SynAD: Enhancing Real-World End-to-End Autonomous Driving Models through Synthetic Data Integration

Jongsuk Kim[1*] Jaeyoung Lee[1*] Gyojin Han[1] Dong-Jae Lee[1] Minki Jeong[2] Junmo Kim[1]
[1]KAIST [2]AI Center, Samsung Electronics
{jskpop, mcneato, hangj0820, jhtwosun, junmo.kim}@kaist.ac.kr minki6.jeong@samsung.com

Abstract

Recent advancements in deep learning and the availability of high-quality real-world driving datasets have propelled end-to-end autonomous driving. Despite this progress, relying solely on real-world data limits the variety of driving scenarios for training. Synthetic scenario generation has emerged as a promising solution to enrich the diversity of training data; however, its application within E2E AD models remains largely unexplored. This is primarily due to the absence of a designated ego vehicle and the associated sensor inputs, such as camera or LiDAR, typically provided in real-world scenarios. To address this gap, we introduce SynAD, the first framework designed to enhance real-world E2E AD models using synthetic data. Our method designates the agent with the most comprehensive driving information as the ego vehicle in a multi-agent synthetic scenario. We further project path-level scenarios onto maps and employ a newly developed Map-to-BEV Network to derive bird's-eye-view features without relying on sensor inputs. Finally, we devise a training strategy that effectively integrates these map-based synthetic data with real driving data. Experimental results demonstrate that SynAD effectively integrates all components and notably enhances safety performance. By bridging synthetic scenario generation and E2E AD, SynAD paves the way for more comprehensive and robust autonomous driving models.

1. Introduction

Autonomous vehicles are moving from research labs to roads as deep learning and comprehensive real-world driving datasets [1] are leading to significant progress. Recent studies leverage LiDAR [16, 33] or multi-camera images [11–13] from these datasets to extract bird's-eye-view (BEV) features [3, 17, 22] for various tasks [10, 18, 21, 35]. These approaches improve the end-to-end autonomous driving (E2E AD) model's performance by incorporating perception tasks (tracking and mapping), prediction tasks (motion forecasting and occupancy prediction), and planning tasks either in parallel [28] or in series [11, 12], in an end-to-end manner. To better capture the complexities of real-world driving environments, some studies [26, 36] have proposed methods incorporating 3D occupancy understanding.

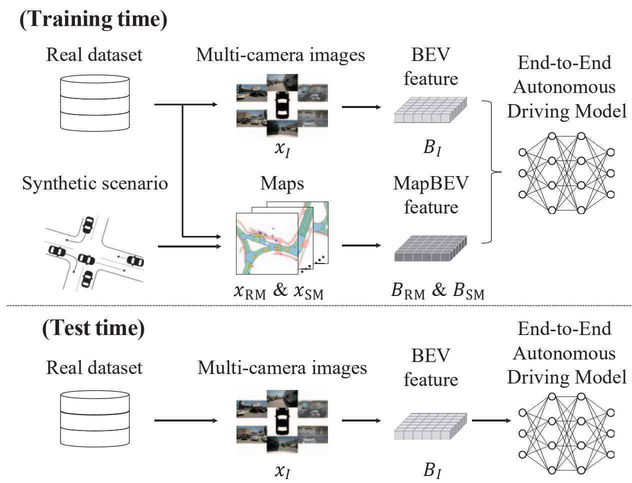

Figure 1. Conceptual illustration of SynAD. During training, both real and synthetic data are used to generate BEV and MapBEV features for the E2E AD model, while only real data is used during testing to ensure practical applicability.

However, relying solely on real-world datasets introduces a fundamental limitation: the high costs of data collection and labeling lead to a lack of diversity, restricting the range of scenarios available for training. To mitigate this issue, some studies [8, 34] generate paths under specific conditions and utilize CARLA simulator [6] to acquire corresponding camera data for additional model training. These approaches enable robust driving in extreme situations but still have limitations since they can only operate in virtual environments. In parallel, several studies have developed methods to generate realistic driving scenarios from real-world datasets without relying on simulators. These

[*]Equal Contribution.

works employ logic-based [38], language-guided [25, 37], and retrieval-based [5] methods to produce high-quality and diverse scenarios that satisfy specific conditions.

Despite the high generative capabilities of these methods, synthetic scenarios have not been effectively integrated into real-world E2E AD model training. A key limitation is that current scenario generation approaches yield only path-level outputs and overlook the designation of an ego vehicle. They also fail to generate the corresponding sensor inputs, such as multi-camera images and LiDAR data, which are required to establish the ego-centric perspective seen in real-world scenarios. Consequently, this absence restricts their integration into real-world E2E AD training pipelines.

To address these challenges, we propose SynAD, a novel framework that enhances real-world E2E AD models by integrating synthetic data. SynAD comprises three key components: First, we introduce an ego-centric scenario generation method specifically tailored for E2E AD training. During scenario generation, we set effective guides while designating the agent with the richest driving information as the ego vehicle. The path of the selected ego vehicle is then set as the target path and serves as additional training data for the E2E AD model. Second, we propose a Map-to-BEV Network to integrate synthetic scenarios into the E2E AD training pipeline. The Map-to-BEV Network encodes BEV features from maps that contain vehicle information from the synthetic scenarios, enabling this integration without relying on sensor data inputs. Finally, we reduce the domain gap between map-based synthetic data and real driving data by also projecting real scenarios onto a map, ensuring consistent integration as shown in Figure 1. Moreover, by selectively utilizing features extracted from each type of map at the most suitable stage, we avoid performance degradation from integrating map data and ensure the model achieves high test time performance with image-only inputs. Extensive ablation studies verify that each component of SynAD contributes effectively to the application of synthetic scenarios in E2E AD training. Our main contributions are summarized as follows:

- To overcome the lack of necessary sensor data and the absence of an ego-centric perspective in synthetic scenarios, we propose SynAD, a novel method that integrates synthetic data into real-world E2E AD models.
- SynAD introduces three key contributions: (1) ego-centric scenario generation method that transforms path-level scenarios into ego-centric maps by designating the most informative agent as the ego vehicle, (2) a Map-to-BEV Network that produces BEV features without relying on any sensor inputs, and (3) a training strategy that effectively utilizes both synthetic and real data.
- Extensive experiments demonstrate that SynAD outperforms existing methods, with ablation studies confirming the effectiveness of each component.

