GenAD: Generative End-to-End Autonomous Driving

Wenzhao Zheng[1,2*], Ruiqi Song[3,5*], Xianda Guo[4,5*†], Chenming Zhang[5,6], and Long Chen[3,5,6†]

[1]Tsinghua University [2]UC Berkeley
[3]Institute of Automation, Chinese Academy of Sciences
[4]School of Computer Science, Wuhan University [5]Waytous
[6]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
{wenzhao.zheng, chmzhang}@outlook.com; xianda_guo@163.com;
{ruiqi.song, long.chen}@ia.ac.cn

Abstract. Directly producing planning results from raw sensors has been a long-desired solution for autonomous driving and has attracted increasing attention recently. Most existing end-to-end autonomous driving methods factorize this problem into perception, motion prediction, and planning. However, we argue that the conventional progressive pipeline still cannot comprehensively model the entire traffic evolution process, e.g., the future interaction between the ego car and other traffic participants and the structural trajectory prior. In this paper, we explore a new paradigm for end-to-end autonomous driving, where the key is to predict how the ego car and the surroundings evolve given past scenes. We propose GenAD, a generative framework that casts autonomous driving into a generative modeling problem. We propose an instance-centric scene tokenizer that first transforms the surrounding scenes into map-aware instance tokens. We then employ a variational autoencoder to learn the future trajectory distribution in a structural latent space for trajectory prior modeling. We further adopt a temporal model to capture the agent and ego movements in the latent space to generate more effective future trajectories. GenAD finally simultaneously performs motion prediction and planning by sampling distributions in the learned structural latent space conditioned on the instance tokens and using the learned temporal model to generate futures. Extensive experiments on the widely used nuScenes benchmark show that the proposed GenAD achieves state-of-the-art performance on vision-centric end-to-end autonomous driving with high efficiency. Code: https://github.com/wzzheng/GenAD.

1 Introduction

Vision-centric autonomous driving has been extensively explored in recent years due to its economic convenience [9,13,20,30,31,45,48]. While researchers have advanced the limit of vision-centric autonomous driving in various tasks including

* Equal contributions.
† Corresponding authors.

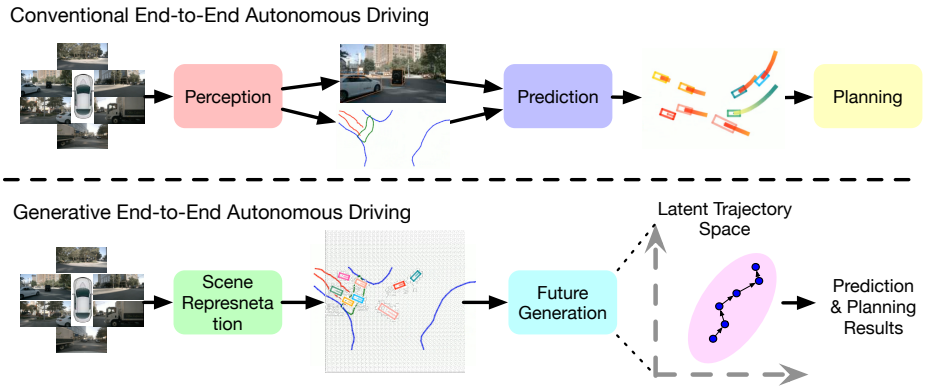

Fig. 1: Comparisons of our generative end-to-end autonomous driving framework with the conventional pipeline. Most existing methods follow a serial design of perception, prediction, and planning. They usually ignore the high-level interactions between the ego car and other agents and the structural prior of realistic trajectories. We model autonomous driving as a future generation problem and conduct motion prediction and ego planning simultaneously in a structural latent space.

3D object detection [20,30,31], map segmentation [29,34,36,59], and 3D semantic occupancy prediction [22, 50, 51, 54, 55, 65], recent advances in vision-centric end-to-end autonomous driving [10, 18, 19, 25, 62] have shed light on a potential and elegant path to directly produce planning results from raw sensors.

Most existing end-to-end models contain several modules and follow a pipeline of perception, motion prediction, and planning [18,19,25]. For example, UniAD [19] further progressively performs map perception, detection, tracking, motion prediction, occupancy prediction, and planning modules to improve the robustness of the system. It is also observed that using a planning objective improves the performance of intermediate tasks [19]. However, the serial design of prediction and planning of existing pipelines ignores the possible future interactions between the ego car and the other traffic participants. We argue that this type of interaction is important for accurate planning. For example, the lane shift of the ego car would affect the action of the rear cars, and further affects the planning of the ego car. This high-order interaction cannot be effectively modeled by the current design of motion prediction before planning. Also, future trajectories are highly structured and share a common prior (e.g., most trajectories are continuous and straight lines). Still, most existing methods fail to consider this structural prior, leading to inaccurate predictions and planning.

In this paper, we propose a **Gen**erative End-to-End **A**utonomous **D**riving (GenAD) framework (shown in Fig. 1), which models autonomous driving as a trajectory generation problem to unleash the full potential of end-to-end methods. We propose a scene tokenizer to obtain instance-centric scene representations, which focus on instances but also integrate map information. To achieve this, we use a backbone network to extract image features for each sur-

rounding camera and then transform them into the 3D bird's eye view (BEV) space [20, 30, 31]. We further use cross-attention to refine high-level map and agent tokens from BEV features. We then add an ego token and use ego-agent self-attention to capture their high-order interactions. We further inject map information with cross-attention to obtain map-aware instance tokens. To model the structural prior of future trajectories, we learn a variational autoencoder to map ground-truth trajectories to Gaussian distributions considering the uncertain nature of motion prediction and driving planning. We then use a simple yet effective gated recurrent unit (GRU) [6] to perform auto-regressing to model instance movement in the latent structural space. During inference, we sample from the learned distributions conditioned on the instance-centric scene representation and can thus predict different possible futures. Our GenAD can simultaneously perform motion prediction and planning using the unified future trajectory generation model. We conduct extensive experiments on the widely used nuScenes benchmark to evaluate the performance of the proposed GenAD framework. Based on generative modeling, our GenAD achieves state-of-the-art vision-based planning performance with high-efficiency.

