GaussianAD: Gaussian-Centric End-to-End Autonomous Driving

Wenzhao Zheng[1,*,†] Junjie Wu[2,*] Yao Zheng[2,*] Sicheng Zuo[1] Zixun Xie[2,3]
Longchao Yang[2] Yong Pan[2] Zhihui Hao[2] Peng Jia[2] Xianpeng Lang[2] Shanghang Zhang[3,‡]
[1]Tsinghua University [2]Li Auto [3]Peking University
wenzhao.zheng@outlook.com; shanghang@pku.edu.cn
Project Page: https://wzzheng.net/GaussianAD
Large Driving Models: https://github.com/wzzheng/LDM

Abstract

Vision-based autonomous driving shows great potential due to its satisfactory performance and low costs. Most existing methods adopt dense representations (e.g., bird's eye view) or sparse representations (e.g., instance boxes) for decision-making, which suffer from the trade-off between comprehensiveness and efficiency. This paper explores a Gaussian-centric end-to-end autonomous driving (GaussianAD) framework and exploits 3D semantic Gaussians to extensively yet sparsely describe the scene. We initialize the scene with uniform 3D Gaussians and use surrounding-view images to progressively refine them to obtain the 3D Gaussian scene representation. We then use sparse convolutions to efficiently perform 3D perception (e.g., 3D detection, semantic map construction). We predict 3D flows for the Gaussians with dynamic semantics and plan the ego trajectory accordingly with an objective of future scene forecasting. Our GaussianAD can be trained in an end-to-end manner with optional perception labels when available. Extensive experiments on the widely used nuScenes dataset verify the effectiveness of our end-to-end GaussianAD on various tasks including motion planning, 3D occupancy prediction, and 4D occupancy forecasting. Code: https://github.com/wzzheng/GaussianAD.

1. Introduction

Vision-based autonomous driving emerges as a promising direction due to its resemblance with human driving and economic sensor configuration [20, 31, 32, 42, 45]. Despite the lack of depth inputs, vision-based methods exploit deep networks to infer structural information from RGB cameras and demonstrate strong performance in various tasks, such as 3D object detection [20, 31, 32], HD map construction [30, 34, 38, 62], and 3D occupancy prediction [21, 22, 49, 50, 55, 56, 66].

Recent autonomous driving research is undergoing a shift from the modular [21, 32, 62] to the end-to-end paradigm, which aims to plan the future trajectory directly from image inputs [18, 19, 28, 59, 64]. The key advantage of the end-to-end pipeline is less information loss from the inputs to the outputs, making it important to design the intermediate 3D scene representation of 2D images. Conventional methods compress the 3D scene in the height dimension to obtain the bird's eye view (BEV) representation [19, 59]. Recent methods explore sparse queries (e.g. instance boxes, map elements) to describe the surrounding scene [48, 61]. Despite their efficiency, they cannot capture the fine-grained structure of the 3D environment, providing less knowledge to the decision-making process. Furthermore, some methods employ tri-perspective view [21, 47, 66] or voxels [50, 56, 63] to represent scenes as 3D occupancy to capture more comprehensive details. However, the dense modeling leads to large computation overhead and thus fewer resources to reason about decision-

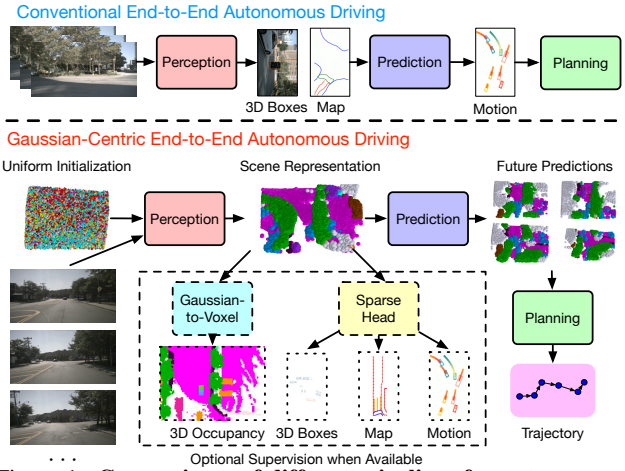

Figure 1. **Comparisons of different pipelines for autonomous driving.** Conventional end-to-end autonomous driving methods usually obtain refined scene descriptions (e.g., 3D boxes, maps) as the interface for prediction and planning, which may omit certain critical information. Differently, the proposed GuassianAD employs sparse yet comprehensive 3D Gaussians to pass information through the pipeline to efficiently preserve more details. We can optionally impose dense or sparse supervision to instruct the learning of scene representations. Our pipeline can adapt to various data with different available annotations.

*Equal contributions. †Project leader. ‡Corresponding author.

making. This raises a natural question: *can we design a comprehensive yet sparse intermediate representation to pass information through the end-to-end model?*

This paper proposes a Gaussian-centric autonomous driving (GaussianAD) framework as a positive answer, as shown in Figure 1. We employ a sparse set of 3D semantic Gaussians [23] from 2D images as the scene representations. Despite the sparsity, it benefits from fine-grained modeling resulting from the universal approximation of Gaussian mixtures and explicit 3D structure facilitating various downstream tasks. We further explore perception, prediction, and planning from the 3D Gaussian representation. For perception, we treat 3D Gaussians as semantic point clouds and employ sparse convolutions and sparse prediction heads to efficiently process the 3D scene. We propose 3D Gaussian flow to comprehensively and explicitly model the scene evolution, where we predict a future displacement for each Gaussian. We then integrate all available information to plan the ego trajectory accordingly. Due to the explicitness of 3D Gaussian representation, we can straightforwardly compute the forecasted future scenes observed by the ego car using affine transformations. We compare the forecasted scenes with ground-truth scene observations as explicit supervision for both prediction and planning. To the best of our knowledge, our GaussianAD is the first to explore the explicitly sparse point-based architecture for vision-centric end-to-end autonomous driving. We conduct extensive experiments on the nuScenes [3] dataset to evaluate the effectiveness of the proposed Gaussian-centric framework. Experimental results demonstrate that our GaussianAD achieves state-of-the-art results on end-to-end motion planning with high efficiency.

2. Related Work

Perception for Autonomous Driving. Accurately perceiving the surrounding environment from sensor inputs is the fundamental step for autonomous driving. As the two main conventional perception tasks, 3D object detection aims to obtain the 3D position, pose, and category of each agent in the surrounding scene [20, 31, 32, 42, 45, 62], which are important for trajectory prediction and planning. Semantic map reconstruction aims to recover the static map elements in the bird's eye view (BEV) to provide additional information for further inference [30, 34, 38, 62]. Both tasks can be efficiently performed in the BEV space, yet they cannot describe the fine-grained 3D structure of the surrounding scene and arbitrary-shape objects [21, 56]. This motivates recent methods to explore other 3D representations like voxel and tri-perspective view (TPV) [21] to perform the 3D occupancy prediction task [49, 50, 55, 56, 66]. 3D occupancy provides more comprehensive descriptions of the surrounding scene including both dynamic and static elements, which can be efficiently learned from sparse LiDAR [21] or video sequences [4]. Gaussianformer [23] proposed to use 3D semantic Gaussians to represent the scene for 3D occupancy sparsely. However, it is still not clear whether the 3D Gaussian representation can be used for general autonomous driving.

