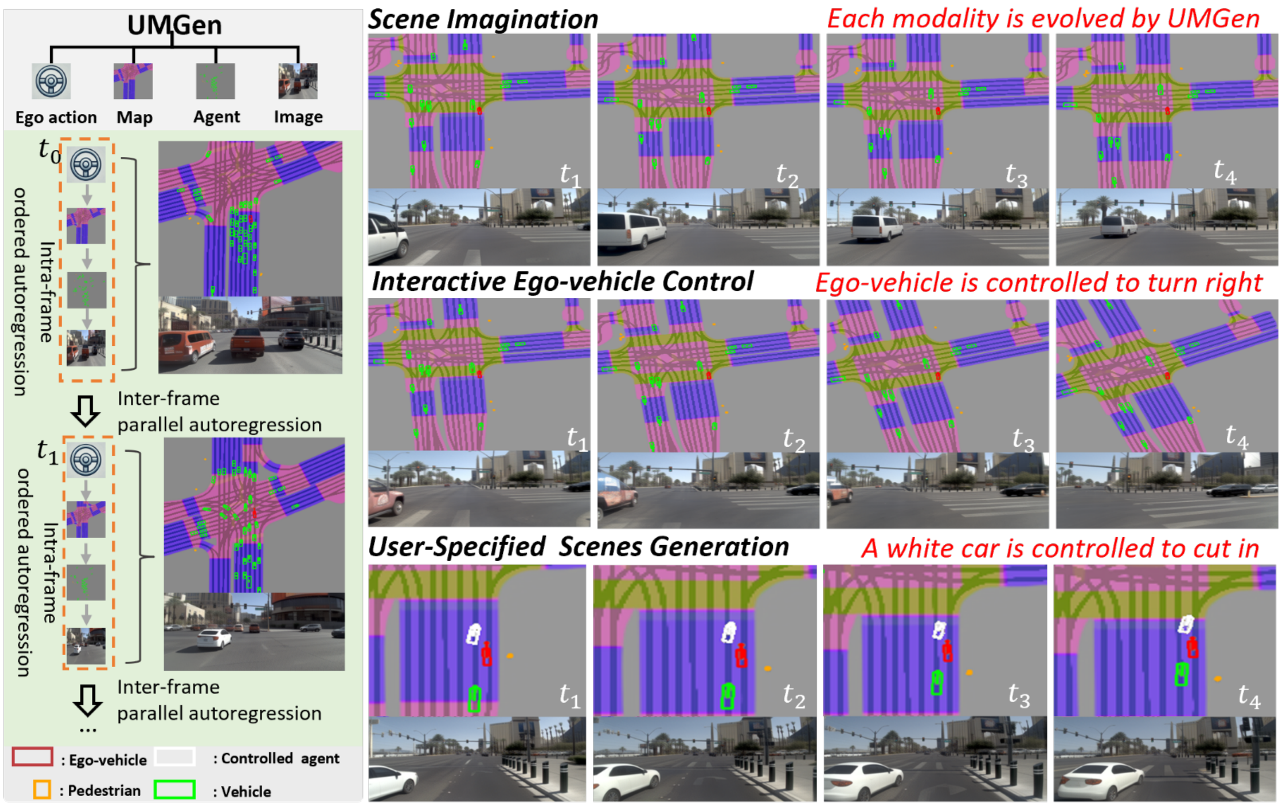
This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Generating Multimodal Driving Scenes via Next-Scene Prediction

Yanhao Wu[1,2] * Haoyang Zhang[2] Tianwei Lin[2] Lichao Huang[2] Shujie Luo[2]
Rui Wu[2] Congpei Qiu[1] Wei Ke[1†] Tong Zhang[3,4]
[1] School of Software Engineering, XJTU [2] Horizon Robotics
[3] School of Computer and Communication Sciences, EPFL
[4] University of Chinese Academy of Sciences



(a) UMGen **(b) Infer four modalities w/o or with conditions on user inputs**

Figure 1. An overview of our proposed driving scene generation paradigm – UMGen. Starting from a random initialization (a) UMGen generates ego-centric, multimodal scenes frame-by-frame. Each scene encompasses four modalities: ego-vehicle action, map, traffic agent, and image; (b) UMGen offers multiple functions. It can autonomously generate multimodal scene sequences based solely on its own historical context, but also predict other modalities based on input ego-vehicle actions provided by users. Furthermore, UMGen can incorporate user-specified agent actions to create customized scene sequences. In three scene sequences, arranged from top to bottom, we demonstrate the ego vehicle autonomously driving straight through an intersection, executing a user-defined right turn, and encountering scenes where a user-specified white car cuts in front of it. For better visualization, a portion of the map in the last row is zoomed in.

Abstract

Generative models in Autonomous Driving (AD) enable diverse scenario creation, yet existing methods fall short by only capturing a limited range of modalities, restricting the capability of generating controllable scenes for comprehensive evaluation of AD systems. In this paper, we introduce a multimodal generation framework that incorporates four major data modalities, including a novel addition of the map modality. With tokenized modalities, our scene sequence generation framework autoregressively predicts each scene while managing computational demands through a two-stage approach. The Temporal AutoRegressive (TAR) component captures inter-frame dynamics for *each modality, while the Ordered AutoRegressive (OAR) component aligns modalities within each scene by sequentially predicting tokens in a fixed order. To maintain coherence between map and ego-action modalities, we introduce the Action-aware Map Alignment (AMA) module, which applies a transformation based on the ego-action to maintain coherence between these two modalities. Our framework effectively generates complex, realistic driving scenes over extended sequences, ensuring multimodal consistency and offering fine-grained control over scene elements. Project page: https://yanhaowu.github.io/UMGen.*

1. Introduction

Generative models are becoming increasingly essential across various domains [1, 12, 13, 21]. In Autonomous Driving (AD) systems, generative models are utilized for producing diverse driving scenes—particularly those that are rare or not well-represented in datasets [6, 28, 31] or real-life. More importantly, generative models can be used to build a closed-loop simulation system that generates realistic interactive scenes to test AD systems without causing potential accidents before deployment [29, 30]. Such capabilities enhance the safety, adaptability, and reliability of autonomous driving [8, 14, 24, 27] systems when dealing with the complexities of real-world driving conditions.