2 Related Work

Perception. Perception is the basic step in autonomous driving, which aims to extract meaningful information from raw sensor inputs. Despite the strong performance of LiDAR-based methods [4, 11, 42, 63], vision-centric methods [22, 30, 31, 55, 60] have emerged as a competitive alternative due to the low costs of RGB cameras. Equipped with large 2D image backbones, vision-centric methods demonstrated great potential in the main perception tasks including 3D object detection [20, 30, 31, 45, 48, 57, 60], HD map reconstruction [29, 34, 36, 59], and 3D semantic occupancy prediction [21, 23, 50, 51, 55, 61, 65]. To accurately complete these 3D tasks, the key procedure is to transform image features to the 3D space. One line of works predicts explicit depths for image features and then projects them into the 3D space using camera parameters [20, 30, 33, 39, 45, 48, 60]. Other methods initialize queries in the 3D space and exploit deformable cross-attention to adaptively aggregate information from 2D images [22, 26, 31]. Some works further design better positional embedding strategies [38], 3D representations [22], or task heads [60] to further improve the perception performance or efficiency. In this paper, we adopt conventional simple designs for 3D perception and focus on motion prediction and planning.

Prediction. Accurate motion prediction for traffic participants is the key to the following motion planning of the ego vehicle. Conventional methods utilized ground-truth agent history and HD map information as inputs and focused on the prediction of future agent trajectories [3, 44, 53]. One straightforward way is to draw agent paths and HD maps on a BEV image and use convolutional neural networks to process them and output motion prediction results [3, 44]. Further methods employed vectors or tokens to represent separate agents or map elements [32, 37, 43, 53]. They then leverage the reasoning ability of graph neural

networks [32] and transformers to infer future motions considering the interaction between agents and map elements. The increase of hardware capacity promotes the emergence of end-to-end motion prediction methods [12,16,24,56,59], which jointly perform perception and prediction to get free of offline HD maps. Despite the challenges, recent end-to-end methods have demonstrated promising performance in this more practical setting [12, 16, 24, 59]. They usually use attention to incorporate agent and map information and leverage temporal networks (e.g., gated recurrent units [16]) to predict futures. However, most existing methods directly decode trajectories from features and ignores the structural nature of realistic trajectories (e.g., most of them are straight lines). Differently, we learn a variational autoencoder from ground-truth trajectories to model the prior in a latent structural space and sample instances in this space for inference.

Planning. Planning is the ultimate goal for the first stage of autonomous driving. Despite the mature development of rule-based planners [1,7,52], learning-based planners [5,46,49] receive increasing attention due to their potential to benefit from large-scale driving data and compatibility with end-to-end autonomous driving methods. Most existing end-to-end planning methods follow a pipeline of perception, prediction, and planning [18, 19, 25]. For example, ST-P3 [18] progressively employs a map perception, a BEV occupancy prediction, and a trajectory planning module to obtain future ego movements from surrounding cameras. UniAD [19] further extends ST-P3 with additional detection, tracking, and motion prediction modules to improve the robustness of the system. VAD [25] simplifies UniAD with vectorized scene representation and only map, motion, and planning modules for end-to-end driving, which achieves state-of-the-art planning performance with better efficiency. However, the serial design of prediction and planning ignores the effect of future ego movements on the agent motion prediction. It also lacks modeling of the uncertain nature of motion prediction and planning. To address this, GenAD models autonomous driving in a generative framework and simultaneously generates the future trajectories for the ego vehicle and other agents in a learned probabilistic latent space.

3 Proposed Approach

This section presents our generative framework of vision-based end-to-end autonomous driving, as shown in Fig. 2. We first introduce an instance-centric scene representation which incorporates high-order map-ego-agent interactions to enable comprehensive yet compact scene descriptions (Sec. 3.1). We then elaborate on the learning of a latent embedding space to model realistic trajectories as prior (Sec. 3.2) and the generation of future motion in this learned latent space (Sec. 3.3). At last, we detail the training and inference of the **Gen**erative end-to-end **A**utonomous **D**riving (GenAD) framework(Sec. 3.4).

3.1 Instance-Centric Scene Representation

The goal of end-to-end autonomous driving can be formulated as obtaining a planned f-frame future trajectory $\mathbf{T}(T, f) = \{\mathbf{w}^{T+1}, \mathbf{w}^{T+2}, \cdots, \mathbf{w}^{T+f}\}$ for the

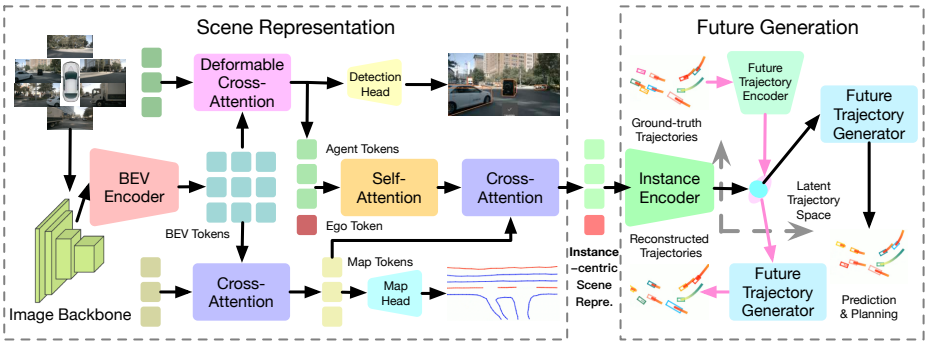

Fig. 2: Framework of our generative end-to-end autonomous driving. Given surrounding images as inputs, we employ an image backbone to extract multi-scale features and then use a BEV encoder to obtain BEV tokens. We then use cross-attention and deformable cross-attention to transform BEV tokens into map and agent tokens, respectively. With an additional ego token, we use self-attention to enable ego-agent interactions and cross-attention to further incorporate map information to obtain the instance-centric scene representation. We map this representation to a structural latent trajectory space which is jointly learned using ground-truth future trajectories. Finally, we employ a future trajectory generator to produce future trajectories to simultaneously complete motion prediction and planning.

ego vehicle given the current and past p-frame sensor inputs $\mathbf{S} = \{\mathbf{s}^T, \mathbf{s}^{T-1}, \cdots, \mathbf{s}^{T-p}\}$ and trajectory $\mathbf{T}(T-p, p+1) = \{\mathbf{w}^T, \mathbf{w}^{T-1}, \cdots, \mathbf{w}^{T-p}\}$.

$$\mathbf{T}(T-p, p+1), \mathbf{S} \rightarrow \mathbf{T}(T, f), \tag{1}$$

where $\mathbf{T}(T, f)$ denotes a f-frame trajectory starting from the T-th frame, $\mathbf{w}^t$ denotes the waypoint at the t-th frame, and $\mathbf{s}^t$ denotes the input at t-th frame.

