

VR-Drive: Viewpoint-Robust End-to-End Driving with Feed-Forward 3D Gaussian Splatting

Hoonhee Cho^{1*} Jae-Young Kang^{1*} Giwon Lee^{1*} Hyemin Yang^{1*}
Heejun Park¹ Seokwoo Jung² Kuk-Jin Yoon¹

¹ KAIST ² 42dot

<https://vrdriveneurips.github.io/>

Abstract

End-to-end autonomous driving (E2E-AD) has emerged as a promising paradigm that unifies perception, prediction, and planning into a holistic, data-driven framework. However, achieving robustness to varying camera viewpoints, a common real-world challenge due to diverse vehicle configurations, remains an open problem. In this work, we propose VR-Drive, a novel E2E-AD framework that addresses viewpoint generalization by jointly learning 3D scene reconstruction as an auxiliary task to enable planning-aware view synthesis. Unlike prior scene-specific synthesis approaches, VR-Drive adopts a feed-forward inference strategy that supports online training-time augmentation from sparse views without additional annotations. To further improve viewpoint consistency, we introduce a viewpoint-mixed memory bank that facilitates temporal interaction across multiple viewpoints and a viewpoint-consistent distillation strategy that transfers knowledge from original to synthesized views. Trained in a fully end-to-end manner, VR-Drive effectively mitigates synthesis-induced noise and improves planning under viewpoint shifts. In addition, we release a new benchmark dataset to evaluate E2E-AD performance under novel camera viewpoints, enabling comprehensive analysis. Our results demonstrate that VR-Drive is a scalable and robust solution for the real-world deployment of end-to-end autonomous driving systems.

1 Introduction

The end-to-end autonomous driving (E2E-AD) system refers to the integration of all modules, including perception, prediction, and planning nodes. The end-to-end driving paradigm [1–10] has consistently gained attention as a holistic approach, wherein the perception and prediction tasks are effectively integrated to support planning. This integration enhances both performance and efficiency, favoring a unified model for the entire driving task. This data-driven approach, compared to traditional rule-based planning, is designed to function robustly in complex scenarios by integrating various perception tasks (*e.g.*, detection, tracking, mapping, *etc.*). During the training process, it incorporates vast amounts of data and annotations to enhance its capabilities.

Despite significant advancements and strong performance across various scenarios, existing end-to-end autonomous driving (E2E-AD) must evolve into scalable and flexible holistic models to become viable industry solutions. Recent E2E-AD systems [11–18], in particular, aim to achieve comparable performance using only raw camera input. However, the viewpoint of the camera [19, 20] can vary depending on the vehicle’s type and make, and systems that can effectively adapt to these changes are crucial from a real-world application perspective. A straightforward solution to this challenge would be to collect data using a variety of vehicles and camera rigs, and then use this data during

*Equal contribution

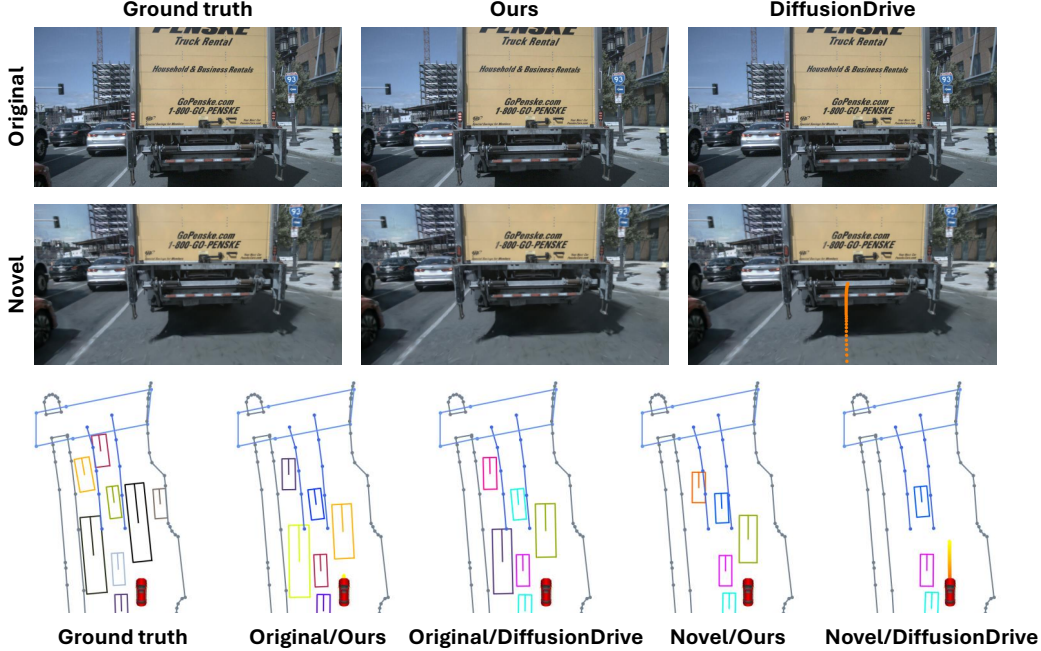


Figure 1: **Example scenario where surrounding vehicles have stopped at a traffic signal.** In the original training view, both our VR-Drive and DiffusionDrive [21] perform well in perceiving nearby vehicles and planning. However, with a lowered camera height, DiffusionDrive fails to detect surrounding vehicles, leading to a trajectory that collides with the front vehicle, posing a safety risk. In contrast, VR-Drive maintains accurate perception (except for those occluded due to the lowered camera height) and plans trajectories as effectively as in the original view.

the training process. However, this approach is impractical because it is impossible to pre-build camera viewpoints for every type of vehicle. Additionally, E2E-AD networks require annotations for various tasks, which incur significant costs, making it an impractical direction. Furthermore, to be deployable across different types of vehicles, the model must be flexible and robust not only to the predefined data but also to out-of-distribution (OOD) data. Therefore, the network must also ensure its generalization ability during the training process.

To this end, we tackle the critical real-world challenge of generalization to diverse camera rigs in end-to-end autonomous driving (E2E-AD) systems. Specifically, we propose VR-Drive, which jointly learns 3D scene reconstruction as an auxiliary modular task within E2E-AD to augment the diversity of camera viewpoints. While numerous prior works [22–25] have explored novel view synthesis through 3D reconstruction, these methods are typically scene-optimized and require significant computational resources, making them unsuitable for real-time downstream tasks. Therefore, we advocate for an online scene reconstruction approach that operates effectively with sparse views. To this end, we adopt a feed-forward inference strategy [26–29] to ensure efficiency. Rather than training a separate novel view synthesis model, we integrate it as a joint modular task within the end-to-end framework, thereby reducing training complexity. Moreover, to prevent errors in view synthesis from propagating and degrading the final planning performance, VR-Drive introduces a unified framework that incorporates 3D reconstruction as an auxiliary task within E2E-AD, enabling novel view synthesis without requiring additional annotations. To learn a viewpoint-robust and consistent feature space, VR-Drive utilizes a viewpoint-mixed memory bank that encourages interaction between features from different viewpoints in the sequential training process by allowing them to mix in 3D space. Additionally, to mitigate the potential noise embedded in the features extracted from viewpoint-augmented images, we propose a distillation strategy that transfers knowledge from the original view features to guide the learning of these synthesized features. Benefiting from its end-to-end joint training, this planning-aware synthesis strategy ensures that the model remains effective under viewpoint shifts and contributes to improved downstream planning. As shown in Fig. 1, VR-Drive maintains robust performance under varying camera viewpoints, unlike existing E2E-AD methods

that are sensitive to such changes, demonstrating its potential as a scalable and reliable end-to-end autonomous driving solution for real-world deployment.