To generate comprehensive driving scenes, as described in Tab. 1, most existing studies [7, 9] focus primarily on generating two modalities. For instance, GUMP [9] and TrafficGen [5] generate ego-action and agent motion within a given static map segment; DriveDreamer [28] and GAIA-1 [7] are capable of generating images given initial conditions. While these approaches offer valuable insights, they face some limitations. The lack of map progression in GUMP and TrafficGen limits their realism, as real-world driving scenes involve dynamically evolving maps from the ego-vehicle's viewpoint. DriveDreamer and GAIA-1 fail to forecast traffic agent motion, which limits control over agent behaviors and, consequently, constrains the generation of user-specific scenes.

In this work, we propose a **U**nified **M**ultimodal driving scene **Gen**eration framework (UMGen) for driving scenes, each containing four key modalities. Our framework integrates map prediction to enhance scene representation, enabling finer control over ego-action and agent behaviors. Generating these modalities scene-by-scene while ensuring consistency presents a significant challenge. To address the diverse domains of each modality, we frame scene generation as a sequence scene prediction task. However, directly applying an AutoRegressive (AR) model is impractical due to the high token count from the four modalities and the length of the video data. To address this, we decompose the sequence prediction task into two stages: inter-frame prediction and intra-frame prediction. For inter-frame prediction, we propose the Temporal AutoRegressive (TAR) module, which models temporal evolution across frames using causal attention, allowing each token to be influenced only by its past states. For intra-scene prediction, we introduce the Ordered AutoRegressive (OAR) module, which captures relationships within each scene by enforcing a structured modality order, ensuring alignment and coherence across modalities. Together, TAR and OAR effectively capture cross-modal dependencies over time, enhancing alignment while reducing computational complexity. To further maintain consistency between ego-action and map data, we introduce an Action-aware Map Alignment (AMA) module. AMA applies an affine transformation, based on predicted ego-action, to prior map features, aligning them with the ego-vehicle's movement. This re-alignment preserves coherence between ego-action and map modalities.

Methods	Modalities			
	Ego-action	Map	Agent	Image
Drivedreamer [28]	✓	✗	✗	✓
TrafficGen [5]	✓	✗	✓	✗
GAIA-1 [7]	✓	✗	✗	✓
GUMP [9]	✓	✗	✓	✗
UMGen (Ours)	✓	✓	✓	✓

Table 1. Comparison of different driving scene generation methods across four modalities.

With these components, our model demonstrates strong capabilities in multimodal driving scene generation, as shown in Fig. 1, producing scene sequences that can span up to 60 seconds in duration. Additionally, thanks to the modality alignment capability provided by the AMA and OAR modules, our model can create specific multimodal-consistent scenes, like a cut-in situation, by controlling ego-vehicle and agents' actions.

In summary, UMGen offers several key contributions:

- We introduce a novel generative framework that integrates four distinct modalities—ego-action, road users, traffic maps, and images—with the flexibility to incorporate additional modalities, enhancing the scene representation and fidelity of driving scene generation.
- We design a computationally efficient AR approach, comprising TAR and OAR modules, to capture both inter-frame and intra-frame dependencies, enabling realistic scene generation with reduced computational costs.
- We introduce an AMA module to apply affine transformations to map features with the ego-vehicle's movement, ensuring consistency between ego-action and map.

Our experimental results provide both quantitative and qualitative evidence of our method's effectiveness, showing that UMGen enables user-defined, ego-centric scenario generation adaptable to specific driving conditions.

*: Work done during internship at Horizon Robotics.
†: Corresponding author. Email: wei.ke[at]mail.xjtu.edu.cn

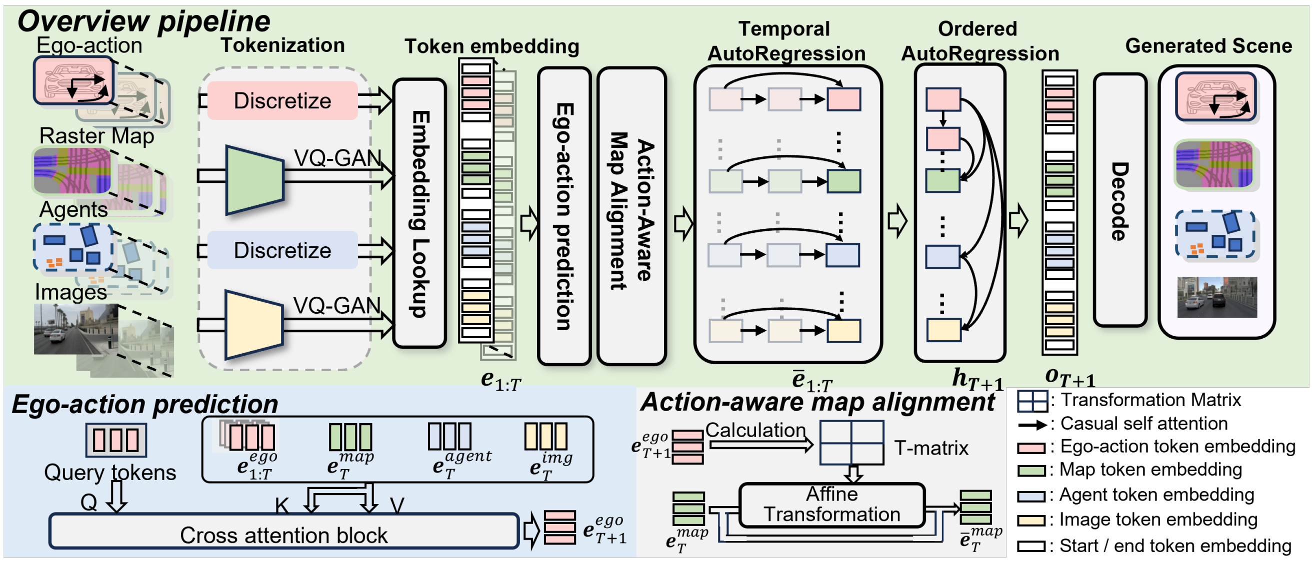


Figure 2. **Pipeline of our UMGen.** Given T past frames of multimodal driving scenes, including ego-action, map, traffic agents, and image in each scene, each modality is tokenized into discrete tokens. The token embeddings are then processed through the Ego-action Prediction module, which forecasts the ego-action for $T+1$ time step. Using this predicted ego-action, the AMA module adjusts the map features. Next, the TAR module aggregates temporal information across sequences, while the OAR module ensures sequential modality prediction within each frame by autoregressively generating each token conditioned on the aggregated history information. Finally, the predicted tokens are fed to the decoder to obtain the next scene.

