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 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 [35], Drive-WM forecasts multiple plausible futures consistent with the initial observation. We generate video samples on the nuScenes [7] 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 [7] *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 [7] 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 [58, 74], 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 [10]. Using Drive-WM for simulation, Figure 18 demonstrates more planned trajectories deviating from the lane center. We find the planner from [35] cannot recover when evaluated on these generated out-of-distribution cases. This high-

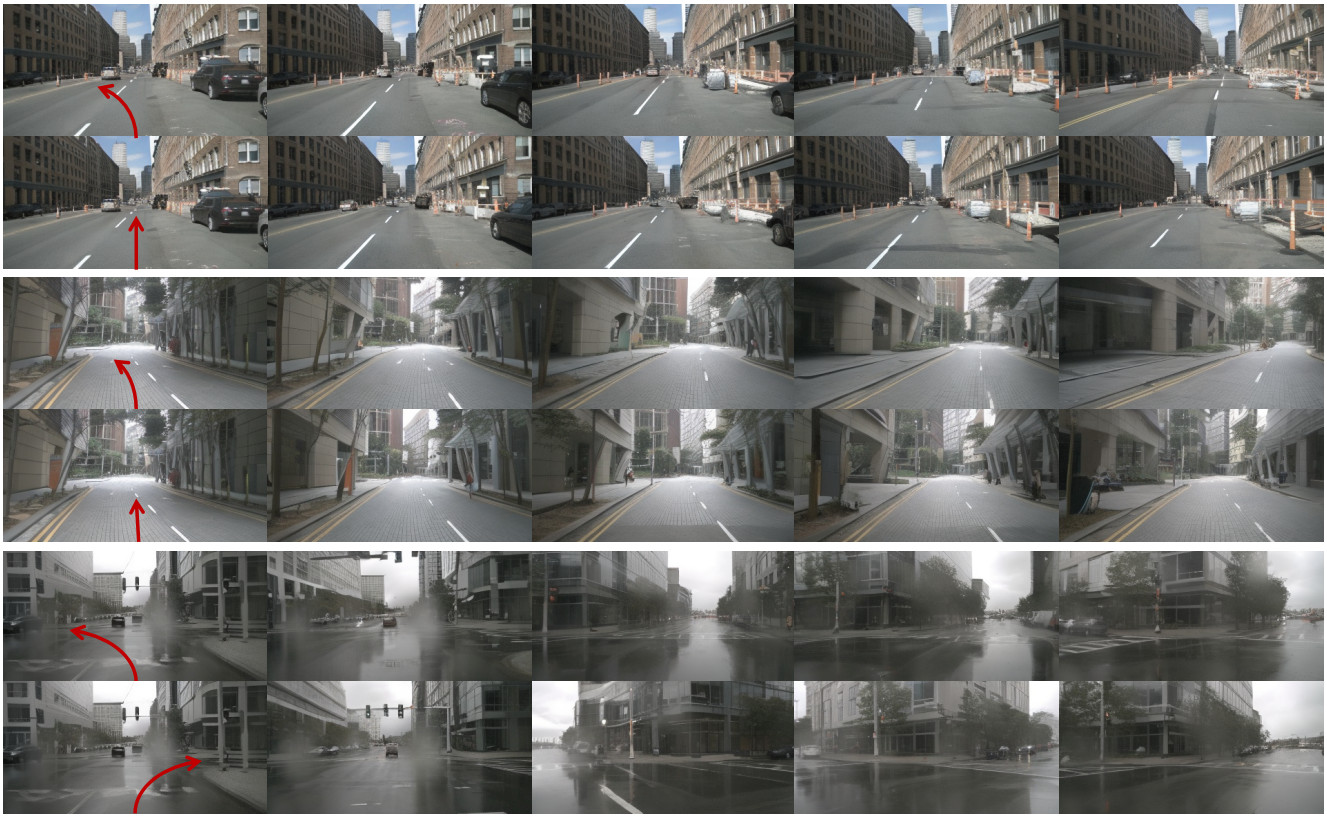

Figure 8. **Generation of multiple plausible futures based on the planning.** Here we only show the front-view videos for better illustration. The video samples are generated based on the first frames from the nuScenes *val* set. In the first row, we show examples of lane changing to the left and straight 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.

lights 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 768×512 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 × 900. We initially crop it to 1600 × 800 by discarding the top area and then resize it to 384 × 192 for model training. Similar to VideoLDM [4], 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 [49] 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 1×10^{-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 [37] with a learning rate 5×10^{-5} for the video model. To sample from our models, we generally use the sampler from Denoising Diffusion Implicit Models (DDIM) [55]. 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 (48GB) GPUs.

