

# Driving into the Future: Multiview Visual Forecasting and Planning with World Model for Autonomous Driving

Yuqi Wang<sup>1\*</sup> Jiawei He<sup>1\*</sup> Lue Fan<sup>1\*</sup> Hongxin Li<sup>1\*</sup> Yuntao Chen<sup>2✉</sup> Zhaoxiang Zhang<sup>1,2✉</sup>  
<sup>1</sup>CASIA    <sup>2</sup>CAIR, HKISI, CAS

Project Page: <https://drive-wm.github.io>

Code: <https://github.com/BraveGroup/Drive-WM>

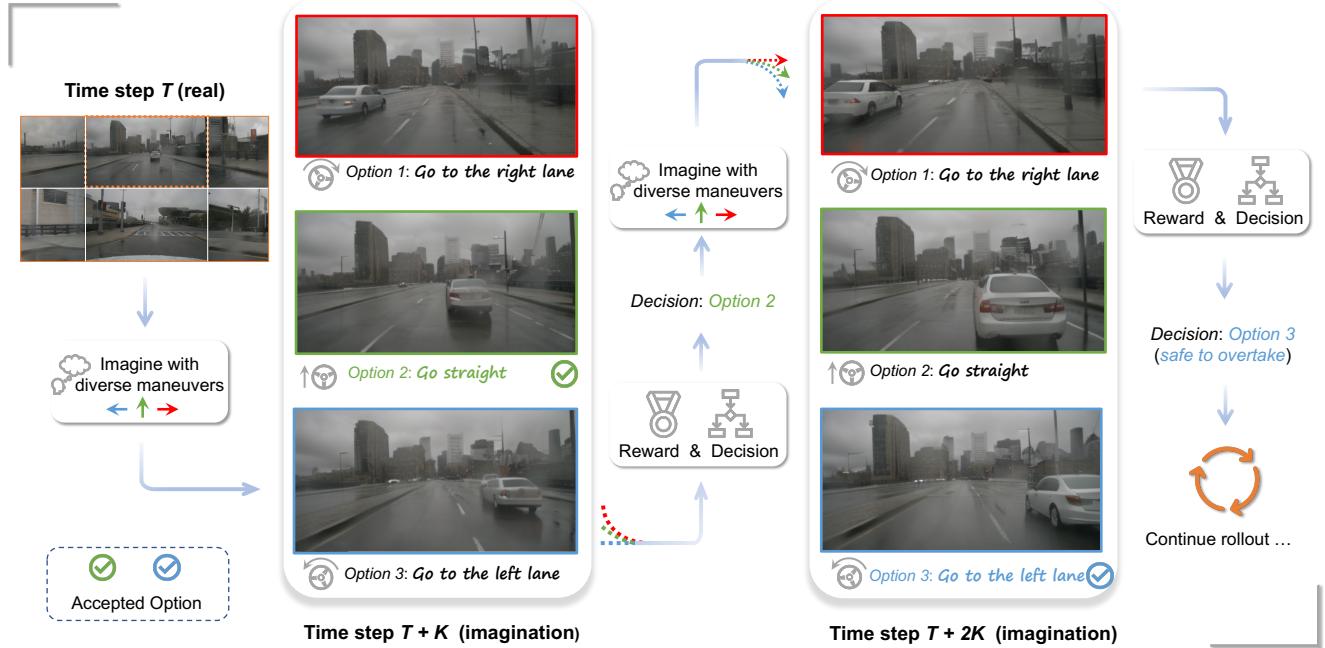


Figure 1. **Multiview visual forecasting and planning by world model.** At time step  $T$ , the world model imagines the multiple futures at  $T + K$ , and finds it is safe to keep going straight at  $T$ . Then the model realizes that the ego car will be too close to the front car according to the imagination of time step  $T + 2K$ , so it decides to change to the left lane for a safe overtaking.

## Abstract

In autonomous driving, predicting future events in advance and evaluating the foreseeable risks empowers autonomous vehicles to better plan their actions, enhancing safety and efficiency on the road. To this end, we propose **Drive-WM**, the first driving world model compatible with existing end-to-end planning models. Through a joint spatial-temporal modeling facilitated by view factorization, our model generates high-fidelity multiview videos in driving scenes. Building on its powerful generation ability, we showcase the potential of applying the world model for safe driving planning for the first time. Particularly, our **Drive-WM** enables driving into multiple futures based on

distinct driving maneuvers, and determines the optimal trajectory according to the image-based rewards. Evaluation on real-world driving datasets verifies that our method could generate high-quality, consistent, and controllable multiview videos, opening up possibilities for real-world simulations and safe planning.

## 1. Introduction

The emergence of end-to-end autonomous driving [29, 30, 54] has recently garnered increasing attention. These approaches take multi-sensor data as input and directly output planning results in a joint model, allowing for joint optimization of all modules. However, it is questionable whether an end-to-end planner trained purely on expert

\*Equal contribution.

driving trajectories has sufficient generalization capabilities when faced with out-of-distribution (OOD) cases. As illustrated in Figure 2, when the ego vehicle’s position deviates laterally from the center line, the end-to-end planner struggles to generate a reasonable trajectory. To alleviate this problem, we propose improving the safety of autonomous driving by developing a predictive model that can foresee planner degradation before decision-making. This model, known as a world model [19, 20, 35], is designed to predict future states based on current states and ego actions. By visually envisioning the future in advance and obtaining feedback from different futures before actual decision-making, it can provide more rational planning, enhancing generalization and safety in end-to-end autonomous driving.

However, learning high-quality world models compatible with existing end-to-end autonomous driving models is challenging, despite successful attempts in game simulations [19–21, 46] and laboratory robotics environments [12, 20]. Specifically, there are three main challenges: (1) The driving world model requires modeling in high-resolution pixel space. The previous low-resolution image [20] or vectorized state space [4] methods cannot effectively represent the numerous fine-grained or non-vectorizable events in the real world. Moreover, vector space world models need extra vector annotations and suffer from state estimation noise of perception models. (2) Generating multiview consistent videos is difficult. Previous and concurrent works are limited to single view video [28, 31, 63] or multiview image generation [17, 53, 69], leaving multiview video generation an open problem for comprehensive environment observation needed in autonomous driving. (3) It is challenging to flexibly accommodate various heterogeneous conditions like changing weather, lighting, ego actions, and road/obstacle/vehicle layouts.

To address these challenges, we propose Drive-WM. Inspired by latent video diffusion models [2, 13, 48, 73], we introduce multiview and temporal modeling for jointly generating multiple views and frames. To further enhance multiview consistency, we propose factorizing the joint modeling to predict intermediate views conditioned on adjacent views, greatly improving consistency between views. We also introduce a simple yet effective unified condition interface enabling flexible use of heterogeneous conditions like images, text, 3D layouts, and actions, greatly simplifying conditional generation. Finally, building on the multiview world model, we explore end-to-end planning applications to enhance autonomous driving safety, as shown in Figure 1. The main contributions of our work can be summarized as follows.

- We propose Drive-WM, a multiview world model, which is capable of generating high-quality, controllable, and consistent *multiview videos* in autonomous driving scenes.



(a) Planning with ego on centerline (b) Planning with ego off centerline

**Figure 2. Ego vehicle’s slight deviation from centerline causes motion planner to struggle generating reasonable trajectories.** We shift the ego location 0.5m to the right to create an out-of-domain case. (a) shows the reasonable trajectory prediction of the VAD [30] method under normal data, and (b) shows the irrational trajectory when encountering out-of-distribution cases.

- Extensive experiments on the nuScenes dataset showcase the leading video quality and controllability. Drive-WM also achieves superior multiview consistency, evaluated by a novel keypoint matching based metric.
- We are the first to explore the potential application of the world model in end-to-end planning for autonomous driving. We experimentally show that our method could enhance the overall soundness of planning and robustness in out-of-distribution situations.

## 2. Related Works

### 2.1. Video Generation and Prediction

Video generation aims to generate realistic video samples. Various generation methods have been proposed in the past, including VAE-based (Variational Autoencoder) [16, 33, 58, 61], GAN-based (generative adversarial networks) [3, 15, 31, 49, 56, 70], flow-based [10, 34] and auto-regressive models [18, 64, 68]. Notably, the recent success of diffusion-based models in the realm of image generation [40, 44, 45] has ignited growing interest in applying diffusion models to the realm of video generation [23, 26]. Diffusion-based methods have yielded significant enhancements in realism, controllability, and temporal consistency. Text-conditional video generation has garnered more attention due to its controllable generation, and a plethora of methods have emerged [2, 25, 48, 66, 73].

Video prediction can be regarded as a special form of generation, leveraging past observations to anticipate future frames [1, 9, 22, 26, 41, 59, 60, 65]. Especially in autonomous driving, DriveGAN [31] learns to simulate a driving scenario with vehicle control signals as its input. GAIA-1 [28] and DriveDreamer [63] further extend to action-conditional diffusion models, enhancing the controllability and realism of generated videos. However, these previous works are limited to monocular videos and fail to comprehend the overall 3D surroundings. We have pioneered the generation of multiview videos, allowing for better integration with current BEV perception and planning models.