2. Related Work

Autoregressive generative models. In recent years, AR generative models have achieved significant success in natural language processing and image generation, as evidenced by models such as GPT-2 [20] and VQGAN [3]. This architecture has since been adapted for autonomous driving scene generation, with methods like [19, 23, 32] leveraging it to predict agent behaviors, and GAIA-1 [7] applying it to generate high-quality driving videos sequentially. Our model, however, extends beyond generating limited-modality data by targeting the generation of multimodal data that collectively compose driving scenes. This extension results in a substantial increase in token count, especially as the number of scene frames grows, making the vanilla AR [7] approach—where tokens from different scene frames are concatenated into a single sequence—computationally prohibitive. Additionally, applying the AR mechanism solely along the temporal dimension—without incorporating an intra-frame AR mechanism—may lead to inconsistencies across modalities within the same frame [23, 32]. By contrast, our method consists of parallel inter-frame temporal prediction with intra-frame AR decoding which balances efficiency and consistency to enable multimodal driving scene generation.

Driving scene sequence generation. Traditional methods rely on hand-crafted rules [2, 11, 14, 15] to generate driving scenes, limiting their ability to capture the diversity and realism of the real world. Recently, data-driven deep learning approaches have gained traction, but most focus on generating a limited set of modalities. For example, models like [10, 16, 26, 32] generate diverse agent trajectories based on map segments but fail to handle the emergence and disappearance of agents. GUMP [9] addresses this challenge through a GPT-like [20] generative framework. However, the lack of map generation confines trajectories to specific map segments with static perspectives, which differs from the real world, where vehicles are observed from the dynamic viewpoint of the ego-vehicle. While DriveDreamer2 [31] addresses this limitation by generating maps, it employs two separate networks for map and video generation, resulting in the loss of explicit per-frame control. Other models [6, 7, 17, 28], like Dreamforge, produce videos based on predefined traffic dynamics, which restricts interactivity and user-defined scene generation. In contrast, UMGen generates four essential modalities—ego-action, map, agent, and image—enriching scene representation within a unified framework and providing finer control over user-specific scene sequence generation.

3. Method

3.1. Problem Setup

Our method generates multimodal driving scenes in an autoregressive scene-by-scene manner, starting from either a provided or self-generated initial scene sequence. To achieve this, we tokenize each scene element into sequential tokens, transforming the generation process into a scene token prediction task. These elements include the ego-vehicle's action, a traffic map, other scene agents, and the camera image. Specifically, the ego-vehicle action $\mathbf{a}_t^{AV}$ represents displacements along the x and y axes, as well as angular changes relative to the previous timestamp. The traffic map $\mathbf{m}_t$ depicts road layouts and situational details surrounding the ego vehicle, while the N_a agents $\{\mathbf{a}_t^{(i)}\}_{i=1}^{N_a}$ represent each agent's coordinates, size, velocity, heading and category. This combination of data provides a comprehensive view of the scene's dynamic and static elements. Finally, the camera image $\mathbf{v}_t$ adds visual context. Thus, the multimodal scene at time t is defined as $s_t = \{\mathbf{a}_t^{AV}, \mathbf{m}_t, \{\mathbf{a}_t^{(i)}\}_{i=1}^{N_a}, \mathbf{v}_t\}$. Therefore, our task is to learn a neural network to generate the next scene s_{t+1}, conditioned on $s_{1:t}$.

3.2. Framework

Throughout the following formulation, we assume the availability of the past T scenes as inputs to predict the scene at the $T+1$ timestep. The pipeline of our model is demonstrated in Fig. 2, which comprises several core components: *Tokenization*, an *Ego-action prediction* module, an *Action-aware Map Alignment (AMA)* module, a *Temporal AutoRegressive (TAR)* module, and an *Ordered Autogressive (OAR)* module. We discuss each component below.

Tokenization. The *ego-action* and *agent* attributes are tokenized through discretization, whereas the *raster map* and *image* data are tokenized using pretrained VQ-GAN models [3]. Discrete tokens from modalities are arranged in a fixed sequence within each frame: ego-action, map, agent, and image. This sequence reflects the causal flow: the ego vehicle's actions modify the observable traffic map and influence the behavior of surrounding agents. These interactions are ultimately captured in the camera view. Using this order and the tokenizer functions, we obtain the ordered token at $t-$th scene as follows:

$$\mathbf{z}_t = g\left(s_t\right) = \left[\mathbf{z}_t^{\text{ego}}, \mathbf{z}_t^{\text{map}}, \mathbf{z}_t^{\text{agent}}, \mathbf{z}_t^{\text{image}}\right], \tag{1}$$

where $\mathbf{z}_t \in \mathbb{N}^N$, N denotes the total number of tokens obtained by concatenating tokens from all modalities and g represents the tokenizer function. Unlike vanilla AR approaches [7] that concatenate tokens from multiple frames into a single sequence, we maintain tokens in a structured per-frame format. Consequently, the tokenized scene sequence across T time steps is organized as: $\mathbf{Z} = \left[\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T\right], \mathbf{Z} \in \mathbb{N}^{T\times N}$, where tokens of each frame retain their intra-frame order. This structured token arrangement enables parallel aggregation of temporal information, forming the basis for subsequent modules. To ensure a fixed token count per frame, we apply padding or jointly sample agents across T frames, preserving a consistent set of objects across all frames for temporal coherence.