Inference Details. During inference, the number of sampling steps is 50, and we use stochasticity η=1.0, CFG=5.0. For video generation, we use the first frame as the condition to generate subsequent video content. Similar to VideoLDM [4], 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 dimension $N = 1$ for the single stitch view $\mathbf{x}_i$ in training.

Inference of factorization. During inference, we predefine 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`[3] as reference views.

[3]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.

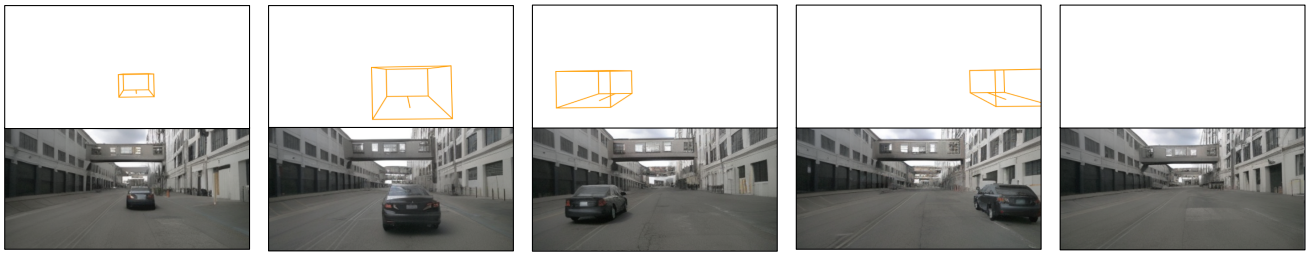

Figure 13. **Control foreground object layouts.** By modifying the positions of 3D boxes, high-fidelity images are produced that correspond to the layout changes specified.

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 [77]. 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 [35] 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 [57] 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 768×512 pixels. For the map condition, we follow the data processing in OpenLane [9].

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 32×11 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

To evaluate the quality of the generated video, we utilize FID (Frechet Inception Distance) [29] and FVD (Frechet Video Distance) [62] as the main metrics. 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 (around 40 frames) and follow the calculation process described in VideoLDM [4]. For FID calculation on videos, we randomly select 10,000 frames for evaluation. We use the official UCF FVD evaluation code[4].

D.2. Controllability Evaluation

To assess the controllability of video content generation, we evaluate generated images using pre-trained perception models. Following the previous method [74], we adopted

[4] https://github.com/SongweiGe/TATS/

Figure 16. **Pedestrian generation examples.** Previous methods are difficult to generate pedestrians.

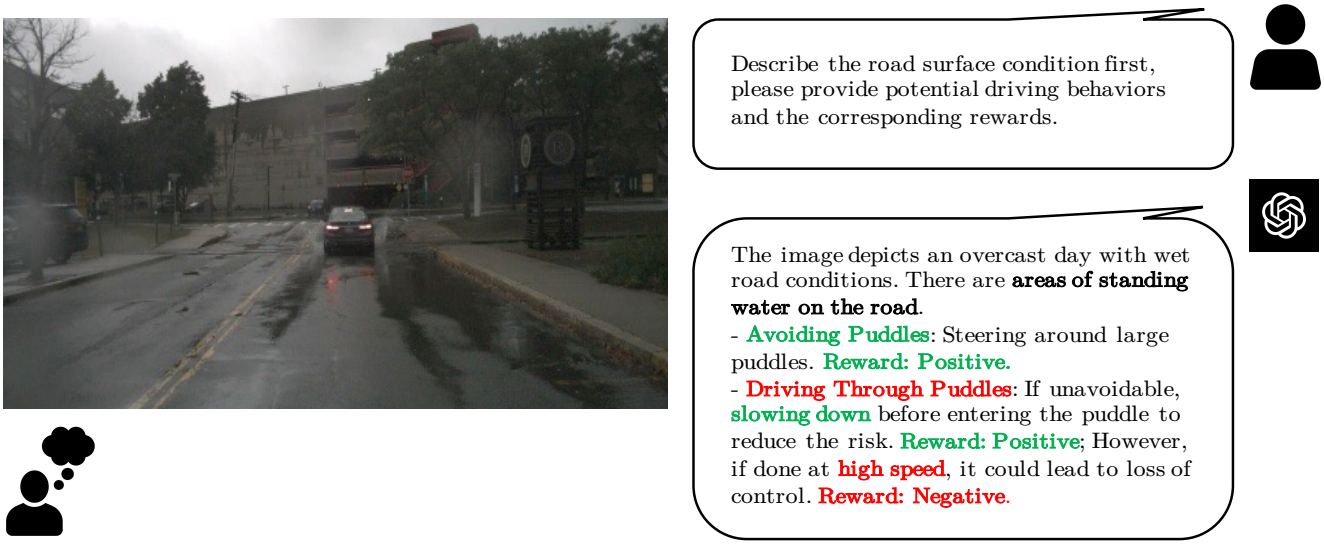

Figure 17. **Using GPT-4V as a reward function.** GPT-4V could give more reasonable rewards in cases such as there is a puddle ahead on the road.