The first step of end-to-end autonomous driving is to perceive the sensor inputs to obtain high-level descriptions of the surrounding scene. These descriptions usually include semantic map [34, 36] and instance bounding box [30, 31]. To achieve this, we follow a conventional vision-centric perception pipeline to extract bird's eye view (BEV) $\mathbf{B} \in \mathbb{R}^{H \times W \times C}$ features first and then build on them to refine map and bounding box features.

Image to BEV. We follow BEVFormer [31] to obtain the BEV features. Specifically, we use a convolutional neural network [14] and feature pyramid network [35] to obtain multi-scale image features $\mathbf{F}$ from camera inputs $\mathbf{s}$. We then initialize $H \times W$ BEV tokens $\mathbf{B}_0$ as queries and use deformable cross-attention [64] to transfer information from the multi-scale image feature $\mathbf{F}$:

$$\mathbf{B} = \mathrm{DA}(\mathbf{B}_0, \mathbf{F}, \mathbf{F}), \tag{2}$$

where $\mathbf{DA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ consists of interleaved self-attention and deformable cross-attention layers using $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ as queries, keys, and values, respectively. We then align the BEV features from the p past frames into the current coordinate system and concatenate them as the final BEV features $\mathbf{B}$.

BEV to map. As semantic map elements are usually sparse in the BEV space, we follow a similar concept [24, 25] and use map tokens $\mathbf{M} \in \mathbb{R}^{N_m * C}$ to represent semantic maps. Each map token $\mathbf{m} \in \mathbf{M}$ can be decoded to a set of points in the BEV space by a map decoder d_m representing a category of map elements and their corresponding positions. Following VAD [25], we consider three categories of map elements (i.e., lane divider, road boundary, and pedestrian crossing). We use the global cross-attention mechanism to update learnable initialized queries $\mathbf{M}_0$ from BEV tokens $\mathbf{B}$:

$$\mathbf{M} = \mathrm{CA}(\mathbf{M}_0, \mathbf{B}, \mathbf{B}), \tag{3}$$

where $\mathbf{CA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ is composed of interleaved self-attention and cross-attention layers using $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ as queries, keys, and values, respectively.

BEV to agent. Similar to the representation of semantic maps, we adopt a set of agent tokens $\mathbf{A}$ to represent the 3D position of each instance in the surroundings. We use deformable cross-attention to obtain the updated agent tokens $\mathbf{A}$ from the BEV tokens $\mathbf{B}$:

$$\mathbf{A} = \mathrm{DA}(\mathbf{A}_0, \mathbf{B}, \mathbf{B}), \tag{4}$$

where $\mathbf{A}_0$ are learnable tokens as initialization. Having obtained the agent tokens $\mathbf{A}$, we employ a 3D object detection head d_a to decode the position, orientation, and category information from each agent token $\mathbf{a}$.

Instance-centric scene representation. As prediction and planning mainly focus on the instances of agents and ego vehicles, respectively, we propose an instance-centric scene representation to comprehensively and efficiently represent the autonomous driving scenario. We first add an ego token $\mathbf{e}$ to the learned agent tokens $\mathbf{A}$ to construct a set of instance tokens $\mathbf{I} = \mathrm{concat}(\mathbf{e}, \mathbf{A})$.

Existing methods [18, 19, 25] usually perform motion prediction and planning in a serial manner, which ignores the effect of future ego movements on the agents. For example, the lane shift of the ego car would possibly affect the action of rear cars, rendering the motion prediction results inaccurate. Differently, we enable high-order interactions between the ego vehicle and other agents by performing self-attention on the instance tokens:

$$\mathbf{I} \leftarrow \mathrm{SA}(\mathbf{I}, \mathbf{I}, \mathbf{I}), \tag{5}$$

where $\mathbf{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ denotes the self-attention block composed of self-attention layers using $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ as queries, keys, and values, respectively.

Furthermore, to perform accurate prediction and planning, both the agents and the ego vehicle need to be aware of the semantic map information. We thus employ cross-attention between the updated instance tokens and the learned map tokens to obtain map-aware instance-centric scene representations:

$$\mathbf{I} \leftarrow \mathrm{CA}(\mathbf{I}, \mathbf{M}, \mathbf{M}). \tag{6}$$

The learned instance tokens $\mathbf{I}$ incorporate high-order agent-ego interactions and are aware of the learned semantic maps, which are compact yet contain all the necessary information to perform motion prediction and trajectory planning.

3.2 Trajectory Prior Modeling

We find that the objectives of motion prediction of other agents and planning of the ego vehicle share the same output space and are essentially the same. They both aim to produce a high-quality realistic trajectory of the concerned instance, given semantic maps and interactions with other agents. The goal for GenAD can be then formulated as inferring the future trajectory $\mathbf{T}$ given the map-aware instance-centric scene representations $\mathbf{I}$. Different from existing methods which directly output the trajectory using a simple decoder, we model it as a trajectory generation problem $\mathbf{T} \sim p(\mathbf{T}|\mathbf{I})$ considering its uncertain nature.

The trajectories of both the ego vehicle and other agents are highly structured (e.g., continuous) and follow certain patterns. For example, most of the trajectories are straight lines as the vehicle driving at a constant speed, and some of them are curved lines with a near-constant curvature when the vehicle turning right or left. Only in very rare cases will the trajectories be zig-zagging. Considering this, we adopt a variational autoencoder (VAE) [28] architecture to learn a latent space $\mathbb{Z}$ to model this trajectory prior. Specifically, we employ a ground-truth trajectory encoder e_f to model $p(\mathbf{z}|\mathbf{T}(T, f))$, which maps the future trajectory $\mathbf{T}(T, f)$ to a diagonal Gaussian distribution on the latent space $\mathbb{Z}$. The encoder e_f outputs two vectors μ_f and σ_f representing the mean and variance of the Gaussian distribution:

$$p(\mathbf{z}|\mathbf{T}(T, f)) \sim N(\mu_f, \sigma_f), \tag{7}$$

where $N(\mu, \sigma^2)$ denotes a Gaussian distribution with a mean of μ and standard deviation of σ.

The learned distribution $p(\mathbf{z}|\mathbf{T}(T, f))$ contains the prior of the ground-truth trajectories, which can be leveraged to improve the authenticity of motion prediction and planning for traffic agents and ego vehicles.

3.3 Latent Future Trajectory Generation

Having obtained the latent distribution of the future trajectories as prior, we need to explicitly decode them from the latent trajectory space $\mathbb{Z}$.