Token Embedding. Each token is embedded into latent features through learnable token embedding codebooks. All embeddings are then summed with a learnable positional embedding and projected to a unified dimension using a Multi-Layer Perceptron (MLP) head:

$$\mathbf{e}_t^i = \text{MLP}\left(\textit{Embed}(\mathbf{z}_t^i) + \textit{PE}(i)\right), \mathbf{e}_t^i \in \mathbb{R}^D \tag{2}$$

where i is the position index in the token sequence of $\mathbf{z_t}$, *PE* is the learnable position embedding and *Embed* is the learnable token embedding. Thus, the token sequence representing an entire scene can be expressed as:

$$\begin{aligned}\mathbf{e}_t = \big[&\underbrace{\mathbf{e}_t^1, \ldots, \mathbf{e}_t^{n_{ego}}}_{\mathbf{e}_t^{ego}}, \underbrace{\mathbf{e}_t^{n_{ego}+1}, \ldots, \mathbf{e}_t^{n_m}}_{\mathbf{e}_t^{map}}, \\ &\underbrace{\mathbf{e}_t^{n_m+1}, \ldots, \mathbf{e}_t^{n_a}}_{\{\mathbf{e}_t^{agent}\}}, \underbrace{\mathbf{e}_t^{n_a+1}, \ldots, \mathbf{e}_t^{n_v}}_{\mathbf{e}_t^{img}}\big],\end{aligned} \tag{3}$$

where $\mathbf{e}_t \in \mathbb{R}^{N\times D}$, and the indices n_{ego}, n_m, n_a, and n_v correspond to the cumulative token counts for each modality. The terms $\mathbf{e}_t^{ego}$, $\mathbf{e}_t^{map}$, $\mathbf{e}_t^{agent}$, and $\mathbf{e}_t^{img}$ represent the tokens derived from each modality at time step t. More details about tokenization and token embedding can be found in the supplementary material section B.

Ego-action Prediction. We first predict the next ego-action $\mathbf{e}_{T+1}^{ego}$, using it as a prior for generating the remaining modalities. Historical ego actions are aggregated via a cross-attention mechanism to capture the ego vehicle's intentions, which is then integrated with the current environment state $\mathbf{E}_T = \left[\mathbf{e}_T^{map}, \mathbf{e}_T^{agent}, \mathbf{e}_T^{img}\right]$ through another cross-attention mechanism:

$$\begin{aligned}\mathbf{u}_T &= \textit{CA}_{\text{hist}}\left(\mathbf{Q}=\mathbf{q}, \mathbf{K}=\{\mathbf{e}_{1:T}^{ego}\}, \mathbf{V}=\{\mathbf{e}_{1:T}^{ego}\}\right), \\ \mathbf{e}_{T+1}^{ego} &= \textit{CA}_{\text{env}}\left(\mathbf{Q}=\mathbf{u}_T, \mathbf{K}=\{\mathbf{E}_T\}, \mathbf{V}=\{\mathbf{E}_T\}\right),\end{aligned} \tag{4}$$

where $\mathbf{q}$ represents the query tokens, and CA_{hist} and CA_{env} denote the two cross-attention blocks. This module serves as a basic planning component.

Action-aware Map Alignment (AMA). The AMA module is motivated by the continuity of map features across adjacent frames: adjusting current map features based on the ego-vehicle's displacement offers a strong prior for the next timestep. As a result, we first rearrange the map feature vector $\mathbf{e}_T^{map}$ to obtain a spatial map representation, denoted as $\mathbf{F}_T = \Gamma(\mathbf{e}^{map})$, where $\Gamma : \mathbb{R}^{N_m\times D} \to \mathbb{R}^{H\times W\times D}$. Here, N_m refers to the number of tokens representing the map, and H and W denote the height and width of the resulting spatial map, respectively. An affine transformation, parameterized by θ, dx, and dy (derived from $\mathbf{e}_{T+1}^{AV}$), generates a sampling grid $\mathbf{G}$ that maps coordinates from the original map representation $\mathbf{F}_T$ to the transformed locations. For each point (x, y) in $\mathbf{F}_T$, the transformed coordinates (x', y') are given by:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{G}(x, y) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}. \tag{5}$$

Using the sampling grid $\mathbf{G}(x, y)$, we interpolate $\mathbf{F}_T$, producing the transformed map representation $\bar{\mathbf{F}}_T \in$

$\mathbb{R}^{H \times W \times D}$ that represents the transformed map features. Finally, we flatten the representation $\bar{\mathbf{F}}_T$ and add it with $\mathbf{e}_T^{map}$ to yield transformed map representation $\bar{\mathbf{e}}_T^{map}$:

$$\bar{\mathbf{e}}_T^{map} = \text{Flatten}(\bar{\mathbf{F}}_T) + \mathbf{e}_T^{map}. \tag{6}$$

Inter-frame Temporal Autoregression (TAR). With the predicted ego-action and transformed map features, the scene state is updated as $\bar{\mathbf{e}}_T = [\, \mathbf{e}_{T+1}^{ego}, \bar{\mathbf{e}}_T^{map}, \mathbf{e}_T^{agent}, \mathbf{e}_T^{img}]$. For previous frames, we similarly obtain the ego-action and apply the AMA module to map features, resulting in $\bar{\mathbf{e}}_{1:T} \in \mathbb{R}^{T \times N \times D}$. A causal self-attention (CSA) block is applied to $\bar{\mathbf{e}}_{1:T}$, where each token attends only to tokens at the same position across previous scenes:

$$\bar{\mathbf{e}}_{T+1}^i = \text{CSA}\left(\bar{\mathbf{e}}_1^i, \bar{\mathbf{e}}_2^i, \ldots, \bar{\mathbf{e}}_T^i\right), \bar{\mathbf{e}}_{T+1}^i \in \mathbb{R}^D. \tag{7}$$

Where i denotes the token's position. This design captures temporal dependencies efficiently by processing tokens in parallel, thereby avoiding the computational costs associated with performing attention mechanism on the long sequence generated by token concatenation [7]. Subsequently, a bidirectional self-attention (SA) block facilitates information exchange between tokens within the same scene, yielding a coarse prediction $\mathbf{h}_{T+1}$ for the next scene:

$$\mathbf{h}_{T+1} = \text{SA}\left(\bar{\mathbf{e}}_{T+1}^1, \ldots, \bar{\mathbf{e}}_{T+1}^N\right), \mathbf{h}_{T+1} \in \mathbb{R}^{N \times D}. \tag{8}$$

