—Supplementary Material—
UniScene: Unified Occupancy-centric Driving Scene Generation

1. Problem Definition and Distinctiveness

1.1. Controllable Occupancy Generation

Previous research on uncontrollable occupancy generation [13, 14, 17] has primarily focused on the creation of static scenes, which limits their applicability to dynamic scenarios due to a lack of controllability. In contrast, our approach introduces controllable generation $\mathbb{P}(\texttt{Occ}|\texttt{BEV})$, effectively incorporating temporal information into the process. Here $\texttt{BEV}$ refers to input Bird's Eye View (BEV) scene layouts and $\texttt{Occ}$ denotes the corresponding generated semantic occupancy, respectively.

1.2. Conditional Video and LiDAR Generation

Existing methods for video [6, 30, 31, 35, 42] and LiDAR [24, 46, 47] generation typically produce data directly from coarse scene layouts (e.g., BEV maps, 3D bounding boxes). However, these approaches often fail to accurately capture the intricate distributions inherent in driving scenes, leading to suboptimal performance. In contrast, our proposed UniScene framework addresses these limitations by decomposing the complex generation task into a hierarchy centered around occupancy. This is formally expressed as:

$$\mathbb{P}(\texttt{Vid}, \texttt{Lid}|\texttt{BEV}) = \mathbb{P}(\texttt{Vid}, \texttt{Lid}|\texttt{Occ}) \cdot \mathbb{P}(\texttt{Occ}|\texttt{BEV}), \tag{1}$$

where $\texttt{Vid}$, and $\texttt{Lid}$ denote the generated video, and LiDAR data, respectively. By leveraging occupancy priors, our method alleviates the learning burden and more accurately captures the underlying distributions for performance enhancement, as demonstrated in Fig. 1(a) of the main paper.

2. More Related Works and Discussions

2.1. Semantic Occupancy Generation

SemCity [14] proposes a 3D semantic scene generation approach with a triplane diffusion framework. PyramidOcc [17] generates large-scale 3D semantic scenes using a coarse-to-fine paradigm with pyramid discrete diffusion [1]. These methods mainly focus on unconditional and static 3D scene generation. More recent work [40] is capable of controlling the generation of 3D scenes through BEV maps. However, it remains confined to static scenes. OccSora [27] generates temporal 3D scene sequences with a diffusion transformer (DiT) [22]. Due to the high compression ratio of occupancy in its designed VQVAE, the reconstruction performance is suboptimal, which to some extent leads to unsatisfactory generation quality. Moreover, it lacks the capability of precisely controlling the generated results. To address the issues in these works, we propose our occupancy generation model, enabling controllable generation of temporal 3D scene sequences with high fidelity while effectively maintaining temporal consistency.

2.2. Driving Video Generation

Vista [7] builds on the architecture of Stable Video Diffusion (SVD) [3] to enable single-view driving video generation with various action controls. However, this method is constrained by its inability to produce multi-view videos and its lack of alignment between the generated outputs and ground-truth labels, which hinders its utility for training downstream tasks. WoVoGen [19] presents a world model that predicts future videos and occupancy based on past observations. In this model, occupancy grids are compressed into low-dimensional features [23], potentially leading to suboptimal generation results (see the seventh row of Table 4 in the main paper). Recent work of SyntheOcc [16] synthesizes multi-view images in driving scenarios using occupancy-based multi-plane semantic images. Nevertheless, this method neglects the explicit geometric information within the semantic occupancy grids and necessitates substantial manual intervention for occupancy grid editing. Our proposed UniScene aims to jointly render semantic and depth maps from the semantic occupancy, thereby providing detailed prior information. Moreover, UniScene simplifies geometric editing by using BEV maps as scene layouts.

2.3. LiDAR Point Cloud Generation

Pioneering works in LiDAR generation [8, 24, 37, 38, 46, 47] utilized GAN or diffusion models to produce LiDAR point clouds. LiDARGen [46] adopts an equirectangular view image as a structured representation of LiDAR point clouds and uses a score-based diffusion model for point cloud generation. Nevertheless, the 2.5D representation may constrain its capacity to accurately generate 3D geometries

of real-world objects. Furthermore, LiDARGen [46] applies the diffusion process directly to LiDAR points rather than in the latent space, significantly slowing down the inference process. UltraLiDAR [38] voxelized LiDAR points and transformed them into a BEV representation, employing VQVAE to learn a compact 3D representation of LiDAR points and a generative transformer for LiDAR points generation. However, using BEV as the representation overlooks the fine geometric details in the LiDAR data, potentially impacting the generation quality. LiDM [8] employs range maps as the representation for LiDAR data, integrating curve-wise compression, patch-wise encoding, and point-wise coordinate supervision in VQVAE to enhance its geometric representation capabilities. However, the use of range map representation can result in a loss of structured LiDAR point information. In this work, we propose to facilitate high-fidelity LiDAR point generation by leveraging fine-grained priors from semantic occupancy grids.

