each contributes meaningfully to the superior performance of the final trajectory.

Table 7: Ablation study on the performance of $\mathcal{T}_{\text{a}}$, $\mathcal{T}_{\text{b}}$, and $\mathcal{T}_{\text{final}}$.

| | NC ↑ | DAC ↑ | TTC ↑ | Comf. ↑ | EP ↑ | PDMS ↑ |
|---|---|---|---|---|---|---|
| $\mathcal{T}_{\text{a}}$ | 98.3 | 96.6 | 94.4 | **100** | 82.7 | 88.3 |
| $\mathcal{T}_{\text{b}}$ | 98.2 | 96.4 | 94.2 | 99.9 | 83.0 | 88.2 |
| $\mathcal{T}_{\text{final}}$ | **98.4** | **97.0** | **94.9** | 99.9 | **83.2** | **88.9** |

**The use of the anchored endpoints to initialize the future ego feature.** The future BEV corresponds to the final planning step. Thus, we initialize the future ego feature using the anchored endpoints, allowing it to encode prior knowledge of the ego vehicle's future state at the final planning step. For more extensive evaluation, we analyze different initialization strategies for the future ego feature. As shown in Table 8, using anchored endpoints as a prior yields the best performance.

Table 8: Ablation study on the use of the anchored endpoints to initialize the future ego feature.

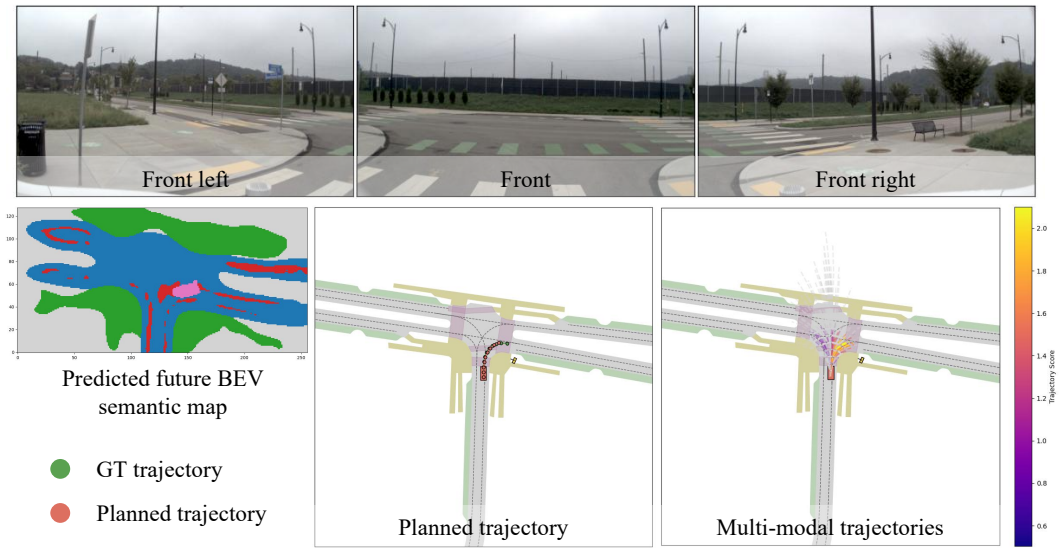
| | NC ↑ | DAC ↑ | TTC ↑ | Comf. ↑ | EP ↑ | PDMS ↑ |
|---|---|---|---|---|---|---|
| Random | 98.3 | 96.9 | 94.5 | **100** | 82.8 | 88.6 |
| Anchored trajectories | 98.3 | 96.8 | 94.0 | **100** | **83.7** | 88.7 |
| Anchored endpoints (Origin) | **98.4** | **97.0** | **94.9** | 99.9 | 83.2 | **88.9** |