2. Related Works

2.1. Traffic Scenario Generation

Traffic scenario generation is crucial for testing and improving autonomous driving systems by enabling safe and comprehensive validation in simulated environments. Recent studies [4, 8, 24, 27, 30, 34] have focused on safety-critical scenarios, which are difficult to capture in real-world driving due to cost and safety constraints. KING [8] uses a kinematic bicycle model to derive gradients of safety-critical objectives, updating paths that make the ego vehicle more likely to cause accidents. It also improves the robustness of E2E AD in synthetic driving environments based on simulators by fine-tuning these generated scenarios. Beyond purely safety-critical contexts, research on controllable scenario generation [14, 20, 23, 30, 37, 38] is also receiving great attention. These works introduce diffusion models that allow users to specify trajectory properties (e.g., reaching a goal, following speed limits) while preserving physical feasibility and natural behaviors. In addition, some studies [19, 25, 29, 37] leverage large language models to convert user queries into realistic traffic scenarios. RealGen [5] highlights a limitation in generative approaches that they often struggle to produce novel scenarios and propose combining behaviors from multiple retrieved examples for creating new scenarios. However, employing these generated scenarios to improve real-world E2E AD models remains largely unexplored.

2.2. End-to-End Autonomous Driving

E2E AD, particularly vision-centric approaches, has become an active area of recent research. Unlike conventional AD methods [2, 7, 15, 32], which separate perception tasks and planning, vision-centric E2E methods integrate these components into a single unified model. These approaches provide interpretability and safety benefits while improving performance in each downstream task through end-to-end optimization. Planning-oriented modular design principles have driven several recent advances in E2E AD. ST-P3 [11] trains semantic occupancy prediction and planning in an end-to-end manner. UniAD [12] proposes a planning-oriented unification of tracking, online mapping, motion forecasting, occupancy prediction, and planning. Paradrive [28] achieves parallel processing across these modules, boosting runtime speed by nearly threefold. VAD [13] replaces dense rasterized scene representations with fully vectorized data to boost efficiency. Meanwhile, OccNet [26] and OccWorld [36] explore 3D occupancy representation by segmenting the scene into structured cells with semantic labels. Despite differences in network design and framework implementation, these methods all rely on BEV features derived from multi-camera inputs.

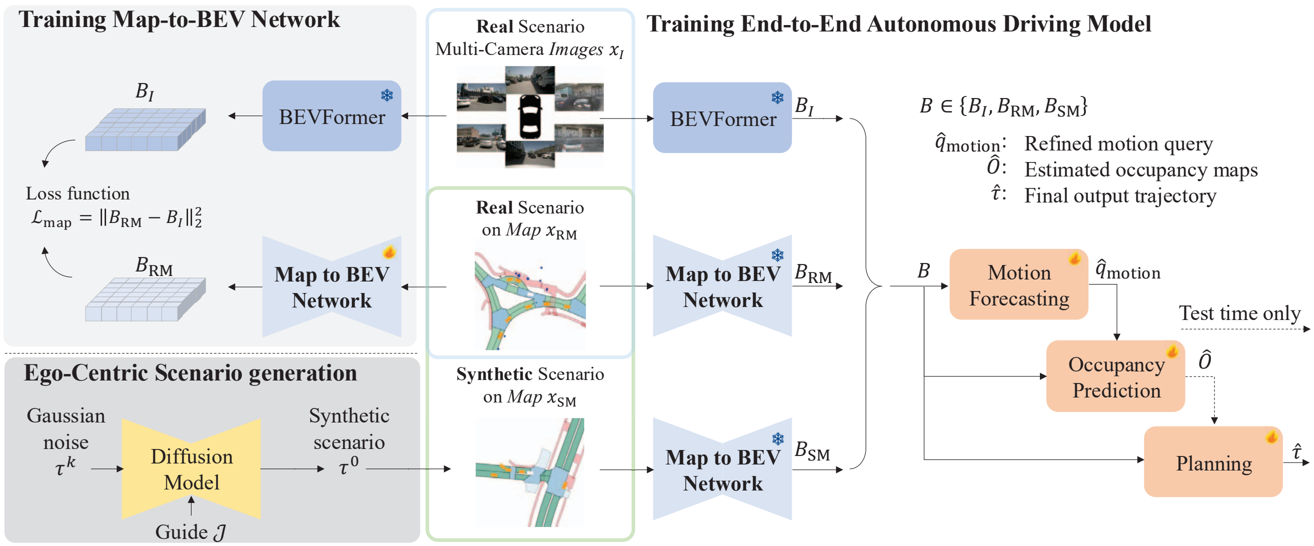

Figure 2. Overview of SynAD. We generate synthetic multi-agent scenarios and convert them into ego-centric map representations x_{SM}, while real scenarios are similarly projected as x_{RM}. To train Map-to-BEV Network, we use paired data from x_{RM} and x_I, ensuring that Map-to-BEV Network produces BEV feature consistent with the output of pretrained BEVFormer applied to multi-camera images. The synthetic scenario x_{SM} can be converted into BEV feature B_{SM} without any multi-camera images using our novel Map-to-BEV network. In the final E2E AD framework, we selectively apply BEV features only to modules that benefit most, thereby improving overall performance.

3. Method

Our method aims to enhance the E2E AD model by leveraging synthetic data. First, we generate multi-agent driving scenarios that satisfy specific conditions and convert them into ego-centric scenarios by designating an ego vehicle and cropping a map centered around it. We denote this synthetic ego-centric map representation as x_{SM}, which is used in E2E AD training. In parallel, we train the Map-to-BEV Network using x_{RM}, constructed by projecting real-world scenarios onto a corresponding map representation. The Map-to-BEV Network aligns BEV features extracted from x_{RM} with multi-camera images x_I, enabling the use of synthetic scenarios without requiring sensor inputs. Finally, we propose a training strategy that incorporates synthetic scenarios into the E2E AD training process. Figure 2 provides an overview of our method.