While a straightforward way is to directly use an MLP-based decoder to output trajectory points in the BEV space to model $p(\mathbf{T}(T, f)|\mathbf{z})$, it fails to model the temporal evolution of the traffic agents and ego vehicle. To consider the temporal relations of instances at different time stamps, we factorized the joint distribution $p(\mathbf{T}(T, f)|\mathbf{z})$ as follows:

$$\begin{aligned} p(\mathbf{T}(T, f)|\mathbf{z}^T) = p(\mathbf{w}^{T+1}|\mathbf{z}^T) \cdot p(\mathbf{w}^{T+2}|\mathbf{w}^{T+1}, \mathbf{z}^T) \\ \cdots p(\mathbf{w}^{T+f}|\mathbf{w}^{T+1}, \cdots, \mathbf{w}^{T+f-1}, \mathbf{z}^T). \end{aligned} \tag{8}$$

We sample a vector from the distribution $N(\mu_f, \sigma_f)$ as the latent state at the current time stamp $\mathbf{z}^T$. Instead of decoding the whole trajectory at once, we adopt a simple MLP-based decoder d_w to decode a waypoint $\mathbf{w} = d_w(\mathbf{z})$ from the latent space $\mathbf{Z}$, i.e., we instantiate $p(\mathbf{w}^{T+1}|\mathbf{z})$ with $\mathbf{w} = d_w(\mathbf{z})$.

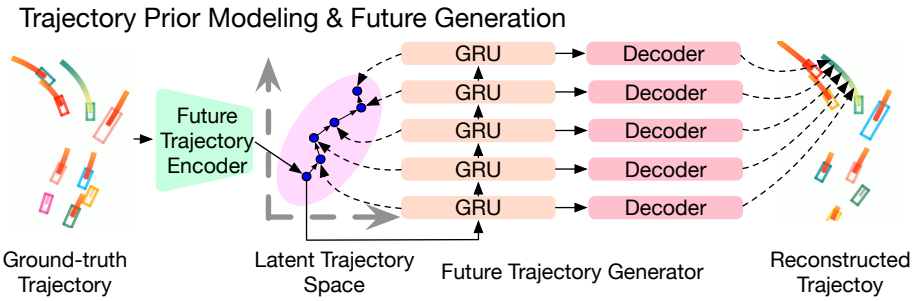

Fig. 3: Illustration of the proposed trajectory prior modeling and future generation. We use a future trajectory encoder to map ground-truth trajectories to a latent trajectory space, where we use the Gaussian distribution to model the trajectory uncertainty. We then employ a gate recurrent unit (GRU) to progressively predict the next future in the latent space and use a decoder to obtain explicit trajectories.

We then adopt a gated recurrent unit (GRU) [6] as the future trajectory generator to model the temporal evolutions of instances. Specifically, the GRU model g takes as inputs the current latent representation $\mathbf{z}_t$ and transforms it into the next state $g(\mathbf{z}_t) = \mathbf{z}_{t+1}$. We can then decode the waypoint $\mathbf{w}^{t+1}$ at the $(t+1)$-th time stamp using the waypoint decoder $\mathbf{w}^{t+1} = d_w(\mathbf{z}_{t+1})$, i.e., we model $p(\mathbf{w}^{t+1}|\mathbf{w}^{T+1}, \cdots, \mathbf{w}^t, \mathbf{z})$ with $d_w(g(\mathbf{z}_t))$.

Compared with a single decoder that directly outputs the whole trajectory, the waypoint decoder performs a simpler task of only decoding a position in the BEV space and the GRU module models the movement of agents in the latent space $\mathbf{Z}$. The produced trajectory is thus more realistic considering the prior knowledge in this learned structured latent space. We illustrate the proposed trajectory prior modeling and latent future trajectory generation in Fig. 3.

3.4 Generative End-to-End Autonomous Driving

In this subsection, we present the overall architecture of the proposed GenAD framework for vision-centric end-to-end autonomous driving. Given surrounding camera signals $\mathbf{s}$ as inputs, we first employ an image backbone to extract multi-scale image features F and then use deformable attention to transform them into the BEV space. We align the BEV features from the past p frames to the current ego-coordinate to obtain the final BEV feature $\mathbf{B}$. We perform global cross-attention and deformable attention to refine a set of map tokens $\mathbf{M}$ and agent tokens $\mathbf{A}$, respectively. To model the high-order interactions between traffic agents and the ego vehicle, we combine agent tokens with an ego token and perform self-attention among them to construct a set of instance tokens $\mathbf{I}$. We also use cross-attention to inject semantic map information into the instance tokens $\mathbf{I}$ to facilitate further prediction and planning.

As realistic trajectories are highly structured, we learn a VAE module to model the trajectory prior and adopt a generative framework for both motion

prediction and planning. We learn an encoder e_t to map ground-truth trajectories to the structural space $\mathbf{Z}$ as Gaussian distributions. We then employ a GRU-based future trajectory generator g to model the temporal evolutions of instances in the latent space $\mathbf{Z}$ and use a simple MLP-based decoder d_w to decode waypoints from latent representations. We can finally reconstruct the trajectories $\hat{\mathbf{T}}_a$ and $\hat{\mathbf{T}}_e$ for traffic agents and the ego vehicle by integrating the decoded waypoint of each time stamp. For training, we additionally use a class decoder d_c to predict the category for each agent $\hat{\mathbf{c}}_a$. To learn the future trajectory encoder e_f, the future trajectory generator g, the waypoint decoder d_w, and the class decoder d_c, we follow VAD [25] to impose trajectory losses on the reconstructed and ground-truth trajectories for both traffic agents and the ego vehicle:

$$J_{prior} = L_{tra}(\hat{\mathbf{T}}_e, \mathbf{T}_e) + \frac{1}{N_a} L_{tra}(\hat{\mathbf{T}}_a, \mathbf{T}_a) + \lambda_c L_{focal}(\hat{\mathbf{C}}_a, \mathbf{C}_a), \tag{9}$$

where $||\cdot||_1$ denotes the L1 norm, N_a is the number of agents, and L_{focal} represents the focal loss to constrain the predicted agent class. L_{tra} denotes the trajectory losses [25] including the L1 discrepancy, ego-agent collision constraint, ego-boundary overstepping constraint, and the ego-lane directional constraint. $\hat{\mathbf{C}}_a$ and $\mathbf{C}_a$ represent the predicted and ground-truth classes for all the agents.