3. More Implementation Details

3.1. BEV Editing Scheme

The modification of BEV layouts for the controllable generation of semantic occupancy, videos, and LiDAR point clouds represents a significant application in the creation of out-of-distribution (OOD) data. To accomplish this, our editing scheme is structured as follows:

- Modify the initial BEV layout, $\mathbf{B}_{\text{ori}}$, to produce a revised layout, $\mathbf{B}_{\text{new}}$, incorporating specific alterations (e.g., the removal of a vehicle).
- Employ DDIM Inversion [20] to transform the original occupancy, $\mathbf{O}_{\text{ori}}$, into a noise latent, ϵ_{ori}, while being guided by the original BEV layout, $\mathbf{B}_{\text{ori}}$.
- Construct the updated occupancy, $\mathbf{O}_{\text{new}}$, via denoising diffusion, utilizing the modified BEV layout, $\mathbf{B}_{\text{new}}$, as a conditioning factor and ϵ_{ori} as the starting noise.
- With the newly generated occupancy, $\mathbf{O}_{\text{new}}$, serve as the addition condition, produce the corresponding video, $\mathbf{V}_{\text{new}}$, and LiDAR data, $\mathbf{L}_{\text{new}}$, with detailed prior guidance.

3.2. Occupancy Generation Model

As illustrated in Fig. 1, the occupancy generation model is composed of a VAE and a DiT, which produces semantic occupancy with a compressed latent space.

Occupancy VAE. We leverage the Occupancy VAE encoder to transform a 3D semantic occupancy $\mathbf{O} \in \mathbb{R}^{H\times W\times D}$ within an occupancy sequence into a BEV representation $\hat{\mathbf{O}} \in \mathbb{R}^{H\times W\times DC'}$ by assigning each category a learnable class embedding C'. A 2D CNN encoder with a 2D axial attention layer is utilized to extract a continuous latent feature with down-sampled resolution $\mathbf{Z}_{\text{occ}} \in \mathbb{R}^{C\times h\times w}$, where $h = \frac{H}{d}$ and $w = \frac{W}{d}$, with d being the down-sampling factor.

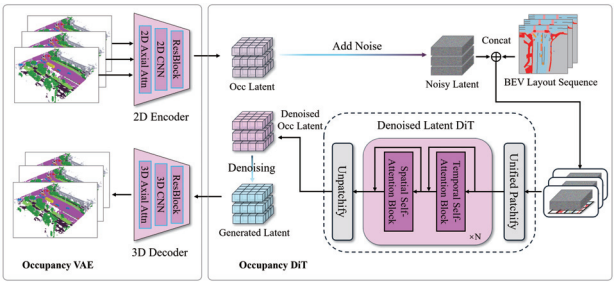

Figure 1. The architecture of the occupancy generation model, which consists of two main components: the Occupancy VAE and the Occupancy DiT. The Occupancy VAE includes a 2D encoder, leveraging ResBlock, 2D CNNs, and axial attention, and a 3D decoder to reconstruct the input data. The Occupancy DiT applies a denoising diffusion process using temporal and spatial self-attention blocks to generate denoised latent representations. Noisy latents, combined with BEV layout sequences, are processed to create unified patchified outputs, enabling robust occupancy generation.

Method	Compression Ratio ↑	mIoU ↑	IoU ↑
OccLLama (VQVAE) [32]	8	75.2	63.8
OccWorld (VQVAE) [43]	16	65.7	62.2
OccSora (VQVAE) [27]	**512**	27.4	37.0
Ours(VQVAE)	32	59.8	58.2
Ours (VAE)	32	**92.1**	**87.0**
Ours(VQVAE)	**512**	55.8	56.8
Ours (VAE)	**512**	72.9	64.1

Table 1. Comparison of VQVAE and VAE performance on occupancy reconstruction, with compression ratios calculated based on the methodology from OccWorld [43]. The results demonstrate the clear superiority of our occupancy VAE over VQVAE under the same architectural design, achieving significantly higher mIoU and IoU scores across various compression ratios.

The VAE decoder reconstructs the latent feature sequence $\mathbf{z}_{\text{occ}}^{\text{seq}} \in \mathbb{R}^{T\times C\times h\times w}$. A 3D CNN network with a 3D axial attention layer is employed to up-sample the latent feature sequence to a BEV representation occupancy sequence $\hat{\mathbf{O}}^{\text{seq}} \in \mathbb{R}^{T\times H\times W\times DC'}$. This sequence is then reshaped to $\mathbb{R}^{THW\times D\times C'}$ and processed through a dot product with the class embeddings to obtain the logits scores. During training, the logits scores and one-hot labels are used to compute the cross-entropy loss and Lovász-softmax loss [2]. In the inference phase, the final reconstructed occupancy sequence $\mathbf{O}^{\text{seq}} \in \mathbb{R}^{T\times H\times W\times D}$ is determined by taking the `argmax` of the logits. We provide more occupancy reconstruction results in Tab. 1 to illustrate the superiority of our occupancy VAE design with continuous compression. The results demonstrate the clear superiority of our occupancy VAE over VQVAE under the same architectural design, which achieves significantly higher mIoU and IoU scores across various compression ratios.

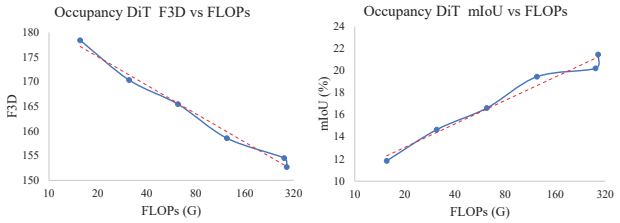

Figure 2. Scalability analysis of Occupancy DiT, illustrating how performance scales with increasing computational cost (FLOPs). The left plot shows a consistent improvement in F3D performance as FLOPs increase, while the right plot highlights the scalability of mIoU, which also improves steadily with higher computational resources, demonstrating the model's capacity to leverage increased computation for better results.