3.1. Ego-centric Scenario Generation

Realistic Scenario Generation. In an autonomous driving system, the trajectory of a vehicle is represented by its state s at each time step t. This state vector comprises four elements: position in 2D coordinates (x, y), speed v, and heading angle θ, represented as $s = (x, y, v, \theta)$. To generate realistic scenarios in ego-centric autonomous driving environments that meet desired conditions, we employ conditional diffusion models that aim to generate trajectory τ, which includes the state of M agents over T timestamps:

$$\tau = [\tau_1, \tau_2, \dots, \tau_M], \text{ where } \tau_i = [s_i^1, s_i^2, \dots, s_i^T]^\top, \quad (1)$$

s_i^t denotes the state of agent i at time t, and $\tau \in \mathbb{R}^{T \times M \times 4}$. The diffusion model adds Gaussian noise in a forward process and then reconstructs it in a reverse process. Defining τ^k as the trajectory at the k-th diffusion step, the forward process is defined as:

$$q\left(\tau^{1:K} \mid \tau^0\right) = \prod_{k=1}^{K} q\left(\tau^k \mid \tau^{k-1}\right), \quad (2)$$

$$q\left(\tau^k \mid \tau^{k-1}\right) = \mathcal{N}\left(\tau^k; \sqrt{1-\beta_k}\tau^{k-1}, \beta_k I\right), \quad (3)$$

and β_k is the variance schedule controlling the amount of noise added at each diffusion step. Note that τ^0 represents the clean trajectory, and τ^K represents the trajectory corrupted by random noise after K diffusion steps.

To incorporate contextual information into the reverse diffusion process, we construct a composite feature $\mathbf{f}$ by aggregating the image features from the past h timestamps. These image features are extracted from maps that display only the road layout and environmental context, without any vehicle depictions. Then, the reverse diffusion process p_φ can be represented as follows:

$$p_\varphi\left(\tau^{0:K} \mid \mathbf{f}\right) = p\left(\tau^K\right) \prod_{k=1}^{K} p_\varphi\left(\tau^{k-1} \mid \tau^k, \mathbf{f}\right), \quad (4)$$

$$p_\varphi\left(\tau^{k-1} \mid \tau^k, \mathbf{f}\right) = \mathcal{N}\left(\tau^{k-1}; \mu_\varphi\left(\tau^k, k, \mathbf{f}\right), \Sigma_\varphi\left(\tau^k, k, \mathbf{f}\right)\right), \quad (5)$$

where $\tau^K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ starts as random noise and is progressively denoised over K steps by p_φ.

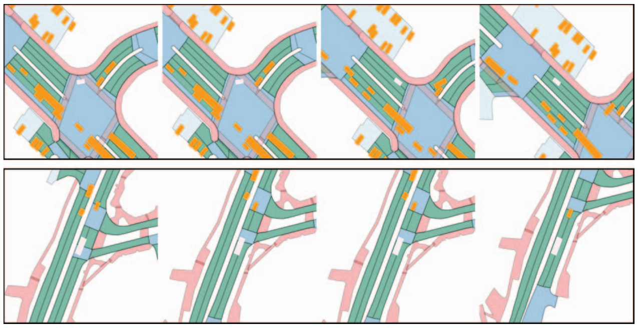

Figure 3. Examples of x_{SM} over time. White box indicates the ego vehicle, while orange boxes denote other vehicles. The synthetic scenarios are conditioned on the existing map representation, then projected using vehicle states and size information.

Guided Sampling. To generate realistic scenarios under diverse conditions, we apply guided sampling during inference. We define a guide $\mathcal{J} = \sum w_i R_i$ as the weighted sum of functions R_i that measure rule satisfaction for the i-th objective. In our work, we employ three specific objectives: preventing agent collisions, preventing map collisions, and adhering to speed limits (detailed in Supp. A.2). We then modify the denoising process by applying the gradient of this guide as follows:

$$p_\varphi\left(\tau^{k-1} \mid \tau^k, \mathbf{f}\right) = \mathcal{N}\left(\tau^{k-1}; \mu_\varphi + \nabla\mathcal{J}(\mu_\varphi), \Sigma_\varphi\right). \quad (6)$$

By modifying the reverse diffusion process, we can dynamically generate trajectories that satisfy each objective.

Ego-centric Scenario. To train the autonomous driving model using multi-agent scenarios generated by the diffusion model, we first determine the ego-vehicle among the agents. Since synthetic scenarios should contain comprehensive driving information, we establish an ego selection rule that designates the vehicle traveling the longest distance as the ego vehicle. Accordingly, the ego index e is determined as:

$$e = \arg\max_i \sum_{t=1}^{T-1} d(s_i^t, s_i^{t+1}), \quad (7)$$

where $d(\cdot,\cdot)$ represents the distance between positions of states. We then crop a fixed-size area centered on the ego vehicle to form the input map x^t_{SM} for timestamp t.