Intra-frame Ordered Autoregression (OAR). The OAR module leverages TAR predictions as temporal priors to enhance inter-frame continuity while maintaining intra-frame consistency through GPT-like autoregressive decoding. To predict the i th token embedding $\mathbf{o}_{T+1}^i$, the OAR combines the previous token embedding $\mathbf{o}_{T+1}^{1:i-1}$ with the TAR-derived feature $\mathbf{h}_{T+1}$, incorporating temporal priors at each prediction step. The OAR then applies a causal self-attention block to sequentially predict each token embedding $\mathbf{o}_{T+1}^i$. Following this, a projection head is applied to $\mathbf{o}_{T+1}^i$, generating a probability vector $\mathbf{p}_{T+1}^{(OAR,i)} \in \mathbb{R}^K$, where K represents the codebook size:

$$w_j = \mathbf{o}_{T+1}^j + \mathbf{h}_{T+1}^{j+1}, \quad j = 1, \ldots, i-1, \tag{9}$$

$$\mathbf{o}_{T+1}^i = \text{CSA}\left(w_1, w_2, \ldots, w_{i-1}\right), \tag{10}$$

$$\mathbf{p}_{T+1}^{(OAR,i)} = \text{Softmax}(\text{MLP}(\mathbf{o}_{T+1}^i)). \tag{11}$$

A Top-k sampling strategy [20] is then used on $\mathbf{p}_{T+1}^{(OAR,i)}$ to select token $\mathbf{z}_{T+1}^i$, which is transformed into embedding for the next prediction. This autoregressive structure aligns tokens contextually, preventing conflicts and enhancing modality coherence.

Loss functions. Additional to the OAR predicted token probability $\mathbf{p}_{T+1}^{OAR}$, we also apply a projection head to $\mathbf{h}_{T+1}$ to obtain $\mathbf{p}_{T+1}^{TAR}$. We then apply cross-entropy loss functions to calculate the total loss:

$$\mathbf{p}_{T+1}^{(TAR)} = \text{Softmax}(\text{MLP}(\mathbf{h}_{T+1})), \tag{12}$$

$$\mathcal{L}_{\text{total}} = CE(\mathbf{p}_{T+1}^{OAR}, \mathbf{z}_{T+1}) + CE(\mathbf{p}_{T+1}^{TAR}, \mathbf{z}_{T+1}), \tag{13}$$

where CE represents the cross-entropy loss function.

4. Experiments

Our experiments evaluate the effectiveness of UMGen in both multimodal driving scene sequence generation and user-guided specific scene sequence generation. We also demonstrate that our model can generate realistic initial scenes, serving as the starting point for sequence generation. Finally, we conduct ablation studies to confirm the critical role of each module.

4.1. Experimental Settings

Training details. We randomly select a 21-frame sequence at each iteration, allowing the model to leverage up to 20 past frames. The training process is conducted over 300 epochs on 32 RTX4090 GPUs for two days.

Evaluation details. Our experiments leverage two public datasets: nuPlan [2] and the Waymo Open Motion Dataset (WOMD) [4]. We assess the realism of the generated initial scenes using Maximum Mean Discrepancy (MMD) [5, 16, 26] scores, following the experimental setup in TrafficGen [5] on both nuPlan and WOMD datasets. The MMD metric quantifies the distributional divergence between the generated and ground-truth agent attributes. When generating initial scenes, we temporarily discard the TAR module since there is no temporal information needed. We present visualizations of the generated multimodal driving scene sequences based on the provided initial frames from nuPlan, verifying both the generation of diverse multimodal scenes and user-specified scene segments. For assessing the computational efficiency of the TAR module relative to the vanilla AR approach, we measure the per-token inference time and peak GPU memory usage during inference. To verify the efficacy of the OAR module in maintaining token consistency, we use both the MMD metric and the average collision rate (CR) among generated road users. Lower CR values indicate reduced conflicts between tokens, demonstrating better consistency. Further training and evaluation details are provided in the Appendix.

4.2. Driving Scene Sequence Generation

Unlike other methods that rely on supplementary conditions and generate limited modalities [6, 9, 28], our model is capable of generating diverse multimodal scene sequences from only given initial frames. To demonstrate the dynamic evolution of our multimodal scenes, as well as the temporal and multimodal consistency of the generated scenes, we present two video samples at different time intervals.

As shown in the first row of Fig 3, we display the generated scenes at 15-second intervals, showing how all modalities evolve as the ego-vehicle moves. Specifically, the ego vehicle performs maneuvers such as lane changes, turns, and straight driving. The map modality generates a variety of road elements, including intersections, curves, and straight paths according to the ego vehicle's movement. The