Prediction for Autonomous Driving. Predicting the scene evolution is also vital to the safety of autonomous driving vehicles. Most existing methods focus on predicting the movement of traffic agents given their past positions and semantic map information [13, 16, 27, 37, 41, 58, 62]. Early methods projected agent and semantic map information onto BEV images and employed 2D image backbones to process them to infer future agent motions [5, 41]. Subsequent methods adopted a more efficient tokenized representation of dynamic agents and used graph neural networks [33] or transformers [37, 40, 52] to aggregate information. Recent works began to explore motion prediction directly from sensor inputs in an end-to-end manner [13, 16, 27, 28, 62]. They usually first perform BEV perception to extract relevant information (e.g., 3D agent boxes, semantic maps, tracklets) and then exploit them to infer future trajectories. Different from existing methods which only model dynamic object motions, we propose Gaussian flows to predict the surrounding scene evolutions including both dynamic and static elements.

Planning for Autonomous Driving. Planning is the essential component of autonomous driving systems, which can be categorized into rule-based [1, 12, 51] and learning-based [8, 43, 46] methods. While the traditional rule-based methods can achieve satisfactory results with high interpretability [12], learning-based methods have received increasing attention in recent years due to their great potential to scale up to large-scale training data [2, 11, 24, 36, 57, 65]. As simple yet effective learning-based solutions, imitation-based planners have been the preferred choices for end-to-end methods [9, 14, 25, 26, 53, 65]. As early attempts, LBC [6] and CILRS [10] employed convolutional neural networks (CNNs) to learn from expert driving data. The following methods incorporated more data [59] or extracted more intermediate features [18, 19, 28, 64] to provide more information for the planner, which achieved remarkable performance. Still, most existing end-to-end autonomous driving methods adopt high-level scene descriptions (e.g., 3D boxes, maps) for downstream prediction and planning and may omit certain critical information. This paper proposes a Gaussian-centric autonomous driving pipeline and uses 3D Gaussians as sparse yet comprehensive information carrier.

3. Proposed Approach

3.1. 3D Scene Representation Matters for Driving

Autonomous driving aims to produce safe and consistent control signals (e.g., accelerator, brake, steer) given a series

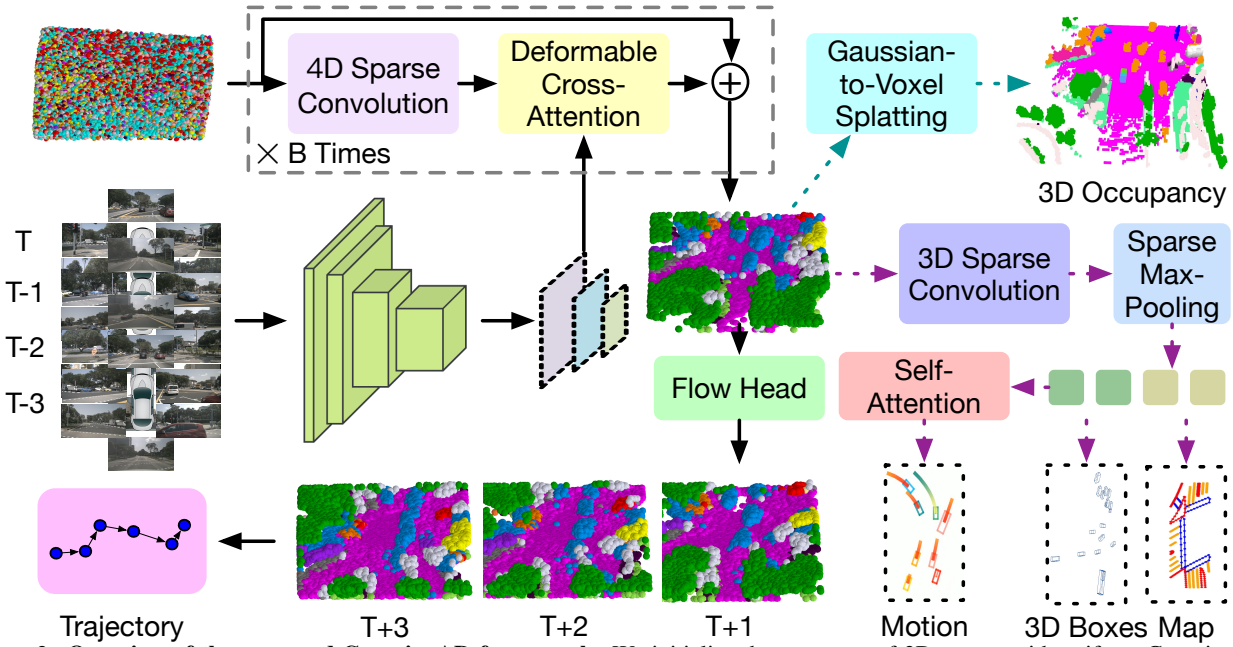

Figure 2. **Overview of the proposed GaussianAD framework.** We initialize the sequence of 3D scenes with uniform Gaussians and employ 4D sparse convolutions to enable interactions between Gaussians. We then extract multi-scale features from surrounding-view multi-frame image observations and use deformable cross-attention to incorporate them into the 3D Gaussians. Having obtained the temporal 3D Gaussians as the scene representation, we can optionally employ Gaussian-to-voxel splatting [23] for dense tasks (e.g., 3D semantic occupancy) or use sparse convolutions and max-pooling [7] for sparse tasks (e.g., 3D object detection, HD map construction, motion prediction). We use a flow head to predict a 3D flow for each Gaussian and aggregate them for trajectory planning.

of scene observations $\{\mathbf{o}\}$. While the scene observations $\{\mathbf{o}\}$ can be obtained from multiple sensors such as cameras and LiDAR, we mainly target vision-based autonomous driving from surrounding cameras due to its high information density and low sensor costs [18, 19, 28, 50, 59, 64].