CVT [77] for foreground and background segmentation. Additionally, we evaluate 3D object detection [67] and online map construction [43]. Here, due to the disparity between the input image size for the perception model and the generated image size, we resize the generated images to match the dimensions of the original images for inference. The performance evaluation of the GT (real image) is conducted by down-sampling real images to match the resolution of the generated images for the fair comparison.

D.3. Planning Evaluation

We follow open-loop evaluation metrics [34, 35] for end-to-end planning, including L2 distance from the GT trajectory and the object collision rate.

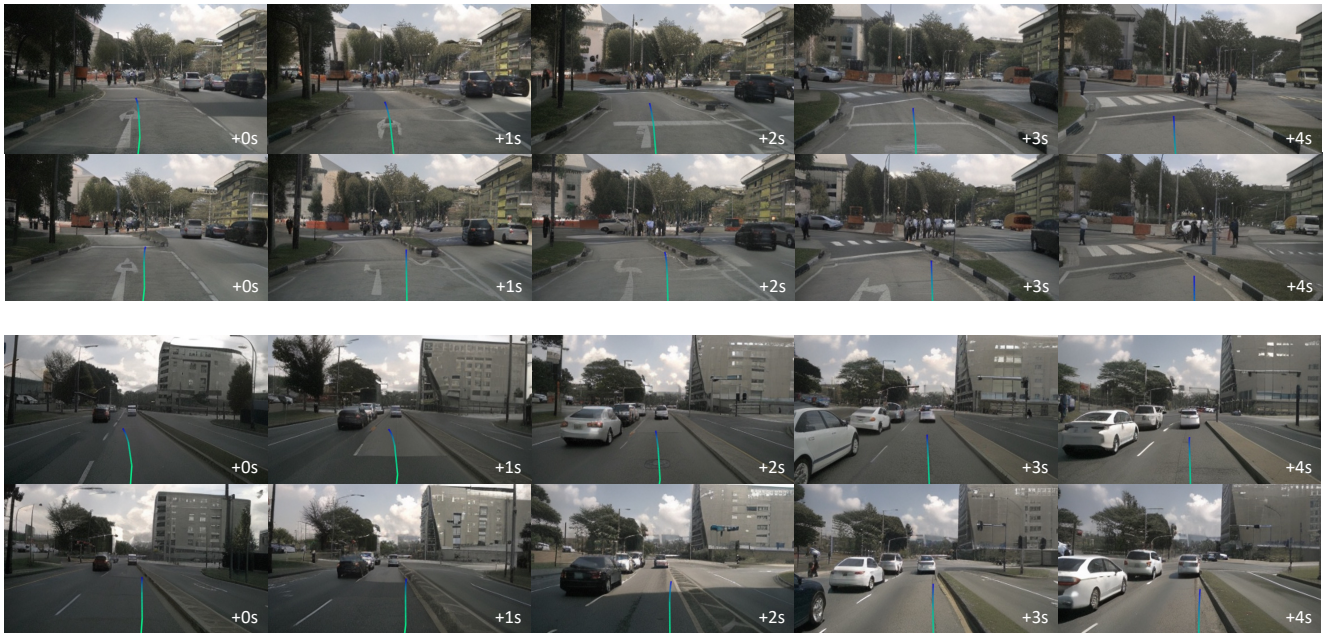

Figure 18. **Videos demonstrating the VAD planning results under normal and out-of-domain cases.** We shift the ego location 0.5m to the right to create an out-of-domain case. The top row of each scene: the reasonable trajectory prediction of the VAD method under normal data. The bottom row of each scene: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