To infer the future trajectory for traffic agents and the ego vehicle from the instance tokens $\mathbf{I}$, we use an instance encoder e_i to map each instance token to the latent space $\mathbb{Z}$. The encoder e_i similarly outputs a mean vector μ_i and variance vector σ_i to parameterize a diagonal Gaussian distribution:

$$p(\mathbf{z}|\mathbf{I}) \sim N(\mu_i, \sigma_i). \tag{10}$$

With the learned latent trajectory space with prior, the motion prediction and planning can be unified and formulated as distribution matching between the instance distribution $p(\mathbf{z}|\mathbf{I})$ and the ground-truth distribution $p(\mathbf{z}|\mathbf{T}(T, f))$. We impose the Kullback-Leibler divergence loss to enforce distribution matching:

$$J_{plan} = D_{KL}(p(\mathbf{z}|\mathbf{I}), p(\mathbf{z}|\mathbf{T}(T, f))), \tag{11}$$

where D_{KL} denotes the Kullback-Leibler divergence.

Additionally, we use two auxiliary tasks to train the proposed GenAD model: map segmentation and 3D object detection. We use a map decoder d_m on the map tokens $\mathbf{M}$ and an object decoder d_o on the agent tokens $\mathbf{A}$ to obtain the predicted maps and 3D object detection results. We follow the task decoder design of VAD [25] and employ bipartite matching for ground truth matching. We then impose semantic map loss [60] J_{map} and 3D object detection loss [31] J_{det} on them to train the network.

The overall training objective of our GenAD can be formulated as:

$$J_{GenAD} = J_{prior} + \lambda_{plan} J_{plan} + \lambda_{map} J_{map} + \lambda_{det} J_{det}, \tag{12}$$

where λ_{plan}, λ_{map}, and λ_{det} are balance factors. Our GenAD can be trained efficiently in an end-to-end manner. For inference, we discard the future trajectory

encoder e_f and sample a latent state following the instance distribution $p(\mathbf{z}|\mathbf{I})$ as input for the trajectory generator g and waypoint decoder d_w. GenAD models end-to-end driving as a generative problem and performs future prediction and planning in a structured latent space, which considers the prior of realistic trajectories to produce high-quality trajectory prediction and planning.

4 Experiments

4.1 Datasets

We conducted extensive experiments on the widely adopted nuScenes [2] dataset for evaluation. The nuScenes dataset contains 1000 driving scenes, where each scene provides RGB and LiDAR video of 20 seconds. The ego vehicle is equipped with 6 surrounding cameras with 360° horizontal FOV and a 32-beam LiDAR sensor. nuScenes provides semantic map and 3D object detection annotations for keyframes at 2Hz. It includes 1.4M 3D bounding boxes of objects from 23 categories. We partitioned the dataset into 700, 150, and 150 scenes for training, validation, and testing, respectively, following the official instructions [2].

4.2 Evaluation Metrics

Following existing end-to-end autonomous driving methods [18, 19], we use the L2 displacement error and collision rate to measure the quality of planning results. The L2 displacement error measures the L2 distance between the planned trajectory and the ground-truth trajectory. The collision rate measures how often the self-driving vehicle collapses with other traffic participants following the planned trajectory. By default, we take as inputs 2s history (i.e., 5 frames) and evaluate the planning performance at the 1s, 2s, and 3s future.

4.3 Implementation Details

We strictly follow the settings of existing methods [25] for fair comparisons. We encode three types of direction instructions (go straight, turn left, and turn right) and incorporate them via cross-attention to the scene representation. We adopted ResNet50 [15] to process images with a resolution of 640×360 and use a 200×200 BEV representation to perceive the surrounding scene. For fair comparisons, we basically use the same hyperparameters as VAD-tiny [25]. We fixed the number of BEV tokens, map tokens, and agent tokens to 100×100, 100, and 300, respectively. Each map token contains 20 point tokens to represent a map point in the BEV space. We set the hidden dimension of each BEV, point, agent, ego, and instance token to 256. We used a latent space with a dimension of 512 to model trajectory prior and also set the hidden dimension of the GRU to 512. We employed 3 layers for each attention block.

For training, we set the loss balance factors to 1 and use the AdamW [41] optimizer with a cosine learning rate scheduler [40]. We set the initial learning rate to 2×10^{-4} and a weight decay of 0.01. By default, we trained GenAD for 60 epochs with 8 NVIDIA RTX 3090 GPUs and adopted a total batch size of 8.

Table 1: Comparisons with state-of-the-art methods in motion planning performance on the nuScenes [2] val dataset. We use bold and underlined numbers to represent the best and second-best results, respectively. † represents that the metrics are computed with an average of all the predicted frames. ‡ denotes FPS measured with the same environment on our machine with a single RTX 3090 GPU.

Method	Input	Aux. Sup.	L2 (m) ↓				Collision Rate (%) ↓				FPS
			1s	2s	3s	Avg.	1s	2s	3s	Avg.	
IL [47]	LiDAR	None	0.44	1.15	2.47	1.35	0.08	0.27	1.95	0.77	-
NMP [58]	LiDAR	B & Mo	0.53	1.25	2.67	1.48	0.04	0.12	0.87	0.34	-
FF [17]	LiDAR	Freespace	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43	-
EO [27]	LiDAR	Freespace	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33	-
ST-P3 [18]	Camera	Ma & B & D	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71	1.6
UniAD [19]	Camera	Ma & B & Mo & T & O	0.48	0.96	1.65	1.03	0.05	**0.17**	**0.71**	**0.31**	1.8
OccNet [51]	Camera	3D-Occ & Ma & Bo	1.29	2.13	2.99	2.14	0.21	0.59	1.37	0.72	2.6
OccWorld [62]	Camera	3D-Occ	0.52	1.27	2.41	1.40	0.12	0.40	2.08	0.87	2.8
VAD-Tiny [25]	Camera	Ma & B & Mo	0.60	1.23	2.06	1.30	0.31	0.53	1.33	0.72	**6.9**‡
VAD-Base [25]	Camera	Ma & B & Mo	0.54	1.15	1.98	1.22	**0.04**	0.39	1.17	0.53	3.6‡
GenAD	Camera	Ma & B & Mo	**0.36**	**0.83**	**1.55**	**0.91**	0.06	0.23	1.00	0.43	6.7‡
VAD-Tiny† [25]	Camera	Ma & B & Mo	0.46	0.76	1.12	0.78	0.21	0.35	0.58	0.38	**6.9**‡
VAD-Base† [25]	Camera	Ma & B & Mo	0.41	0.70	1.05	0.72	**0.07**	0.17	0.41	0.22	3.6‡
GenAD†	Camera	Ma & B & Mo	**0.28**	**0.49**	**0.78**	**0.52**	0.08	**0.14**	**0.34**	**0.19**	6.7

4.4 Results and Analysis

Main results. We compared GenAD with state-of-the-art end-to-end methods in Table 1. Note that VAD [25] reported results computed with an average of all the previous predicted frames instead of results at each predicted frame. For fair comparisons, we reported both metrics.