## 2.2. World Model for Planning

The world model [35] learns a general representation of the world and predicts future world states resulting from a sequence of actions. Learning world models in either game [19–21, 43, 47] or lab environments [12, 14, 67] has been widely studied. Dreamer [20] learns a latent dynamics model from past experience to predict state values and actions in a latent space. It is capable of handling challenging visual control tasks in the DeepMind Control Suite [55]. DreamerV2 [21] improves upon Dreamer to achieve human-level performance on Atari games. DreamerV3 [22] uses larger networks and learns to obtain diamonds in Minecraft from scratch given sparse rewards, which is considered a long-standing challenge. Day-Dreamer [67] applies Dreamer [20] to training 4 robots online in the real world and solves locomotion and manipulation tasks without changing hyperparameters. Recently, learning world models in driving scenes has gained attention. MILE [27] employs a model-based imitation learning method to jointly learn a dynamics model and driving behaviour in CARLA [11]. The aforementioned works are limited to either simulators or well-controlled lab environments. In contrast, our world model can be integrated with existing end-to-end driving planners to improve planning performance in real-world scenes.

## 3. Multi-view Video Generation

In this section, we first present how to jointly model the multiple views and frames, which is presented in Sec 3.1. Then we enhance multiview consistency by factorizing the joint modeling in Sec 3.2. Finally, Sec. 3.3 elaborates on how we build a unified condition interface to integrate the multiple heterogeneous conditions.

### 3.1. Joint Modeling of Multiview Video

To jointly model multiview temporal data, we start with the well-studied image diffusion model and adapt it into multiview-temporal scenarios by introducing additional temporal layers and multiview layers. In this subsection, we first present the overall formulation of joint modeling and elaborate on the temporal and multiview layers.

**Formulation.** We assume access to a dataset  $p_{\text{data}}$  of multiview videos, such that  $\mathbf{x} \in \mathbb{R}^{T \times K \times 3 \times H \times W}$ ,  $\mathbf{x} \sim p_{\text{data}}$  is a sequence of  $T$  images with  $K$  views, with height and width  $H$  and  $W$ . Given encoded video latent representation  $\mathcal{E}(\mathbf{x}) = \mathbf{z} \in \mathbb{R}^{T \cdot K \times C \times \hat{H} \times \hat{W}}$ , diffused inputs  $\mathbf{z}_\tau = \alpha_\tau \mathbf{z} + \sigma_\tau \epsilon$ ,  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , here  $\alpha_\tau$  and  $\sigma_\tau$  define a noise schedule parameterized by a diffusion time step  $\tau$ . A denoising model  $\mathbf{f}_{\theta, \phi, \psi}$  (parameterized by spatial parameters  $\theta$ , temporal parameters  $\phi$  and multiview parameters  $\psi$ )

receives the diffused  $\mathbf{z}_\tau$  as input and is optimized by minimizing the denoising score matching objective

$$\mathbb{E}_{\mathbf{z} \sim p_{\text{data}}, \tau \sim p_\tau, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\mathbf{y} - \mathbf{f}_{\theta, \phi, \psi}(\mathbf{z}_\tau; \mathbf{c}, \tau)\|_2^2], \quad (1)$$

where  $\mathbf{c}$  is the condition, and target  $\mathbf{y}$  is the random noise  $\epsilon$ .  $p_\tau$  is a uniform distribution over the diffusion time  $\tau$ .

**Temporal encoding layers.** We first introduce *temporal layers* to lift the pretrained image diffusion model into a temporal model. The temporal encoding layer is attached after the 2D spatial layer in each block, following established practice in VideoLDM [2]. The spatial layer encodes the latent  $\mathbf{z} \in \mathbb{R}^{T \cdot K \times C \times \hat{H} \times \hat{W}}$  in a frame-wise and view-wise manner. Afterward, we rearrange the latent to hold out the temporal dimension, denoted as  $(TK) \text{CHW} \rightarrow KCTHW$ , to apply the 3D convolution in spatio-temporal dimensions  $THW$ . Then we arrange the latent to  $(KHW) \text{TC}$  and apply standard multi-head self-attention to the temporal dimension, enhancing the temporal dependency. The notation  $\phi$  in Eq. 1 stands for the parameters of this part.

**Multiview encoding layers.** To jointly model the multiple views, there must be information exchange between different views. Thus we lift the single-view temporal model to a multi-view temporal model by introducing *multiview encoding layers*. In particular, we rearrange the latent as  $(KHW) \text{TC} \rightarrow (THW) \text{KC}$  to hold out the view dimension. Then a self-attention layer parameterized by  $\psi$  in Eq. 1 is employed across the view dimension. Such multiview attention allows all views to possess similar styles and consistent overall structure.

**Multiview temporal tuning.** Given the powerful image diffusion models, we do not train the temporal multiview network from scratch. Instead, we first train a standard image diffusion model with single-view image data and conditions, which corresponds to the parameter  $\theta$  in Eq. 1. Then we freeze the parameters  $\theta$  and fine-tune the additional temporal layers ( $\phi$ ) and multiview layers ( $\psi$ ) with video data.

### 3.2. Factorization of Joint Multiview Modeling

Although the joint distributions in Sec. 3.1 could yield similar styles between different views, it is hard to ensure strict consistency in their overlapped regions. In this subsection, we introduce the distribution factorization to enhance multiview consistency. We first present the formulation of factorization and then describe how it cooperates with the aforementioned joint modeling.

**Formulation.** Let  $\mathbf{x}_i$  denote the sample of  $i$ -th view, Sec. 3.1 essentially model the joint distribution  $p(\mathbf{x}_1, \dots, K)$ , which can be into

$$p(\mathbf{x}_1, \dots, K) = p(\mathbf{x}_1)p(\mathbf{x}_2 | \mathbf{x}_1) \dots p(\mathbf{x}_K | \mathbf{x}_1, \dots, \mathbf{x}_{K-1}). \quad (2)$$

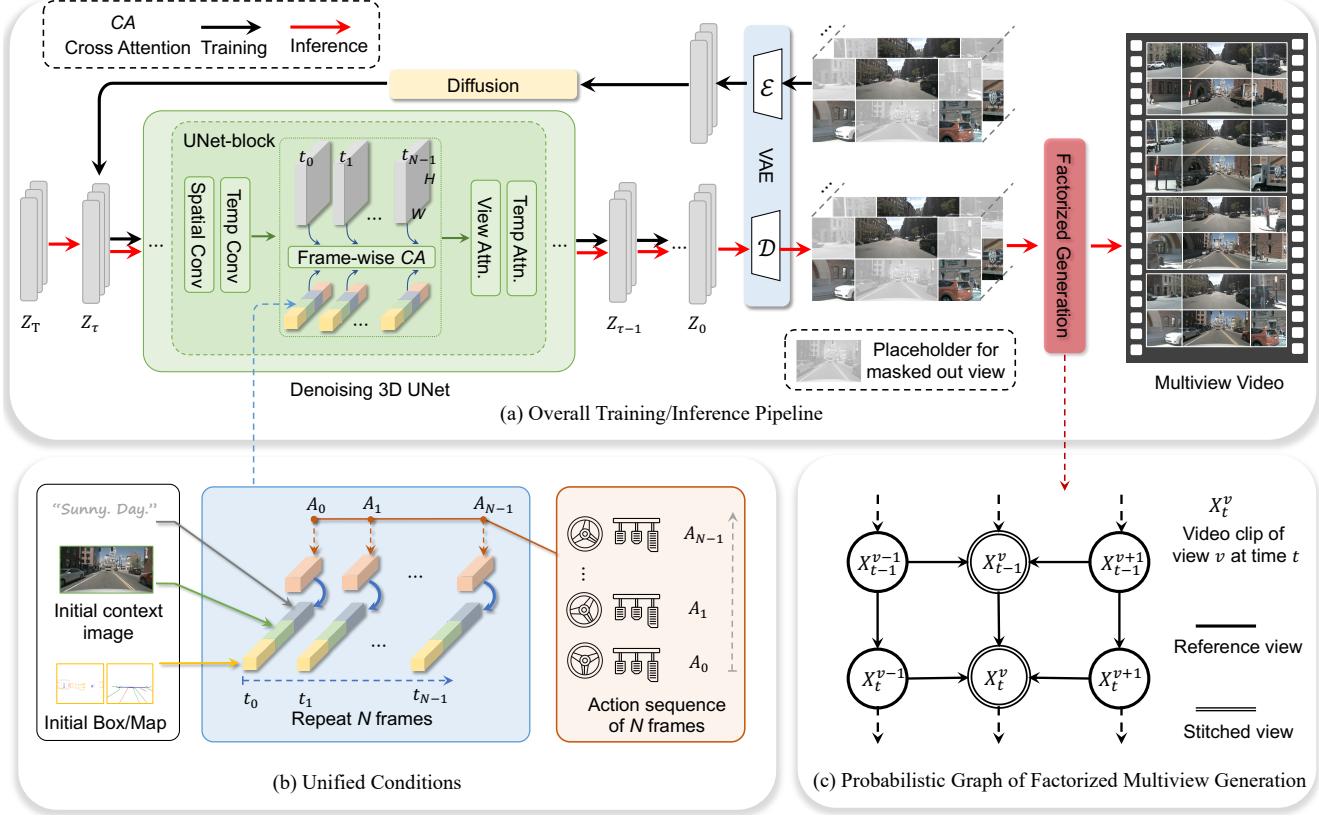


Figure 3. **Overview of the proposed framework.** (a) illustrates the training and inference pipeline of the proposed method. (b) visualizes the unified conditions leveraged to control the generation of multi-view video. (c) represents the probabilistic graph of factorized multiview generation. It takes the 3-view output from (a) as input to generate other views, enhancing the multi-view consistency.