Training the planning module requires the future trajectory of the ego vehicle and the bounding boxes of other vehicles to ensure collision-free predictions. Since the generated trajectories are in the absolute coordinate system, we transform them into coordinate systems relative to the selected ego vehicles to align them with the real data. The transformation aligns the driving direction of the ego vehicle with the positive y-axis and sets its center as the origin. To transform an arbitrary position (x, y) in the absolute coordinate system relative to the state of ego vehicle s, we define the transformation function as:

$$T(x, y; s) = \begin{pmatrix} \sin s_\theta & -\cos s_\theta \\ \cos s_\theta & \sin s_\theta \end{pmatrix} \times \begin{pmatrix} x - s_x \\ y - s_y \end{pmatrix}. \quad (8)$$

The derivation of this specific form of the rotation matrix is included in the Supp. A.3. To obtain the target path over the next T_p timestamps in the ego-centric coordinate frame at time t, we apply the transformation $T(\cdot; s_e^t)$ to the positions of the ego vehicle. We also transform the heading angle relative to the ego vehicle's orientation at time t as:

$$\mathcal{T}^t = \left\{T\left(s_x, s_y; s_e^t\right) \mid s = s_e^{t+t'}, t' \in [T_p]\right\}, \quad (9)$$

$$\Theta^t = \left\{s_\theta - s_{e,\theta}^t \mid s = s_e^{t+t'}, t' \in [T_p]\right\}, \quad (10)$$

where $[T_p]$ denotes the set of integers from 1 to T_p. To process the bounding box information, we first obtain the bounding box coordinates $b_i^{t+t'}$, for each vehicle i at time $t + t'$. We then apply the transformation across other vehicles and timestamps, resulting in:

$$\mathcal{B}^t = \left\{T(x, y;\ s_e^t) \mid (x, y) \in b_i^{t+t'}, i \in [M]\backslash\{e\}, t' \in [T_p]\right\}. \quad (11)$$

Finally, each scenario provides $(\mathcal{T}^t, \Theta^t, \mathcal{B}^t, w_e, h_e)$ for the input x^t_{SM} where $t \in [T - T_p]$. Unlike real driving datasets, the ego vehicle in synthetic data can vary in size across scenarios as shown in Figure 3. Therefore, ego vehicle's width and height (w_e, h_e) are also included in each instance.

3.2. Map-to-BEV Network

To address the absence of sensor inputs (e.g., multi-camera images or LiDAR) in synthetic scenarios, we introduce a Map-to-BEV Network f_B that generates BEV features directly from ego-centric map inputs. Consistent with our synthetic data generation pipeline, we derive the map input x_{RM} from the real scenario. A map encoder f_M processes x_{RM} into a spatial feature, which then serves as the key and value in a Transformer encoder. A learnable query Q_B is used as the query input, producing the mapBEV feature. The encoding process is as follows:

$$B_{\text{RM}} = f_B(Q_B, x_{\text{RM}}) \quad (12)$$

$$= \text{TransformerEncoder}(Q_B, f_M(x_{\text{RM}})). \quad (13)$$

This design enables the Transformer encoder to capture spatial relationships within the map feature, producing accurate map-based BEV representation. Concurrently, we utilize the pre-trained BEVFormer [17] to extract BEV features B_I from multi-camera images x_I, which correspond to the map input x_{RM}. To align the BEV features extracted from the map (B_{RM}) with those extracted from multi-camera images (B_I), we employ an L2 loss function. Formally, the

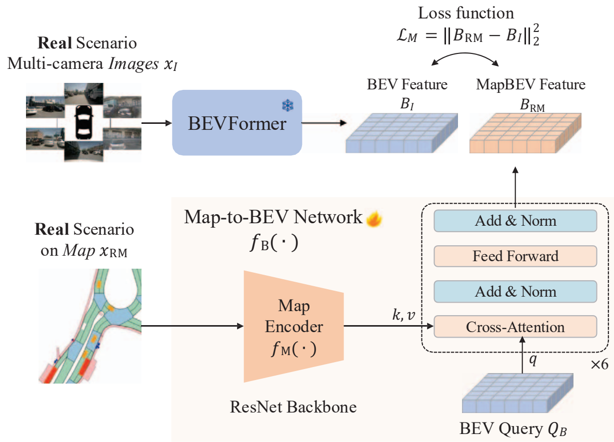

Figure 4. Overview of the Map-to-BEV training. We freeze pre-trained BEVFormer and align B_{RM} with B_I, enabling the network to generate BEV representations without sensor inputs.

loss function for training the Map-to-BEV Network can be expressed as follows:

$$\mathcal{L}_{\text{map}} = \|B_{\text{RM}} - B_I\|_2^2. \tag{14}$$

This step allows the Map-to-BEV Network to generate BEV features without depending on sensor inputs. Consequently, we can encode the BEV features from x_{SM}, enabling E2E AD model training using synthetic scenarios.

3.3. Training E2E AD with Generated Scenario

To train our model, we use generated data x_{SM} alongside multi-camera image data x_I and real data x_{RM}. The image data x_I is fed into BEVFormer to produce BEV features B_I, while map data x_{RM} and x_{SM} pass through the Map-to-BEV Network to obtain B_{RM} and B_{SM}, respectively. E2E AD models typically comprise three main BEV-based modules: a perception module that handles tracking and mapping, a prediction module for motion forecasting and occupancy prediction, and a planning module. Since our map input already includes much of the perception-level information, we do not incorporate it into the perception module, and focus on integrating map data into prediction and planning modules. In the following, we provide a brief overview of each module, with additional details in Supp. B.

Motion Forecasting. To predict trajectories for multi-agents over multi-timestamps with possible N series of waypoints, we employ MotionEncoder and MotionDecoder based on deformable cross-attention [39]. MotionEncoder produces a motion query embedding q_{motion} that represents general motion patterns, agent-centered motion offsets, and relationships to ego vehicle. The MotionDecoder refines this embedding using the BEV feature, yielding multiple predicted trajectories $\hat{\mathbf{x}}_n$ and their probabilities p_n. To handle multi-hypothesis output, we find trajectory $\hat{\mathbf{x}}_{n^*}$, closest to ground-truth trajectory $\mathbf{x}_{\text{GT}}$ by minimizing average displacement error over time. We then calculate joint negative log-likelihood loss as:

$$\mathcal{L}_{\text{JNLL}} = -\log\left(p_{n^*} \cdot P(\mathbf{x}_{\text{GT}} \mid \hat{\mathbf{x}}_{n^*})\right), \tag{15}$$

$$\text{where} \quad n^* = \arg\min_n \left(\frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{x}}_n^t - \mathbf{x}_{\text{GT}}^t\|_2^2\right). \tag{16}$$

With the minimum final displacement error (minFDE), total motion forecasting loss $\mathcal{L}_{\text{motion}}$ is defined as:

$$\mathcal{L}_{\text{minFDE}} = \min_n \left(\|\hat{\mathbf{x}}_n^T - \mathbf{x}_{\text{GT}}^T\|_2^2\right), \tag{17}$$

$$\mathcal{L}_{\text{motion}} = \lambda_{\text{JNLL}}\mathcal{L}_{\text{JNLL}} + \lambda_{\text{minFDE}}\mathcal{L}_{\text{minFDE}},$$

where λ_{JNLL} and λ_{minFDE} are corresponding loss weights.