Assuming a good-performing controller, most autonomous driving models mainly focus on learning the mapping f from the current and history observations $\{\mathbf{o}\}$ to the future ego trajectories $\{\mathbf{w}\}$:

$$\{\mathbf{o}^{T-H}, \cdots, \mathbf{o}^{T}\} \xrightarrow{f} \{\mathbf{w}^{T+1}, \cdots, \mathbf{w}^{T+F}\}, \tag{1}$$

where T denotes the current time stamp, H is the number of history frames, and F is the number of predicted future frames. Each waypoint $\mathbf{w} = \{x, y, \psi\}$ is determined by the 2D position $\{x, y\}$ and the yaw angle ψ (i.e., the advancing direction of the ego vehicle) in the bird's eye view (BEV).

Conventional autonomous driving methods decompose f into perception, prediction, and planning modules and train them separately before connecting [5, 12, 21, 32, 41]:

$$\begin{aligned} &\text{Perception: } \{\mathbf{o}^{T-H}, \cdots, \mathbf{o}^{T}\} \rightarrow \mathbf{d}^{T}, \\ &\text{Prediction: } \mathbf{d}^{T} \rightarrow \{\mathbf{d}^{T+1}, \cdots, \mathbf{d}^{T+F}\}, \\ &\text{Planning: } \{\mathbf{d}^{T+1}, \cdots, \mathbf{d}^{T+F}\} \rightarrow \{\mathbf{w}^{T+1}, \cdots, \mathbf{w}^{T+F}\}, \end{aligned} \tag{2}$$

where $\mathbf{d}$ is the scene description such as instance bounding boxes of other agents or map elements of the surroundings. The scene description $\mathbf{d}$ usually only provides a partial representation of the scene, resulting in information loss.

The separate training of these modules further aggravates this issue as different tasks focus on extracting different information. The incomprehensive information provided to the planning module might bias the decision-making process of the autonomous driving model. This motivates the shift from the modular framework to the end-to-end framework [19, 61, 64], which differentiably bridges and jointly learns the perception, prediction, and planning modules:

$$\begin{aligned} &\{\mathbf{o}^{T-H}, \cdots, \mathbf{o}^{T}\} \rightarrow \mathbf{r}^{T} \rightarrow \mathbf{r}^{T}, \mathbf{d}^{T} \rightarrow \\ &\mathbf{r}^{T}, \{\mathbf{d}^{T+1}, \cdots, \mathbf{d}^{T+F}\} \rightarrow \{\mathbf{w}^{T+1}, \cdots, \mathbf{w}^{T+F}\}, \end{aligned} \tag{3}$$

where $\mathbf{r}$ is the scene representation. $\mathbf{r}$ is usually composed of a set of continuous features and provides a more comprehensive representation of the 3D scenes than $\mathbf{d}$.

The scene representation $\mathbf{r}$ conveys information throughout the model, making the choice of $\mathbf{r}$ critical to the performance of the end-to-end system. As autonomous driving needs to make decisions in the 3D space, the scene representation should be 3D-structured and contain 3D structural information inferred from the input images. On the other hand, 3D space is usually sparse, resulting in a tradeoff between comprehensiveness and efficiency when designing $\mathbf{r}$.

For comprehensiveness, the conventional bird's eye view (BEV) representation [31, 32, 62] uses dense grid features in the map view and compresses the height dimension to reduce redundancy. Subsequent methods further explore more dense representations such as voxels [56] or tri-

perspective view (TPV) [21] to capture more detailed and fine-grained 3D information. For efficiency, recent methods [48, 61] adopt sparse queries and focus on modeling instance boxes and map elements, which are the most important factors for decision-making. Still, the discarded information can still be important (e.g., irregular obstacles, traffic lights, human poses) and is contradictory to the philosophy of end-to-end autonomous driving (i.e., comprehensive information flow). This paper explores the 3D Gaussians as a comprehensive yet sparse scene representation and proposes a fully sparse framework for end-to-end perception, prediction, and planning, as shown in Figure 2.

3.2. Gaussian-Centric Autonomous Driving

3D Gaussian Representation. Existing methods typically build a dense 3D feature to represent the surrounding environment and processes every 3D voxel with equal storage and computation resources, which often leads to intractable overhead because of unreasonable resource allocation. At the same time, this dense 3D voxel representation cannot distinguish objects of different scales. Unlike these methods, we follow GaussianFormer [23] which represents an autonomous driving scene with a number of sparse 3D semantic Gaussians. Each Gaussian instantiates a semantic Gaussian distribution characterized by mean, covariance, and semantic logits. This sparse explicit feature representation is more beneficial for downstream tasks.

Gaussians From Images. We first represent 3D Gaussians and their high-dimensional queries as learnable vectors. We then employ a Gaussian encoder to iteratively enhance these representations. Each Gaussian Encoder block is composed of three modules: a self-encoding module facilitating interactions between Gaussians, an image cross-attention module for aggregating visual information, and a refinement module to fine-tune Gaussian properties. Different from GaussianFormer [23], we utilize a temporal encoder consisting of 4D sparse convolutions to integrate Gaussian features from the previous frame with the corresponding features in the current frame.

Sparse 3D Detection from Gaussians. As 3D Gaussian representation is a sparse scene representation, we follow VoxelNeXt [7] which predicts 3D objects directly based on sparse voxel features. Specially, we conduct a 3D sparse CNN network $\mathbf{V}$ to encode 3D Gaussian representation $\mathbf{r}$. Following GenAD [64], we decode 3D objects $\mathbf{a}$ with a set of agent tokens $\mathbf{D}$ on $\mathbf{V}(\mathbf{r})$:

$$\mathbf{a} = f_a(\mathbf{D}, \mathbf{V}(\mathbf{r})), \tag{4}$$

where f_a represents a combine of global cross-attention mechanism to learn 3D objects tokens and a 3D object decoder head $\mathbf{d_a}$ on learned 3D objects tokens.

Sparse Map Construction from Gaussians. Similar to the representation of 3D detection from Gaussian, we adopt a set of map tokens $\mathbf{M}$ to represent semantic maps. We focus on three categories of map elements(i.e., lane divider, road boundary, and pedestrian crossing).

$$\mathbf{m} = f_m(\mathbf{M}, \mathbf{V}(\mathbf{r})), \tag{5}$$

where f_m represents a combination of a global cross-attention mechanism to learn map tokens and a semantic map elements decoder head $\mathbf{d_m}$ on learned map tokens.

Motion Prediction. Motion prediction module assists ego trajectories planning by forecasting the future trajectories of other traffic participants [19]. We obtain motion tokens $\mathbf{M_o}$ by make agent tokens $\mathbf{D}$ interact with map tokens $\mathbf{M}$ through cross-attention layers CA:

$$\mathbf{M_o} = CA(\mathbf{D}, \mathbf{M}). \tag{6}$$

A motion decoder $\mathbf{d_{mo}}$ can be applied on motion tokens $\mathbf{M_o}$, meanwhile the learned motion tokens $\mathbf{M_o}$ are fed to ego trajectory planning head.