Eq. 2 indicates that different views are generated in an autoregressive manner, where a new view is conditioned on existing views. These conditional distributions can ensure better view consistency because new views are aware of the content in existing views. However, such an autoregressive generation is inefficient, making such full factorization infeasible in practice.

To simplify the modeling in Eq. 2, we partition all views into two types: *reference views*  $\mathbf{x}_r$  and *stitched views*  $\mathbf{x}_s$ . For example, in nuScenes, reference views can be the {F, BL, BR}<sup>2</sup>, and stitched views can be {FL, B, FR}. We use the term “stitched” because a stitched view appears to be “stitched” from its two neighboring reference views. Views belonging to the same type do not overlap with each other, while different types of views may overlap. This inspires us to first model the joint distribution of reference views. Here the joint modeling is effective for those non-overlapped reference views since they do not necessitate strict consistency. Then the distribution of  $\mathbf{x}_s$  is modeled as a conditional distribution conditioned on the  $\mathbf{x}_r$ . Figure 4 illustrates the basic concept of multiview factorization in nuScenes. In this sense, we

<sup>2</sup>F: front, B: back, L: left, R: right.

simplify Eq. 2 into

$$p(\mathbf{x}) = p(\mathbf{x}_s, \mathbf{x}_r) = p(\mathbf{x}_r)p(\mathbf{x}_s|\mathbf{x}_r). \quad (3)$$

Considering the temporal coherence, we incorporate previous frames as additional conditions. The Eq. 3 can be re-written as

$$p(\mathbf{x}) = p(\mathbf{x}_s, \mathbf{x}_r | \mathbf{x}_{pre}) = p(\mathbf{x}_r | \mathbf{x}_{pre})p(\mathbf{x}_s | \mathbf{x}_r, \mathbf{x}_{pre}), \quad (4)$$

where  $\mathbf{x}_{pre}$  is context frames (e.g., the last two frames) from previously generated video clips. The distribution of reference views  $p(\mathbf{x}_r | \mathbf{x}_{pre})$  is implemented by the pipeline in Sec. 3.1. As for  $p(\mathbf{x}_s | \mathbf{x}_r, \mathbf{x}_{pre})$ , we adopt the similar pipeline but incorporate neighboring reference views as an additional condition as Figure 4 shows. We introduce how to use conditions in the following subsection.

### 3.3. Unified Conditional Generation

Due to the great complexity of the real world, the world model needs to leverage multiple heterogeneous conditions. In our case, we utilize initial context frames, text descriptions, ego actions, 3D boxes, BEV maps, and reference views. More conditions can be further included for better

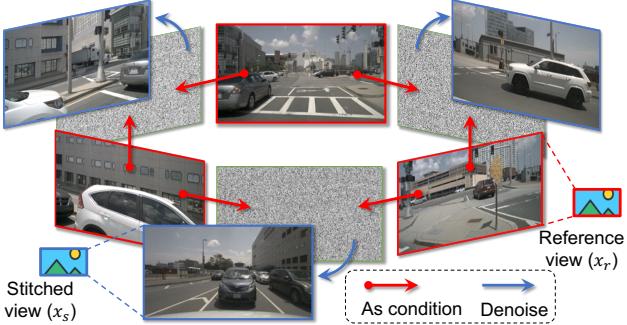


Figure 4. **Illustration of factorized multi-view generation.** We take the sensor layout in nuScenes as an example.

controllability. Developing specialized interface for each one is time-consuming and inflexible to incorporate more conditions. To address this issue, we introduce a unified condition interface, which is simple yet effective in integrating multiple heterogeneous conditions. In the following, we first introduce how we encode each condition, and then describe the unified condition interface.

**Image condition.** We treat initial context frames (i.e., the first frame of a clip) and reference views as image conditions. A given image condition  $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$  is encoded and flattened to a sequence of  $d$ -dimension embeddings  $\mathbf{i} = (i_1, i_2, \dots, i_n) \in \mathbb{R}^{n \times d}$ , using ConvNeXt as encoder [39]. Embeddings from different images are concatenated in the first dimension of  $n$ .

**Layout condition.** Layout condition refers to 3D boxes, HD maps, and BEV segmentation. For simplicity, we project the 3D boxes and HD maps into a 2D perspective view. In this way, we leverage the same strategy with image condition encoding to encode the layout condition, resulting in a sequence of embeddings  $\mathbf{l} = (l_1, l_2, \dots, l_k) \in \mathbb{R}^{k \times d}$ .  $k$  is the total number of embeddings from the projected layouts and BEV segmentation.

**Text condition.** We follow the convention of diffusion models to adopt a pre-trained CLIP [42] as the text encoder. Specifically, we combine view information, weather, and light to derive a text description. The embeddings are denoted as  $\mathbf{e} = (e_1, e_2, \dots, e_m) \in \mathbb{R}^{m \times d}$ .

**Action condition.** Action conditions are indispensable for the world model to generate the future. To be compatible with the existing planning methods [30], we define the action in a time step as  $(\Delta x, \Delta y)$ , which represents the movement of ego location to the next time step. We use an MLP to map the action into a  $d$ -dimension embedding  $\mathbf{a} \in \mathbb{R}^{2 \times d}$ .

**A unified condition interface.** So far, all the conditions are mapped into  $d$ -dimension feature space. We take the concatenation of required embeddings as input for the denoising UNet. Taking action-based joint video generation as an example, this allows us to utilize the initial context

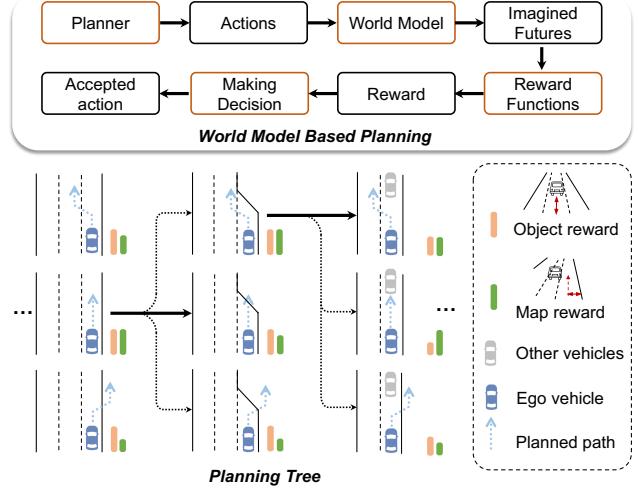


Figure 5. **End-to-end planning pipeline with our world model.** We display the components of our planning pipeline at the top and illustrate the decision-making process in the planning tree using image-based rewards at the bottom.

images, initial layout, text description, and frame-wise action sequence. So we have unified condition embeddings in a certain time  $t$  as

$$\mathbf{c}_t = [\mathbf{i}_0, \mathbf{l}_0, \mathbf{e}_0, \mathbf{a}_t] \in \mathbb{R}^{(n+k+m+2) \times d}, \quad (5)$$

where subscript  $t$  stands for the  $t$ -th generated frame and subscript 0 stands for the current *real* frame. We emphasize that such a combination of different conditions offers a unified interface and can be adjusted by the request. Finally,  $\mathbf{c}_t$  interacts with the latent  $\mathbf{z}_t$  in 3D UNet by cross attention in a frame-wise manner (Figure 3 (a)).

## 4. World Model for End-to-End Planning

Blindly planning actions without anticipating consequences is dangerous. Leveraging our world model enables comprehensive evaluation of possible futures for safer planning. In this section, we explore end-to-end planning using the world model for autonomous driving, an uncharted area.

### 4.1. Tree-based Rollout with Actions

We describe planning with world models in this section. At each time step, we leverage the world model to generate predicted future scenarios for trajectory candidates sampled from the planner, evaluate the futures using an image-based reward function, and select the optimal trajectory to extend the planning tree.

As shown in Figure 5, we define the planning tree as a series of predicted ego trajectories that evolve over time. For each time, the real multiview images can be captured by the camera. The pre-trained planner takes the real multiview images as input and samples possible trajectory candidates. To be compatible with the input of mainstream planner, we define its action  $\mathbf{a}_t$  at time  $t$  as  $(x_{t+1} - x_t, y_{t+1} - y_t)$  for

each trajectory, where  $x_t$  and  $y_t$  are the ego locations at time  $t$ . Given the actions, we adopt the condition combination in Eq. 5 for the video generation. After the generation, we leverage an image-based reward function to choose the optimal trajectory as the decision. Such a generation-decision process can be repeated to form a tree-based rollout.