### 4.5 Qualitative results

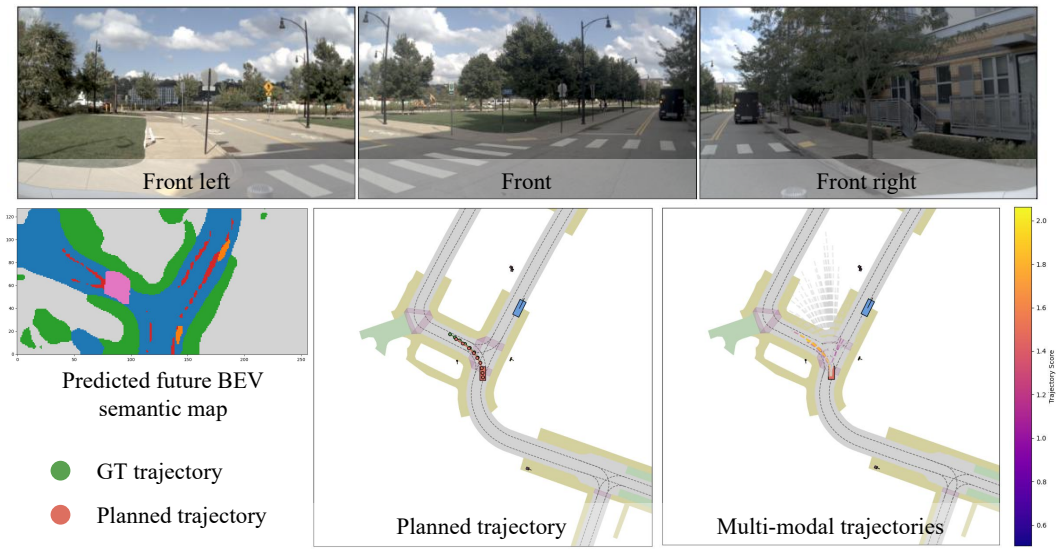
As shown in Figure 3, we visualize two cases: a right turn and a left turn. In the bottom middle figure, the final planned trajectory generated by our model closely aligns with the ground-truth trajectory. The bottom right figure presents the planned multi-modal trajectories, capturing multiple possible future motions. The bottom left figure shows the predicted future BEV semantic maps, which reflect how the scene evolves and how the ego vehicle's position changes after the turning behaviors.

## 5 Conclusion

This paper presents SeerDrive, a unified end-to-end framework that combines future scene modeling and trajectory planning. Unlike traditional one-shot approaches that rely only on the current scene, SeerDrive predicts future BEV representations to guide planning. It introduces two core components: Future-Aware Planning, which incorporates predicted future context into trajectory




**(a) Right-turn case**




**(b) Left-turn case**

Figure 3: Qualitative results on the NAVSIM dataset. The visualization includes three front-facing camera views: front left, front, and front right, as well as model outputs including the predicted future BEV semantic map, the planned trajectory, and multi-modal predicted trajectories.

generation, and Iterative Scene Modeling and Vehicle Planning, which refines both scene predictions and plans through repeated interaction. This design enables more informed and adaptive decisions. Extensive experiments on NAVSIM and nuScenes show our approach achieves state-of-the-art results.

**Limitations and future work.** The BEV world model adopts a transformer architecture specifically tailored to our framework, making it both effective and efficient for planning. However, it does not benefit from the generalization capabilities of foundation models. On the other hand, using off-the-shelf foundation models as world models often suffers from slow inference speed and challenges in joint optimization with the planner. Therefore, developing a tightly integrated paradigm of planning and world modeling represents a promising direction for future work.

## Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Grant No. 62376060).