The main contributions and unique aspects of our work are summarized as follows:

- We tackle viewpoint robustness in end-to-end autonomous driving (E2E-AD) by jointly learning 3D reconstruction for planning-aware view synthesis, enabling training data augmentation across diverse viewpoints and improving generalization to unseen camera configurations.
- We propose a viewpoint-mixed memory bank that enables temporal interaction between features from different viewpoints, and introduce a viewpoint-consistent distillation strategy that transfers knowledge from original viewpoint images to their corresponding augmented novel view synthesis images in a 3D space.
- We introduce a new benchmark dataset for E2E-AD to evaluate robustness under novel camera viewpoints unseen during training.

2 Related Works

2.1 End-to-End Autonomous Driving

End-to-end autonomous driving (E2E-AD) aims to generate final driving plans directly from raw sensor inputs within an integrated framework, in contrast to conventional methods that separately train perception, prediction, and planning modules. Previous E2E-AD works can be largely categorized into two major directions: (1) focusing on architecture and task exploration, and (2) leveraging high-level information distillation. Architecture-based approaches, such as [30–32], demonstrate that submodules within an integrated framework can be optimized to enhance the final planning performance. The following works [33, 34] further improved planning accuracy by removing certain auxiliary tasks, such as occupancy prediction and motion prediction. In contrast, [35] reorganized traditionally sequential auxiliary tasks into a parallel structure, while [34] proposed a task-aware training strategy to better leverage task relationships in parallel settings.

Architecture-based methods rely on large-scale annotated data, but often struggle in diverse scenarios due to biased training distributions, leading to issues such as causal confusion and long-tail errors. To address this, several studies have explored distilling actions and feature information from rule-based or reinforcement learning (RL)-based experts trained in privileged settings [36–39]. Additionally, there has been research on utilizing language models for scene representation, prediction, and planning, enhancing situational understanding and adaptability through the general knowledge embedded in large-scale foundation models [40–45].

Despite various research directions in E2E-AD, no prior work has addressed the development of model architectures that are robust to novel sensor viewpoints. This challenge is particularly critical, as sensor viewpoint variation is an inevitable and realistic factor in real-world deployments, arising from differences in vehicle types, sensor configurations, and mounting positions. However, it remains difficult to address within existing E2E-AD architectures and training paradigms, which are heavily dependent on the sensor inputs seen during training. In this work, we take the first step toward overcoming this limitation by proposing a method that enhances robustness to unseen sensor views.

2.2 Viewpoint-Robust Representations and Scene Reconstruction

Early studies [46–49] have shown that neural networks are vulnerable to viewpoint changes, especially under distribution shifts. While these studies explored adversarial viewpoints in 2D perception, more recent efforts [19, 50, 51] have extended this line of research to address viewpoint robustness in 3D perception tasks. They typically leverage novel view synthesis to generate images under varying camera viewpoints, aiming to train perception algorithms that are robust across diverse views. Research on novel view synthesis via 3D scene reconstruction [52, 24, 53–55] has advanced significantly, particularly with the emergence of Neural Radiance Fields (NeRF) [23] and 3D Gaussian Splatting (3DGS) [56]. However, most methods are scene-specific and require long training times, as they rely on scene-by-scene optimization.

To be applicable to scalable E2E-AD, a view augmentation strategy must satisfy two key requirements. (1) Since the test-time camera viewpoint is not fixed and can vary widely, the model must be robust to arbitrary views. This requires synthesizing diverse novel views during training, which in turn demands

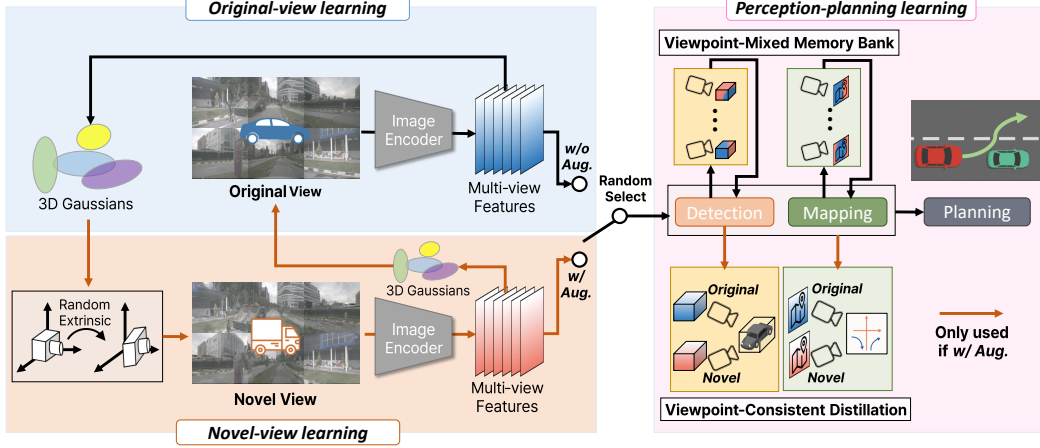


Figure 2: Overall framework of VR-Drive. Our overall framework consists of three main components, as follows: (1) original-view learning, (2) novel-view learning, and (3) perception-planning learning. For novel-view learning, the perception-planning head is randomly assigned to either the original or a novel view during training, allowing the model to generalize across different viewpoints.

real-time online processing for both training and inference. (2) To be effective in driving scenes, the method must support 3D reconstruction even with sparse or low-overlap observations. To meet these requirements, we adopt a feed-forward 3D gaussian splatting [57, 58, 29, 28, 26] that is both generalizable and capable of online training and inference. By incorporating 3D scene reconstruction as a sub-task within E2E-AD, we enhance scene-level understanding and achieve performance gains even for the original viewpoints. Furthermore, by jointly training the view synthesis and driving tasks in an end-to-end manner, we account for potential synthesis errors and demonstrate the feasibility of extending novel view synthesis as a practical means to improve viewpoint robustness in E2E-AD.

3 Methods

3.1 Overall Framework

Given multi-view images, end-to-end autonomous driving (E2E-AD) models jointly learn perception and motion prediction to produce accurate motion plans for the ego vehicle. In addition to the standard pipeline of existing E2E-AD approaches, the proposed VR-Drive incorporates scene reconstruction as an auxiliary task, leveraging 3D Gaussian Splatting (3DGS) [56]. The overall framework of VR-Drive is shown in Fig. 2. VR-Drive comprises three components, each targeting a distinct objective: (1) original-view learning, (2) novel-view learning, and (3) perception-planning learning.