Occupancy Prediction. An occupancy prediction module forecasts future occupancy maps. Using embedding $\hat{q}_{\text{motion}}$ from motion forecasting module, we first derive temporal queries q_{temp}. Then, temporal queries refine a down-scaled BEV feature B_{state}^{t-1} through a transformer-based OccDecoder, producing updated BEV feature B_{state}^t. Once all timestamps have been processed, we combine final BEV features with instance queries to produce occupancy maps $\hat{O}^t$. The predicted occupancy maps $\hat{O} = \{\hat{O}^1, ..., \hat{O}^T\}$ are compared with ground-truth occupancy maps O_{GT} to compute occupancy prediction loss $\mathcal{L}_{\text{occ}}$, which consists of dice loss $\mathcal{L}_{\text{dice}}$ and binary cross-entropy loss $\mathcal{L}_{\text{bce}}$ as follows:

$$\mathcal{L}_{\text{occ}} = \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}(\hat{O}, O_{\text{GT}}) + \lambda_{\text{bce}}\mathcal{L}_{\text{bce}}(\hat{O}, O_{\text{GT}}), \tag{18}$$

where λ_{dice} and λ_{bce} are the corresponding loss weights.

Planning. Following prior works [11, 12], we concatenate a high-level command embedding with a learnable parameter and pass them through a linear layer to form the initial planning query q_{plan}. A Transformer-based PlanDecoder refines this query using an adapted BEV feature B_a as the key and value inputs:

$$\hat{q}_{\text{plan}} = \text{PlanDecoder}(q_{\text{plan}}, B_a). \tag{19}$$

Then $\hat{q}_{\text{plan}}$ passes through an MLP to obtain displacement vectors $\Delta\hat{\tau} = \text{MLP}(\hat{q}_{\text{plan}})$. Taking the cumulative sum of these displacements across timesteps produces the final predicted trajectory $\hat{\tau}$. The planning loss is defined as the sum of the imitation loss and the collision loss, both computed from $\hat{\tau}$. The imitation loss measures the L2 distance between the $\hat{\tau}$ and the ground-truth trajectory τ. For the collision loss, we obtain the ego vehicle's bounding box at timestamp t as:

$$\hat{b}^t(\delta) = \text{box}(\hat{\tau}^t, w_e + \delta, h_e + \delta). \tag{20}$$

where δ is a safety margin. We then compute collision loss using IoU between $\hat{b}^t(\delta)$ and each other vehicle's bounding box b_i^t across all timesteps. Combining these losses for multiple values of δ yields the planning loss as below:

$$\mathcal{L}_{\text{col}}(\delta) = \sum_{i,t} \text{IoU}\left(\hat{b}^t(\delta), b_i^t\right), \tag{21}$$

$$\mathcal{L}_{\text{plan}} = \|\tau - \hat{\tau}\|_2^2 + \sum_{(\lambda_\delta, \delta)} \lambda_\delta \mathcal{L}_{\text{col}}(\delta). \tag{22}$$

Note that in training with generated scenarios, (w_e, h_e) may vary, the ground-truth trajectory τ is taken from $\mathcal{T}$ (Eq. 9), and each bounding box b_i^t is an element of $\mathcal{B}$ (Eq. 11).

Finally, the loss function for E2E AD training can be expressed by incorporating scaling factors as follows:

$$\mathcal{L}_{\text{E2E}} = \lambda_{\text{motion}}\mathcal{L}_{\text{motion}} + \lambda_{\text{occ}}\mathcal{L}_{\text{occ}} + \lambda_{\text{plan}}\mathcal{L}_{\text{plan}}. \tag{23}$$

Map-Data Integration. We incorporate map-based BEV features into the motion forecasting and planning modules, where additional contextual information (e.g., road geometry, traffic structure) proves beneficial. In contrast, occupancy prediction requires high spatial precision, making 2D map data less helpful [36]; we thus exclude map inputs for this module to prevent performance degradation. Experimental results confirm that this selective integration avoids degrading overall performance and maintains strong test-time accuracy with image-only data.

4. Experiments

4.1. Implementation Details

Scenario Generation. We conduct all experiments using nuScenes [1], a real-world driving dataset containing 1,000 scenes. Each nuScenes scene consists of 40 video frames, capturing a 20-second video at 2Hz. We set the future prediction timestamp T_p to 6, which yields 34 training instances per scene. For our main results, we train the model from scratch for 5 epochs while adding 500 synthetic scenes, which is equivalent to 7.5 epochs if training solely on the original nuScenes dataset. Despite this additional data, our total training cost remains lower than that of other E2E AD methods. For ablation studies, unless otherwise noted, we use 100 synthetic scenes for training. To maintain sufficient interaction complexity, we exclude any instance that contains only a single driving agent.

Training Details. For training the Map-to-BEV Network, we freeze the pre-trained BEVFormer [17] and update only the Map-to-BEV network parameters over 20 epochs. For the E2E AD model, we train each module from scratch while keeping the BEVFormer and the Map-to-BEV network frozen. At test time, we apply the occupancy-based optimization from UniAD [12]. All experiments are conducted on 8 NVIDIA RTX 4090 GPUs with batch size 1 per GPU. More details can be found in the Supp. C.

Method	no collision ↓			no offroad ↓		
	rule	real	rel real	rule	real	rel real
BITS [31]	0.065	0.099	0.352	0.018	0.099	0.355
BITS+opt [31]	0.041	0.070	0.353	0.005	0.100	0.358
CTG [38]	0.052	0.044	0.346	**0.002**	0.042	0.346
CTG++ [37]	0.036	**0.040**	0.332	0.004	**0.038**	0.328
SynAD(Ours)	**0.033**	0.045	**0.330**	**0.002**	0.040	**0.324**

Table 1. Evaluation of synthetic scenarios with varying guidance.