Method	mIoU ↑				IoU ↑			
	1s	2s	3s	Avg	1s	2s	3s	Avg
OccWorld [43]	25.75	15.14	10.51	17.13	34.63	25.07	20.19	26.63
OccLLama [32]	25.05	19.49	15.26	19.93	34.56	25.83	24.41	29.17
Ours-Fore. (1 ref)	30.93	24.87	20.75	27.33	35.15	31.79	29.24	31.62
Ours-Fore. (2 ref)	35.37	29.59	25.08	31.76	38.34	32.70	29.09	34.84

Table 2. Comparison of occupancy forecasting performance, where Ours-Fore. (1 ref) and Ours-Fore. (2 ref) represent our forecasting model conditioned on 1 and 2 reference occupancy frames, respectively. The results highlight significant improvements in both the mIoU and IoU metrics over baseline methods (OccWorld and OccLLama) across different time horizons (1s, 2s, 3s), demonstrating the effectiveness of our approach in generating accurate and consistent occupancy predictions.

Occupancy DiT. As shown in Fig. 1, we employ an Occupancy DiT to denoise occupancy latent sequence features from noisy occupancy latents and BEV layout sequences. To align the BEV layout sequence with the occupancy latent features, we introduce a unified patchify module. Specifically, the BEV layout at time step i is down-sampled into $\mathbf{B}^i_{down} \in \mathbb{R}^{(C_b)\times h\times w}$ to match the shape of the latent feature and then concatenated with the latent $\mathbf{Z}^i_{\text{occ}} \in \mathbb{R}^{(C_o)\times h\times w}$. A unified patch embedder transforms the concatenated latent $\mathbf{Z}_{\text{cat}} \in \mathbb{R}^{(C_o+C_b)\times h\times w}$ into unified latent tokens $\mathbf{Z} \in \mathbb{R}^{L\times E_d}$, where L is the number of patches and E_d is the embedding dimension.

The backbone of Occupancy DiT is the Spatial-Temporal Latent Diffusion Transformer, comprising stacked spatial and temporal blocks. Spatial blocks aggregate features across different positions within the same latent, whereas temporal blocks capture features across different latent frames at the same position. Additionally, 2D positional embeddings and 1D temporal embeddings are used to account for relative relationships. The output of the backbone, with dimensions $\mathbb{R}^{T\times L\times E_d}$, is passed through an unpatchify layer to yield a denoised occupancy latent sequence of size $\mathbb{R}^{T\times H\times W\times D}$. During the training phase, we randomly drop the BEV layout condition with a probability of 0.1, enabling the diffusion model to learn unconditional generation, which is essential for occupancy editing. In the sampling phase, the classifier-free guidance scale is set to 1.0 by default.

Occupancy Forecasting Variant. To facilitate comparison with other occupancy forecasting works [32, 43], the occupancy generation model is adapted into a generative forecasting model, which predicts T_f future frames based on T_c conditional frames. Specifically, during the training phase, the conditional occupancy latent frames are concatenated directly with the BEV layouts without the addition of noise. The unified latent tokens for both T_c and T_f frames are then fed into the DiT backbone, following the same procedure as in the generation model. The model outputs denoised occupancy latent frames for both T_c and T_f frames, while the loss is computed only on the T_f frames. In the inference phase, the T_f frames are initialized with pure noise, while the T_c frames are initialized with conditional occupancy latents sampled from the Occupancy VAE. The future occupancy frame number T_f is set to 6 to align with previous works [32, 43]. To improve computational efficiency and reduce dependence on the conditional frames, we set $T_c = 1$ or $T_c = 2$ instead of $T_c = 5$ in previous works [32, 43]. The results of the forecasting model are presented in Tab. 2. As we can see from the table, our method effectively surpasses other works across different time horizons (1s, 2s, 3s), demonstrating the effectiveness of our approach in generating accurate and consistent occupancy generation. Moreover, we visualize the scalability of the Occupancy DiT in Fig. 2 to demonstrate the potential of our proposed method. The performance improves steadily with higher computational resources, which demonstrates the model's capacity to leverage increased computation for better results as discussed in [21].

3.3. Video Generation Model

As shown in Fig. 3, the video generation model combines video latent representations with occupancy grids and textual prompts to guide video generation. The VAE encoder extracts video latent features, which are combined with noise before passing through the Diffusion UNet.

Preliminaries with Stable Video Diffusion. The Stable Video Diffusion (SVD) [3] is a latent diffusion model specifically designed for image-to-video (I2V) generation. To increase sampling flexibility, SVD utilizes a continuous-timestep formulation. The model converts data samples x into noise n through a diffusion process, where $p(n|x)$ follows a Gaussian distribution $\mathcal{N}(x, \sigma^2 I)$. New samples are then generated by gradual denoising the latent space from Gaussian noise until $\sigma_0 = 0$. SVD generates videos by processing a series of noisy latent representations, with the generation being guided by a conditional image. The latent

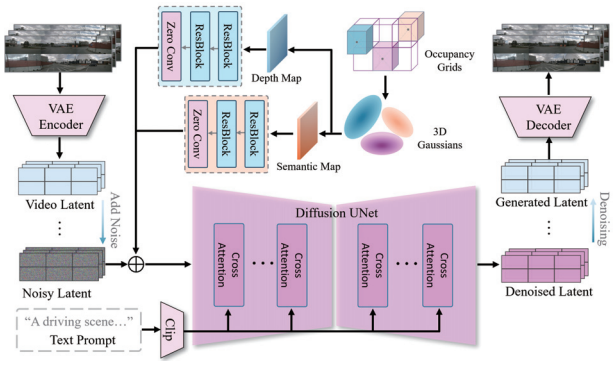

Figure 3. The architecture of the video generation model, which combines video latent representations with textual prompts to guide video generation. The VAE encoder extracts video latent features, which are combined with noise before passing through the Diffusion UNet. The Diffusion UNet employs cross-attention mechanisms to integrate textual guidance and refine the latent representation. Occupancy grids serve as the basis for generating depth maps and semantic maps, which act as guidance for the video generation process. The VAE decoder reconstructs the video from the denoised latent, producing realistic outputs guided by the input prompts.

representation of this image is channel-wise concatenated to the input, acting as a reference for the content creation.