Original-view learning: During training, we use the original view as the default input of the pipeline. Given multi-view images, the image encoder (ResNet50 [59]) first extracts original multi-view feature maps, $I \in \mathbb{R}^{N \times C \times H \times W}$, where N is the number of camera views. These generated feature maps are utilized not only for perception and planning in autonomous driving, but also for learning and rendering novel views via 3DGS. We build on the original 3DGS framework [56], which represents a scene using Gaussian primitives $g = (\mu, \Sigma, \alpha, c)$, defined by position μ , covariance Σ , opacity α , and spherical harmonics for color c . The covariance matrix Σ is constructed by combining the scaling factor s and rotation quaternion r . Unlike the original 3DGS that relies on structure-from-motion for optimizing μ , we predict primitives in a feed-forward [26], pixel-wise manner directly from input images. Similar to previous work [60] that treated depth estimation as an auxiliary task within E2E-AD, we jointly learn depth as part of the E2E-AD framework. The estimated depth D is then used to infer the position of Gaussian primitives $\mu \in \mathbb{R}^3$. We use the predicted depth map D and the image feature map I , as input to a Gaussian network composed of multiple convolutional layers. This network predicts the remaining parameters of each Gaussian primitive, including the scaling factor $s \in \mathbb{R}_+^3$, rotation quaternion $r \in \mathbb{R}^4$, opacity $\alpha \in [0, 1]$, and color $c \in \mathbb{R}^k$ represented by k -degree spherical harmonics. To ensure valid ranges, we apply softplus to H_s and softmax to H_α ,

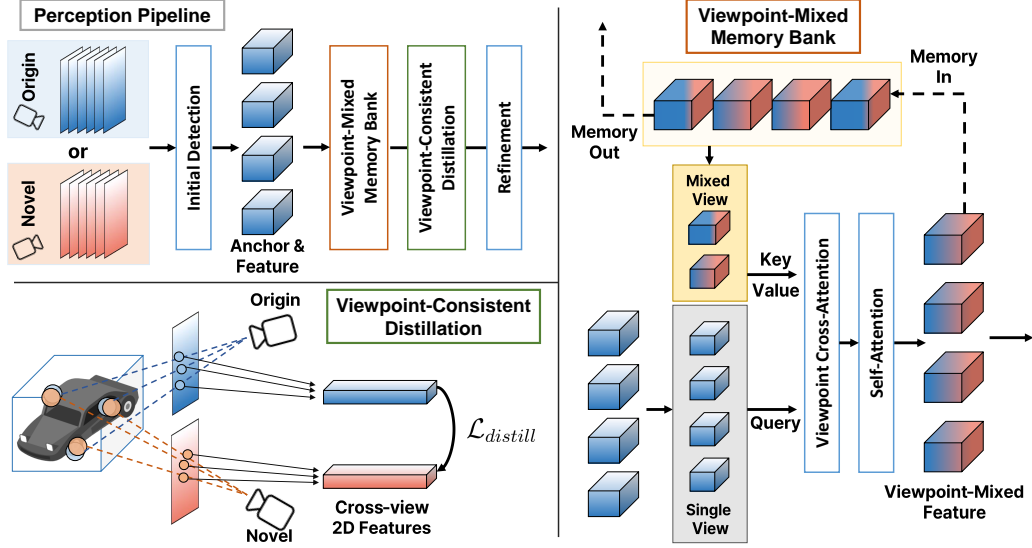


Figure 3: Illustration of the perception pipeline. VR-Drive includes two complementary techniques to ensure consistent feature representations across camera viewpoints: Viewpoint-Mixed Memory Bank and Viewpoint-Consistent Distillation.

enforcing $s \in \mathbb{R}_+^3$ and $\alpha \in [0, 1]$. The feed-forward design enables online inference on novel views and generalization to new inputs without scene-specific constraints.

Novel-view learning: VR-Drive aims to achieve robust planning performance by generating consistent feature representations even for camera viewpoints that were not observed during training. Specifically, at test time, it seeks to replicate the feature space of the original view across diverse, unseen viewpoints. To this end, we randomly sample camera extrinsics and render multi-view feature maps from arbitrary perspectives using the Gaussian primitives generated from the original view. Given the rendered multi-view images from a novel view, we generate novel view features, $\tilde{I} \in \mathbb{R}^{N \times C \times H \times W}$, using a shared image encoder with the original view. Since the novel view features may differ in distribution from the original, we guide the model to generate feature representations that closely align with those of the original view. We observe that feed-forward 3DGS facilitates scene-level 3D understanding, which proves beneficial even under novel viewpoints. To encourage robustness, we additionally employ a cyclic reconstruction loss that trains the model to regenerate the original view from a novel one.

Perception-planning learning: VR-Drive selectively trains on original and novel views during the training to achieve robustness across diverse camera viewpoints. The image features extracted from the selected view are passed to the perception and planning heads, enabling planning based on the corresponding perception representation. Following [21, 61, 33], we adopt 3D object detection and mapping as our perception tasks. More specifically, to achieve efficient representation, we utilize a same sparse architecture that leverages anchor- and instance feature-based designs [62, 63] for both detection and mapping tasks. Since the two tasks differ only in the dimensionality of the anchors, we provide all descriptions and definitions in the context of detection, which are equally applicable to mapping. We first generate initial bounding box proposals using the detection module [62], denoted as $B = \{B^1, B^2, \dots, B^M\} \in \mathbb{R}^{M \times N_B}$, where M is the number of anchors and N_B is the dimensionality of each anchor. For each proposal, we also extract the corresponding instance features $F = \{F^1, F^2, \dots, F^M\} \in \mathbb{R}^{M \times N_i}$, where N_i is the dimension of the instance feature. This allows us to encode the surrounding agents in the 3D space based on the extracted image features. As illustrated in Fig. 3, we insert viewpoint-robust modules into the perception pipeline for detection and mapping, in addition to the conventional detection components. Specifically, we introduce two dedicated components within the perception stage of VR-Drive: the **Viewpoint-Mixed Memory Bank** and the **Viewpoint-Consistent Distillation** strategy, designed to address feature variations across viewpoints and promote canonical feature learning. We obtain the final perception results by refining the viewpoint-robust features through an additional detection decoder. Finally, to enable planning that interacts with the predicted agents, we adopt the motion planner proposed in [21].

3.2 Viewpoint-Mixed Memory Bank

As mentioned in Sec. 3.1, the perception and planning pipeline randomly receives features from either the original view or a novel view during training. To promote canonical 3D feature learning from image inputs across diverse viewpoints with different distributions, we encourage interaction between 3D features extracted from diverse view during training. Rather than simply using a single pair of original and novel views, limiting the model to observing only two viewpoints within a single forward pass, we adopt a memory bank strategy that stores and updates features from continuously changing novel views to promote broader viewpoint generalization. Let $F' \in \mathbb{R}^{M' \times N_i}$ be the instance features retrieved from the viewpoint-mixed memory bank, where M' is the number of sampled features. Following the method proposed in [62], we align F' to the current frame by leveraging the velocities of the anchor box and the status of the ego vehicle to compensate for temporal shifts between viewpoints. Our objective is to generate interactive features between F' and the instance features from the current view F . To achieve this, we leverage attention mechanisms [64] to fuse features from the memory bank and the current view, resulting in the following mixed feature representation:

$$\mathbf{F} = \text{Cross-Attention}(\text{Query} = F, \text{Key} = F', \text{Value} = F'). \quad (1)$$

The mixed feature, \mathbf{F} , is further processed through a self-attention mechanism to model interactions among agents, and are then passed to the viewpoint-consistent distillation module. The viewpoint-mixed memory bank is updated by selecting the top- K high-confidence instances after the final refinement, while the oldest instances in the bank are discarded in an FIFO manner.