Method	L2(m) ↓				Colllsion Rate(%) ↓			
	$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
ST-P3† [11]	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD [12]	0.48	0.74	1.07	0.76	0.12	0.13	0.28	0.17
VAD [13]	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22
OCCNet† [26]	1.29	2.31	2.99	2.14	0.21	0.59	1.37	0.72
Paradrive [28]	**0.25**	**0.46**	**0.74**	**0.48**	0.14	0.23	0.39	0.25
OCCWorld [36]	0.32	0.61	0.98	0.64	0.06	0.21	0.47	0.24
SynAD (Ours)	0.52	0.78	1.10	0.80	**0.04**	**0.10**	**0.20**	**0.11**

Table 2. Planning performance on the nuScenes validation set. † denotes results evaluated under the ST-P3 metric.

4.2. Main Results

We evaluate our method on the nuScenes validation set, adopting the CTG++ [37] metrics for scenario generation and the VAD [13] evaluation protocol for the E2E AD task, ensuring consistency with existing methods. Details on the reporting rules can be found in Supp. D

Scenario Generation. In Table 1, we evaluate our generated paths using three metrics: *rule*, *real*, and *rel real*. The *rule* metric indicates how strictly the generated trajectories adhere to given rules. The *real* metric measures absolute similarity to real-world data using the Wasserstein distance, while *rel real* assesses the realism of scene-level interactions between vehicles. Our method demonstrates robust compliance with traffic constraints, as indicated by its substantial *rule* score. Although it has a slightly lower *real* score, suggesting a looser correspondence to exact real-world trajectories, it achieves a higher *rel real* score that highlights more sophisticated multi-agent interactions. These results show that the generated trajectories deviate from real-world paths while still capturing diverse driving behaviors, which is advantageous for building more robust E2E AD systems.

Planning. Table 2 presents our planning performance from two perspectives: trajectory accuracy, measured by the L2 distance error from the ground truth path, and safety, represented by the collision rate with other vehicles. While

Method	Motion Forecasting ↓			Occupancy. ↑	
	minADE	minFDE	MR	IoU-n	IoU-f
UniAD [12]	0.75	1.10	0.166	**61.9**	**39.7**
VAD* [13]	0.78	1.11	0.169	-	-
Paradrive [28]	0.73	1.08	0.162	60.0	36.4
SynAD (Ours)	**0.69**	**1.01**	**0.154**	60.5	39.6

Table 3. Prediction performance on the nuScenes validation set. Results reproduced in our environments. *VAD does not have an occupancy prediction module.

Updated Modules				Motion Forecasting ↓			Occupancy. ↑		Plan.(avg.) ↓	
x_{RM}			x_{SM}							
Mot.	Occ.	Plan	Plan	minADE	minFDE	MR	IoU-n	IoU-f	L2	Col.
				0.76	1.11	0.162	60.1	38.9	1.15	0.25
			✓	0.75	1.13	0.166	59.3	38.3	0.82	0.19
		✓	✓	0.77	1.15	0.168	60.2	39.0	0.79	0.18
	✓	✓	✓	0.77	1.14	0.167	58.4	37.6	0.78	0.18
✓		✓	✓	**0.73**	**1.06**	**0.157**	**60.2**	**39.2**	**0.77**	**0.14**

Table 4. Prediction and planning performance variations based on the incorporation of x_{RM} and x_{SM} in each E2E AD module.

# Synthetic scenes	Motion Forecasting ↓			Occupancy. ↑		Plan.(avg.) ↓	
	minADE	minFDE	MR	IoU-n	IoU-f	L2	Col.
Baseline							
0	0.76	1.11	0.162	60.1	38.9	1.15	0.25
Same step (Fair comparison)							
100	0.73	1.06	0.157	**60.2**	**39.2**	**0.77**	0.14
300	**0.72**	**1.02**	**0.153**	59.4	38.6	0.81	**0.13**
500	0.73	1.03	0.155	59.4	38.7	0.85	0.14
Same epoch (Longer training)							
100	0.72	1.04	0.156	60.3	39.1	**0.76**	0.13
300	0.71	1.02	0.155	60.3	39.4	0.77	0.12
500	**0.69**	**1.01**	**0.154**	**60.5**	**39.6**	0.80	**0.11**

Table 5. Performance under different numbers of generated scenes, comparing two training protocols.

Arch.	input res.	$\mathcal{L}^{\text{val}}_{\text{map}}$ ↓ $(\times 10^{-2})$	Motion Forecasting ↓			Plan.(avg.) ↓	
			minADE	minFDE	MR	L2	Col.
SwinUNETR	800	9.55	0.75	1.11	0.158	1.08	0.26
Ours	224	**8.96**	**0.73**	**1.06**	**0.157**	**0.77**	**0.14**

Table 6. Performance variations based on Map-to-BEV network architectures.

our SynAD model exhibits slightly higher L2 distance errors due to the broader distribution of generated behaviors, it achieves the lowest collision rate among all baselines, indicating superior collision avoidance. This trade-off stems from emphasizing more diverse, realistic interactions during scenario generation, which yields safer but not necessarily GT-matching trajectories. Since real-world driving prioritizes collision avoidance over precise path replication, our approach is particularly well-suited for practical deployment. Additionally, SynAD is the only method to incorporate variations in vehicle sizes during training, further enhancing its adaptability to real-world driving conditions.

Prediction. Motion forecasting and occupancy prediction results provide insights into the E2E AD model's ability to interpret and anticipate the behavior of surrounding objects and agents. The results in Table 3 show that SynAD excels in accurately predicting the movements of surrounding agents and maintains a solid understanding of environmental occupancy. Even when synthetic data is introduced as a new input type, the model demonstrates robust performance during testing with image-only input, validating the effectiveness of the integration strategy.