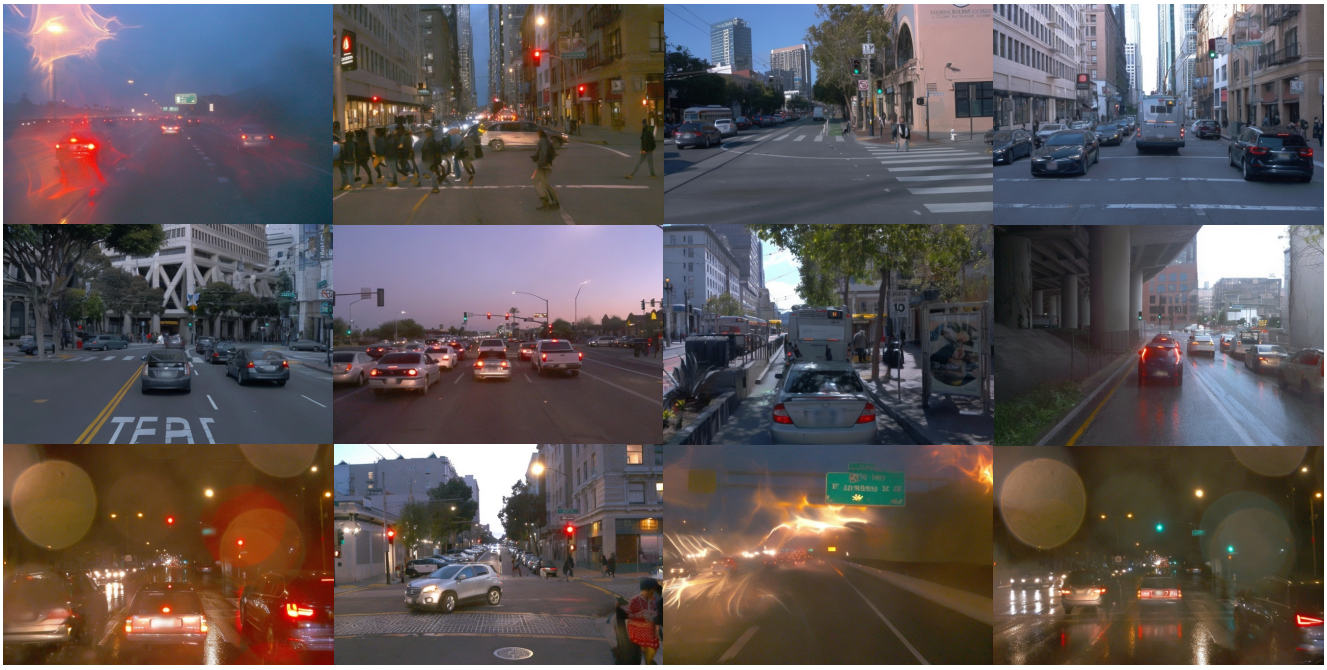

Figure 19. **Generation of High-Resolution Image on Waymo Open Dataset.** We showcase the image generation results for a wide range of traffic density, driving scenarios, lighting, and weather conditions.

D.4. 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. 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 cal-

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.

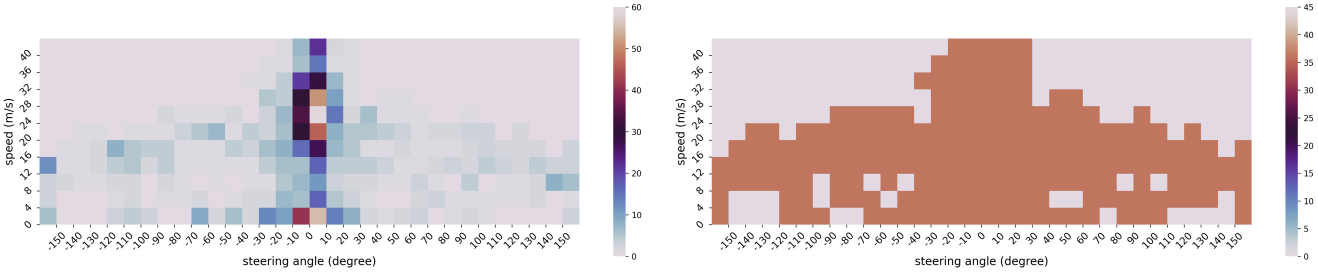

Figure 21. **The ego-action distribution before (left) and after re-sampling (right).** We re-sample rare combinations of speeds and steering angles, obtaining a balanced training dataset.

culated 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.

As shown in Figure 22, we demonstrate the keypoint matching process. The blue points are the keypoints in the overlapping regions on the left/right side of the image. The green lines are the matched points between the current view and its two adjacent views using the LoFTR [56] matching algorithm.

E. Future Work

In the future, there are two main research directions. Inspired by GAIA-1 [33], one is to scale up data; richer data can enable models to possess stronger generation capabilities, thereby better envisioning various out-of-distribution (OOD) cases. The second is to consider the practicality of world models, which requires compatibility with more planners and faster inference speeds. Currently, the computation cost of video generation remains a challenging issue, and we will explore further in the future.

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 [56] matching algorithm.