3.3 Viewpoint-Consistent Distillation

One potential challenge in learning viewpoint robustness through novel view synthesis is that the synthesized images may contain rendering artifacts, especially in occluded or texture-less regions. Moreover, novel view settings often involve more extreme or side-facing camera angles, which can be more challenging for autonomous driving due to reduced visibility or increased uncertainty in object localization. To address this, we adopt a distillation strategy in which the original view, typically containing more reliable and informative features due to better visibility and camera positioning, guides the learning of novel views. One simple strategy is to force alignment between two view features by projecting one onto the other using depth and pose. However, such alignment often excludes regions that are perceptually important for downstream tasks. Instead, we utilize the instance features \mathbf{F} and their corresponding anchor boxes B to selectively distill information that is crucial from a planning perspective. Motivated by [65], we aim to extract representative object features by computing a learnable offset \mathbf{p} and weight \mathbf{w} for each instance i based on its instance feature \mathbf{F}_i , defined as $\mathbf{p}_i = f(\mathbf{F}_i) \in \mathbb{R}^{s \times 3}$ and $\mathbf{w}_i = g(\mathbf{F}_i) \in \mathbb{R}^{N \times s}$, where f and g are a learnable keypoint and weight generations. Here, s and N denote the number of sampled points and cameras, respectively. Then, we compute the j -th 3D sampled point as follows:

$$\mathbf{p}_{i,j}^* = \mathbf{p}_{i,j} + \text{position}(B_i), \quad (2)$$

where $\text{position}(B_i)$ denotes the 3D center coordinates (x, y, z) of the bounding box B_i . We project the sampled 3D points onto the image plane of each camera view using the corresponding transformation matrix, and extract image features at the n -th camera view via bilinear sampling, defined as:

$$\mathbf{f}_{n,i,j} = \text{BilinearSample}(I_n, \Pi_n \mathbf{p}_{i,j}^*) \in \mathbb{R}^C \quad (3)$$

where Π_n is the camera transformation matrix and I_n is the original view 2D image feature map from the n -th camera. Then, we define the aggregated feature at anchor index i as:

$$S_i = \sum_n^N \sum_j^s \mathbf{w}_{n,i,j} \cdot \mathbf{f}_{n,i,j}. \quad (4)$$

The same procedure is applied to the novel view image feature map \tilde{I} , resulting in \tilde{S} . To align the sampled features \tilde{S}_i from the novel view with the corresponding features S_i from the original view, we apply a mean squared error (MSE) loss between features. We restrict the loss of distillation to high-confidence anchors to avoid distillation in the background or noisy boxes. Let \mathcal{I}^* be the set of anchors whose confidence scores exceed a predefined threshold τ . The viewpoint-consistent distillation loss is defined as:

$$\mathcal{L}_{\text{distill}} = \frac{1}{|\mathcal{I}^*|} \sum_{i \in \mathcal{I}^*} \left\| \tilde{S}_i - \text{stopgrad}(S_i) \right\|_2^2, \quad (5)$$

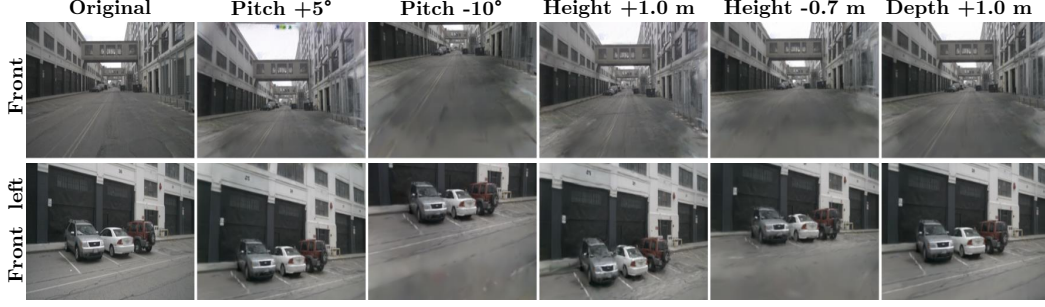


Figure 4: Variant camera viewpoints at test time, differing from the original training distribution.

where $\text{stopgrad}(\cdot)$ indicates gradient detachment.

Note that the viewpoint-mixed memory bank is always used, whereas the viewpoint-consistent distillation is only applied when a novel view image is used as the input for perception and planning.

3.4 Loss Functions

The loss functions consist of various tasks. For motion prediction and planning, we apply the winner-takes-all strategy [66]. In the planning task, an extra regression loss is introduced to handle ego status. For classification, we utilize focal loss [67], while L1 loss is used for regression in both detection and mapping tasks. Furthermore, L1 loss is also employed for depth estimation. Additionally, we incorporate the viewpoint-consistent distillation loss. We also use a rendering loss for scene reconstruction, as described below.

Rendering Loss. We use both L2 and LPIPS [68] losses as the rendering objective. Since ground truth for various viewpoints is unavailable during training, we apply rendering loss through two alternative strategies, depending on whether novel view augmentation is used.

- **Original Reconstruction Loss.** The reconstruction loss encourages the model to render novel views from input images using Gaussian primitives. As real data lacks paired novel views, we simulate them by synthesizing adjacent-time views via splat-based rendering and apply the loss to the generated outputs.
- **Cyclic Reconstruction Loss.** When a novel view is given as input for perception-planning heads, supervision using adjacent time-step images, as done with the original view, is not feasible due to the absence of paired frames. To support effective 3D scene learning with Gaussian primitives and depth, we adopt a cyclic rendering strategy that reconstructs the original view from the novel view.

The overall loss function for end-to-end training is:

$$\mathcal{L} = \mathcal{L}_{det} + \mathcal{L}_{map} + \mathcal{L}_{depth} + \mathcal{L}_{motion} + \mathcal{L}_{plan} + \mathcal{L}_{render}. \quad (6)$$

4 End-to-End Autonomous Driving Benchmark with Viewpoint Variations

4.1 Training and Evaluation Setup

Our work pioneers research on camera viewpoint variations in end-to-end autonomous driving (E2E-AD) and aims to establish a framework for training and evaluation in future studies. Considering the challenges of acquiring data with varying rigs during vehicle operation in real-world applications, we fix the rig to a single setup during the training process. Furthermore, our goal is to evaluate the model’s robustness across various out-of-distribution data and assess its performance under different camera settings with distinct distributions. To achieve this, we introduce sensor variations at test time, deviating from the original camera configuration used during training, including: +5° pitch, -10° pitch, +1.0m height, -0.7m height, and +1.0m depth. These variations are configured based on the sensor settings from [19] to evaluate robustness.