Our GenAD achieves the best L2 errors with an efficient inference speed. Though UniAD [19] outperforms our method with respect to the collision rates, it employs additional supervision signals during training such as tracking and occupancy information, which has been verified to be vital to avoid collision [19]. However, these labels in the 3D space are difficult to annotate, rendering it not trivial to use less labels to achieve competitive performance. Also, the complexity of the proposed GenAD model is comparable (an FPS of 6.7 v.s. 6.9 with total parameters of 47.8M v.s. 39.9M) with VAD-tiny but we achieve even better results than VAD-base, demonstrating a strong performance/speed trade-off.

Closed-loop Testing. We followed the suggestion to conduct additional closed-loop testing based on the CARLA Town05 Long benchmark [8], as shown in Table 2. Our model performs best on vision-based methods and even suppresses the LiDAR-based Transfuser.

Method	CILRS	LBC	Transfuser†	ST-P3	VAD-Base	**GenAD**
DS ↑	3.68	7.05	33.15	11.45	30.31	**34.53**

Table 2: Closed-loop simulation results on Carla [8]. † LiDAR-based methods.

Table 3: Comparisons of perception, prediction, and planning performance. † denotes FPS measured with the same environment with a single RTX 3090 GPU.

Method	Detection	Map Segmentation			Motion Prediction		Planning		FPS
	mAP ↑	mAP@0.5↑	mAP@1.0↑	mAP@1.5↑	EPA (car)↑	EPA (ped.)↑	Avg. L2↓	Avg. CR.↓	
VAD [25]	0.27	0.15	0.44	0.61	0.56	0.29	1.30	0.72	**6.9**†
GenAD	**0.29**	**0.24**	**0.56**	**0.71**	**0.59**	**0.34**	**0.91**	**0.43**	6.7†

Table 4: Effect of the instance-centric scene representation. E → A represents our ego-to-agent interaction to obtain the instance-centric scene representation.

Setting	L2 (m) ↓				Collision Rate (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
VAD [25]	0.60	1.23	2.06	1.30	0.31	0.53	1.33	0.72
w/ E → A	**0.39**	**0.99**	**1.91**	**1.10**	**0.14**	**0.39**	**1.35**	**0.63**
GenAD	**0.36**	**0.83**	**1.55**	**0.91**	**0.06**	**0.23**	**1.00**	**0.43**
w/o E → A	0.40	0.93	1.74	1.02	0.70	0.98	1.97	1.22

Perception and prediction performance. We further evaluated the perception and prediction performance of GenAD and compared it with VAD-tiny [25], as shown in Table 3. We use mean average precision (mAP) to measure the 3D object detection performance, and mAP@0.5, mAP@1.0, and mAP@1.5 to evaluate the quality of predicted maps. For motion prediction, we report the end-to-end prediction accuracy (EPA) for both cars and pedestrians to avoid the influence of falsely detected agents. For motion planning, we report the average L2 error and collision rate (CR) over 1s, 2s, and 3s.

We observe that GenAD outperforms VAD on all the tasks with a similar inference speed. Specifically, GenAD achieves better motion prediction performances by considering the influence of the ego vehicle on the other agents. GenAD also demonstrates superior performance in 3D detection and map segmentation, showing a better consistency between perception, prediction, and planning. Though our model does not specifically target perception, the generative modeling would affect the perception modules, and the consistent structural modeling of the planning improves the performance of previous tasks.

Effect of the instance-centric scene representation. We conducted an ablation study to analyze the effectiveness of the instance-centric scene representation, as shown in Table 4. We first added the ego-to-agent interaction with

Table 5: Effect of the generative framework for driving. TPM and LFTG denote trajectory prior modeling and latent future trajectory generation, respectively.

TPM	LFTG	L2 (m) ↓				Collision Rate (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
×	×	0.42	1.02	1.89	1.11	0.06	0.43	1.60	0.70
✓	×	0.38	0.92	1.76	1.02	0.23	0.37	1.19	0.60
×	✓	0.39	0.92	1.74	1.02	0.18	0.45	1.05	0.56
✓	✓	**0.36**	**0.83**	**1.55**	**0.91**	**0.06**	**0.23**	**1.00**	**0.43**

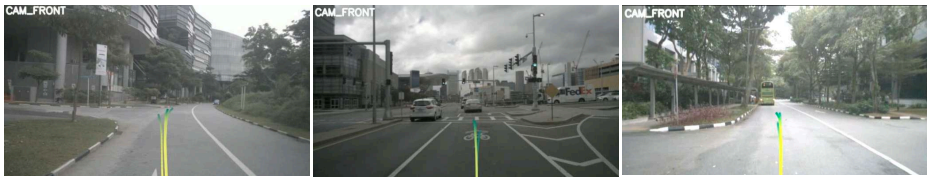

Fig. 4: Planning results with different sampled embedding.

the proposed method to VAD-tiny [25], and observe a large improvement in both the L2 error and the collision rate. We also removed the ego-to-agent interaction in our GenAD model by masking the self-attention matrix to dissect its effect. We see that the collision rate performance drops greatly. This is because without considering the high-order interactions between the ego car and the other agents, it becomes very difficult to learn the true underlying distributions of trajectories.

Effect of the generative framework of autonomous driving. We also analyzed the designs of the proposed future trajectory generation model, which is composed of two modules: trajectory prior modeling (TPM) and latent future trajectory generation (LFTG). When using TPM alone, we directly decode the entire trajectory from the latent space. With only the LFTG module, we use a gated recurrent unit to gradually produce waypoints given the instance-centric scene representation. We see that both modules are effective and improve planning performance. Combining the two modules further improves the performance by a large margin. This verifies the importance of factorizing the joint distribution as in (8) to release the full potential of latent trajectory prior modeling.

Generation Diversity. We visualized the predicted futures by sampling differently from the Gaussian distribution as shown in Figure 4. Given the direction instruction, the generated futures are diverse yet realistic.