Gaussian Flow for Scene Prediction. Furthermore, it shows that the scene prediction of the intermediate representation $\mathbf{r}$ plays a significant role in end-to-end autonomous driving [63]. We predict the future Gaussian representation as Gaussian flow $\mathbf{r^{T+N}}$ from current Gaussian representation $\mathbf{r^T}$ and the predicted ego trajectories $\mathbf{w}^{T+N}$:

$$\mathbf{r}^{T+N} = f_r(\mathbf{r}^T, \mathbf{w}^{T+N}). \tag{7}$$

We then feed the predicted future Gaussian representation $\mathbf{r^{T+N}}$ to an occupancy decoder $\mathbf{d}_{occ}$ [23] to predict future occupancy. The supervision of future occupancy on the intermediate Gaussian representation guarantees the scene forecasting ability which finally improves the performance of ego trajectory prediction.

3.3. End-to-End GaussianAD Framework

This subsection presents the overall end-to-end framework of our proposed GaussianAD. We first initialize the scenes with a set of uniformly distributed 3D Gaussians $\mathbf{G}_0$ and then progressively refine them by incorporating information from the surrounding-view images $\mathbf{o}$ to obtain the Gaussian scene representation $\mathbf{r}$. We can then optionally extract various scene descriptions $\mathbf{d}$ from $\mathbf{r}$ as auxiliary tasks if the corresponding annotations are available. Concretely, we employ Gaussian-to-voxel splatting [23] to obtain dense voxel features for dense descriptions (e.g., 3D occupancy prediction) and fully sparse convolutions [7] to obtain sparse queries for sparse descriptions (e.g., 3D bounding boxes, map elements). The use of auxiliary perception supervisions introduces additional constraints and prior knowledge on the scene representation $\mathbf{r}$ to guide its learning process. Still, we predict future evolutions directly on the 3D Gaussians $\mathbf{r}$ to reduce information loss and plan the ego trajectory $\{\mathbf{w}\}$ accordingly. GaussianAD passes information

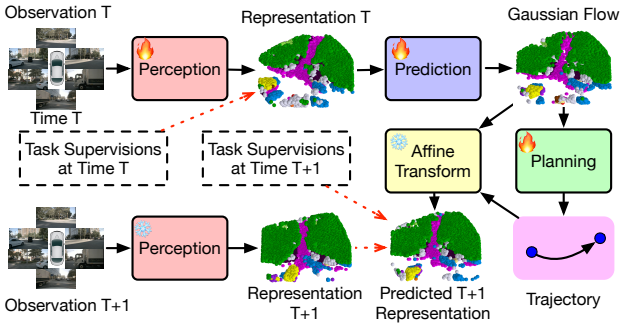

Figure 3. **Illustration of the training of our GaussianAD.** Our framework can accommodate training data with different annotations by optionally imposing the corresponding supervisions on the scene representation. Due to the explicit and structural nature of 3D Gaussians, we use global affine transformation to predict the future scene representations observed by the ego vehicle following the planned trajectory. We can then use future perception labels or future scene representations obtained from future observations as the supervision. They impose stronger constraints on the planned trajectory than the low-dimension trajectory discrepancy loss.

throughout the model with the sparse yet comprehensive 3D Gaussian representation, providing more knowledge to the decision-making process. The overall framework of our GaussianAD is formulated as follows:

$$\begin{aligned}&\{\mathbf{o}^{T-H},\cdots,\mathbf{o}^{T}\}\rightarrow\mathbf{r}^{T}(\rightarrow\mathbf{r}^{T},\mathbf{d}^{T})\rightarrow\\&\{\mathbf{r}^{T},\mathbf{r}^{T+1},\cdots,\mathbf{r}^{T+F}\}\rightarrow\{\mathbf{w}^{T+1},\cdots,\mathbf{w}^{T+F}\},\end{aligned}\tag{8}$$

where $(\rightarrow\mathbf{r}^{T},\mathbf{d}^{T})$ means that it is optional to incorporate additional perception supervision with $\mathbf{d}$ when available.

For training, we adaptively impose different perception losses on the scene descriptions $\mathbf{d}$ extracted from $\mathbf{r}$:

$$\begin{aligned}&J_{perc}(\mathbf{d},\hat{\mathbf{d}})=\lambda_{occ}J_{occ}(\mathbf{d},\hat{\mathbf{d}})+\lambda_{det}J_{det}(\mathbf{d},\hat{\mathbf{d}})\\&+\lambda_{map}J_{map}(\mathbf{d},\hat{\mathbf{d}})+\lambda_{motion}J_{motion}(\mathbf{d},\hat{\mathbf{d}}),\end{aligned}\tag{9}$$

where λ_{occ}, λ_{det}, λ_{map}, and λ_{motion} are balance factors and equal 0 if the supervision is not available. $\hat{\mathbf{d}}$ denotes the ground-truth descriptions. We use 3D occupancy prediction loss [23] as J_{occ}, 3D detection loss [32] as J_{det}, semantic map loss [62] as J_{map}, and motion loss [28] as J_{motion}.

Due to the explicit representation of 3D Gaussians, we can use global affine transformation t to simulate the scene representation $\widetilde{\mathbf{r}}$ observed at a certain given ego position $\mathbf{w}$. Having obtained the predicted future scene representations $\{\mathbf{r}^{T},\mathbf{r}^{T+1},\cdots,\mathbf{r}^{T+F}\}$ with the proposed Gaussian flows, we simulate the future ego scene representations using the planned waypoints $\{\mathbf{w}^{T+1},\cdots,\mathbf{w}^{T+F}\}$:

$$\{\widetilde{\mathbf{r}}=t(\mathbf{r},\mathbf{w})\}^{F},\tag{10}$$

where ^{F} denotes the future F frames. We then use the discrepancy between the simulated representations $\{\widetilde{\mathbf{r}}\}^{F}$ and ground truth representations $\{\hat{\mathbf{r}}\}^{F}$ as the loss:

$$\begin{aligned}&J_{pred}(\{\mathbf{r}\}^{F},\{\hat{\mathbf{r}}\}^{F},\{\hat{\mathbf{d}}\}^{F})=\lambda_{re}J_{re}(\{\widetilde{\mathbf{r}}\}^{F},\{\hat{\mathbf{r}}\}^{F})\\&+\lambda_{perc}J_{perc}(\{\widetilde{\mathbf{d}}(\widetilde{\mathbf{r}})\}^{F},\{\hat{\mathbf{d}}\}^{F}),\end{aligned}\tag{11}$$

where λ_{re} and λ_{perc} are balance factors, and J_{re} computes the discrepancy between two Gaussian representations. $\{\hat{\mathbf{r}}\}^{F}$ can be computed from future observations $\{\mathbf{o}\}$. $\widetilde{\mathbf{d}}(\widetilde{\mathbf{r}})$ denotes the predicted descpritions $\widetilde{\mathbf{d}}$ extracted from $\widetilde{\mathbf{r}}$.