Table 1: Open-loop planning performance in nuScenes dataset. Metric calculation follows ST-P3 [69]. The best performance in each setting is highlighted in **bold**. * denotes the usage of ego-status.

| Camera Setting | | Methods | L2 (m) ↓ | | | | Collision Rate (%) ↓ | | | |
|----------------|---------------|---------------------|-------------|-------------|-------------|-------------|----------------------|-------------|-------------|-------------|
| | | | 1s | 2s | 3s | Avg. | 1s | 2s | 3s | Avg. |
| Original | | AD-MLP* [70] | 0.20 | 0.26 | 0.41 | 0.29 | 0.17 | 0.18 | 0.24 | 0.19 |
| | | BEV-Planner* [34] | 0.28 | 0.52 | 0.84 | 0.55 | 0.13 | 0.17 | 0.36 | 0.22 |
| | | VAD [33] | 0.41 | 0.70 | 1.05 | 0.72 | 0.07 | 0.17 | 0.41 | 0.22 |
| | | SparseDrive [60] | 0.29 | 0.58 | 0.96 | 0.61 | 0.01 | 0.05 | 0.18 | 0.08 |
| | | DiffusionDrive [21] | 0.27 | 0.54 | 0.90 | 0.57 | 0.03 | 0.05 | 0.16 | 0.08 |
| | | VR-Drive (Ours) | 0.29 | 0.57 | 0.95 | 0.60 | 0.01 | 0.03 | 0.14 | 0.06 |
| Unseen | pitch +5° | AD-MLP* [70] | 0.20 | 0.26 | 0.41 | 0.29 | 0.17 | 0.18 | 0.24 | 0.19 |
| | | BEV-Planner* [34] | 0.29 | 0.56 | 0.91 | 0.59 | 0.27 | 0.31 | 0.54 | 0.37 |
| | | VAD [33] | 0.38 | 0.66 | 1.00 | 0.68 | 0.11 | 0.21 | 0.51 | 0.28 |
| | | SparseDrive [60] | 0.32 | 0.63 | 1.03 | 0.66 | 0.02 | 0.08 | 0.35 | 0.15 |
| | | DiffusionDrive [21] | 0.33 | 0.64 | 1.04 | 0.67 | 0.00 | 0.09 | 0.24 | 0.11 |
| | | VR-Drive (Ours) | 0.29 | 0.57 | 0.94 | 0.60 | 0.00 | 0.02 | 0.14 | 0.06 |
| | pitch -10° | AD-MLP* [70] | 0.20 | 0.26 | 0.41 | 0.29 | 0.17 | 0.18 | 0.24 | 0.19 |
| | | BEV-Planner* [34] | 0.27 | 0.51 | 0.86 | 0.54 | 0.64 | 0.73 | 0.93 | 0.76 |
| | | VAD [33] | 0.70 | 1.01 | 1.35 | 1.02 | 0.55 | 0.82 | 1.27 | 0.88 |
| | | SparseDrive [60] | 0.46 | 0.91 | 1.50 | 0.96 | 0.03 | 0.15 | 0.50 | 0.23 |
| | | DiffusionDrive [21] | 0.45 | 0.91 | 1.52 | 0.96 | 0.02 | 0.16 | 0.55 | 0.24 |
| | | VR-Drive (Ours) | 0.34 | 0.66 | 1.10 | 0.70 | 0.02 | 0.08 | 0.24 | 0.11 |
| | height +1.0 m | AD-MLP* [70] | 0.20 | 0.26 | 0.41 | 0.29 | 0.17 | 0.18 | 0.24 | 0.19 |
| | | BEV-Planner* [34] | 0.28 | 0.54 | 0.88 | 0.57 | 0.20 | 0.22 | 0.44 | 0.29 |
| | | VAD [33] | 0.41 | 0.70 | 1.07 | 0.73 | 0.14 | 0.45 | 0.80 | 0.47 |
| | | SparseDrive [60] | 0.42 | 0.83 | 1.36 | 0.87 | 0.10 | 0.45 | 1.08 | 0.54 |
| | | DiffusionDrive [21] | 0.81 | 1.44 | 2.14 | 1.46 | 0.17 | 0.78 | 1.47 | 0.81 |
| | | VR-Drive (Ours) | 0.34 | 0.66 | 1.07 | 0.69 | 0.00 | 0.05 | 0.28 | 0.11 |
| | height -0.7 m | AD-MLP* [70] | 0.20 | 0.26 | 0.41 | 0.29 | 0.17 | 0.18 | 0.24 | 0.19 |
| | | BEV-Planner* [34] | 0.29 | 0.55 | 0.89 | 0.58 | 0.49 | 0.61 | 0.82 | 0.64 |
| | | VAD [33] | 0.41 | 0.71 | 1.09 | 0.74 | 0.09 | 0.17 | 0.39 | 0.22 |
| | | SparseDrive [60] | 0.50 | 0.97 | 1.56 | 1.01 | 0.01 | 0.20 | 0.68 | 0.30 |
| | | DiffusionDrive [21] | 0.64 | 1.18 | 1.82 | 1.21 | 0.00 | 0.12 | 0.49 | 0.20 |
| | | VR-Drive (Ours) | 0.34 | 0.66 | 1.09 | 0.69 | 0.03 | 0.11 | 0.28 | 0.14 |
| | depth +1.0 m | AD-MLP* [70] | 0.20 | 0.26 | 0.41 | 0.29 | 0.17 | 0.18 | 0.24 | 0.19 |
| | | BEV-Planner* [34] | 0.29 | 0.55 | 0.89 | 0.58 | 0.17 | 0.23 | 0.43 | 0.28 |
| | | VAD [33] | 0.39 | 0.68 | 1.05 | 0.71 | 0.09 | 0.19 | 0.48 | 0.26 |
| | | SparseDrive [60] | 0.66 | 1.23 | 1.91 | 1.27 | 0.05 | 0.25 | 0.62 | 0.31 |
| | | DiffusionDrive [21] | 0.87 | 1.55 | 2.30 | 1.57 | 0.12 | 0.37 | 0.75 | 0.41 |
| | | VR-Drive (Ours) | 0.37 | 0.69 | 1.11 | 0.72 | 0.02 | 0.11 | 0.27 | 0.13 |
| | Average | AD-MLP* [70] | 0.20 | 0.26 | 0.41 | 0.29 | 0.17 | 0.18 | 0.24 | 0.19 |
| | | BEV-Planner* [34] | 0.28 | 0.54 | 0.88 | 0.57 | 0.36 | 0.42 | 0.63 | 0.47 |
| | | VAD [33] | 0.46 | 0.75 | 1.11 | 0.78 | 0.20 | 0.37 | 0.69 | 0.42 |
| | | SparseDrive [60] | 0.47 | 0.91 | 1.47 | 0.95 | 0.04 | 0.23 | 0.65 | 0.31 |
| | | DiffusionDrive [21] | 0.62 | 1.14 | 1.76 | 1.17 | 0.07 | 0.36 | 0.80 | 0.41 |
| | | VR-Drive (Ours) | 0.34 | 0.65 | 1.06 | 0.68 | 0.01 | 0.07 | 0.24 | 0.11 |