4.3. Ablation Studies

Training Strategy. To effectively leverage the synthetic scenarios in our E2E AD framework, we use x_{RM}, the real scenario projected onto the map, as a training bridge. Table 4 presents the results of this approach. First, incorporating x_{SM} into the planning module training significantly improves planning performance, while adding x_{RM} provides a modest additional gain. However, when we extend real map to the occupancy prediction module, performance declines, suggesting that 2D map representations alone are insufficient for this task. These results indicate that BEV features extracted from map data suffice for motion forecasting and planning but fall short for occupancy prediction. The latter often requires richer spatial information, as evidenced by OCCNet [26] and OCCWorld [36], which leverage 3D data to improve performance. Consequently, our main training strategy updates only the motion forecasting and planning modules through the map data.

Scale of Synthetic Scenarios. Table 5 illustrates how varying their number influences performance under two training protocols: one with the same number of training steps and another with the same number of epochs. When no synthetic scenes are used, the model relies solely on multi-camera image data, forming our baseline. In the same-step protocol, incorporating a moderate amount of synthetic scene improves results, although further increases yield diminishing results. Under the same-epoch protocol, introducing more synthetic scenes consistently enhances performance, demonstrating that the model benefits from broader coverage given sufficient training iterations. Across both protocols, L2 distance tends to increase with more scenes, reflecting the broader distribution of synthetic scenarios. In particular, faster convergence under the same training steps as the baseline underscores the advantages of using the synthetic scenario.

Map-to-BEV Network Architecture. Table 6 shows the ablation results on the different network architectures for the Map-to-BEV Network. We compare our model with SwinUNETR [9], which preserves spatial correspondence between the map and BEV features. One observation is that

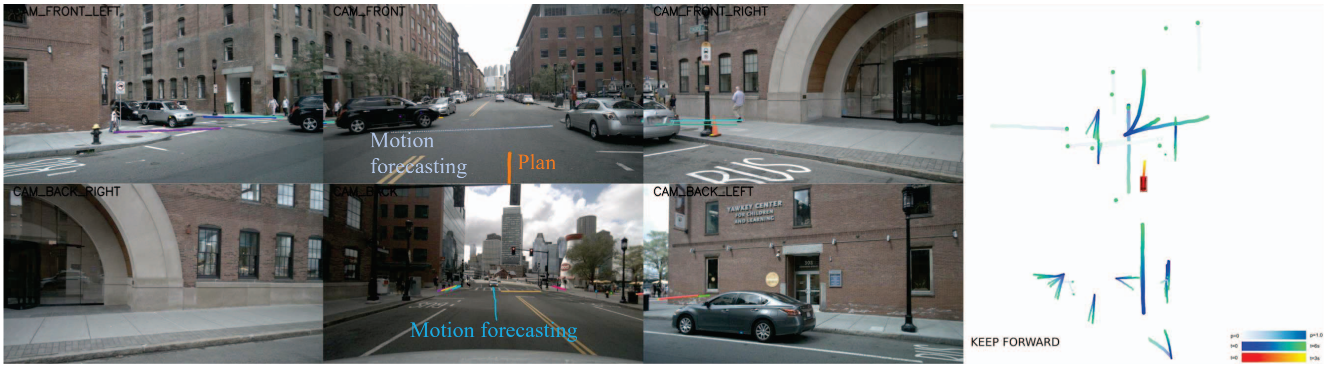

Figure 5. Qualitative result of SynAD. The performance of SynAD in an urban driving scenario is presented through six views capturing the surroundings. The front and back vehicles' motion forecasting are visualized with color-coded trajectories, where warmer colors (red) indicate more immediate movements and cooler colors (blue) represent later positions.

Guide			L2(m) ↓				Collision Rate(%) ↓			
agent	map	speed	$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
✓			**0.48**	**0.73**	**1.05**	**0.75**	0.05	0.15	0.35	0.18
✓	✓		0.49	0.74	1.06	0.76	0.05	0.12	0.27	0.15
✓	✓	✓	0.50	0.75	1.07	0.77	**0.05**	**0.11**	**0.26**	**0.14**

Table 7. Planning performance with varying guide functions for sampling synthetic scenarios.

Rule	Dist.	L2(m) ↓				Colllsion Rate(%) ↓			
		$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
Random	3.67	0.51	0.77	1.09	0.79	0.06	0.11	0.31	0.16
Dynamic	3.68	0.50	0.77	1.11	0.79	**0.04**	**0.10**	0.31	0.15
Longest	3.70	**0.50**	**0.75**	**1.07**	**0.77**	0.05	0.11	**0.26**	**0.14**

Table 8. Planning performance with different ego vehicle selection rules in synthetic scenarios. *Dist.* represents the average distance traveled between consecutive frames.

SwinUNETR requires high-resolution inputs to achieve sufficient performance, leading to higher computational costs. After training each Map-to-BEV Network variant, we evaluate BEV feature quality using an L2 loss on the validation dataset (i.e. $\mathcal{L}_{\text{map}}^{\text{val}}$). Then, we compare the performance of motion forecasting and planning in E2E AD training, which are influenced by map data. The results demonstrate that our design outperforms SwinUNETR while incurring lower computational costs.

Guide Composition for Scenario Generation. Table 7 presents the impact of different guide compositions from Equation 6 on planning performance. The agent guide aims to prevent collisions between agents, and the map guide prevents collisions with map components, and the speed guide enforces both minimum and maximum speeds. Incorporating the map and speed guides progressively decreases collision rates, demonstrating that these additional constraints enhance safety. While we focus on specific guide functions, extending the approach, such as using LLM- or retrieval-based guidance, is left for future work.