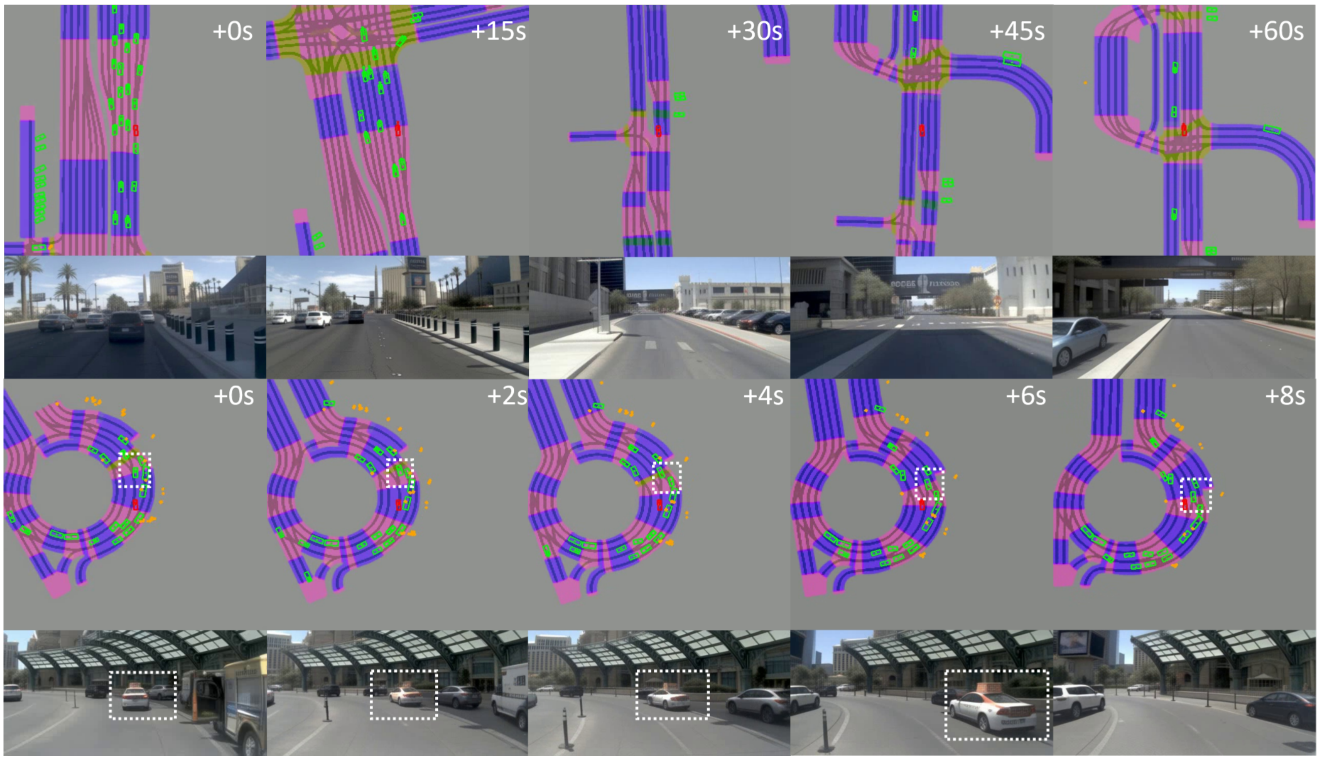


Figure 3. **Generated multimodal driving scenes by UMGen**: The generated scenes evolve continuously from the ego vehicle's perspective. Red Box: ego-vehicle, Green Box: cars, Orange Box: pedestrians or cyclists, Arrow: agent velocities.

number of agents decreases from numerous at the beginning to fewer on narrower roads. Furthermore, the image modality presents the corresponding visualization that reflects these dynamics. This capability of on-the-fly multimodal generation allows our approach to simulate a broader spectrum of driving worlds. The second row of Fig 3 shows 2-second intervals of the ego vehicle passing a hotel entrance. Dashed boxes are used to mark the same car across these frames, showing that its position and movement on the map remain consistent with those in the generated images. This multimodal consistency and temporal continuity provide a solid foundation for generating specified multimodal scenes in the subsequent sections.

Method	Position	Heading	Size	Velocity
TrafficGen[5]	0.1451	0.1325	0.0926	0.1733
LCTGen[26]	0.1319	0.1418	0.1092	0.1938
SceneGen[25]	0.1362	0.1307	0.1190	0.1772
UniGen[16]	0.1208	0.1104	0.0815	0.1591
Ours	**0.0730**	**0.0550**	**0.0533**	**0.1502**

Table 2. **MMD results on Waymo Open Motion Dataset**

Method	Position	Heading	Size	Velocity
TrafficGen*	0.0926	0.0799	0.0856	0.0988
Ours	**0.0828**	**0.0682**	**0.0674**	**0.0760**

Table 3. **MMD results on nuPlan.** $\star$: Results reproduced by us under the same experimental setting

4.3. Initial Scene Generation

The ability to generate realistic initial scenes is paramount for facilitating the generation of driving scene sequences. To evaluate the generated initial scenes, we follow the experimental setting of TrafficGen [5] and evaluate the MMD score on the WOMD [4] and nuPlan [2]. As shown in Table 2 and Table 3, our UMGen achieved lower MMD values compared to other approaches, indicating that it generates more realistic scenes.

4.4. User-Guided Scene Sequence Generation

The ability to generate interactive scene sequences that replicate real-world conditions, along with control over the ego vehicle's movements within these scenes, is essential for validating AD systems. This section, therefore, focuses on user-guided scene generation, highlighting our model's flexibility in adapting to diverse ego-vehicle controls and simulating various traffic interactions.

Interactive ego-vehicle control. To illustrate the model's ability to control ego-vehicle actions, we present visualization from two distinct generated scene sequences. As shown in the first row of Fig.4, we actively control the ego-vehicle to execute either a left turn or a right turn. The map undergoes corresponding rotations and transformations based on the ego vehicle's actions, while the image modality provides a visualization of these changes. Notably, although the scene observed after the right turn does not exist in the dataset, our model generates a corresponding multi-