4.2 Dataset Generation Protocol

We use the nuScenes [71] benchmark, which is widely used in recent E2E-AD works [62, 61, 44]. However, since the nuScenes dataset does not provide images from variant camera viewpoints, we performed offline scene optimization as a method to obtain data from various viewpoints. We performed offline scene optimization [72], showcasing high-performance and strong geometric alignment, on nuScenes test sequences. This process enabled rendering various views for each sequence, as shown in Fig. 4. After manually inspecting the test sequences, we excluded 4 sequences from the original 150 due to unsatisfactory quality, leaving 146 test sequences for unseen viewpoints. Note that this offline scene optimization requires significant training time for each scene, making it impractical for datasets with a large number of sequences. On an NVIDIA TITAN RTX, each sequence took over 8 hours to train, and the total time of optimization and rendering took more than 3 weeks. This underscores the practicality of our online novel view synthesis approach for training.

5 Experiments

Experiment Setup. We evaluate the model using Average Displacement Error (ADE) and Collision Rate. For comparison, we use existing end-to-end models, including AD-MLP [70], BEV-

Table 2: **Ablation for design choices.** “SR” and “VMM” indicate scene reconstruction and viewpoint-mixed memory bank. “CR” and “VCD” indicate cyclic reconstruction loss and viewpoint-consistent distillation, respectively. \triangle indicates that scene reconstruction is learned jointly, but the generated novel view images are not used as perception and planning input.

| Modules | | | | | Seen | | | | | | | | Unseen Average | | | | | | | |
|---------|-------------|-----|----|-----|-------------|-------------|-------------|-------------|-----------------|-------------|-------------|-------------|----------------|-------------|-------------|-------------|-----------------|-------------|-------------|-------------|
| ID | SR | VMM | CR | VCD | L2 (m) ↓ | | | | Collision (%) ↓ | | | | L2 (m) ↓ | | | | Collision (%) ↓ | | | |
| | | | | | 1s | 2s | 3s | Avg. | 1s | 2s | 3s | Avg. | 1s | 2s | 3s | Avg. | 1s | 2s | 3s | Avg. |
| 1 | - | - | - | - | 0.31 | 0.60 | 0.98 | 0.63 | 0.13 | 0.10 | 0.19 | 0.14 | 0.47 | 0.88 | 1.38 | 0.91 | 0.17 | 0.25 | 0.48 | 0.30 |
| 2 | \triangle | - | - | - | 0.28 | 0.56 | 0.93 | 0.59 | 0.00 | 0.05 | 0.17 | 0.07 | 0.46 | 0.87 | 1.36 | 0.90 | 0.04 | 0.20 | 0.53 | 0.26 |
| 3 | ✓ | - | - | - | 0.31 | 0.60 | 0.97 | 0.63 | 0.03 | 0.08 | 0.20 | 0.10 | 0.40 | 0.76 | 1.20 | 0.79 | 0.04 | 0.16 | 0.36 | 0.19 |
| 4 | ✓ | ✓ | - | - | 0.31 | 0.59 | 0.95 | 0.62 | 0.02 | 0.06 | 0.19 | 0.09 | 0.37 | 0.70 | 1.12 | 0.73 | 0.03 | 0.13 | 0.36 | 0.17 |
| 5 | ✓ | ✓ | ✓ | - | 0.29 | 0.56 | 0.93 | 0.59 | 0.04 | 0.06 | 0.19 | 0.09 | 0.33 | 0.64 | 1.06 | 0.68 | 0.04 | 0.13 | 0.31 | 0.16 |
| 6 | ✓ | ✓ | - | ✓ | 0.31 | 0.59 | 0.94 | 0.61 | 0.02 | 0.05 | 0.16 | 0.08 | 0.37 | 0.70 | 1.14 | 0.73 | 0.02 | 0.09 | 0.31 | 0.14 |
| 7 | ✓ | ✓ | ✓ | ✓ | 0.29 | 0.57 | 0.95 | 0.60 | 0.01 | 0.03 | 0.14 | 0.06 | 0.34 | 0.65 | 1.06 | 0.68 | 0.01 | 0.07 | 0.24 | 0.11 |

Table 3: Analysis of the range of random extrinsics for novel views during the training process.

| Settings | | | | | Seen | | | | | | | | Unseen Average | | | | | | | |
|----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------------|-------------|-------------|-------------|----------------|-------------|-------------|-------------|-----------------|----|----|------|
| | 1s | 2s | 3s | Avg. | L2 (m) ↓ | | | | Collision (%) ↓ | | | | L2 (m) ↓ | | | | Collision (%) ↓ | | | |
| | | | | | 1s | 2s | 3s | Avg. | 1s | 2s | 3s | Avg. | 1s | 2s | 3s | Avg. | 1s | 2s | 3s | Avg. |
| - | 0.29 | 0.57 | 0.95 | 0.60 | 0.01 | 0.03 | 0.14 | 0.06 | 0.34 | 0.65 | 1.06 | 0.68 | 0.01 | 0.07 | 0.24 | 0.11 | | | | |
| Superset | 0.29 | 0.57 | 0.95 | 0.60 | 0.00 | 0.03 | 0.15 | 0.06 | 0.33 | 0.65 | 1.06 | 0.68 | 0.03 | 0.09 | 0.25 | 0.12 | | | | |
| Subset | 0.30 | 0.60 | 0.99 | 0.63 | 0.00 | 0.05 | 0.16 | 0.07 | 0.41 | 0.79 | 1.27 | 0.82 | 0.02 | 0.09 | 0.27 | 0.13 | | | | |

Planner [34], VAD [33], SparseDrive [60], and DiffusionDrive [21]. During training, we rendered random novel view images with pitch in the range $[-10^\circ, 5^\circ]$, height in $[-0.7\text{m}, 1.0\text{m}]$, and depth in $[-0.2\text{m}, 1.0\text{m}]$, which broadly covers the test configurations. Additional implementation details will be described in the supplementary material.

Experimental Results. Table 1 shows the performance of E2E-AD models on both original and novel views, where “unseen” refers to data that was not provided during training. When focusing on the performance in both the original and unseen domains, we begin by comparing the performance of our proposed VR-Drive with DiffusionDrive as an example. On the original domain, both models show similar performance. However, when evaluated on the unseen domain, DiffusionDrive experiences a significant increase in both ADE and collision rate. In contrast, our method demonstrates performance comparable to the original view, even in more challenging camera viewpoints under previously unseen distributions.