## References

[1] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE TPAMI*, 2024. 1

[2] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 1, 2

[3] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *CVPR*, 2023. 1

[4] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *ICCV*, 2023. 1, 2, 5

[5] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *NeurIPS*, 2022. 1, 2

[6] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *IEEE TPAMI*, 2024. 1

[7] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *CVPR*, 2023. 1, 2

[8] Bozhou Zhang, Nan Song, and Li Zhang. Decoupling motion forecasting into directional intentions and dynamic states. In *NeurIPS*, 2024. 1

[9] Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking imitation-based planners for autonomous driving. In *ICRA*, 2024. 1, 2

[10] Jie Cheng, Yingbing Chen, and Qifeng Chen. Pluto: Pushing the limit of imitation learning-based planning for autonomous driving. *arXiv preprint*, 2024. 1

[11] Yinan Zheng, Ruiming Liang, Kexin ZHENG, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-based planning for autonomous driving with flexible guidance. In *ICLR*, 2025. 1, 2

[12] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023. 1, 2, 7, 23

[13] 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 *ICCV*, 2023. 1, 2, 3, 6, 7, 23

[14] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *ICRA*, 2025. 1, 2, 3, 4, 7, 21

[15] 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 *CVPR*, 2025. 1, 2, 3, 4, 6, 7, 21

[16] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *ICLR*, 2025. 1, 3

[17] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1, 2, 6, 21

[18] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *NeurIPS*, 2024. 1, 3, 22, 23

[19] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *NeurIPS*, 2024. 1, 3, 6, 21, 22, 23

[20] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, 2017. 1, 2

[21] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021. 2, 4, 7

[22] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022. 2, 7

[23] 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 *ICCV*, 2023. 2

[24] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *CVPR*, 2023. 2

[25] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint*, 2024. 3, 7

[26] 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 *ICLR*, 2025. 3, 7, 22

[27] Peidong Li and Dixiao Cui. Navigation-guided sparse scene representation for end-to-end autonomous driving. In *ICLR*, 2025. 3, 5, 22

[28] 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 *CVPR*, 2025. 3, 7

[29] Ruiyuan Gao, Kai Chen, Enze Xie, HONG Lanqing, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *ICLR*, 2024. 3

[30] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *CVPR*, 2024. 3

[31] Jiachen Lu, Ze Huang, Zeyu Yang, Jiahui Zhang, and Li Zhang. Wovogen: World volume-aware diffusion for controllable multi-camera driving scene generation. In *ECCV*, 2024. 3

[32] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *ECCV*, 2024. 3

[33] 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 *CVPR*, 2024. 3

[34] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint*, 2023. 3

[35] 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 *NeurIPS*, 2024. 3

[36] Kashyap Chitta, Daniel Dauner, and Andreas Geiger. Sledge: Synthesizing driving environments with generative models and rule-based traffic. In *ECCV*, 2024. 3, 5

[37] Yihan Hu, Siqi Chai, Zhening Yang, Jingyu Qian, Kun Li, Wenxin Shao, Haichao Zhang, Wei Xu, and Qiang Liu. Solving motion planning tasks with a scalable generative model. In *ECCV*, 2024. 3

[38] Yanhao Wu, Haoyang Zhang, Tianwei Lin, Lichao Huang, Shujie Luo, Rui Wu, Congpei Qiu, Wei Ke, and Tong Zhang. Generating multimodal driving scenes via next-scene prediction. In *CVPR*, 2025. 3

[39] Luke Rowe, Roger Girgis, Anthony Gosselin, Liam Paull, Christopher Pal, and Felix Heide. Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In *CVPR*, 2025. 3

[40] Jingyu Li, Bozhou Zhang, Xin Jin, Jiankang Deng, Xiatian Zhu, and Li Zhang. Imagidrive: A unified imagination-and-planning framework for autonomous driving. *arXiv preprint*, 2025. 3

[41] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint*, 2025. 3

[42] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. In *NeurIPS*, 2025. 3

[43] 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*, 2025. 3, 4, 5, 6, 7, 21, 22

[44] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *ECCV*, 2024. 3

[45] Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *AAAI*, 2025. 3

[46] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint*, 2024. 4, 7

[47] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint*, 2021. 6

[48] I Loshchilov. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7

[50] Zhenxin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Zuxuan Wu, and Jose M Alvarez. Hydra-next: Robust closed-loop driving with open-loop training. *arXiv preprint*, 2025. 6, 7