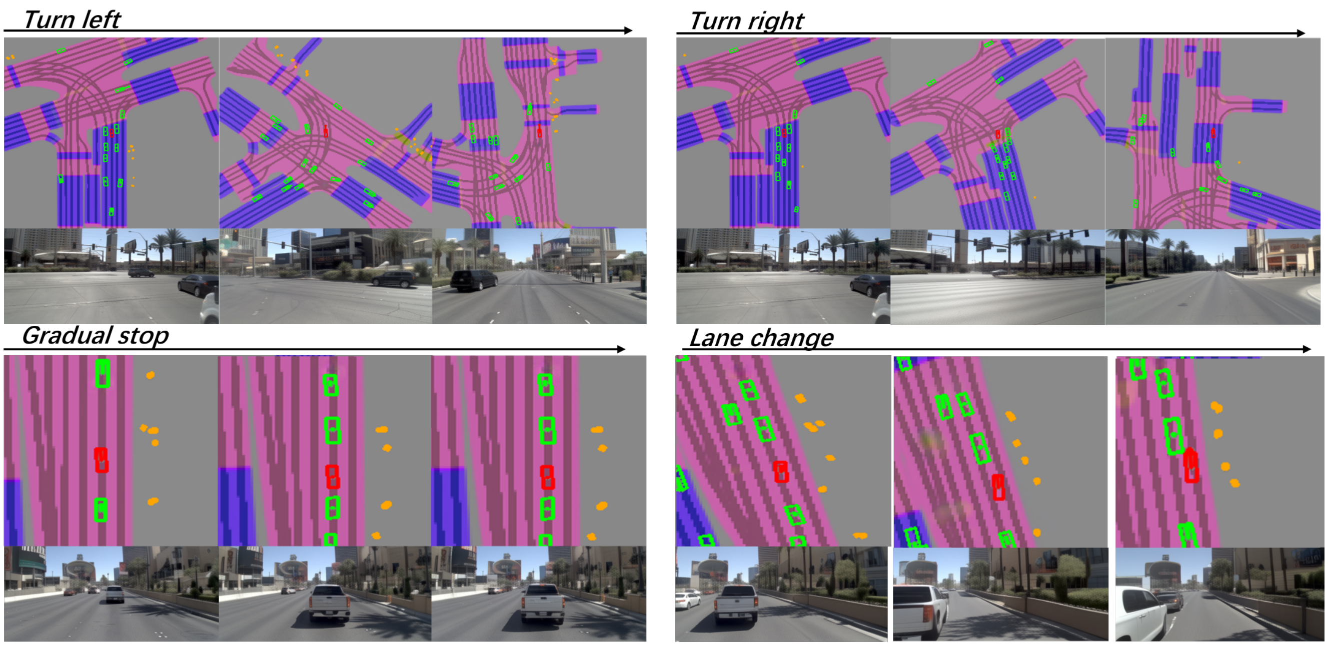


Figure 4. **Generated scenes with input ego actions**. The first row shows the interactive control of the ego vehicle to perform left and right turns. The second row shows the ego vehicle, initialized with a right-turn velocity or a gradual deceleration to a stop. Red box: ego-vehicle, green box: vehicle, orange box: pedestrians or cyclists, arrow: agent velocity.

modal scene. In another scene (second row), instead of following the dataset's "gradual stop" action (left), we assign the ego vehicle a one-frame right-turn velocity. The model autonomously executes maneuvers of lane change and overtaking (right), showcasing its ability to interpret intention, predict maneuvers, and update modalities. Leveraging multimodal generation, we observe the ego vehicle avoids collision with the car ahead despite close proximity—an insight that would be challenging to achieve with models restricted to image-only predictions.

User-specified agent control. UMGen also provides flexible control over other road users, enabling the generation of specific scenes. Moreover, it can simultaneously simulate how these controls impact the behavior of other road users. For example, in the scene shown in Fig.5, we simulate a sudden cut-in maneuver by assigning a forward-left velocity to the agent marked by yellow dashed boxes. In response, the ego-vehicle executes emergency braking (the velocity arrow quickly turns shorter), as shown in the second row, with a trailing agent (highlighted with white dashed boxes) also decelerating in reaction to the ego-vehicle's braking. Compared to the ego's spontaneous braking action, we can also simultaneously control the ego-vehicle to perform a lane change to avoid the collision, as shown in the third row. This shows the flexible controllability of our model. Notably, such agent control through speed settings and realistic interaction simulation are two key features that simplify the creation of specific scenes. Speed control offers a human-like way to adjust agent behavior, while interaction simulation enables agents to respond dynamically to each other. Without these capabilities, generating such situations would require manual calculations of agent positions and interactions between road users.

Method	CR(%)	MMD			
		Posi	Size	Head	Vel
UMGen-T	5.6	0.040	0.043	0.041	0.049
UMGen	**5.1**	**0.038**	**0.037**	**0.040**	**0.036**

Table 4. **MMD and collision rate results.** UMGen-T represents UMGen without OAR module. CR: averaged agent collision rates. Vel: Velocity.

4.5. Ablation study

In this section, we analyze each component of UMGen.

Effectiveness of TAR. Our TAR module applies causal self-attention exclusively to tokens at the same position across historical frames, reducing the time complexity from $O((TN)^2)$ in the vanilla AR model to $O(T^2)$ for modeling temporal dependencies, where T is the number of frames and N is the number of tokens per frame. This design enhances computational efficiency and significantly reduces GPU memory usage. We conducted comparative experiments to validate these benefits using the vanilla AR model and UMGen, monitoring peak GPU memory usage and per-token inference time. As shown in Fig. 6, UMGen maintains stable performance, in contrast to the vanilla AR model, whose GPU memory consumption and inference time increase sharply with the addition of frames, highlighting the challenges of long-horizon scene generation.

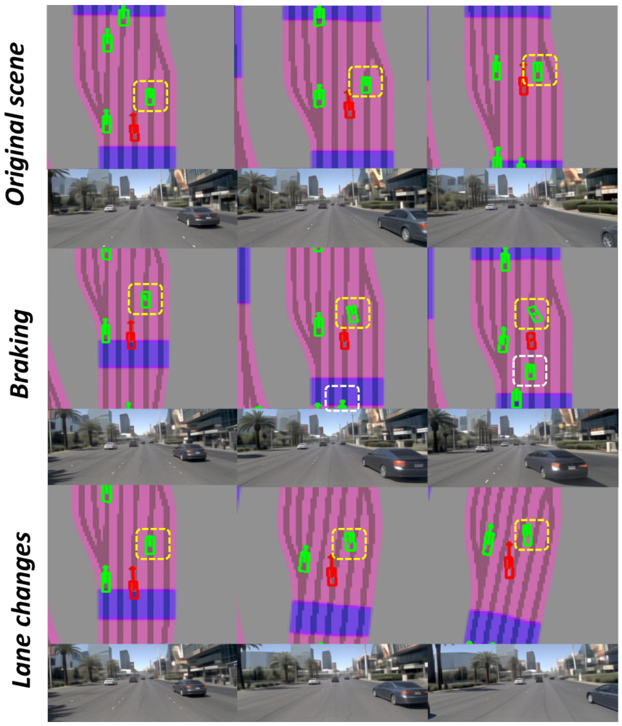


Figure 5. **Customized scenario generation by UMGen**: The first row presents the original scene from the dataset. We assign a forward-left velocity to the vehicle highlighted by the yellow dashed line box and the ego-vehicle spontaneously takes a braking action (second row). Alternatively, we can actively control the ego-vehicle to perform a lane change (third row). Red box: ego-vehicle, green box: vehicle, arrow: agent velocity.

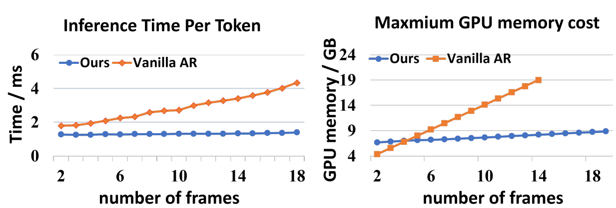


Figure 6. **Comparison of peak GPU memory usage and per-token inference time** for the vanilla AR model and our UMGen.