6 Ablation Study

Effect of the components. We conducted an ablation study on each module, as shown in Table 2. Notably, comparing ID-1 and ID-2 reveals that simply enabling joint learning of scene reconstruction already improves performance on both original viewpoints. This suggests that online joint optimization with 3DGS contributes to improving the scalability of E2E-AD systems, likely by encouraging a more precise comprehension of 3D geometry. Such enhanced geometric understanding facilitates more informed and reliable planning decisions. The most significant performance gain emerges at ID-3, where the novel view generated via scene reconstruction is used as an additional input to the model. Beyond this, the proposed modules further contribute to performance improvements. Interestingly, our method does not suffer from a trade-off where improved performance on novel views comes at the cost of degraded performance on original views. Instead, the proposed components enhance the model’s overall capability, even in the original views. This suggests that novel views serve as an effective form of augmentation during training, and the introduced modules help guide the model to learn better representations, ultimately benefiting both original and novel view settings.

Range of random extrinsics. We study the distribution shift between training and testing in terms of camera viewpoint diversity, as summarized in Table 3. For the experiments in Table 1, we set the training-time random extrinsic ranges to pitch $\in [-10^\circ, 5^\circ]$, height $\in [-0.7\text{m}, 1.0\text{m}]$, and depth $\in [-0.2\text{m}, 1.0\text{m}]$. To examine generalization beyond the test distribution, the “Superset” setting expands the training sensor range to pitch $\in [-15^\circ, 10^\circ]$, height $\in [-1.0\text{m}, 1.5\text{m}]$, and depth $\in [-0.5\text{m}, 1.5\text{m}]$, covering viewpoints that go beyond the test distribution. This allows us to investigate whether the model remains robust when trained with a broader range of viewpoints.

Conversely, the “Subset” setting limits the sensor range to pitch $\in [-5^\circ, 2^\circ]$, height $\in [-0.3\text{m}, 0.5\text{m}]$, and depth $\in [-0.1\text{m}, 0.5\text{m}]$, ensuring that the training views do not overlap with any of the test-time configurations. Our model performs consistently across the Superset, Subset, and original settings, demonstrating robustness to continuous viewpoint variation.

7 Closed-loop Evaluation on the CARLA dataset

Table 4: Closed-loop test on CARLA dataset.

| Methods | Original | | Unseen | | | | | | | | | | | |
|------------------------|----------|--------|-----------|--------|------------|--------|---------------|--------|---------------|--------|--------------|--------|---------|--------|
| | | | pitch +5° | | pitch -10° | | height +1.0 m | | height -0.7 m | | depth +1.0 m | | Average | |
| | DS | RC | DS | RC | DS | RC | DS | RC | DS | RC | DS | RC | DS | RC |
| Town05-Nov | | | | | | | | | | | | | | |
| ST-P3 [69] | 44.24 | 100.00 | 41.00 | 100.00 | 23.85 | 100.00 | 25.83 | 100.00 | 28.60 | 100.00 | 32.06 | 100.00 | 30.27 | 100.00 |
| TCP [36] | 92.73 | 92.73 | 70.33 | 80.33 | 4.65 | 4.65 | 88.51 | 88.51 | 0.00 | 0.00 | 91.11 | 91.11 | 50.92 | 52.92 |
| AD-MLP [70] | 13.59 | 32.83 | 13.59 | 32.83 | 13.59 | 32.83 | 13.59 | 32.83 | 13.59 | 32.83 | 13.59 | 32.83 | 13.59 | 32.83 |
| BEV-Planner [34] | 17.25 | 28.70 | 7.30 | 28.89 | 7.74 | 28.83 | 8.51 | 28.95 | 7.69 | 28.70 | 7.75 | 28.95 | 7.80 | 28.86 |
| Baseline | 76.47 | 99.20 | 69.41 | 89.60 | 45.65 | 99.38 | 48.67 | 100.00 | 41.59 | 86.76 | 35.95 | 98.60 | 48.25 | 94.87 |
| VR-Drive (Ours) | 84.04 | 99.04 | 75.00 | 100.00 | 91.26 | 98.76 | 98.44 | 98.99 | 80.67 | 97.32 | 95.88 | 96.35 | 88.25 | 98.28 |

We use the CARLA 0.9.10.1 simulator [73] for closed-loop testing. For the closed-loop test, we evaluate performance using the Town05short benchmark. We collect the training data from Town01, 02, 03, 04, 06, 07, and 10, using scenario routes based on previous work [74]. For the evaluation, we assessed each model’s performance based on two key metrics: Driving Score (DS) and Route Completion (RC). To provide a comprehensive comparison, we included several established end-to-end autonomous driving models. Specifically, we evaluated ST-P3[69], TCP[36], AD-MLP [70], BEV-Planner [34], and baseline alongside our proposed method. As the baseline, we adopt the ID-1 setting from Table 2, removing all proposed modules.

Following existing works [9, 69, 33, 74], we adopt the Town05 benchmark for simulation. However, to enable training and evaluation on novel viewpoints, we establish a new benchmark. Specifically, we sample 20% of sequences from Town05 Short to construct Town05-Nov, which serves as our novel-view evaluation set. For training data, we follow Transfuser [74] and collect samples using the autopilot, but only from original viewpoints. For fair comparison with prior works, we handle baselines based on their available resources. In the case of ST-P3 [69] and TCP [36], since pretrained checkpoints on Town05 are publicly released, we directly evaluate these models without retraining.

Table 4 shows the closed-loop evaluation results on the Town05-Nov benchmark. Existing end-to-end autonomous driving approaches tend to struggle with planning in unseen test scenarios, sometimes failing to initiate driving in particularly challenging cases. Notably, the DS metric is more adversely affected compared to RC, experiencing degradation in perception performance when faced with novel viewpoint inputs. In contrast, our method demonstrates performance on unseen tests that is comparable to that of the original viewpoint.

8 Conclusion

In this work, we present VR-Drive, a unified end-to-end autonomous driving framework that leverages novel view synthesis and viewpoint-robust learning. To the best of our knowledge, we are the first to study camera viewpoint variation in E2E-AD for real-world applications. We benchmark VR-Drive on the nuScenes dataset and the CARLA simulator, achieving state-of-the-art performance across diverse camera viewpoints and out-of-distribution conditions.

Limitation and Future Work. The performance of VR-Drive is influenced by the accuracy of camera calibration. While errors in calibration may lead to suboptimal results, the system could be made more robust to such errors. Addressing this issue and enhancing the system’s robustness to calibration inaccuracies could be an important focus for future work.

Acknowledgement

We thank 42dot for funding this research. We also appreciate the support and valuable discussions from the members of the 42dot research team. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF2022R1A2B5B03002636).

References

- [1] Zesong Chen, Ze Yu, Jun Li, Linlin You, and Xiaojun Tan. Dualat: Dual attention transformer for end-to-end autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16353–16359. IEEE, 2024.
- [2] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700, 2018. doi: 10.1109/ICRA.2018.8460487.
- [3] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [4] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15793–15803, 2021.
- [5] Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13723–13733, 2023.
- [6] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [7] Yiqun Duan, Qiang Zhang, and Renjing Xu. Prompting multi-modal tokens to enhance end-to-end autonomous driving imitation learning with llms. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6798–6805. IEEE, 2024.
- [8] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8406–8415, 2023.
- [9] Kaituo Feng, Changsheng Li, Dongchun Ren, Ye Yuan, and Guoren Wang. On the road to portability: Compressing end-to-end motion planner for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15099–15108, 2024.
- [10] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European conference on computer vision*, pages 55–72. Springer, 2024.
- [11] Simon Doll, Niklas Hanselmann, Lukas Schneider, Richard Schulz, Marius Cordts, Markus Enzweiler, and Hendrik Lensch. Dualad: Disentangling the dynamic and static world for end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14728–14737, 2024.
- [12] Bozhou Zhang, Nan Song, Xin Jin, and Li Zhang. Bridging past and future: End-to-end autonomous driving with historical prediction and planning. *arXiv preprint arXiv:2503.14182*, 2025.
- [13] Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. *arXiv preprint arXiv:2503.05689*, 2025.