Preliminaries with Gaussian Splatting. The Gaussian Splatting [11] approach defines a set of 3D Gaussian primitives as $\mathcal{G} = \{G_i\}_{i=1}^{N}$, where each primitive G_i contains attributes of position $\boldsymbol{\mu}_i$, color c_i, opacity α_i, rotation matrix $\mathbf{R}_i$, and scale matrix $\mathbf{S}_i$. The associated 3D covariance matrix $\boldsymbol{\Sigma}_i$ for each primitive is derived as:

$$\boldsymbol{\Sigma}_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^{\mathbf{T}} \mathbf{R}_i^{\mathbf{T}}, \tag{2}$$

when transformed by a viewing transformation $\mathbf{W}$, the covariance matrix in camera coordinates $\boldsymbol{\Sigma}_i^{'}$ is computed as:

$$\boldsymbol{\Sigma}_i^{'} = \mathbf{J}\mathbf{W}\boldsymbol{\Sigma}_i\mathbf{W}^{\mathbf{T}}\mathbf{J}^{\mathbf{T}}, \tag{3}$$

where $\mathbf{J}$ is the Jacobian of the affine approximation of the projective transformation. This process yields a 2×2 covariance matrix, consistent with the previous work [11]. For a given projected 3D Gaussian center $\boldsymbol{\mu} \in \mathbb{R}^{2\times 1}$ and a point $\mathbf{x} \in \mathbb{R}^{2\times 1}$ in camera coordinates, the opacity α' of projected 2D Gaussian is formulated as:

$$\alpha' = \alpha \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \left(\boldsymbol{\Sigma}'\right)^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right). \tag{4}$$

The accumulated color map $\mathbf{C}$ is obtained via tile-based rasterization:

$$\mathbf{C} = \sum_{i\in\mathcal{N}} c_i \alpha_i' \prod_{j=1}^{i-1} \left(1-\alpha_j'\right). \tag{5}$$

Gaussian-based Joint Rendering. In this work, to generate depth and semantic maps, we start by transforming an input semantic occupancy grid, with dimensions $\mathbb{R}^{H\times W\times D}$, into a set of 3D Gaussian primitives $\mathcal{G}$. Each primitive G_i is assigned an additional attribute: the semantic label s_i. The position of each 3D Gaussian is initialized using the center of the corresponding occupancy grid cell, its scale is set according to the size of the grid cell, and its semantic attribute is initialized with the semantic label of the grid cell. To improve the effectiveness of the 2D conditions, we render depth and semantic maps that align with the camera's viewpoint. Specifically, given the semantic label s_i and depth value d_i, the depth map $\mathbf{D}$ and the semantic map $\mathbf{S}$ are rendered in a manner analogous to the color rendering:

$$\mathbf{D} = \sum_{i\in N} d_i \alpha_i' \prod_{j=1}^{i-1} \left(1-\alpha_j'\right), \tag{6}$$

$$\mathbf{S} = \texttt{argmax}\Big(\sum_{i\in N} \texttt{onehot}(s_i)\alpha_i' \prod_{j=1}^{i-1} \left(1-\alpha_j'\right)\Big). \tag{7}$$

As discussed in Section 3.2 of the main paper, we incorporate an encoder branch inspired by ControlNet [41] to condition the rendering maps. The specifics of this conditioning are depicted in Fig. 3. The encoder branch consists of two modules, each designed for either depth map conditioning or semantic map conditioning, with similar architectures. Each conditioning module comprises two ResNet blocks followed by a zero convolution, as recommended by ControlNet [41], to preserve the original functionality of the diffusion model.

3.4. LiDAR Generation Model

As illustrated in Fig. 4, the LiDAR generation model comprises a LiDAR Sparse UNet and a Prior Guided Sampling mechanism. The LiDAR Sparse UNet processes input occupancy data to extract spatial features, whereas the Prior Guided Sampling mechanism generates LiDAR points by sampling along rays according to the spatial structure. The LiDAR Head utilizes weighted summation and multilayer perceptron (MLP) modules to compute the intensity, ray drop probabilities, and final positions of the LiDAR points. This process involves assigning weights to the generated LiDAR point features, refining these features through an MLP, and aggregating the results to produce high-quality LiDAR outputs with realistic point distributions.