Decoding without OAR. Our OAR module autoregressively decodes tokens within a frame aiming to prevent inconsistency between tokens. To analyze the impact of this component, we created a variant of UMGen by removing the OAR module, directly decoding from the TAR output. This modified version, referred to as UMGen-T, was evaluated by generating scene sequences based on the initial frames provided by the nuPlan validation set. Performance was measured using the average collision rates between agents and MMD scores on generated scene sequences. As shown in Table 4, UMGen-T exhibits higher MMD values, indicating that its generated scenarios appear less realistic. More importantly, the elevated agent collision rates suggest that UMGen-T struggles to effectively capture intra-frame token relationships, resulting in more frequent conflicts.

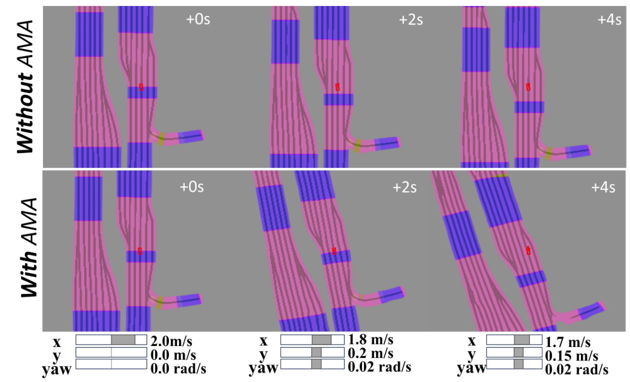


Figure 7. **Map predictions with and without the AMA module**. The bar below illustrates the ego vehicle's velocity over time.

Improved coherence by AMA module. Given the role of the AMA module in maintaining map consistency relative to the ego vehicle's movement, we conducted experiments to assess its impact. We trained an alternative model without this module. As illustrated in Figure 7, we apply a forward-right velocity to the ego-vehicle while it was traveling along a straight road. With the AMA module enabled, the map adjusts accurately to reflect the ego-vehicle's actions. In contrast, without the AMA module, the map reveals a lack of responsiveness. This experiment underscores the importance of the AMA module in enhancing coherence between ego actions and map modalities.

5. Conclusion and Discussion

Conclusion. We have presented UMGen, a generative framework that enhances multimodal driving scene generation by integrating ego-action, road users, traffic maps, and images. Addressing existing limitations, UMGen formulates scene generation as a sequence prediction task, utilizing Temporal AutoRegressive (TAR) and Ordered AutoRegressive (OAR) modules to achieve temporal coherence and cross-modal alignment with reduced computational costs. Additionally, we introduce an Action-aware Map Alignment (AMA) module to ensure consistency between ego-action and map data by dynamically aligning map features with the ego-vehicle's movement. Experimental results have demonstrated the effectiveness of UMGen in generating diverse and realistic scenarios. In addition, it has been experimentally validated that UMGen can generate user-specified scenarios by controlling both the ego vehicle and other vehicles. This further highlights its potential as an interactive closed-loop simulator for AD systems.

Discussion. Building on this, to further improve the quality of the generated images, one promising approach is to incorporate generated multimodal scenes as conditions for a diffusion model [18, 22]. Results demonstrating this approach are provided in the Appendix.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 3, 5, 6

[3] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 3, 4

[4] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. 5, 6

[5] Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3567–3575. IEEE, 2023. 2, 5, 6

[6] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. *arXiv preprint arXiv:2310.02601*, 2023. 2, 3, 5

[7] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023. 2, 3, 4, 5

[8] 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. 2

[9] Yihan Hu, Siqi Chai, Zhening Yang, Jingyu Qian, Kun Li, Wenxin Shao, Haichao Zhang, Wei Xu, and Qiang Liu. Solving motion planning tasks with a scalable generative model. *arXiv preprint arXiv:2407.02797*, 2024. 2, 3, 5

[10] Maximilian Igl, Daewoo Kim, Alex Kuefler, Paul Mougin, Punit Shah, Kyriacos Shiarlis, Dragomir Anguelov, Mark Palatucci, Brandyn White, and Shimon Whiteson. Symphony: Learning realistic and diverse agents for autonomous driving simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2445–2451. IEEE, 2022. 3

[11] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Meta-sim: Learning to generate synthetic datasets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4551–4560, 2019. 3

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[13] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014. 2

[14] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022. 2, 3

[15] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. IEEE, 2018. 3

[16] Reza Mahjourian, Rongbing Mu, Valerii Likhosherstov, Paul Mougin, Xiukun Huang, Joao Messias, and Shimon Whiteson. Unigen: Unified modeling of initial agent states and trajectories for generating autonomous driving scenarios. *arXiv preprint arXiv:2405.03807*, 2024. 3, 5, 6

[17] Jianbiao Mei, Xuemeng Yang, Licheng Wen, Tao Hu, Yu Yang, Tiantian Wei, Yukai Ma, Min Dou, Botian Shi, and Yong Liu. Dreamforge: Motion-aware autoregressive video generation for multi-view driving scenes. *arXiv preprint arXiv:2409.04003*, 2024. 3

[18] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 8

[19] Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[20] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3, 5

[21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 2

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 8

[23] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023. 3

[24] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end au-

tonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024. 2

[25] Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Scenegen: Learning to generate realistic traffic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 892–901, 2021. 6

[26] Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Kraehenbuehl. Language conditioned traffic generation. *arXiv preprint arXiv:2307.07947*, 2023. 3, 5, 6

[27] Junming Wang, Xingyu Zhang, Zebin Xing, Songen Gu, Xiaoyang Guo, Yang Hu, Ziying Song, Qian Zhang, Xiaoxiao Long, and Wei Yin. He-drive: Human-like end-to-end driving with vision language models. *arXiv preprint arXiv:2410.05051*, 2024. 2

[28] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023. 2, 3, 5

[29] Xuemeng Yang, Licheng Wen, Yukai Ma, Jianbiao Mei, Xin Li, Tiantian Wei, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, Botian Shi, Liang He, Yong Liu, and Yu Qiao. Drivearena: A closed-loop generative simulation platform for autonomous driving. *arXiv preprint arXiv:2408.00415*, 2024. 2

[30] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023. 2

[31] Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. *arXiv preprint arXiv:2403.06845*, 2024. 2, 3

[32] Zikang Zhou, Haibo Hu, Xinhong Chen, Jianping Wang, Nan Guan, Kui Wu, Yung-Hui Li, Yu-Kai Huang, and Chun Jason Xue. Behaviorgpt: Smart agent simulation for autonomous driving with next-patch prediction. *arXiv preprint arXiv:2405.17372*, 2024. 3