Visualizations. We provide a visualization of our GenAD model with comparisons with VAD-tiny [25] with a similar model size, as shown in Fig. 5. We visualize the map segmentation, detection, motion prediction, and planning results on a single image and provide the surrounding camera inputs as references. We see that GenAD produces better and safer trajectories than VAD in various scenarios including going straight, overtaking, and turning. For the challenging

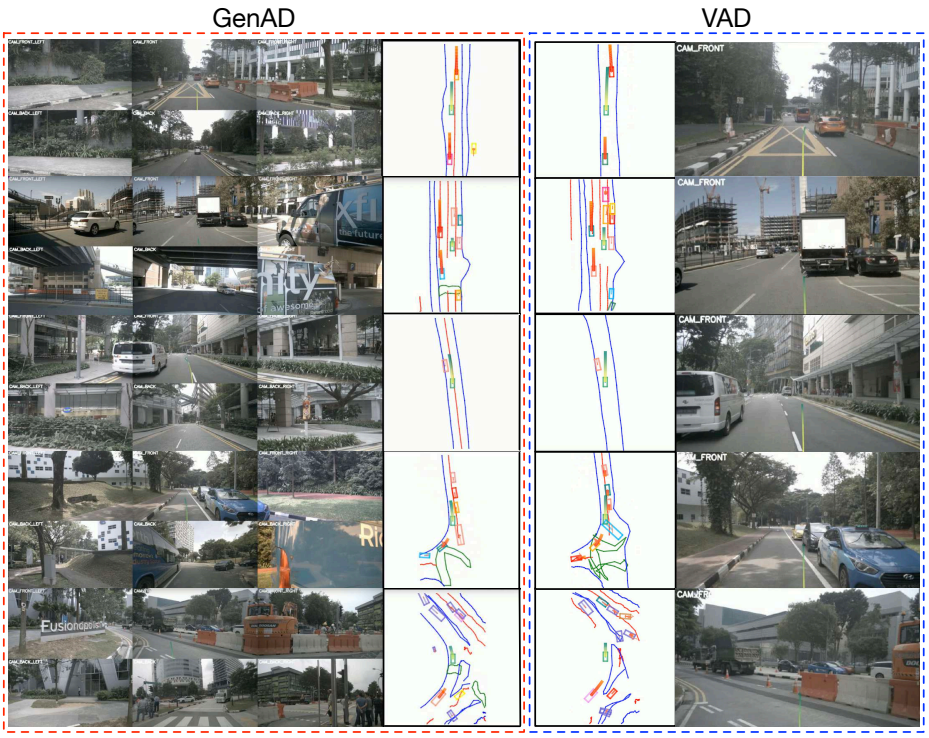

Fig. 5: Visualizations of the results of GenAD with comparisons with VAD [25]. We observe that GenAD still performs well on challenging scenarios involving multiple agents in complex traffic scenes.

scenarios when multiple agents are involved in complex traffic scenes, our GenAD still demonstrates good results while VAD cannot safely move through.

5 Conclusion and Discussions

In this paper, we have presented a generative end-to-end autonomous driving (GenAD) framework for driving from vision inputs. We have investigated the conventional serial design of perception, prediction, and planning for autonomous driving and proposed a generative framework to enable high-order ego-agent interactions and produce more accurate future trajectories with learned structural prior. We have conducted extensive experiments on the widely adopted nuScenes dataset and demonstrated the state-of-the-art planning performance of the proposed GenAD. In the future, it is interesting to explore other generative modeling methods such as diffusion models for end-to-end autonomous driving.

Limitations. One of the advantages of generative modeling is the ability to scale up. However, due to the device memory limit, we cannot conduct experiments with a larger model. The scaling ability of our GenAD is still unclear.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 62373356 and the Open Projects Program of the State Key Laboratory of Multimodal Artificial Intelligence Systems.

References

1. Bouchard, F., Sedwards, S., Czarnecki, K.: A rule-based behaviour planner for autonomous driving. In: IJCRR. pp. 263–279 (2022) 4
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020) 10, 11
3. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. arXiv preprint arXiv:1910.05449 (2019) 3
4. Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J.: Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. arXiv preprint arXiv:2303.11301 (2023) 3
5. Cheng, J., Xin, R., Wang, S., Liu, M.: Mpnp: Multi-policy neural planner for urban driving. In: IROS. pp. 10549–10554 (2022) 4
6. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: EMNLP. p. 1724 (2014) 3, 8
7. Dauner, D., Hallgarten, M., Geiger, A., Chitta, K.: Parting with misconceptions about learning-based vehicle motion planning. In: CoRL (2023) 4
8. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: CoRL (2017) 12
9. Duan, Y., Guo, X., Zhu, Z.: Diffusiondepth: Diffusion denoising approach for monocular depth estimation. arXiv preprint arXiv:2303.05021 (2023) 1
10. Duan, Y., Guo, X., Zhu, Z., Wang, Z., Wang, Y.K., Lin, C.T.: Maskfuser: Masked fusion of joint multi-modal tokenization for end-to-end autonomous driving. arXiv preprint arXiv:2405.07573 (2024) 2
11. Graham, B., Engelcke, M., Van Der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: CVPR. pp. 9224–9232 (2018) 3
12. Gu, J., Hu, C., Zhang, T., Chen, X., Wang, Y., Wang, Y., Zhao, H.: Vip3d: End-to-end visual trajectory prediction via 3d agent queries. arXiv preprint arXiv:2208.01582 (2022) 4
13. Guo, X., Lu, J., Zhang, C., Wang, Y., Duan, Y., Yang, T., Zhu, Z., Chen, L.: Openstereo: A comprehensive benchmark for stereo matching and strong baseline. arXiv preprint arXiv:2312.00343 (2023) 1
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 5
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016) 10
16. Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., Cipolla, R., Kendall, A.: Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In: ICCV (2021) 4