## 4.2. Image-based Reward Function

After generating the future videos for planned trajectories, reward functions are required to evaluate the soundness of the multiple futures.

We first get the rewards from perception results. Particularly, we utilize image-based 3D object detector [37] and online HDMap predictor [38] to obtain the perception results on the generated videos. Then we define map reward and object reward, inspired by traditional planner [6, 30]. The map reward includes two factors, distance away from the curb, encouraging the ego vehicle to stay in the correct drivable area, and centerline consistency, preventing ego from frequently changing lanes and deviating from the lane in the lateral direction. The object reward means the distance away from other road users in longitudinal and lateral directions. This reward avoids the collision between the ego vehicle and other road users. The total reward is defined as the product of the object reward and the map reward. We finally select the ego prediction with the maximum reward. Then the planning tree forwards to the next timestamp and plans the subsequent trajectory iteratively.

Since the proposed world model operates in pixel space, it can further get rewards from the *non-vectorized* representation to handle more general cases. For example, the sprayed water from the sprinkler and damaged road surface are hard to be vectorized by the supervised perception models, while the world model trained from massive unlabeled data could generate such cases in pixel space. Leveraging the recent powerful foundational models such as GPT-4V, the planning process can get more comprehensive rewards from the non-vectorized representation. In the appendix, we showcase some typical examples.

## 5. Experiments

### 5.1. Setup

**Dataset.** We adopt the nuScenes [5] dataset for experiments, which is one of the most popular datasets for 3D perception and planning. It comprises a total of 700 training videos and 150 validation videos. Each video includes around 20 seconds captured by six surround-view cameras.

**Training scheme.** We crop and resize the original image from  $1600 \times 900$  to  $384 \times 192$ . Our model is initialized with Stable Diffusion checkpoints [44]. All experiments are conducted on A40 (48GB) GPUs. For additional details, please refer to the appendix.

**Quality evaluation.** To evaluate the quality of the generated video, we utilize FID (Frechet Inception Distance) [24] and FVD (Frechet Video Distance) [57] as the main metrics.

**Multiview consistency evaluation.** We introduced a novel metric, the Key Points Matching (KPM) score, to evaluate multi-view consistency. This metric utilizes a pre-trained matching model [51] to calculate the average number of matching key points, thereby quantifying the KPM score. In the calculation process, for each image, we first compute the number of matched key points between the current view and its two adjacent views. Subsequently, we calculated the ratio between the number of matched points in generated data and the number of matched points in real data. Finally, we averaged these ratios across all generated images to obtain the KPM score. In practice, we uniformly selected 8 frames per scene in the validation set to calculate KPM.

**Controllability evaluation.** To assess the controllability of video content generation, we evaluate generated images using pre-trained perception models. Following the previous method [69], we adopted CVT [72] for foreground and background segmentation. Additionally, we evaluate 3D object detection [62] and online map construction [38].

**Planning evaluation.** We follow open-loop evaluation metrics [29, 30] for end-to-end planning, including L2 distance from the GT trajectory and the object collision rate.

**Model variants.** We support action-based video generation and layout-based video generation. The former gives the ego action of each frame as the condition, while the latter gives the layout (3D box, map information) of each frame.

### 5.2. Main Results of Multi-view Video Generation

We first demonstrate our superior generation quality and controllability. Here the generation is conditioned on frame-wise 3D layouts. Our model is trained in nuScenes *train* split, and evaluated with the conditions in *val* split.

**Generation quality.** Since we are the first one to explore multi-view video generation, we make separate comparisons with previous methods in *multi-view images* and *single-view videos*, respectively. For multi-view image generation, we remove the temporal layers in Sec. 3.1. Table 1a showcases the main results. In single-view image generation, we achieve **12.99** FID, achieving a significant improvement over previous methods. For video generation, our method exhibits a significant quality improvement compared to past single-view video generation methods, achieving **15.8** FID and **122.7** FVD. Additionally, our method is the first work to generate consistent multi-view videos, which is quantitatively demonstrated in Sec. 5.3.

**Controllability.** In Table 1b, we examine the controllability of our method on the nuScenes validation set. For foreground controllability, we evaluate the performance of 3D

Method	Multi-view	Video	FID↓	FVD↓
BEVGen [53]	✓		25.54	-
BEVControl [69]	✓		24.85	-
MagicDrive [17]	✓		16.20	-
Ours	✓		<b>12.99</b>	-
DriveGAN [31]		✓	73.4	502.3
DriveDreamer [63]		✓	52.6	452.0
Ours	✓	✓	<b>15.8</b>	<b>122.7</b>

(a) Generation quality.

Method	mAP <sub>obj</sub> ↑	mAP <sub>map</sub> ↑	mIoU <sub>fg</sub> ↑	mIoU <sub>bg</sub> ↑
GT	37.78	59.30	36.08	72.36
BEVGen [53]	-	-	5.89	50.20
LayoutDiffusion [71]	3.68	-	15.51	35.31
GLIGEN [36]	15.42	-	22.02	38.12
BEVControl [69]	19.64	-	26.80	60.80
MagicDrive [17]	12.30	-	27.01	61.05
Ours	<b>20.66</b>	<b>37.68</b>	<b>27.19</b>	<b>65.07</b>

(b) Generation controllability.

Table 1. **Multi-view video generation performance on nuScenes.** For each task, we test the corresponding models trained on the nuScenes training set. Our Drive-WM surpasses all other methods in both quality and controllability evaluation.

Temp emb.	Layout Cond.	FID↓	FVD↓	KPM(%)↑
✓		20.3	212.5	31.5
	✓	18.9	153.8	44.6
✓	✓	<b>15.8</b>	<b>122.7</b>	<b>45.8</b>

(a) Ablations of unified condition.

Temp Layers	View Layers	FID↓	FVD↓	KPM(%)↑
		23.3	228.5	40.8
	✓	16.2	127.1	40.9
✓	✓	<b>15.8</b>	<b>122.7</b>	<b>45.8</b>

(b) Ablations of multiview temporal tuning.

Method	KPM(%)↑	FVD↓	FID↓
Joint Modeling	45.8	122.7	<b>15.8</b>
Factorized Generation	<b>94.4</b>	<b>116.6</b>	16.4

(c) Ablations of factorized generation.

Table 2. **Ablations of the components in model design.** The experiments are conducted under the layout-based video generation (See model variants in Sec. 5.1) from nuScenes validation set.

object detection on the generated multiview videos, reporting mAP<sub>obj</sub>. Additionally, we segment the foreground on the BEV layouts, reporting mIoU<sub>fg</sub>. Regarding background control, we report the mIoU of road segmentation. Furthermore, we evaluate mAP<sub>map</sub> for HDMap performance. This superior controllability highlights the effectiveness of the unified condition interface (Sec. 3.3) and demonstrates the potential of the world model as a neural simulator.

### 5.3. Ablation Study for Multiview Video Generation

To validate the effectiveness of our design decisions, we conduct ablation studies on the key features of the model, as illustrated in Table 2. The experiments are conducted under layout-based video generation.

**Unified condition.** In Table 2a, we find that the layout condition has a significant impact on the model’s ability, improving both the quality and consistency of the generated videos. Additionally, temporal embedding can enhance the quality of the generated videos.

**Model design.** In Table 2b, we explore the role of the temporal and view layers in multiview temporal tuning. The experiment shows that the temporal and view layers simply adopting the multiview layer without factorization (Sec. 3.2) slightly improve the KPM.

**Factorized multiview generation.** As indicated in Table 2c, factorized generation notably improves the consistency among multiple views, increasing from 45.8% to 94.4%, in contrast to joint modeling. This enhancement is achieved while ensuring the quality of both images and videos. Qualitative results are illustrated in Figure 6.

### 5.4. Exploring Planning with World Model

In this subsection, we explore the application of the world model in end-to-end planning, which is under-explored in recent works for autonomous driving. Our attempts lie in two aspects. (1) We first demonstrate that evaluating the generated futures is helpful in planning. (2) Then we showcase that the world model can be leveraged to improve planning in some out-of-domain cases.

Method	L2 (m) ↓				Collision (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
VAD (GT cmd)	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22
VAD (rand cmd)	0.51	0.97	1.57	1.02	0.34	0.74	1.72	0.93
Ours	<b>0.43</b>	<b>0.77</b>	<b>1.20</b>	<b>0.80</b>	<b>0.10</b>	<b>0.21</b>	<b>0.48</b>	<b>0.26</b>

Table 3. **Planning performance on nuScenes.** Instead of using the ground truth driving command, we use our tree-based planning to select the best out of three commands.

Map Reward	Object Reward	L2 (m) ↓				Collision (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
✓		0.51	0.97	1.57	1.02	0.34	0.74	1.72	0.93
	✓	0.45	0.82	1.29	0.85	0.12	0.33	0.72	0.39
	✓	0.43	0.77	1.20	0.80	0.12	0.21	0.48	0.27
✓	✓	<b>0.43</b>	<b>0.77</b>	<b>1.20</b>	<b>0.80</b>	<b>0.10</b>	<b>0.21</b>	<b>0.48</b>	<b>0.26</b>

Table 4. **Image-based reward function design.** We use two sub-rewards, map reward and object reward.