The predicted future ego scene representations $\{\widetilde{\mathbf{r}}\}^{F}$ also depend on the planned trajectories $\{\mathbf{w}\}^{F}$ as in (10). Therefore, we further adopt the prediction loss (11) for planning in addition to the conventional trajectory loss:

$$\begin{aligned}&J_{plan}(\{\mathbf{w}\}^{F},\{\hat{\mathbf{w}}\}^{F})=\lambda_{tra}J_{tra}(\{\mathbf{w}\}^{F},\{\hat{\mathbf{w}}\}^{F})\\&+\lambda_{pred}J_{pred}(\{\mathbf{r}\}^{F},\{\hat{\mathbf{r}}\}^{F},\{\hat{\mathbf{d}}\}^{F}),\end{aligned}\tag{12}$$

where λ_{tra} and λ_{pred} are balance factors, and $\hat{\mathbf{w}}$ denotes the ground truth waypoint. We adopt the trajectory losses from GenAD [64] as J_{tra}.

The proposed GaussianAD is a flexible framework and can accommodate various cases with different available supervisions, as shown in Figure 3. We train GaussianAD jointly with the following overall objective:

$$J_{GaussianAD}=J_{perc}+J_{pred}+J_{plan},\tag{13}$$

where J_{perc}, J_{pred}, and J_{plan} can be customized for different scenarios.

For inference, GaussianAD accomplishes end-to-end driving using 3D Gaussian representation to efficiently pass information throughout the pipeline. It provides comprehensive knowledge for the decision-making process and maintains high efficiency with sparse computing.

4. Experiments

4.1. Datasets

We conducted a series of experiments using the widely used nuScenes [3] dataset to evaluate our GaussianAD. The nuScenes dataset consists of 1000 driving sequences, each providing 20 seconds of video captured by both RGB and LiDAR sensors. They provide data with a rate of 20Hz but only supply annotations for the keyframes at 2Hz, including labels for the semantic map construction and 3D object detection tasks. The recent SurroundOcc [56] further complements nuScenes with 3D semantic occupancy annotations. It assigns each voxel with a label of 18 categories including 16 semantic classes, 1 empty class, and 1 unknown class.

4.2. Evaluation Metrics

We evaluate the planning performance of our GaussianAD using the L2 displacement error and collision rate for fair comparisons with existing end-to-end methods [18, 19, 63,

Table 1. **Open-looped motion planning results in comparison with state-of-the-art methods on the validation set of nuScenes [3].** † denotes the results computed with an average of previous frames as adopted in VAD [28]. Aux. Sup. represents auxiliary supervision in addition to planning. Avg. computes the average result of 1s, 2s, and 3s. Bold numbers represent the best results.

Method	Input	Aux. Sup.	L2 (m) ↓				Collision Rate (%) ↓			
			1s	2s	3s	Avg.	1s	2s	3s	Avg.
IL [44]	LiDAR	None	0.44	1.15	2.47	1.35	0.08	0.27	1.95	0.77
NMP [60]	LiDAR	Box & Motion	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 [29]	LiDAR	Freespace	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
ST-P3 [18]	Camera	Map & Box & Depth	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD [19]	Camera	Map & Box & Motion & Tracklets & Occ	0.48	0.96	1.65	1.03	0.05	**0.17**	0.71	**0.31**
VAD-Tiny [28]	Camera	Map & Box & Motion	0.60	1.23	2.06	1.30	0.31	0.53	1.33	0.72
VAD-Base [28]	Camera	Map & Box & Motion	0.54	1.15	1.98	1.22	**0.04**	0.39	1.17	0.53
GenAD [64]	Camera	Map & Box & Motion	**0.36**	0.83	1.55	0.91	0.06	0.23	1.00	0.43
GaussianAD	Camera	3D-Occ & Map & Box & Motion	0.40	**0.64**	**0.88**	**0.64**	0.09	0.38	0.81	0.42
OccWorld [63]	Camera	3D-Occ	0.52	1.27	2.41	1.40	**0.12**	0.40	2.08	0.87
OccNet [50]	Camera	3D-Occ & Map & Box	1.29	2.13	2.99	2.14	0.21	0.59	1.37	0.72
GaussianAD	Camera	3D-Occ & Map & Box	0.40	0.66	0.92	0.66	0.49	0.38	**0.61**	0.49
VAD-Tiny† [28]	Camera	Map & Box & Motion	0.46	0.76	1.12	0.78	0.21	0.35	0.58	0.38
VAD-Base† [28]	Camera	Map & Box & Motion	0.41	0.70	1.05	0.72	**0.07**	0.17	0.41	0.22
OccWorld-D† [63]	Camera	3D-Occ	0.39	0.73	1.18	0.77	0.11	0.19	0.67	0.32
GenAD† [64]	Camera	Map & Box & Motion	**0.28**	0.49	0.78	0.52	0.08	**0.14**	**0.34**	**0.19**
GaussianAD†	Camera	3D-Occ & Map & Box	0.34	**0.47**	**0.60**	**0.47**	0.49	0.49	0.51	0.50

Table 2. **Comparisons on 3D perception.** We report results on the 3D object detection, 3D occupancy prediction, and planning tasks. † denotes using a perception region of 51.2m × 51.2m.

Method	Detection	Occupancy		Planning	
	mAP ↑	mIoU ↑	IoU ↑	Avg. L2 ↓	Avg. CR. ↓
VAD [28]	0.27	-	-	1.30	0.72
GenAD [64]	**0.29**	-	-	0.91	0.43
TPVFormer† [21]	-	17.10	30.86	-	-
GaussianFormer† [23]	-	19.10	29.83	-	-
SurroundOcc† [56]	-	20.30	31.49	-	-
GaussianAD	0.19	**22.12**	**33.81**	**0.64**	**0.42**

64]. The L2 displacement error quantifies the difference between the planned and the ground-truth trajectory, computed as the L2 distance. The collision rate indicates the frequency with which the autonomous vehicle collides with other agents while following the planned path. For evaluation, we use a 2-second 5-frame history as input and compute the metric at future time steps of 1s, 2s, and 3s.

4.3. Implementation Details

We employ ResNet101-DCN [15] with pre-trained weights from FCOS3D [54] as the backbone and additionaly use a feature pyramid network [35] to generate multi-scale image features. Our model takes as input images with a resolution of 1600 × 900 and sets the default number of Gaussians to 25600. In the training stage, we take the AdamW [39] with a weight decay of 0.01 as the optimizer. The learning rate begins with 2e-4 and decreases according to a cosine schedule. By default, our models are trained on 32 A100 GPUs with a batch size of 8 for 20 epochs.