Specifically, for n points on a LiDAR ray, the feature of the i-th point, u_i, is sampled from the output of the LiDAR Sparse UNet with the occupancy-based prior guidance. This feature is then fed into an MLP to predict the corresponding Signed Distance Function (SDF) value. The weight for the point, w_i, is subsequently computed and used to determine the ray depth d_i of the generated LiDAR points according to Eq. 8 and Eq. 9 in the main paper. Additionally, the ray feature, v_r, is obtained by performing a weighted sum of

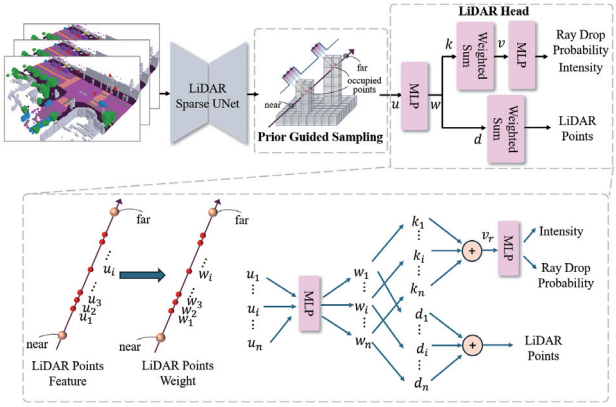

Figure 4. The architecture of the LiDAR generation model, which integrates a LiDAR Sparse UNet and a Prior Guided Sampling mechanism. The LiDAR Sparse UNet processes input occupancy data to extract spatial features, while the Prior Guided Sampling generates LiDAR points by sampling along rays based on the spatial structure. The LiDAR Head employs weighted summation and MLP modules to calculate the intensity, ray drop probabilities, and final positions of the LiDAR points. The process involves assigning weights to the generated LiDAR point features, refining them through an MLP, and aggregating the results to produce high-quality LiDAR outputs with realistic point distributions.

the features of all points along the ray, using their respective weights, that is,

$$v_r = \sum_{i=1}^{n} k_i = \sum_{i=1}^{n} w_i \cdot u_i. \tag{8}$$

Finally, v_r is passed through another MLP to simultaneously predict the intensity and drop probability of the LiDAR ray.

4. Datasets

The NuScenes benchmark [4] is widely used in autonomous driving research. To enrich this dataset with more detailed annotations, the NuScenes-Occupancy dataset [29] has introduced dense semantic occupancy labels. This dataset encompasses comprehensive LiDAR sweep data across 850 scenes, comprising 34,000 key-frames at a frame rate of 2 Hz, each annotated with one of 17 semantic categories. Following the methodology described in [29], our study allocates 28,130 frames for the training set and 6,019 frames for validation.

However, the 2 Hz data alone does not meet our requirements. To enhance our results and ensure a fair comparison with previous works [6, 25, 31, 35, 42], we further extend it into 12 Hz data. Specifically, we utilize the interpolated 12 Hz annotations from ASAP [28] and the LiDAR data from the NuScenes dataset to generate semantic occupancy labels at 12 Hz. We begin the generation of the 12 Hz occupancy labels by concatenating the LiDAR points from the entire scene, following the method outlined in [33]. Next, we reconstruct the mesh of the driving scene using the NKSR algorithm [10]. We then extract the vertices of the mesh and assign semantic labels to these vertices based on the LiDARseg tags from NuScenes [4] and the 12 Hz annotation information from ASAP [28]. Finally, we convert the mesh vertices, along with their semantic labels, into semantic occupancy data.

5. Evaluation Metrics

To evaluate the fidelity of occupancy generation, we adopt the F3D metric and Maximum Mean Discrepancy (MMD) as [17]. F3D represents a three-dimensional adaptation of the two-dimensional Fréchet Inception Distance (FID), utilizing a pre-trained occupancy auto-encoder. The MMD is calculated within the feature space of this same auto-encoder. Given that BEV layouts are used for generation, Mean Intersection over Union (mIoU) serves as the metric for evaluating the accuracy of the generated outputs. For occupancy forecasting, we report both IoU and mIoU, aligning with previous works [43]. For video generation assessment, we utilize the FID and Frechet Video Distance (FVD) to evaluate the quality of the generated content, following previous works [9, 26]. For the evaluation of LiDAR generation, we apply the Jensen-Shannon Divergence (JSD) and MMD following LiDARDM [47].

For the downstream perception task of Semantic Occupancy Prediction (SOP), we adopt mIoU and IoU as evaluation metrics, following previous research [5, 15]. For BEV segmentation, we use CVT [44] as the baseline and assess the mIoU for road and vehicle classes, following the approach in [6]. For 3D object detection, we employ BEVFusion [18] as the baseline model and measure performance using mean Average Precision (mAP) and the NuScenes Detection Score (NDS), as employed in [6].

6. Model Setup

The UniScene framework undergoes a two-stage training process implemented with PyTorch on NVIDIA A100 GPUs. Initially, the occupancy generative models are trained using ground-truth labels. Subsequently, the occupancy generative model is fixed to generate occupancy grids from the BEV maps, while the video and LiDAR generation models are jointly trained with occupancy-based conditions.

The occupancy generation model is first trained with a batch size of 16 on 8 NVIDIA A100 GPUs for about 3 days. The AdamW optimizer is adopted. The number of occupancy frames is fixed at 8 during training. For Occupancy VAE, the learnable class embedding C' is set to 8 following [43]. The learning rate is set to 1×10^{-3} over 200 epochs. For Occupancy DiT, the diffusion transformer model is based on [22], with a learning rate of 1×10^{-4} over 600 epochs.

In the second stage, the video generation model and the Li-

Figure 5. Visualization of Out-of-Distribution (OOD) occupancy generation, showcasing the model's ability to generalize and produce realistic occupancy outputs in unseen scenarios. The top row displays the generated 3D occupancy grids, while the bottom row presents the corresponding BEV layouts, highlighting the preservation of structural and semantic consistency in OOD cases.