Ego Selection Rule. For synthetic scenario generations, we experiment with three ego selection rules: random, dynamic, and longest. Each rule selects vehicles with a minimum movement of $1m$ over the generated timestamps to ensure meaningful training. The random rule selects any vehicle meeting this criterion, while the dynamic rule designates the vehicle with the largest lateral (x-axis) movement. Lastly, the longest rule selects the ego vehicle that traveled the longest distance, following Equation 7. Table 8 presents both the planning performance and the average distance traveled by the ego vehicle over future timestamp T_p. When selecting the ego vehicle based on the longest rule, the model shows the lowest collision rate. Furthermore, the longest rule achieves the lowest L2 errors with sufficient trajectory distance. Based on the results, the longest rule is set as the ego selection rule for the synthetic scenario that we incorporate into our E2E AD training.

5. Conclusion

We propose SynAD, a novel method that integrates synthetic scenarios into real-world E2E AD models. SynAD overcomes the limitations that previously confined such integrations to virtual environments like simulators. By utilizing map-based BEV feature encoding, we enable the training of synthetic scenarios without relying on sensor data such as multi-camera images or LiDAR data. Also, we propose ego-centric scenario generation methods and strategic integration approaches. Meanwhile, integrating synthetic scenarios holds significant potential for incorporation into existing E2E AD pipelines. We leave applying our integration strategy to other E2E AD methods for future work.

Acknowledgments

This work was supported by Samsung Electronics Co., Ltd (IO231005-07280-01).

References

[1] 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. 1, 6

[2] Long Chen, Lukas Platinsky, Stefanie Speichert, Błażej Osiński, Oliver Scheel, Yawei Ye, Hugo Grimmett, Luca Del Pero, and Peter Ondruska. What data do we need for training an av motion planner? In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1066–1072. IEEE, 2021. 2

[3] Shaoyu Chen, Tianheng Cheng, Xinggang Wang, Wenming Meng, Qian Zhang, and Wenyu Liu. Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer. *arXiv preprint arXiv:2206.04584*, 2022. 1

[4] Wenhao Ding, Baiming Chen, Minjun Xu, and Ding Zhao. Learning to collide: An adaptive safety-critical scenarios generating method. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2243–2250. IEEE, 2020. 2

[5] Wenhao Ding, Yulong Cao, Ding Zhao, Chaowei Xiao, and Marco Pavone. Realgen: Retrieval augmented generation for controllable traffic scenarios. *arXiv preprint arXiv:2312.13303*, 2023. 2

[6] 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. 1

[7] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on intelligent transportation systems*, 17(4):1135–1145, 2015. 2

[8] Niklas Hanselmann, Katrin Renz, Kashyap Chitta, Apratim Bhattacharyya, and Andreas Geiger. King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients. In *European Conference on Computer Vision*, pages 335–352. Springer, 2022. 1, 2

[9] Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R Roth, and Daguang Xu. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI brainlesion workshop*, pages 272–284. Springer, 2021. 7

[10] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021. 1

[11] 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. 1, 2, 5, 6

[12] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 1, 2, 5, 6, 7, 3

[13] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023. 1, 2, 6, 7

[14] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9644–9653, 2023. 2

[15] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 international conference on robotics and automation (ICRA)*, pages 8248–8254. IEEE, 2019. 2

[16] Tarasha Khurana, Peiyun Hu, Achal Dave, Jason Ziglar, David Held, and Deva Ramanan. Differentiable raycasting for self-supervised occupancy forecasting. In *European Conference on Computer Vision*, pages 353–369. Springer, 2022. 1

[17] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022. 1, 4, 6, 3

[18] Zhi Liu, Shaoyu Chen, Xiaojie Guo, Xinggang Wang, Tianheng Cheng, Hongmei Zhu, Qian Zhang, Wenyu Liu, and Yi Zhang. Vision-based uneven bev representation learning with polar rasterization and surface estimation. In *Conference on Robot Learning*, pages 437–446. PMLR, 2023. 1

[19] Zhiyuan Liu, Leheng Li, Yuning Wang, Haotian Lin, Zhizhe Liu, Lei He, and Jianqiang Wang. Controllable traffic simulation through llm-guided hierarchical chain-of-thought reasoning. *arXiv preprint arXiv:2409.15135*, 2024. 2

[20] Jack Lu, Kelvin Wong, Chris Zhang, Simon Suo, and Raquel Urtasun. Scenecontrol: Diffusion for controllable traffic scene generation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16908–16914. IEEE, 2024. 2

[21] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14074–14083, 2020. 1

[22] 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. 1

[23] Ethan Pronovost, Meghana Reddy Ganesina, Noureldin Hendy, Zeyu Wang, Andres Morales, Kai Wang, and Nick Roy. Scenario diffusion: Controllable driving scenario generation with diffusion. *Advances in Neural Information Processing Systems*, 36:68873–68894, 2023. 2

[24] Davis Rempe, Jonah Philion, Leonidas J Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17305–17315, 2022. 2

[25] Bo-Kai Ruan, Hao-Tang Tsui, Yung-Hui Li, and Hong-Han Shuai. Traffic scene generation from natural language description for autonomous vehicles with large language model. *arXiv preprint arXiv:2409.09575*, 2024. 2

[26] 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. 1, 2, 6, 7, 3

[27] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9909–9918, 2021. 2

[28] 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. 1, 2, 6, 7, 3

[29] Junkai Xia, Chenxin Xu, Qingyao Xu, Chen Xie, Yanfeng Wang, and Siheng Chen. Language-driven interactive traffic trajectory generation. *arXiv preprint arXiv:2405.15388*, 2024. 2

[30] Chejian Xu, Ding Zhao, Alberto Sangiovanni-Vincentelli, and Bo Li. Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023. 2

[31] Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Bits: Bi-level imitation for traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2929–2936. IEEE, 2023. 6

[32] Wenda Xu, Qian Wang, and John M Dolan. Autonomous vehicle motion planning via recurrent spline optimization. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7730–7736. IEEE, 2021. 2

[33] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019. 1

[34] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15459–15469, 2024. 1, 2

[35] 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

[36] 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, 2025. 1, 2, 6, 7, 3

[37] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-guided traffic simulation via scene-level diffusion. In *Conference on Robot Learning*, pages 144–177. PMLR, 2023. 2, 6, 3

[38] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3560–3566. IEEE, 2023. 2, 6

[39] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 5