4.4. Results and Analysis

End-to-End Planning Results. We provide comparisons with state-of-the-art end-to-end autonomous driving models in Table 1. Bold numbers and underlined numbers denote the best and next-best results, respectively. We also report the metrics used in VAD [28], which computes the average results of all the previous frames at each time stamp.

Note that different methods use different input modalities and auxiliary supervision signals that may influence the performance. Generally, LiDAR provides additional depth information that is critical for planning, especially when measuring the collision rate. However, the LiDAR point clouds, though accurate, are usually sparse and lack more fine-grained information, yielding inferior performance. For auxiliary supervision, motions are usually considered the most effective labels as they provide ground truth for safety-critical future predictions. Still, motions are relatively expensive to annotate while 3D occupancy labels can be automatically annotated using multi-frame LiDAR and 3D bounding boxes [56]. Though our GaussianAD can accommodate different supervision signals, we replace motion with 3D occupancy as the most practical setting.

Table 1 shows that our method achieves the best perfor-

Table 3. **Results of the scene prediction performance.** We report results on the 4D occupancy forecasting task [63]. Aux. Sup. represents auxiliary supervision in addition to planning. Avg. computes the average result of 1s, 2s, and 3s. * denotes using an end-to-end model.

Method	Input	Aux. Sup.	mIoU (%) ↑					IoU (%) ↑				
			0s	1s	2s	3s	Avg.	0s	1s	2s	3s	Avg.
Copy&Paste	3D-Occ	None	66.38	14.91	10.54	8.52	11.33	62.29	24.47	19.77	17.31	20.52
OccWorld-O [63]	3D-Occ	None	**66.38**	**25.78**	**15.14**	**10.51**	**17.14**	**62.29**	**34.63**	**25.07**	**20.18**	**26.63**
OccWorld-T [63]	Camera	Semantic LiDAR	7.21	4.68	3.36	2.63	3.56	10.66	9.32	8.23	7.47	8.34
OccWorld-S [63]	Camera	None	0.27	0.28	0.26	0.24	0.26	4.32	5.05	5.01	4.95	5.00
OccWorld-D [63]	Camera	3D-Occ	18.63	11.55	8.10	6.22	8.62	22.88	18.90	16.26	14.43	16.53
GaussianAD*	Camera	3D-Occ	15.87	6.29	5.36	4.58	5.41	29.35	14.13	14.09	14.04	14.30

Table 4. **Effect of using different auxiliary supervision signals.** We analyze the effect of using additional 3D occupancy, 3D detection, map construction, motion prediction, and scene prediction labels as supervision. Bold numbers represent the best results. * represents that the use of the proposed flow-based prediction does not require additional annotations.

Occupancy	Detection	Map	Motion	Prediction*	L2 (m) ↓				Collision Rate (%) ↓			
					1s	2s	3s	Avg.	1s	2s	3s	Avg.
×	✓	✓	×	×	0.49	0.75	1.00	0.74	0.28	0.51	1.12	0.63
✓	×	×	×	×	0.41	0.67	0.91	0.66	0.26	0.43	1.19	0.62
✓	✓	×	×	×	0.41	0.67	0.92	0.66	0.24	0.68	0.83	0.58
✓	✓	✓	×	×	0.48	0.67	0.89	0.68	0.16	0.78	0.81	0.58
✓	✓	✓	✓	×	**0.40**	**0.64**	**0.88**	**0.64**	**0.09**	**0.38**	0.81	**0.42**
✓	✓	✓	×	✓	**0.40**	0.66	0.92	0.66	0.49	**0.38**	**0.61**	0.49

mance on the L2 metric and competitive results on the collision rate metric. In particular, GaussianAD outperforms OccNet [50] using the same supervision signals (i.e., 3D occupancy, maps, and 3D bounding boxes) by a large margin. Despite the lack of motion labels, our GaussianAD predicts Gaussian flows to simulate future scenes, enabling the exploitation of perception labels for the motion task. This forces the model to consider more about future interactions, resulting in the large improvements over OccNet [50].

3D Occupancy Prediction. We also provide results on other perception tasks though they are not the focus of this paper. We adopt the mean average precision (mAP) for the 3D object detection task [31, 32]. We use the mean intersection-over-union (mIoU) and intersection-over-union (IoU) for 3D occupancy prediction to measure the semantic and structural reconstruction quality, respectively.

Table 2 compares our GaussianAD with state-of-the-art end-to-end and 3D occupancy prediction methods. GaussianAD shows good results on the 3D occupancy prediction task but underperforms existing end-to-end methods on 3D object detection. This is because different perception tasks focus on different aspects of scene descriptions and could interfere with one another. This explains the inferior performance of our method on the collision metric, which requires accurate perception of other agents to avoid collision.

4D Occupancy Forecasting. By predicting a 3D flow for each Gaussian and performing the affine transformation using the planned trajectory, GaussianAD is able to forecast future scenes and perform perception on them. We evaluate the prediction ability of GaussianAD on the 4D occupancy forecasting task [63] and measure the 3D occupancy quality (mIoU and IoU) at the future 1s, 2s, and 3s.

Table 3 shows that our GaussianAD can effectively predict forecast future 3D occupancy. Note that our GaussianAD is an end-to-end model that performs multiple tasks simultaneously, while OccWorld [63] specifically targets this task. Also, our forecasting does not consider the completion of newly observed areas (due to the ego car moving forward), leading to inferior performance. GaussianAD still demonstrates non-trivial 4D forecasting results, verifying the effectiveness of the proposed Gaussian flows.

Effect of Different Supervision Signals. As our model can adapt to different training signals for different tasks, we conducted an ablation study to analyze the effect of using different auxiliary supervision, as shown in Table 4. We study the planning performance with a combination of 3D occupancy, 3D detection, map construction, motion prediction, and scene prediction supervision. We see that our GaussianAD delivers consistent performance with different supervision combinations, and using more supervision signals generally improves the performance. The use of motion supervision is particularly effective for the collision rate metric since it provides guidance on potential future overlap of trajectories. Still, using the proposed flow-based scene prediction supervision achieves similar improvements, which only requires future perception labels and introduces no additional annotations.

3D Gaussian Pruning. We also analyze the effect of further pruning the Gaussians to reduce redundancy, as shown in Table 5. We perform pruning by ordering the Gaus-

Table 5. **Effect of further Gaussian pruning.** We report results on the 3D occupancy and planning tasks.