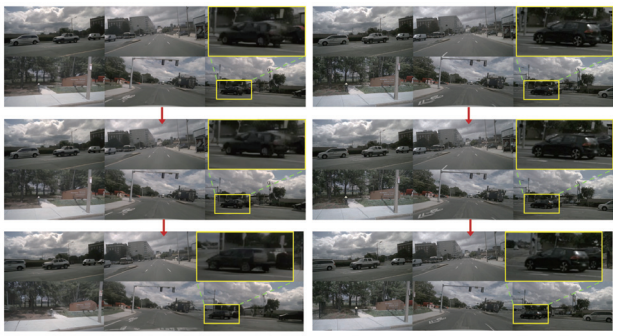

Figure 6. Visualization of the effect on geometric aware noise prior. Our method injects dense appearance priors and incorporates explicit geometric awareness in the sampling process, resulting in high-fidelity and consistent moving cars across different frames.

DAR generation model are jointly trained for 90 epochs with a total batch size of 96 and an initial learning rate of 1×10^{-5}. The training is implemented on 96 NVIDIA A100 GPUs for about 7 days. The video generation model is initialized with pre-trained weights from SVD [3] with a fixed video VAE. The image resolution for training is set to 256×512 following [34], and the number of frames is fixed at 8. Note that we apply the roll-out sampling strategy following [7] in the sampling process to enable long-term generation. The LiDAR generation model is trained from scratch. The AdamW optimizer is leveraged during the training.

7. Comparison Baselines

The occupancy generation method is compared with OccLlama [32], OccWorld [43], and OccSora [27]. OccWorld relies on historical occupancy data and the trajectory of the ego vehicle to predict future occupancy. Building upon OccWorld, OccLlama incorporates a Large Language Model (LLM) to interpret the semantic scene, thereby enhancing forecasting accuracy.

We compare our video generation results with other image and video generation methods, including BEVGen [25], BEVControl [39], DriveGAN [12], DriveDreamer [42], WoVoGen [19], Panacea [34], MagicDrive [6], DriveWM [31], Vista [7]. As mentioned in Sec.4.1 of the main paper, the initial model of Vista [7] only supports single-view video generation. Thus, we implement the multi-view variant of Vista* with spatial-temporal attention following [6, 36] for a fair comparison.

The LiDAR generation results are compared with Open3D [45] and LiDARDM [47]. Open3D employs a hard ray-casting function to generate LiDAR point clouds from occupancy grids. This process involves converting the occupancy grid into a mesh, where each occupied voxel is represented as a cube. The ray depth is determined by calculating the intersection between the ray and the nearest triangular face of the mesh. Note that this method does not provide intensity values or ray drop probabilities. LiDARDM begins the generation process by aggregating multi-frame point clouds and removing dynamic objects using 3D object detection labels. The remaining data is used to reconstruct a mesh via the NKSR algorithm [10]. A mesh diffusion model is trained using this mesh as a 3D representation and the BEV layout as a condition. During inference, a mesh is sampled from the diffusion model based on a BEV layout map. Mesh models of dynamic objects, selected from predefined assets, are then inserted into the generated mesh according to the detection labels of the corresponding frame. Hard ray casting is subsequently applied to obtain the point cloud. Similar to Open3D, LiDARDM does not generate intensity values and requires an additional network to predict ray drop probabilities.

8. More Visualization Results

Out-of-Distribution Occupancy Generation. To further illustrate the strong generalization capability and controllability of our occupancy generation model, we present the Out-of-Distribution (OOD) occupancy generation results, as shown in Fig. 5. These visualizations demonstrate the model's ability to generate realistic occupancy outputs in previously unseen scenarios, while maintaining structural and semantic consistency in OOD cases.

Generalization on other Autonomous Driving Datasets. As shown in Fig. 10, we evaluate the generalization ability

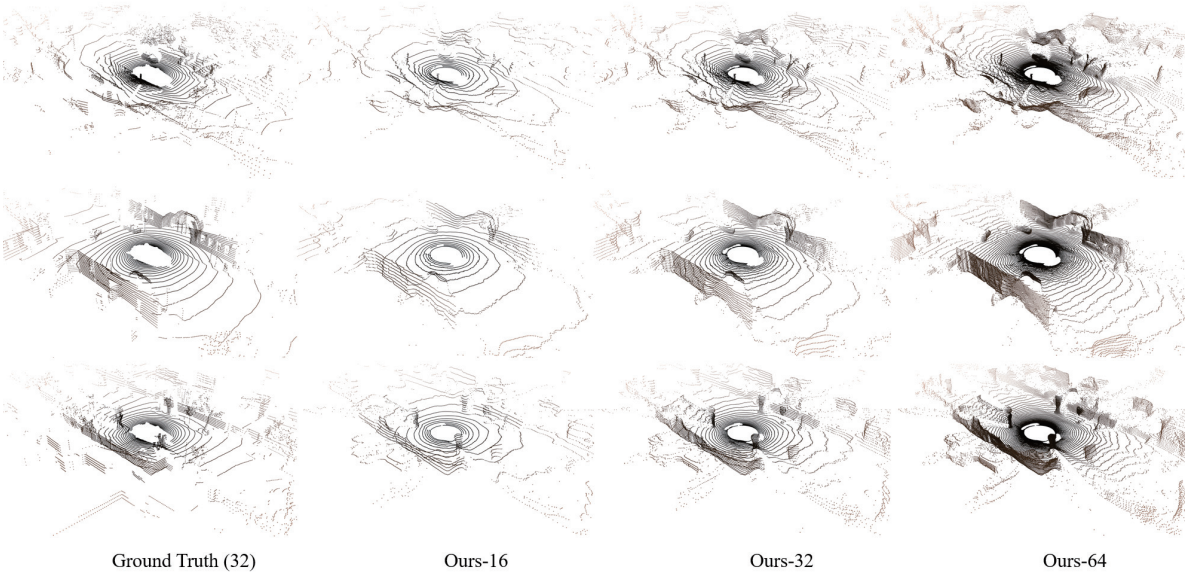

Figure 7. Visualization of LiDAR beam scanning patterns across different configurations. The ground truth utilizes a 32-beam LiDAR setup, while the proposed methods ('Ours-16', 'Ours-32', and 'Ours-64') demonstrate varying levels of beam density. The comparison highlights the model's ability to simulate realistic LiDAR patterns and preserve scene geometry across different beam resolutions.