[51] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, 2020. 7

[52] Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *CVPR*, 2025. 7, 23

[53] Bozhou Zhang, Nan Song, Xin Jin, and Li Zhang. Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In *CVPR*, 2025. 7, 23

[54] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *CVPR*, 2024. 7

[55] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*, 2024. 7

[56] Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint*, 2024. 7

[57] Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint*, 2025. 7

[58] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *ECCV*, 2024. 7

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See the experiment section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The details of the model and experiments are provided in the main paper and the appendix.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our main idea is specifically designed for applications in real-world autonomous driving.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See the experiment section.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

# Appendix

## A Notations

As shown in Table 9, we provide a lookup table for notations used in the paper.

Table 9: Notations used in the paper.

| Notation | Description |
|---|---|
| $\mathcal{I}$ | multi-view images |
| $\mathcal{P}$ | LiDAR |
| $\mathcal{T}$ | anchored multi-modal trajectories |
| $\mathcal{E}$ | ego status |
| $H\&W$ | spatial dimensions of the BEV feature |
| $M$ | the number of trajectory modes |
| $C$ | the number of feature channels |
| $N$ | the number of iterations |
| $F_{\text{ego}}^{\text{curr}}$ | current ego feature |
| $F_{\text{bev}}^{\text{curr}}$ | current BEV feature map |
| $F_{\text{scene}}^{\text{curr}}$ | current scene feature |
| $F_{\text{ego}}^{\text{fut}}$ | future ego feature |
| $F_{\text{bev}}^{\text{fut}}$ | future BEV feature map |
| $F_{\text{scene}}^{\text{fut}}$ | future scene feature |
| $\mathcal{B}_{\text{curr}}$ | predicted current BEV semantic map |
| $\mathcal{B}_{\text{fut}}$ | predicted future BEV semantic map |
| $\mathcal{T}_{\text{a}}$ | planned multi-modal trajectories by current BEV |
| $\mathcal{T}_{\text{b}}$ | planned multi-modal trajectories by future BEV |
| $\mathcal{T}_{\text{final}}$ | final planned multi-modal trajectories |

## B Implementation details

In addition to the implementation details provided in the main paper, we offer further clarifications here. The feature channels $C$ are set to 256 for both the NAVSIM [19] and nuScenes [17] datasets. Regarding the spatial dimensions $H \times W$ of the BEV features, they are $8 \times 8$ for NAVSIM and $100 \times 100$ for nuScenes. However, for future BEV map prediction, the features are uniformly downsampled to $8 \times 8$ in nuScenes. For the anchored multi-modal trajectories, we follow previous works [14, 15, 43] and use the K-Means algorithm to cluster them from the ground-truth trajectories.

## C Evaluation metrics

In the NAVSIM dataset, trajectories spanning 4 seconds at 2Hz are first produced and then upsampled to 10Hz using an LQR controller. These refined trajectories are evaluated using a set of closed-loop metrics, which include No At-Fault Collisions ($S_{\text{NC}}$), Drivable Area Compliance ($S_{\text{DAC}}$), Time to Collision with bounds ($S_{\text{TTC}}$), Ego Progress ($S_{\text{EP}}$), Comfort ($S_{\text{CF}}$), and Driving Direction Compliance ($S_{\text{DDC}}$). The overall performance score is computed by aggregating these individual metrics. However, due to implementation limitations, $S_{\text{DDC}}$ is excluded from the final evaluation[1].

$$S_{\text{PDM}} = S_{\text{NC}} \times S_{\text{DAC}} \times s_{\text{TTC}} \times \left( \frac{5 \times S_{\text{EP}} + 5 \times S_{\text{CF}} + 2 \times S_{\text{DDC}}}{12} \right). \tag{6}$$

[1] https://github.com/autonomousvision/navsim/issues/14

## D Qualitative results

In addition to the qualitative results, we provide additional visualizations in Figure 4 and Figure 5 to further demonstrate the effectiveness of our model.