**Tree-based planning.** We conduct the experiments to show the performance of our tree-based planning. Instead

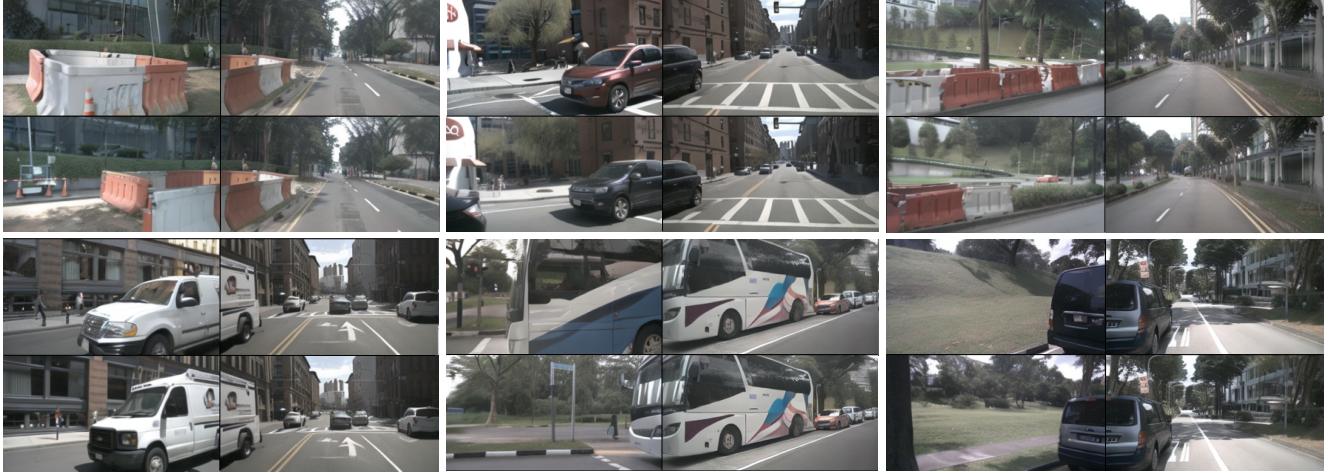


Figure 6. **Qualitative results of factorized multiview generation.** For each compared pair, the upper row is generated without factorization, and the lower row is generated with factorization.



Figure 7. **Counterfactual events generation.** Top: turning around at a T-shape intersection on a rainy day. Note that our training set does not contain any turning-around samples. Bottom: running over a non-drivable area.

of using the ground truth driving command, we sample planned trajectories from VAD according to the three commands “Go straight”, “Turn left”, and “Turn right”. Then the sampled actions are used for our tree-based planning (Sec. 4). As shown in Table 3, our tree-based planner outperforms random driving commands and even achieves performance close to the GT command. Besides, in Table 4, we ablate two adopted rewards and the results indicate that the combined reward outperforms each sub-reward, particularly in terms of the object collision metric.

OOD	World Model f.t.	L2 (m) ↓				Collision (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
✓		0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22
✓	✓	0.73	0.99	1.33	1.02	1.25	1.62	1.91	1.59
		0.50	0.79	1.17	0.82	0.72	0.84	1.16	0.91

Table 5. **Out-of-domain planning.** We define OOD location with a lateral deviation of 0.5 meters from the ego vehicle.

**Recovery from OOD ego deviation.** Using our world model, we can simulate the out-of-domain ego locations in pixel space. In particular, we shift the ego location laterally by 0.5 meters, like the right one in Figure 2. In this

situation, the performance of the existing end-to-end planner VAD [30] undergoes a significant decrease, shown in Table 5. To alleviate the problem, we fine-tune the planner with generated video supervised by the trajectory that the ego-vehicle drives back to the lane. Learning from these OOD data, the performance of the planner can be better and near normal levels.

## 5.5. Counterfactual Events

Given an initial observation and the action, our Drive-WM can generate counterfactual events, such as turning around and running over non-drivable areas (Figure 7), which are significantly different from the training data. The ability to generate such counterfactual data reveals again that our Drive-WM has the potential to foresee and handle out-of-domain cases.

## 6. Conclusion

We introduce Drive-WM, the first multiview world model for autonomous driving. Our method exhibits the capability to generate high-quality and consistent multiview videos under diverse conditions, leveraging information from textual descriptors, layouts, or ego actions to control video

generation. The introduced factorized generation significantly enhances spatial consistency across various views. Besides, extensive experiments on nuScenes dataset show that our method could enhance the overall soundness of planning and robustness in out-of-distribution situations.

## References

- [1] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021. [2](#)
- [2] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, pages 22563–22575, 2023. [2, 3, 14, 17](#)
- [3] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei Efros, and Tero Karras. Generating long videos of dynamic scenes. *NeurIPS*, 35:31769–31781, 2022. [2](#)
- [4] Lars Buesing, Theophane Weber, Sébastien Racaniere, SM Eslami, Danilo Rezende, David P Reichert, Fabio Viola, Frederic Besse, Karol Gregor, Demis Hassabis, et al. Learning and querying fast generative models for reinforcement learning. *arXiv preprint arXiv:1802.03006*, 2018. [2](#)
- [5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Lioung, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. [6, 12](#)
- [6] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *CVPR*, pages 14403–14412, 2021. [6](#)
- [7] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, and Junchi Yan. Performer: 3d lane detection via perspective transformer and the openlane benchmark. In *ECCV*, 2022. [16](#)
- [8] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *ICCV*, pages 9329–9338, 2019. [12](#)
- [9] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *ICML*, pages 1174–1183. PMLR, 2018. [2](#)
- [10] Michael Dorkenwald, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G Derpanis, and Bjorn Ommer. Stochastic image-to-video synthesis using cinns. In *CVPR*, pages 3742–3753, 2021. [2](#)
- [11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *CoRL*, pages 1–16. PMLR, 2017. [3](#)
- [12] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018. [2, 3](#)
- [13] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *ICCV*, pages 7346–7356, 2023. [2](#)
- [14] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. *ICRA*, pages 2786–2793, 2016. [3](#)
- [15] Geron Fox, Ayush Tewari, Mohamed Elgarib, and Christian Theobalt. Stylevideogan: A temporal generative model using a pretrained stylegan. *arXiv preprint arXiv:2107.07224*, 2021. [2](#)
- [16] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *ICML*, pages 3233–3246. PMLR, 2020. [2](#)
- [17] 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, 7](#)
- [18] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *ECCV*, pages 102–118. Springer, 2022. [2](#)
- [19] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *NeurIPS*, 31, 2018. [2, 3](#)
- [20] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *ICLR*, 2020. [2, 3](#)
- [21] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *ICLR*, 2021. [2, 3](#)
- [22] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. [2, 3](#)
- [23] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *NeurIPS*, 35:27953–27965, 2022. [2](#)
- [24] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. [6](#)
- [25] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. [2](#)
- [26] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *TMLR*, 2022. [2](#)
- [27] Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zachary Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. *NeurIPS*, 35:20703–20716, 2022. [3](#)
- [28] 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](#)

- [29] Yihan Hu, Jiazh Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, pages 17853–17862, 2023. 1, 6
- [30] 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 *ICCV*, pages 8340–8350, 2023. 1, 2, 5, 6, 8, 12, 16
- [31] Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation. In *CVPR*, pages 5820–5829, 2021. 2, 7
- [32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 14
- [33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2
- [34] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. In *ICLR*, 2020. 2
- [35] Yann LeCun. A path towards autonomous machine intelligence. 2022. 2, 3
- [36] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, pages 22511–22521, 2023. 7
- [37] 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 *ECCV*, pages 1–18. Springer, 2022. 6
- [38] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. In *ICLR*, 2022. 6
- [39] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 5
- [40] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, pages 16784–16804, 2022. 2
- [41] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. *NeurIPS*, 28, 2015. 2
- [42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 5
- [43] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *TMLR*, 2022. 3
- [44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2, 6, 14
- [45] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, pages 22500–22510, 2023. 2
- [46] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. 2
- [47] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *ICML*, pages 8583–8592, 2020. 3
- [48] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2022. 2
- [49] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhosiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*, pages 3626–3636, 2022. 2
- [50] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2020. 14
- [51] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 6, 18, 20
- [52] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yunling Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 16
- [53] Alexander Swerdfog, Runsheng Xu, and Bolei Zhou. Street-view image generation from a bird’s-eye view layout. *arXiv preprint arXiv:2301.04634*, 2023. 2, 7, 12
- [54] Ardi Tamppuu, Tambe Matiisen, Maksym Semikin, Dmytro Fishman, and Naveed Muhammad. A survey of end-to-end driving: Architectures and training methods. *IEEE Trans. Neural Netw. Learn. Syst.*, 33(4):1364–1384, 2020. 1
- [55] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018. 3
- [56] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, pages 1526–1535, 2018. 2
- [57] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 6
- [58] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. *NeurIPS*, 32, 2019. 2