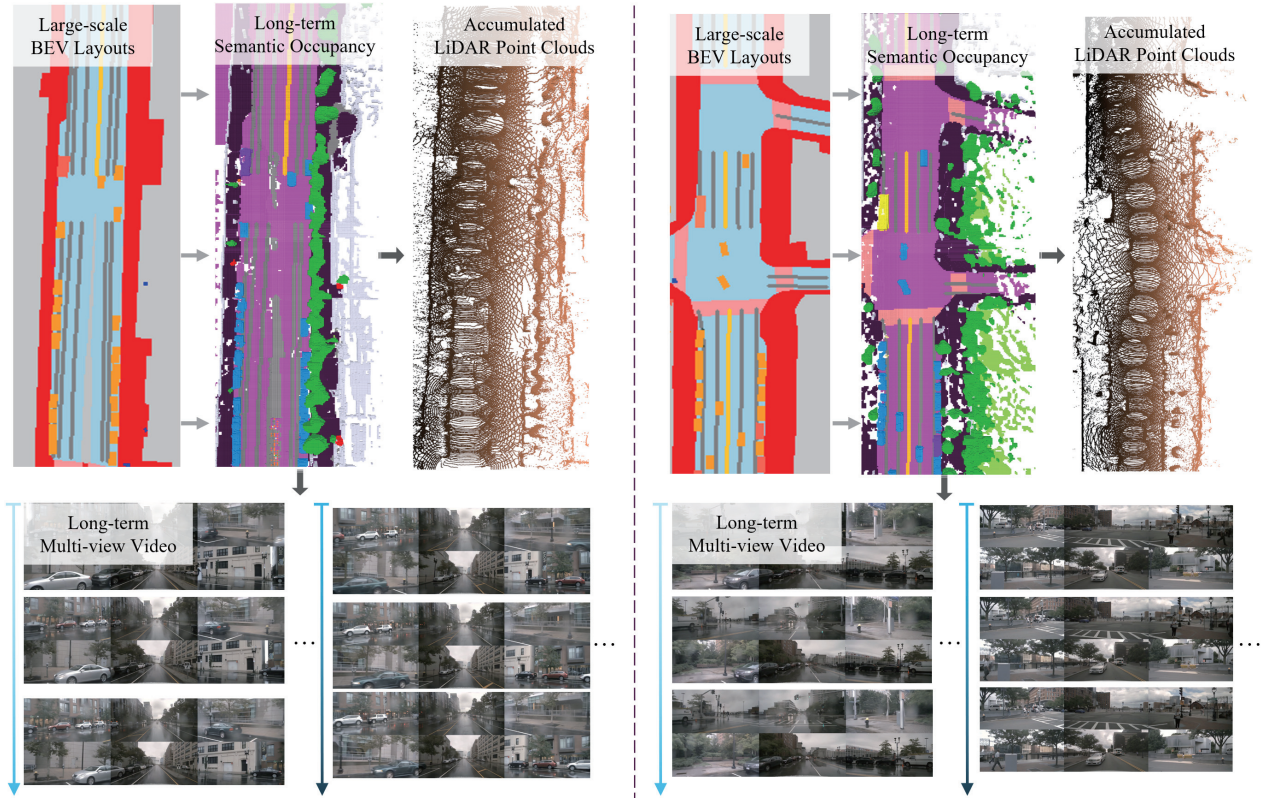

Figure 8. Visualization of large-scale coherent generation, where a large-scale BEV layout serves as the input. The long-term semantic occupancy from the given BEV layouts is first produced, which subsequently guides the generation of LiDAR point clouds and multi-view videos. The results demonstrate the model's ability to produce temporally and spatially consistent outputs in large-scale environments.