## E Failure cases

Although our SeerDrive demonstrates strong performance, it still encounters some failure cases.

As illustrated in Figure 6 (a), the model fails to choose the correct lane after a right turn. In the right subfigure of (a), which shows the multi-modal planning result, one of the predicted trajectories closely aligns with the ground truth. However, the classification score fails to select it, suggesting that the multi-modal trajectory selection process requires further improvement.

As illustrated in Figure 6 (b), the model fails to infer the correct driving intention for a right turn. Most of the planned trajectories instead tend to change lanes to the left and proceed straight. This suggests that incorporating high-level driving intentions, such as explicit driving commands, is necessary for achieving more accurate planning outcomes.

## F Discussions

**Comparison with existing methods.** Table 10 summarizes the differences between our work and prior approaches in terms of how predicted future scenes from world models are utilized.

Table 10: Comparison between SeerDrive and existing methods.

| Method | Usage of predicted future scene from world model |
|---|---|
| LAW [26] | Used as an auxiliary supervision signal during training |
| SSR [27] | Used as an auxiliary supervision signal during training |
| WoTE [43] | Used to assist in selecting the best trajectory among multiple candidates |
| SeerDrive (Ours) | Used as feature-level reference for planning, enabling iterative interaction with planner |

**Experiment on complex scenarios.** Following the reviewer's valuable suggestions, we evaluate our model on the test split of NAVSIM [19] under scenarios with more than 10 agents. As shown in Table 11, our model still demonstrates strong performance.

Table 11: Experiment on complex scenarios.

| | NC ↑ | DAC ↑ | TTC ↑ | Comf. ↑ | EP ↑ | PDMS ↑ |
|---|---|---|---|---|---|---|
| Complex scenarios | 98.2 | 96.8 | 94.3 | 99.9 | 83.1 | 88.5 |

**Experiment beyond NAVSIM and nuScenes.** Following the reviewer's valuable suggestions, we have further evaluated the more complex Bench2Drive [18] dataset with many rare and high-entropy scenarios such as emergency braking, merging, and overtaking, along with a longer planning horizon and closed-loop testing. As shown in Table 12, our model consistently achieves the best performance.

**Uncertainty in future predictions and planning.** To tackle the uncertainty caused by the multi-modal characteristics of future trajectories, our framework performs prediction and planning while explicitly considering the multi-modal nature of the future. The predicted future BEV feature is multi-modal and shares the same modality as the ego feature used for planning. Accordingly, we also predict a multi-modal future BEV representation. For supervision, we adopt a winner-takes-all strategy: the best trajectory and the corresponding future BEV are selected using the same probability generated from the current ego feature, and the loss is computed based on the selected mode.

**Parameter size and inference cost.** We evaluate our model on the NAVSIM [19] dataset. The model has 66 M parameters, and the average inference time is 24 ms under the same configuration as in the main paper.

Table 12: Experiment on the Bench2Drive [18] dataset.The first metric corresponds to open-loop testing, while the latter two are used for closed-loop evaluation.

| Method | Avg. L2 Error ↓ | Driving Score ↑ | Success Rate (%) ↑ |
|---|---|---|---|
| VAD [13] | 0.91 | 42.35 | 15.00 |
| UniAD-Tiny [12] | 0.80 | 40.73 | 13.18 |
| UniAD-Base [12] | 0.73 | 45.81 | 16.46 |
| MomAD [52] | 0.82 | 47.91 | 18.11 |
| BridgeAD [53] | 0.71 | 50.06 | 22.73 |
| SeerDrive (Ours) | **0.66** | **58.32** | **30.17** |

**Quantitative metric for predicted future BEV.** The mIoU of the predicted future BEV map on the NAVSIM [19] dataset is 38.87.