#Gaussians	Occupancy		Detection	Planning L2 (m) ↓				Planning Collision Rate (%) ↓			
	mIoU↑	IoU↑	mAP↑	1s	2s	3s	Avg.	1s	2s	3s	Avg.
25600	**21.22**	**33.81**	**0.16**	0.48	**0.67**	**0.89**	**0.68**	0.16	0.78	0.81	0.58
20480 (-20%)	21.01	33.61	**0.16**	0.45	0.69	0.94	0.69	**0.08**	**0.29**	1.29	0.55
15360 (-40%)	20.57	33.77	0.15	**0.43**	0.68	0.93	**0.68**	0.28	0.46	**0.59**	**0.44**

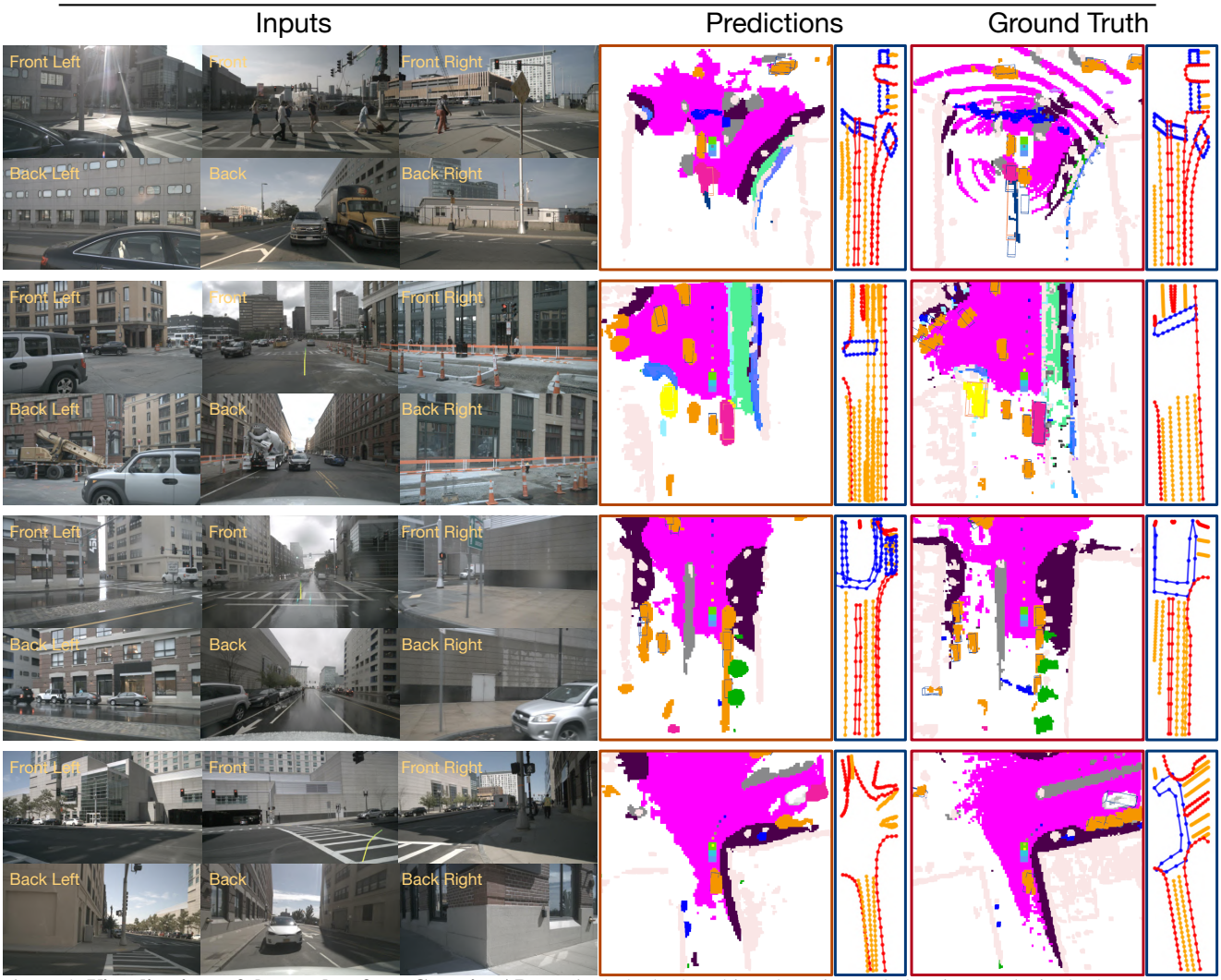

Figure 4. **Visualizations of the results of our GaussianAD.** We include the 3D object detection and planning results in the 3D occupancy visualizations. We also provide map visualizations. (Better viewed on a monitor when zoomed in.)

sians according to their semantic confidence (i.e., the largest probability in the logits) and pruning the smallest ones. We observe that Gaussian pruning slightly decreases the performance of perception tasks and yet improves the planning performance, demonstrating the potential of our framework.

Visualizations. Figure 4 provides a visualization of the outputs of GaussianAD, which effectively perceives the surroundings and makes correct decisions in various scenarios.

5. Conclusion

We have presented a Gaussian-centric framework for vision-based end-to-end autonomous driving. To preserve more comprehensive information, we employ 3D Gaussians as the scene representation and adopt Gaussian flows to effectively predict future evolutions. Our framework offers flexibility to accommodate different training data with various annotations. We have conducted extensive experiments on the widely used nuScenes and demonstrated competitive performance on various tasks including ent-to-end planning and 4D occupancy forecasting. It is interesting to explore larger-scale end-to-end models based on 3D Gaussian scene representation trained with more diverse data.

Limitations. GaussianAD cannot predict accurate scene evolutions since it does not consider newly observed areas.

References

[1] Frédéric Bouchard, Sean Sedwards, and Krzysztof Czarnecki. A rule-based behaviour planner for autonomous driving. In *IJCRR*, pages 263–279, 2022. 2

[2] Eli Bronstein, Mark Palatucci, Dominik Notz, Brandyn White, Alex Kuefler, Yiren Lu, Supratik Paul, Payam Nikdel, Paul Mougin, Hongge Chen, et al. Hierarchical model-based imitation learning for planning in autonomous driving. In *IROS*, pages 8652–8659, 2022. 2

[3] 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. 2, 5, 6

[4] Anh-Quan Cao and Raoul de Charette. Scenerf: Self-supervised monocular 3d scene reconstruction with radiance fields. In *ICCV*, pages 9387–9398, 2023. 2

[5] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 2, 3

[6] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. 2020. 2

[7] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. *arXiv preprint arXiv:2303.11301*, 2023. 3, 4

[8] Jie Cheng, Ren Xin, Sheng Wang, and Ming Liu. Mpnp: Multi-policy neural planner for urban driving. In *IROS*, pages 10549–10554, 2022. 2

[9] Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking imitation-based planner for autonomous driving. *arXiv preprint arXiv:2309.10443*, 2023. 2