17. Hu, P., Huang, A., Dolan, J., Held, D., Ramanan, D.: Safe local motion planning with self-supervised freespace forecasting. In: CVPR (2021) 11
18. Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D.: St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In: ECCV (2022) 2, 4, 6, 10, 11
19. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: CVPR. pp. 17853–17862 (2023) 2, 4, 6, 10, 11
20. Huang, J., Huang, G., Zhu, Z., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021) 1, 2, 3
21. Huang, Y., Zheng, W., Zhang, B., Zhou, J., Lu, J.: Selfocc: Self-supervised vision-based 3d occupancy prediction. In: CVPR. pp. 19946–19956 (2024) 3
22. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: CVPR. pp. 9223–9232 (2023) 2, 3
23. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. In: ECCV (2024) 3
24. Jiang, B., Chen, S., Wang, X., Liao, B., Cheng, T., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C.: Perceive, interact, predict: Learning dynamic and static clues for end-to-end motion prediction. arXiv preprint arXiv:2212.02181 (2022) 4, 6
25. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. arXiv preprint arXiv:2303.12077 (2023) 2, 4, 6, 9, 10, 11, 12, 13, 14
26. Jiang, Y., Zhang, L., Miao, Z., Zhu, X., Gao, J., Hu, W., Jiang, Y.G.: Polarformer: Multi-camera 3d object detection with polar transformers. arXiv preprint arXiv:2206.15398 (2022) 3
27. Khurana, T., Hu, P., Dave, A., Ziglar, J., Held, D., Ramanan, D.: Differentiable raycasting for self-supervised occupancy forecasting. In: ECCV (2022) 11
28. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) 7
29. Li, Q., Wang, Y., Wang, Y., Zhao, H.: Hdmapnet: An online hd map construction and evaluation framework. In: ICRA (2022) 2, 3
30. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. arXiv preprint arXiv:2206.10092 (2022) 1, 2, 3, 5
31. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV (2022) 1, 2, 3, 5, 9
32. Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., Urtasun, R.: Learning lane graph representations for motion forecasting. In: ECCV (2020) 4
33. Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., Tang, T., Wang, B., Tang, Z.: Bevfusion: A simple and robust lidar-camera fusion framework. arXiv preprint arXiv:2205.13790 (2022) 3
34. Liao, B., Chen, S., Wang, X., Cheng, T., Zhang, Q., Liu, W., Huang, C.: Maptr: Structured modeling and learning for online vectorized hd map construction. arXiv preprint arXiv:2208.14437 (2022) 2, 3, 5
35. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017) 5
36. Liu, Y., Wang, Y., Wang, Y., Zhao, H.: Vectormapnet: End-to-end vectorized hd map learning. arXiv preprint arXiv:2206.08920 (2022) 2, 3, 5

37. Liu, Y., Zhang, J., Fang, L., Jiang, Q., Zhou, B.: Multimodal motion prediction with stacked transformers. In: CVPR (2021) 4
38. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. arXiv preprint arXiv:2203.05625 (2022) 3
39. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. arXiv preprint arXiv:2205.13542 (2022) 3
40. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 11
41. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 11
42. Mao, J., Xue, Y., Niu, M., Bai, H., Feng, J., Liang, X., Xu, H., Xu, C.: Voxel transformer for 3d object detection. In: ICCV. pp. 3164–3173 (2021) 3
43. Ngiam, J., Caine, B., Vasudevan, V., Zhang, Z., Chiang, H.T.L., Ling, J., Roelofs, R., Bewley, A., Liu, C., Venugopal, A., et al.: Scene transformer: A unified architecture for predicting multiple agent trajectories. arXiv preprint arXiv:2106.08417 (2021) 4
44. Phan-Minh, T., Grigore, E.C., Boulton, F.A., Beijbom, O., Wolff, E.M.: Covernet: Multimodal behavior prediction using trajectory sets. In: CVPR (2020) 3
45. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: ECCV. pp. 194–210 (2020) 1, 3
46. Pini, S., Perone, C.S., Ahuja, A., Ferreira, A.S.R., Niendorf, M., Zagoruyko, S.: Safe real-world autonomous driving by learning to predict and plan with a mixture of experts. In: ICRA. pp. 10069–10075 (2023) 4
47. Ratliff, N.D., Bagnell, J.A., Zinkevich, M.A.: Maximum margin planning. In: Proceedings of the 23rd international conference on Machine learning. pp. 729–736 (2006) 11
48. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: CVPR (2021) 1, 3
49. Scheel, O., Bergamini, L., Wolczyk, M., Osiński, B., Ondruska, P.: Urban driver: Learning to drive from real-world demonstrations using policy gradients. In: Conference on Robot Learning. pp. 718–728. PMLR (2022) 4
50. Tian, X., Jiang, T., Yun, L., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. arXiv preprint arXiv:2304.14365 (2023) 2, 3
51. Tong, W., Sima, C., Wang, T., Chen, L., Wu, S., Deng, H., Gu, Y., Lu, L., Luo, P., Lin, D., et al.: Scene as occupancy. In: ICCV. pp. 8406–8415 (2023) 2, 3, 11
52. Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Physical review E **62**(2), 1805 (2000) 4
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017) 3, 4
54. Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., Ye, Y., Du, D., Lu, J., Wang, X.: Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. arXiv preprint arXiv:2303.03991 (2023) 2
55. Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Zhou, J., Lu, J.: Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In: ICCV. pp. 21729–21740 (2023) 2, 3
56. Yuan, Y., Weng, X., Ou, Y., Kitani, K.M.: Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In: ICCV. pp. 9813–9823 (2021) 4
57. Zeng, S., Zheng, W., Lu, J., Yan, H.: Hardness-aware scene synthesis for semi-supervised 3d object detection. TMM (2024) 3

58. Zeng, W., Luo, W., Suo, S., Sadat, A., Yang, B., Casas, S., Urtasun, R.: End-to-end interpretable neural motion planner. In: CVPR (2019) 11
59. Zhang, Y., Zhu, Z., Zheng, W., Huang, J., Huang, G., Zhou, J., Lu, J.: Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. arXiv preprint arXiv:2205.09743 (2022) 2, 3, 4
60. Zhang, Y., Zhu, Z., Zheng, W., Huang, J., Huang, G., Zhou, J., Lu, J.: Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. arXiv preprint arXiv:2205.09743 (2022) 3, 9
61. Zhao, L., Xu, X., Wang, Z., Zhang, Y., Zhang, B., Zheng, W., Du, D., Zhou, J., Lu, J.: Lowrankocc: Tensor decomposition and low-rank recovery for vision-based 3d semantic occupancy prediction. In: CVPR. pp. 9806–9815 (2024) 3
62. Zheng, W., Chen, W., Huang, Y., Zhang, B., Duan, Y., Lu, J.: Occworld: Learning a 3d occupancy world model for autonomous driving. In: ECCV (2024) 2, 11
63. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: CVPR. pp. 4490–4499 (2018) 3
64. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020) 5
65. Zuo, S., Zheng, W., Huang, Y., Zhou, J., Lu, J.: Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. arXiv preprint arXiv:2308.16896 (2023) 2, 3