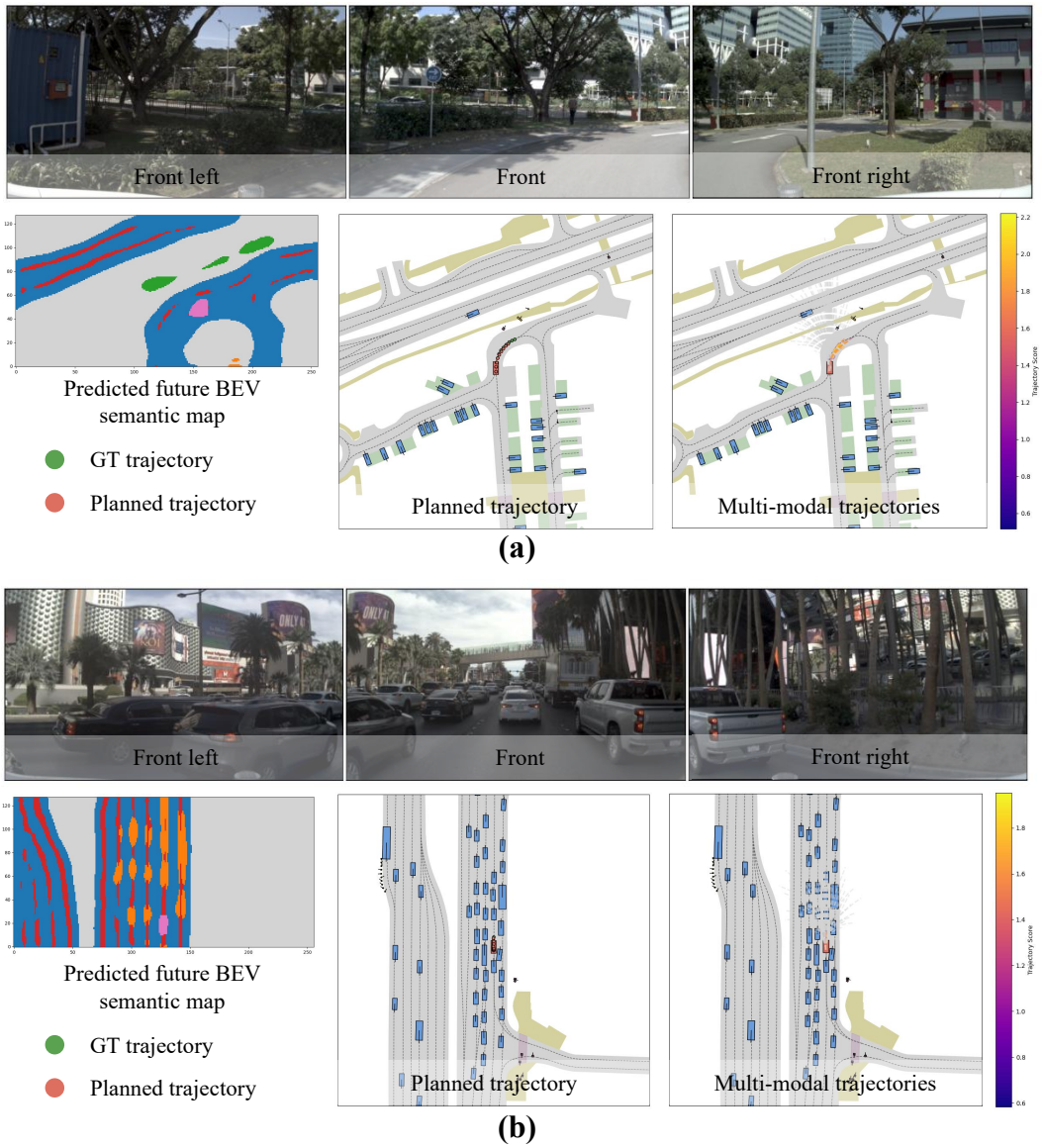
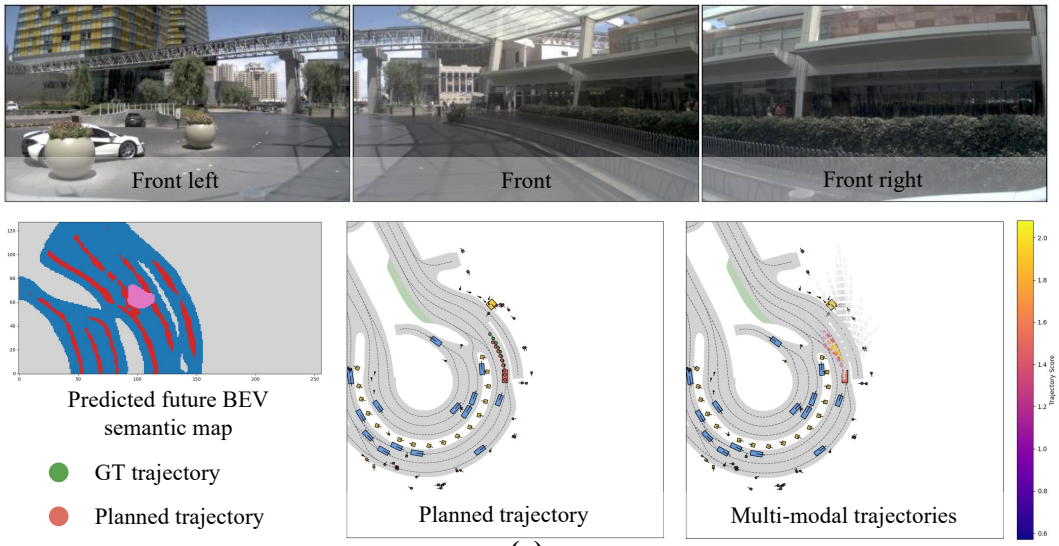