- [59] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *NeurIPS*, 35:23371–23385, 2022. 2
- [60] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *NeurIPS*, 29, 2016. 2
- [61] Jacob Walker, Ali Razavi, and Aäron van den Oord. Predicting video with vqvae. *arXiv preprint arXiv:2103.01950*, 2021. 2
- [62] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *ICCV*, pages 3621–3631, 2023. 6
- [63] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023. 2, 7
- [64] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*, 2020. 2
- [65] Bohan Wu, Suraj Nair, Roberto Martin-Martin, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *CVPR*, pages 2318–2328, 2021. 2
- [66] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, pages 7623–7633, 2023. 2
- [67] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *CoRL*, pages 2226–2240. PMLR, 2023. 3
- [68] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 2
- [69] Kairui Yang, Enhui Ma, Jibin Peng, Qing Guo, Di Lin, and Kaicheng Yu. Bevcontrol: Accurately controlling street-view elements with multi-perspective consistency via bev sketch layout. *arXiv preprint arXiv:2308.01661*, 2023. 2, 6, 7, 12
- [70] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*, 2021. 2
- [71] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *CVPR*, pages 22490–22499, 2023. 7
- [72] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, pages 13760–13769, 2022. 6, 16
- [73] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 2

# Driving into the Future: Multiview Visual Forecasting and Planning with World Model for Autonomous Driving

## Supplementary Material

### A. Qualitative Results

In this section, we present qualitative examples that demonstrate the performance of our model. For the full set of video results generated by our model, please see our anonymized project page at <https://drive-wm.github.io>.

#### A.1. Generation of Multiple Plausible Futures

Drive-WM can predict diverse future outcomes according to maneuvers from planners, as shown in Figure 8. Based on plans from VAD [30], Drive-WM forecasts multiple plausible futures consistent with the initial observation. We generate video samples on the nuScenes [5] validation set. The rows in Figure 8 show predicted futures for lane changes to left or keep the current lane (row 1), driving towards the roadside and straight ahead (row 2), and left/right turns at intersections (row 3).

#### A.2. Generation of Diverse Multiview Videos

Drive-WM can function as a multiview video generator conditioned on temporal layouts. This enables applications as a neural simulator for Drive-WM. Although trained on nuScenes [5] *train* set, Drive-WM exhibits creativity on the *val* set by generating novel combinations of objects, motions, and scenes.

**Normal scenes generation.** Drive-WM can generate diverse multiview video forecasts based on layout conditions, as shown for the nuScenes [5] validation set in Figure 9.

**Rare scenes generation.** It can also produce high-quality videos for rare driving conditions like nighttime and rain, despite limited exposure during training, as illustrated in Figure 10. This demonstrates the model’s ability to generalize effectively beyond the daytime scenarios dominant in the training data distribution.

#### A.3. Visual Element Control

Drive-WM allows conditional generation through various forms of control, including text prompts to modify global weather and lighting, ego-vehicle actions to change driving maneuvers, and 3D boxes to alter foreground layouts. This section demonstrates Drive-WM’s flexible control mechanisms for interactive video generation based on user-specified conditions.

**Weather & Lighting change.** As shown in Figure 11 and Figure 12, we demonstrate the ability of our model to change weather or lighting conditions while maintaining the same scene layout (road structure and foreground objects). Video examples are generated based on the layout conditions from nuScene *val* set. This ability has great potential for future data augmentation. By generating diverse scenes under various weather and lighting conditions, our model can significantly expand the training dataset, thereby improving the generalization performance and robustness of the model.

**Action control.** Our model is capable of generating high-quality street views consistent with given ego-action signals. For instance, as shown in Fig. 14, our model correctly generates turning-left and turning-right videos from the same initial frame according to the input steering signals. In Fig. 15, our model successfully predicts the positions of the surrounding vehicles conforming with both the accelerating and decelerating signals. These qualitative results demonstrate the high controllability of our world model.

**Foreground control.** As Figure 13 shows, Drive-WM enables fine-grained control of foreground layouts in generated videos. By modifying lateral and longitudinal conditions, high-fidelity images are produced that correspond to the layout changes specified.

Pedestrian generation poses challenges for street-view synthesis methods. However, unlike previous work [53, 69], Figure 16 shows Drive-WM can effectively generate pedestrians. The first six images displays a vehicle waiting for pedestrians to cross, while the second six images shows pedestrians waiting at a bus stop. This demonstrates our model’s potential to produce detailed multi-agent interactions.

#### A.4. End-to-end Planning Results for Out-of-domain Scenarios

Existing end-to-end planners are trained on expert trajectories aligned to lane centers. As Figure 2 shows, this causes difficulties generating off-center deviations, known as the “lack-of-exploration” problem in behavior cloning [8]. Using Drive-WM for simulation, Figure 18 demonstrates more planned trajectories deviating from the lane center. We find the planner from [30] cannot recover when evaluated on these generated out-of-distribution cases. This highlights



**Figure 8. Generation of multiple plausible futures based on the planning.** Here we only show the front-view videos for better illustration. The video samples are generated based on the first frames from the nuScenes *val* set. In the first row, we show examples of lane changing to the left and straight ahead. In the second row, we show cases of driving towards the roadside and driving straight ahead. In the last row, we present examples of making left and right turns at the intersection.

the utility of Drive-WM for exploring corner cases and improving robustness.

#### A.5. Using GPT-4V as a Reward Function

To assess the safety of different futures forecasted under different plans, we leverage the recent GPT-4V model as an evaluator. Specifically, we use Drive-WM to synthesize diverse future driving videos with varying road conditions and agent behaviors. We then employ GPT-4V to analyze these simulated videos and provide holistic rewards in terms of driving safety. As illustrated in Figure 17, it demonstrates different driving behaviors that GPT-4V plans when there is a puddle ahead on the road. Compared to reward functions with vectorized inputs, GPT-4V provides a more generalized understanding of hazardous situations in the Drive-WM videos. By deploying GPT-4V’s multimodal reasoning capacity for future scenario assessment, we enable enhanced evaluation that identifies risks not directly represented but inferred through broader scene understanding. This demonstrates the value of combining generative world models like Drive-WM with reward-generating models like GPT-4V. By using GPT-4V to critique Drive-WM’s

forecasts, more robust feedback can be achieved to eventually improve autonomous driving safety under diverse real-world conditions.

#### A.6. Video Generation on Other Datasets

**Waymo Open Dataset.** To showcase the wide applicability of Drive-WM, we apply it to generate high-resolution 768x512 images on the Waymo Open Dataset. As seen in Figure 19 and Figure 20, Drive-WM produces realistic and diverse driving forecasts at this resolution with the same hyper-parameters for nuScenes. By generalizing effectively to new datasets and resolutions, these Waymo examples verify that Drive-WM provides a widely adaptable approach to high-fidelity video synthesis across different driving datasets.

## B. Implementation Details

In this section, we introduce the training & inference details of *joint multiview video model* and *factorization model*.



Figure 9. **Conditional generation of diverse multiview videos.** Given layout conditions (3D box, HD map, and BEV segmentation) from the nuScenes val set, our model is able to generate spatio-temporal consistent multiview videos.

## B.1. Joint Multiview Video Model

**Training Details.** The original image resolution of nuScenes is  $1600 \times 900$ . We initially crop it to  $1600 \times 800$  by discarding the top area and then resize it to  $384 \times 192$  for model training. Similar to VideoLDM [2], we begin by training a conditional image latent diffusion model. The model is conditioned on various scene elements, such as HD maps, BEV segmentation, 3D bounding boxes, and text descriptions. All the conditions are concatenated in the token-length dimension. The image model is initialized with Stable Diffusion checkpoints [44]. This *conditional image model* is trained for 60,000 iterations with a total batch size of 768. We use the AdamW optimizer with a learning rate  $1 \times 10^{-4}$ . Subsequently, we build the *multiview video model* by introducing temporal and multiview parameters (Sec. 3.1) and fine-tune this model for 40,000 iterations with a batch size of 32, with video frame length  $T = 8$ . For action-based video generation, the difference lies only in the change of condition information for each frame, while the rest of the training and model structure are the same. We use the AdamW optimizer [32] with a learning rate  $5 \times 10^{-5}$  for

the video model. To sample from our models, we generally use the sampler from Denoising Diffusion Implicit Models (DDIM) [50]. Classifier-free guidance (CFG) reinforces the impact of conditional guidance. For each condition, we randomly drop it with a probability of 20% during training. All experiments are conducted on A40 GPUs.

**Inference Details.** During inference, the number of sampling steps is 50, and we use stochasticity  $\eta=1.0$ , CFG=5.0. For video generation, we use the first frame as the condition to generate subsequent video content. Similar to VideoLDM [2], we use the generated frame as the subsequent condition for long video generation.

## B.2. Factorization Model

**Training implementation of factorization.** The implementation is overall similar to the implementation of the joint modeling in Sec. B.1. For the factorized generation, we additionally use reference views as extra image conditions.

Taking nuScenes data as an example, we first sort the six multiview video clips clockwise, denoted as



Figure 10. **Rare scenes generation.** Top two rows: night scenarios. Bottom two rows: rainy scenarios.