- [14] Peidong Li and Dixiao Cui. Navigation-guided sparse scene representation for end-to-end autonomous driving. In *The Thirteenth International Conference on Learning Representations*.
- [15] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 87–104. Springer, 2024.
- [16] Ziyang Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don’t shake the wheel: Momentum-aware planning in end-to-end autonomous driving. *arXiv preprint arXiv:2503.03125*, 2025.
- [17] Zhili Chen, Maosheng Ye, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Ppad: Iterative interactions of prediction and planning for end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 239–256. Springer, 2024.
- [18] Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. *arXiv preprint arXiv:2406.08481*, 2024.
- [19] Tzofi Klinghoffer, Jonah Philion, Wenzheng Chen, Or Litany, Zan Gojcic, Jungseock Joo, Ramesh Raskar, Sanja Fidler, and Jose M Alvarez. Towards viewpoint robustness in bird’s eye view segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8515–8524, 2023.
- [20] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020.
- [21] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. *arXiv preprint arXiv:2411.15139*, 2024.
- [22] Nan Huang, Xiaobao Wei, Wenzhao Zheng, Pengju An, Ming Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. S3gaussian: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2405.20323*, 2024.
- [23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [24] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21634–21643, 2024.
- [25] Mustafa Khan, Hamidreza Fazlali, Dhruv Sharma, Tongtong Cao, Dongfeng Bai, Yuan Ren, and Bingbing Liu. Autosplat: Constrained gaussian splatting for autonomous driving scene reconstruction. *arXiv preprint arXiv:2407.02598*, 2024.
- [26] Qijian Tian, Xin Tan, Yuan Xie, and Lizhuang Ma. Drivingforward: Feed-forward 3d gaussian splatting for driving scene reconstruction from flexible surround-view input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7374–7382, 2025.
- [27] Letian Wang, Seung Wook Kim, Jiawei Yang, Cunjun Yu, Boris Ivanovic, Steven Waslander, Yue Wang, Sanja Fidler, Marco Pavone, and Peter Karkus. Distillnerf: Perceiving 3d scenes from single-glance images by distilling neural fields and foundation model features. *Advances in Neural Information Processing Systems*, 37:62334–62361, 2024.
- [28] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024.

- [29] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19457–19467, 2024.
- [30] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.
- [31] Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden biases of end-to-end driving models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8240–8249, 2023.
- [32] Yunpeng Zhang, Deheng Qian, Ding Li, Yifeng Pan, Yong Chen, Zhenbao Liang, Zhiyao Zhang, Shurui Zhang, Hongxu Li, Maolei Fu, et al. Graphad: Interaction scene graph for end-to-end autonomous driving. *arXiv preprint arXiv:2403.19098*, 2024.
- [33] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.
- [34] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahao Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14864–14873, 2024.
- [35] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024.
- [36] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022.
- [37] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15222–15232, 2021.
- [38] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21983–21994, 2023.
- [39] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7953–7963, 2023.
- [40] Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15120–15130, 2024.
- [41] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beßwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European Conference on Computer Vision*, pages 256–274. Springer, 2024.
- [42] Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023.

- [43] Chenbin Pan, Burhaneddin Yaman, Tommaso Nesti, Abhirup Mallik, Alessandro G Allievi, Senem Velipasalar, and Liu Ren. Vlp: Vision language planning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14760–14769, 2024.
- [44] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.
- [45] Yi Xu, Yuxin Hu, Zaiwei Zhang, Gregory P Meyer, Siva Karthik Mustikovela, Siddhartha Srinivasa, Eric M Wolff, and Xin Huang. Vlm-ad: End-to-end autonomous driving through vision-language model supervision. *arXiv preprint arXiv:2412.14446*, 2024.
- [46] Spandan Madan, Timothy Henry, Jamell Dozier, Helen Ho, Nishchal Bhandari, Tomotake Sasaki, Frédo Durand, Hanspeter Pfister, and Xavier Boix. When and how cnns generalize to outof-distribution category-viewpoint combinations. *arXiv preprint arXiv:2007.08032*, 2021.
- [47] Spandan Madan, Tomotake Sasaki, Tzu-Mao Li, Xavier Boix, and Hanspeter Pfister. Small in-distribution changes in 3d perspective and lighting fool both cnns and transformers. *arXiv preprint arXiv:2106.16198*, 3, 2021.
- [48] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Nova: Learning to see in novel viewpoints and domains. In *2019 International Conference on 3D Vision (3DV)*, pages 116–125. IEEE, 2019.
- [49] Tien Do, Khiem Vuong, Stergios I Roumeliotis, and Hyun Soo Park. Surface normal estimation of tilted images via spatial rectifier. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 265–280. Springer, 2020.
- [50] Ruiyuan Gao, Kai Chen, Zhihao Li, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magic-drive3d: Controllable 3d generation for any-view rendering in street scenes. *arXiv preprint arXiv:2405.14475*, 2024.
- [51] Gyusam Chang, Jiwon Lee, Donghyun Kim, Jinkyu Kim, Dongwook Lee, Daehyun Ji, Sujin Jang, and Sangpil Kim. Unified domain generalization and adaptation for multi-view 3d object detection. *Advances in Neural Information Processing Systems*, 37:58498–58524, 2024.
- [52] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pages 156–173. Springer, 2024.
- [53] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024.
- [54] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, et al. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5166–5175, 2024.
- [55] Haithem Turki, Vasu Agrawal, Samuel Rota Bulò, Lorenzo Porzi, Peter Kotschieder, Deva Ramanan, Michael Zollhöfer, and Christian Richardt. Hybridnerf: Efficient neural rendering via adaptive volumetric surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19647–19656, 2024.
- [56] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- [57] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splat image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10208–10217, 2024.

- [58] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19680–19690, 2024.
- [59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [60] 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.
- [61] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. *arXiv preprint arXiv:2503.07656*, 2025.
- [62] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. *arXiv preprint arXiv:2211.10581*, 2022.
- [63] Xuewu Lin, Zixiang Pei, Tianwei Lin, Lichao Huang, and Zhizhong Su. Sparse4d v3: Advancing end-to-end 3d detection and tracking. *arXiv preprint arXiv:2311.11722*, 2023.
- [64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [65] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022.
- [66] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 541–556. Springer, 2020.
- [67] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [68] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [69] 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 *European Conference on Computer Vision*, pages 533–549. Springer, 2022.
- [70] Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023.
- [71] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- [72] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024.
- [73] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.
- [74] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7077–7087, 2021.