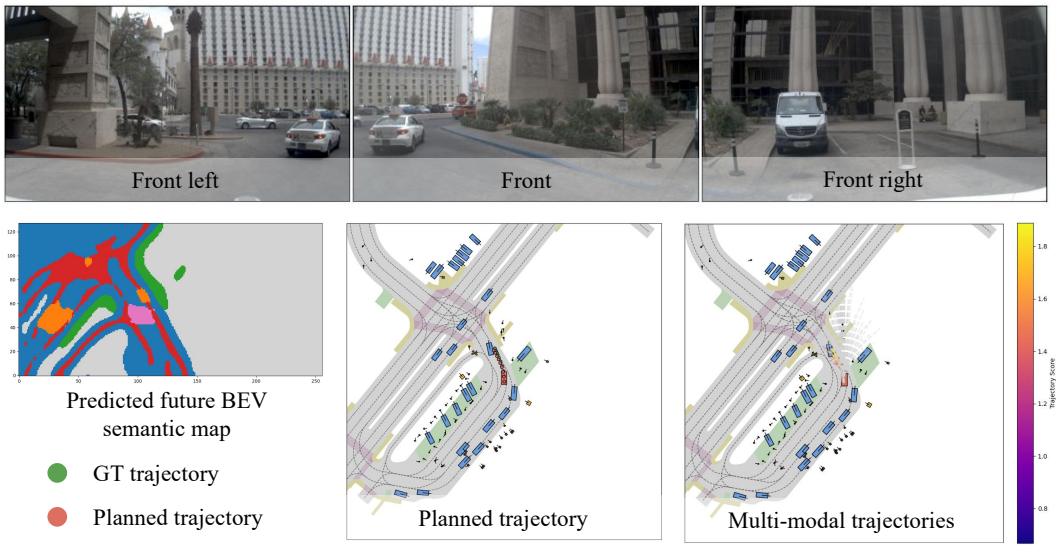