Figure 11. **Weather change generation.** The top row displays sunny daylight scenes. The bottom row shows the same layouts rendered as rainy scenes, demonstrating conditional generation capabilities.

$\mathbf{x}_0$  to  $\mathbf{x}_5$ . Then a training sample is defined as  $\{\mathbf{x}_{(i-1) \bmod 6}, \mathbf{x}_i, \mathbf{x}'_i, \mathbf{x}_{(i+1) \bmod 6}\}$ , where  $\mathbf{x}_i$  is the stitched view randomly sampled from all six views, and  $\{\mathbf{x}_{(i-1) \bmod 6}, \mathbf{x}_{(i+1) \bmod 6}\}$  is a pair of reference view.  $\mathbf{x}'_i$  is the previously generated frame of  $i$ -th view, which also serves as an additional image condition. The training pipeline is very similar to the joint modeling, while here we generate a single view every iteration instead of multiple views. As can be seen from the training sample  $\{\mathbf{x}_{(i-1) \bmod 6}, \mathbf{x}_i, \mathbf{x}'_i, \mathbf{x}_{(i+1) \bmod 6}\}$ , we have view dimen-

sion  $N = 1$  for the single stitch view  $\mathbf{x}_i$  in training.

**Inference of factorization.** During inference, we pre-define some views as the reference views, and the corresponding videos of these reference views are first generated by the joint model. Then we generate the videos of stitched views conditioned on the paired reference video clips and previously generated view (i.e.,  $\mathbf{x}'_i$ ). Particularly, in nuScenes, we select F, BL, BR<sup>3</sup> as reference views.

<sup>3</sup>F: front; B: back; L: left; R: right

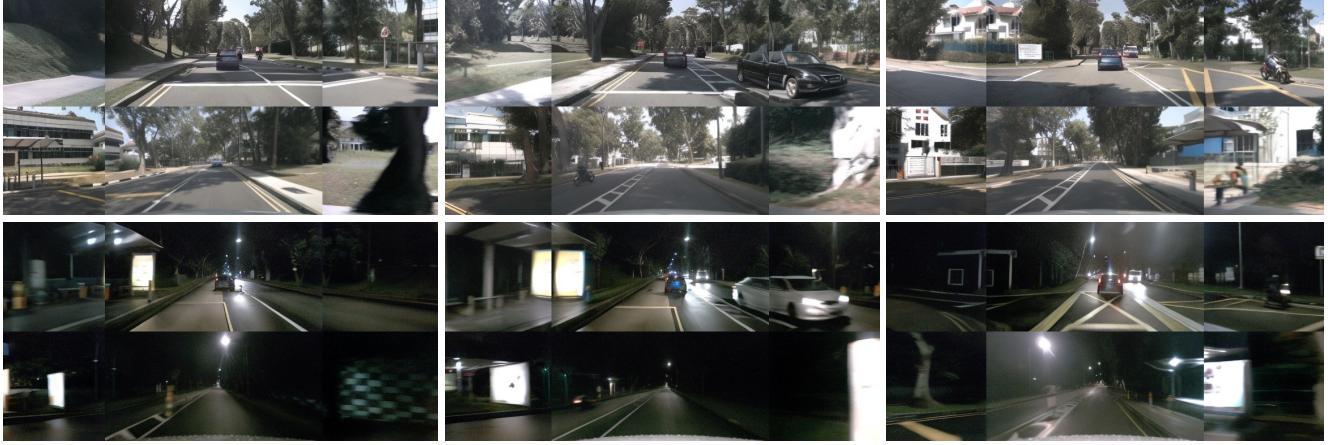


Figure 12. **Lighting change generation.** The top row displays daytime scenes. The bottom row shows the same layouts rendered as nighttime scenes, demonstrating conditional generation capabilities.

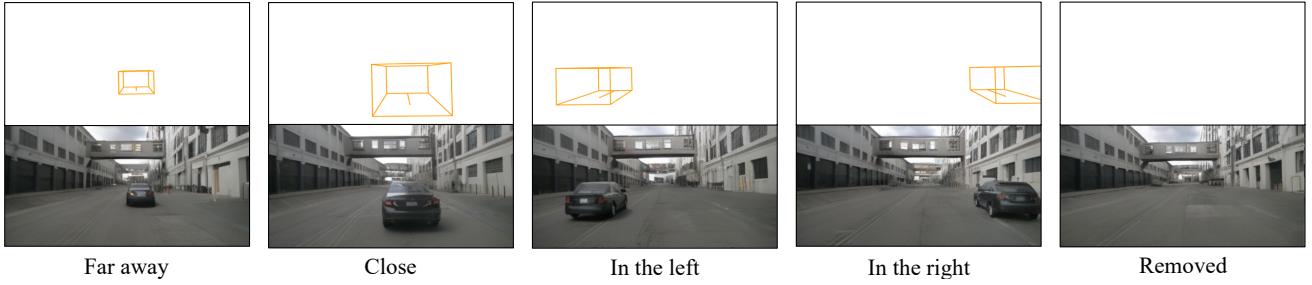


Figure 13. **Control foreground object layouts.** By modifying the positions of 3D boxes, high-fidelity images are produced that correspond to the layout changes specified.

The three reference views constitute three pairs, serving as the condition for three stitched views. For example, our model generates front-left views conditioning on front views, back-left views, and previously generated front-left views. The inference parameters are the same with Sec. B.1.

## C. Data

In this section, we first describe the dataset preparation and then introduce the curation of the dataset to enhance the action-based generation.

### C.1. Data Preparation

**NuScenes Dataset.** The nuScenes dataset provides full 360-degree camera coverage and is currently a primary dataset for 3D perception and planning. Following the official configuration, we use 700 street-view scenes for training and 150 for validation. Next, we introduce the processing of each condition. For the 3D box condition, we project the 3D bounding box onto the image plane, utilizing octagonal corner points to depict the object’s position and dimensions, while colors are employed to distinguish different categories. This ensures accurate object localization and

discrimination between different objects. For the HD map condition, we project the vector lines onto the image plane, with colors indicating various types. In terms of the BEV segmentation, we adhere to the generation process outlined in CVT [72]. This process generates a bird’s-eye view segmentation mask, which represents the distribution of different objects and scenery in the scene. For the text condition, we sift through information provided in each scene description. For the planning condition, we utilize the ground truth movement of the ego locations for training, and the planned output from VAD [30] for inference. This allows the model to learn from accurate ego-motion information and make predictions that are consistent with the planned trajectory. Finally, for the ego-action condition, we extracted the information of vehicle speed and steering for each frame.

**The Waymo Open Dataset.** The Waymo Open Dataset [52] is a well-known large-scale dataset for autonomous driving. We only utilize data from the “front” camera to train the video model, with an image resolution of  $768 \times 512$  pixels. For the map condition, we follow the data processing in OpenLane [7].



**Figure 14. Diverse turning behaviors.** Utilizing an identical initial frame, we provide our model with sequences of positive steering angles (indicating a left turn) and negative values (indicating a right turn). The figure demonstrates the model’s proficiency in generating consistent street views for both turning behaviors. Each frame is accompanied by a blue bar, indicating the corresponding steering angle in degrees. A longer bar correlates with a more substantial steering angle. For clarity, only the front view is shown.



**Figure 15. Diverse speeding behaviors.** We input different patterns of speed signals into our model to assess controllability in terms of speed. The top series shows that the ego car decelerates and then accelerates while the bottom one shows a contrary behavior. These two results highlight the realism of our model’s prediction.

## C.2. Data Curation

The ego action distribution of the nuScenes dataset is heavily imbalanced: a large proportion of its frames exhibit small steering angles (less than 30 degrees) and a normal speed in the range of 10-20 m/s. This imbalance leads to weak generalizability to rare combinations of steering angles and speeds.

To alleviate this negative impact, we balance our training dataset by re-sampling rare ego actions. Firstly, we split each trajectory into several clips, each of which demonstrates only one type of driving behavior (i.e., turning left, going straight, or turning right). This process results in 1048 unique clips. Afterward, we cluster these clips by digitizing the combination of average steering angles and speeds. The speed range [0, 40] (m/s) is divided into 10 bins with equal lengths. Extreme speeds greater than 40 m/s will fall into the 11th bin. The steering angle range [-150, 150] (degree) is divided into 30 bins with equal lengths. Likewise, extreme angles greater than 150 degrees or less than -150 degrees will fall into another two bins, respectively. We plot the ego-action distribution resulting from this categorization in Fig. 21.

To balance the action distribution of these clips, we sample  $N = 36$  clips from each bin of the 2D  $32 \times 11$  grid. For a bin containing more than  $N$  clips, we randomly sample

$N$  clips; For a bin containing fewer than  $N$  clips, we loop through these clips until  $N$  samples are collected. Consequently, 7272 clips are collected. The action distribution after re-sampling can be seen in Fig. 21.

## D. Metric Evaluation Details

### D.1. Video Quality

The FID and FVD calculations are performed on 150 validation video clips from the nuScenes dataset. Since our model can generate multiview video, we break it down into six views of video for evaluation. We have a total of 900 video segments (40 frames) and follow the calculation process described in VideoLDM [2]. We use the official UCF FVD evaluation code<sup>4</sup>.