[10] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. 2019. 2

[11] Gustavo Claudio Karl Couto and Eric Aislan Antonelo. Hierarchical generative adversarial imitation learning with mid-level input generation for autonomous driving on urban environments. *arXiv preprint arXiv:2302.04823*, 2023. 2

[12] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *CoRL*, 2023. 2, 3

[13] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. *arXiv preprint arXiv:2208.01582*, 2022. 2

[14] Ke Guo, Wei Jing, Junbo Chen, and Jia Pan. Ccil: Context-conditioned imitation learning for urban driving. *arXiv preprint arXiv:2305.02649*, 2023. 2

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

[16] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *ICCV*, 2021. 2

[17] Peiyun Hu, Aaron Huang, John Dolan, David Held, and Deva Ramanan. Safe local motion planning with self-supervised freespace forecasting. In *CVPR*, 2021. 6

[18] 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. 1, 2, 3, 5, 6

[19] 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*, pages 17853–17862, 2023. 1, 2, 3, 4, 5, 6

[20] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 2

[21] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, pages 9223–9232, 2023. 1, 2, 3, 4, 6

[22] Yuanhui Huang, Amonnut Thammatadatrakoon, Wenzhao Zheng, Yunpeng Zhang, Dalong Du, and Jiwen Lu. Gaussianformer-2: Probabilistic gaussian superposition for efficient 3d occupancy prediction. *arXiv preprint arXiv:2412.04384*, 2024. 1

[23] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2405.17429*, 2024. 2, 3, 4, 5, 6

[24] Zhiyu Huang, Jingda Wu, and Chen Lv. Efficient deep reinforcement learning with imitative expert priors for autonomous driving. *TNNLS*, 2022. 2

[25] Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. *arXiv preprint arXiv:2303.05760*, 2023. 2

[26] Zhiyu Huang, Haochen Liu, Jingda Wu, and Chen Lv. Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving. *TNNLS*, 2023. 2

[27] Bo Jiang, Shaoyu Chen, Xinggang Wang, Bencheng Liao, Tianheng Cheng, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, and Chang Huang. Perceive, interact, predict: Learning dynamic and static clues for end-to-end motion prediction. *arXiv preprint arXiv:2212.02181*, 2022. 2

[28] 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. *arXiv preprint arXiv:2303.12077*, 2023. 1, 2, 3, 5, 6

[29] Tarasha Khurana, Peiyun Hu, Achal Dave, Jason Ziglar, David Held, and Deva Ramanan. Differentiable raycasting for self-supervised occupancy forecasting. In *ECCV*, 2022. 6

[30] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, 2022. 1, 2

[31] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*, 2022. 1, 2, 3, 7

[32] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 1, 2, 3, 5, 7

[33] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 2

[34] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. *arXiv preprint arXiv:2208.14437*, 2022. 1, 2

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 6

[36] Haochen Liu, Zhiyu Huang, Jingda Wu, and Chen Lv. Improved deep reinforcement learning with expert demonstrations for urban autonomous driving. In *TIV*, pages 921–928, 2022. 2

[37] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *CVPR*, 2021. 2

[38] Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920*, 2022. 1, 2

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[40] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021. 2

[41] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *CVPR*, 2020. 2, 3

[42] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, pages 194–210, 2020. 1, 2

[43] Stefano Pini, Christian S Perone, Aayush Ahuja, Ana Sofia Rufino Ferreira, Moritz Niendorf, and Sergey Zagoruyko. Safe real-world autonomous driving by learning to predict and plan with a mixture of experts. In *ICRA*, pages 10069–10075, 2023. 2

[44] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *ICML*, pages 729–736, 2006. 6

[45] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 1, 2

[46] Oliver Scheel, Luca Bergamini, Maciej Wolczyk, Błażej Osiński, and Peter Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *CoRL*, pages 718–728. PMLR, 2022. 2

[47] Sathira Silva, Savindu Bhashitha Wannigama, Roshan Ragel, and Gihan Jayatilaka. S2tpvformer: Spatio-temporal tri-perspective view for temporally coherent 3d semantic occupancy prediction. *arXiv e-prints*, pages arXiv–2401, 2024. 1

[48] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024. 1, 4

[49] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *arXiv preprint arXiv:2304.14365*, 2023. 1, 2

[50] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *ICCV*, pages 8406–8415, 2023. 1, 2, 3, 6, 7

[51] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000. 2

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2

[53] Matt Vitelli, Yan Chang, Yawei Ye, Ana Ferreira, Maciej Wołczyk, Błażej Osiński, Moritz Niendorf, Hugo Grimmett, Qiangui Huang, Ashesh Jain, et al. Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies. In *ICRA*, pages 897–904, 2022. 2

[54] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *ICCV*, 2021. 6

[55] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. *arXiv preprint arXiv:2303.03991*, 2023. 1, 2

[56] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, pages 21729–21740, 2023. 1, 2, 3, 5, 6

[57] Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. *NeurIPS*, 33:2564–2575, 2020. 2

[58] Zixun Xie, Sicheng Zuo, Wenzhao Zheng, Yunpeng Zhang, Dalong Du, Jie Zhou, Jiwen Lu, and Shanghang Zhang. Gpd-1: Generative pre-training for driving. *arXiv preprint arXiv:2412.08643*, 2024. 2

[59] Tengju Ye, Wei Jing, Chunyong Hu, Shikun Huang, Lingping Gao, Fangzhen Li, Jingke Wang, Ke Guo, Wencong

Xiao, Weibo Mao, et al. Fusionad: Multi-modality fusion for prediction and planning tasks of autonomous driving. *arXiv preprint arXiv:2308.01006*, 2023. 1, 2, 3

[60] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *CVPR*, 2019. 6

[61] Diankun Zhang, Guoan Wang, Runwen Zhu, Jianbo Zhao, Xiwu Chen, Siyu Zhang, Jiahao Gong, Qibin Zhou, Wenyuan Zhang, Ningzi Wang, et al. Sparsead: Sparse query-centric paradigm for efficient end-to-end autonomous driving. *arXiv preprint arXiv:2404.06892*, 2024. 1, 3, 4

[62] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. 1, 2, 3, 5

[63] 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. 1, 4, 5, 6, 7

[64] Wenzhao Zheng, Ruiqi Song, Xianda Guo, and Long Chen. Genad: Generative end-to-end autonomous driving. In *ECCV*, 2024. 1, 2, 3, 4, 5, 6

[65] Jinyun Zhou, Rui Wang, Xu Liu, Yifei Jiang, Shu Jiang, Jiaming Tao, Jinghao Miao, and Shiyu Song. Exploring imitation learning for autonomous driving with feedback synthesizer and differentiable rasterization. In *IROS*, pages 1450–1457, 2021. 2

[66] Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2308.16896*, 2023. 1, 2