Figure 4: Qualitative results 1 on the NAVSIM dataset. The visualization includes three front-facing camera views: front left, front, and front right, as well as model outputs including the predicted future BEV semantic map, the planned trajectory, and multi-modal predicted trajectories.

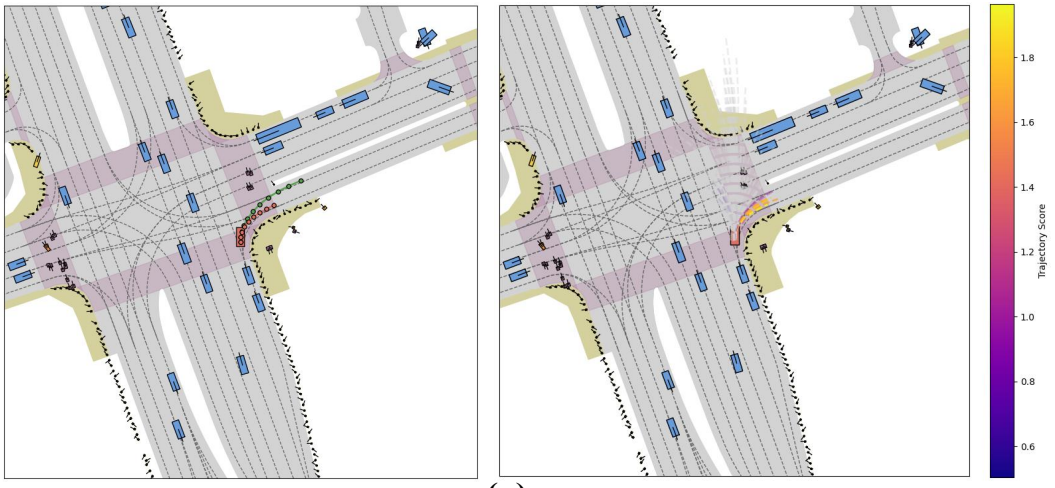



**(c)**

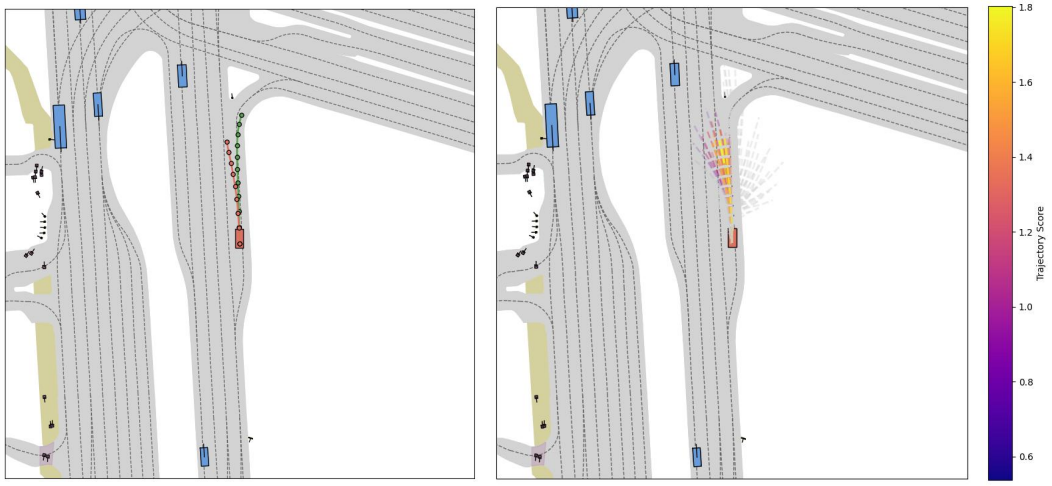



**(d)**

Figure 5: Qualitative results 2 on the NAVSIM dataset. The visualization includes three front-facing camera views: front left, front, and front right, as well as model outputs including the predicted future BEV semantic map, the planned trajectory, and multi-modal predicted trajectories.




**(a)**




**(b)**

Figure 6: Failure cases from the NAVSIM dataset. In the left figure, the ground-truth trajectory is shown in green, while the top-ranked planning trajectory, selected based on the classification score, is displayed in orange. The right figure illustrates the predicted multi-modal trajectories.