### D.2. KPM Illustration

As mentioned in Sec. 5.1, we introduced the KPM score metric for measuring multiview consistency for generated images, which are not considered in both FID and FVD metrics. As shown in Figure 22, we demonstrate the key-point matching process. The blue points are the keypoints in the overlapping regions on the left/right side of the image.

<sup>4</sup><https://github.com/SongweiGe/TATS/>



Figure 16. Pedestrian generation examples.



Describe the road surface condition first, please provide potential driving behaviors and the corresponding rewards.



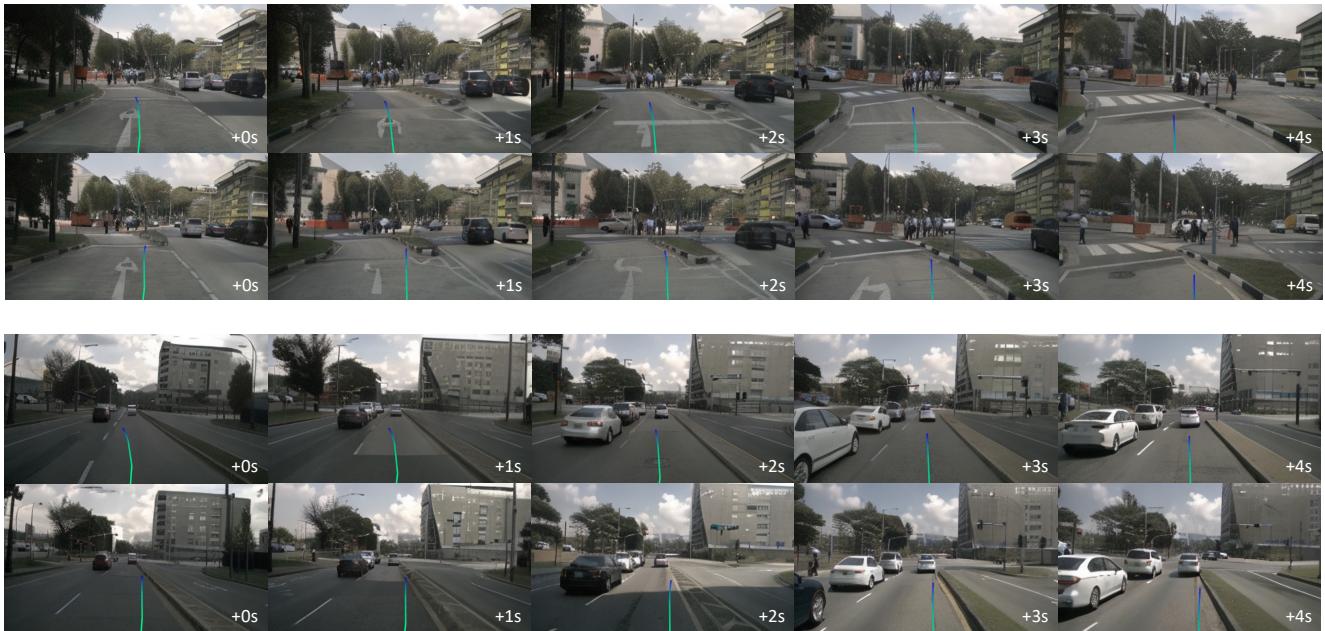
The image depicts an overcast day with wet road conditions. There are **areas of standing water on the road**.



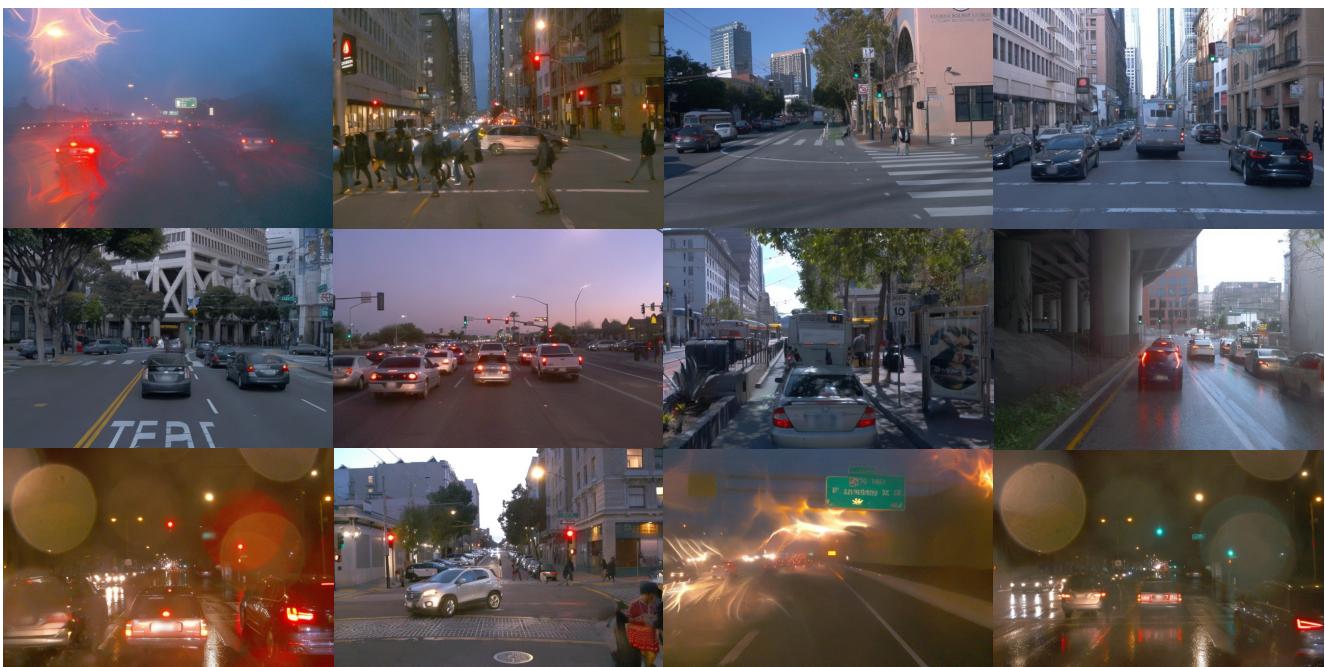
- **Avoiding Puddles:** Steering around large puddles. **Reward: Positive.**  
 - **Driving Through Puddles:** If unavoidable, **slowing down** before entering the puddle to reduce the risk. **Reward: Positive;** However, if done at **high speed**, it could lead to loss of control. **Reward: Negative.**

Figure 17. **GPT-4V as a reward function.** GPT-4V could give more reasonable rewards in cases such as there is a puddle ahead on the road.

The green lines are the matched points between the current view and its two adjacent views using the LoFTR [51] matching algorithm.



**Figure 18. Videos demonstrating the VAD planning results under normal and out-of-domain cases.** We shift the ego location 0.5m to the right to create an out-of-domain case. The top row of each scene: the reasonable trajectory prediction of the VAD method under normal data. The bottom row of each scene: the irrational trajectory when encountering out-of-distribution cases.



**Figure 19. Generation of High-Resolution Image on Waymo Open Dataset.** We showcase the image generation results for a wide range of traffic density, driving scenarios, lighting, and weather conditions.



Figure 20. **Generation of High-Resolution Video on Waymo Open Dataset.** We showcase the video generation results in highly interactive driving scenarios like closely following the front car in heavy traffic (top) or waiting for pedestrians to cross the road.

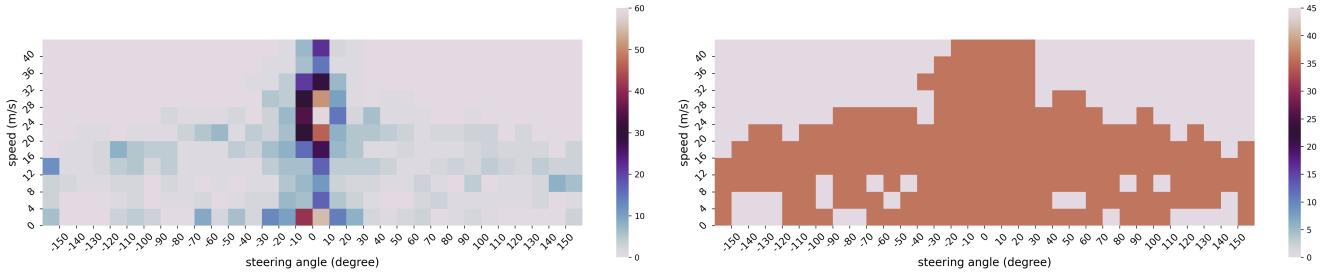


Figure 21. **The ego-action distribution before (left) and after re-sampling (right).** We re-sample rare combinations of speeds and steering angles, obtaining a balanced training dataset.



Figure 22. **Illustration of keypoint matching in KPM calculation.** The blue points are the image keypoints in the overlapping regions on the left/right side of the image. The green lines are the matched keypoints between the current view and its two adjacent views using the LoFTR [51] matching algorithm.