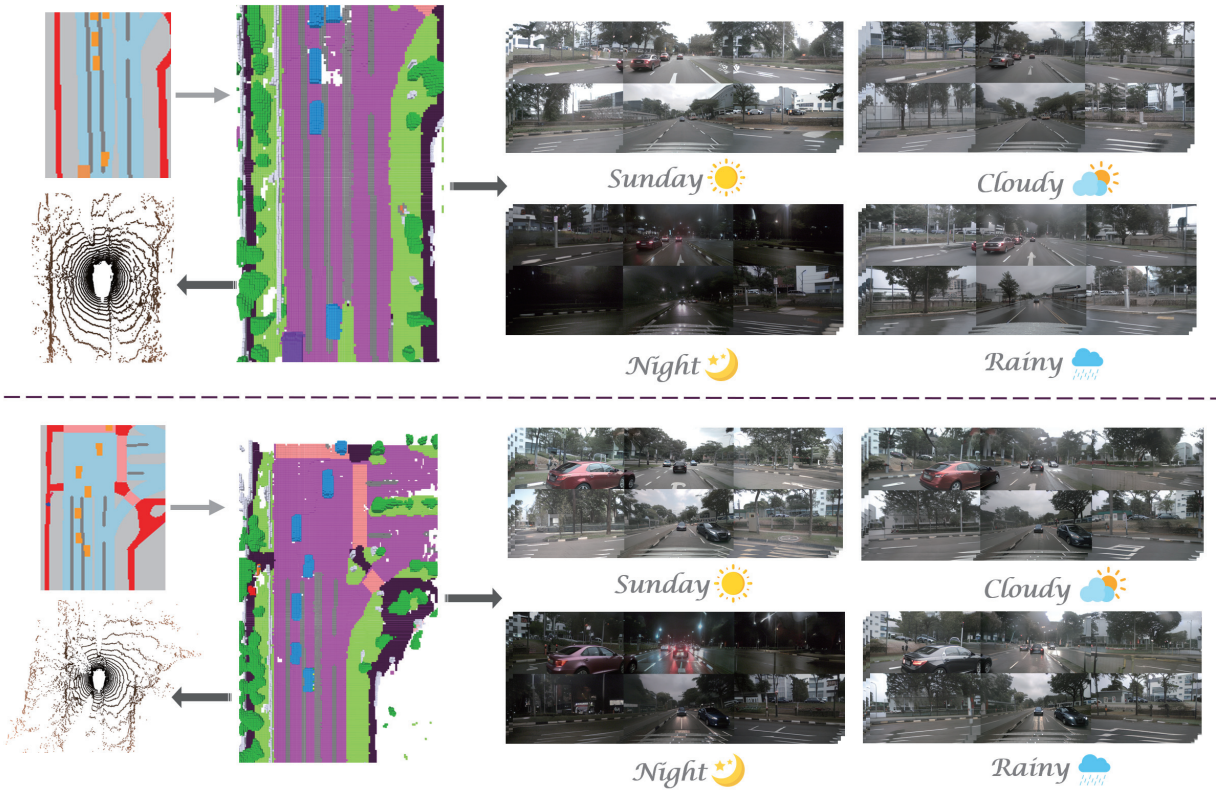

Figure 9. Visualization of controllable video generation with diverse attributes. The pipeline demonstrates the ability to generate videos conditioned on BEV layouts, with control over various attributes such as weather (sunny, cloudy, rainy) and time of day (day, night). The top and bottom rows showcase different input configurations, resulting in realistic video outputs with the desired attribute variations.

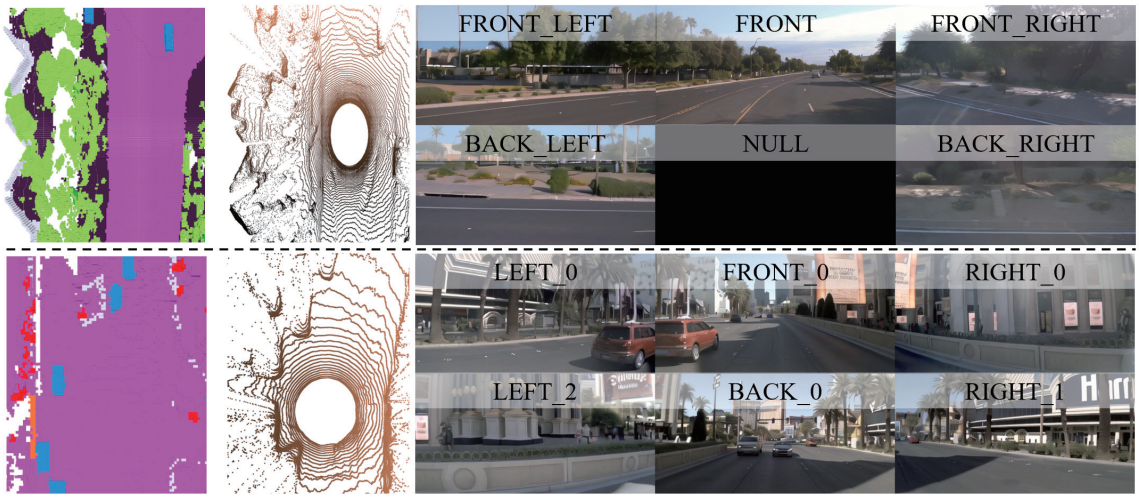

Figure 10. Generalization on the Waymo (upper) and nuPlan (lower) datasets. Due to unavailability, the back view on the Waymo dataset is set to null.

of our model on the Waymo and nuPlan datastes. Given scene layouts, our model can be directly transferred to different datasets with conditional reference images. Note that due to unavailability, the back view on the Waymo

dataset is set to null.

Effect on Geometric Aware Noise Prior. As mentioned in Sec.3.2 of the main paper, the Geometric-aware Noise Prior strategy injects dense appearance priors and incorporates explicit geometric awareness in the sampling process of video generation. The Visualization results are illustrated in Fig. 6. We can see that the video generation quality is obviously improved, resulting in high-fidelity and consistent moving cars across different frames.

Diverse LiDAR Beam Scanning Patterns. Owing to the flexibility of our ray-based LiDAR heads, the scanning pattern of the LiDAR beam can be freely configured. This capability facilitates the generation of LiDAR point clouds with various scanning patterns without necessitating retraining. As illustrated in Fig. 7, whereas the ground truth LiDAR beam is constrained to 32, our method is capable of generalizing to 16, 32, and 64 LiDAR beams. The visualization underscores the model's capability to simulate realistic LiDAR patterns while preserving scene geometry across various beam resolutions.

Versatile Generation Ability of UniScene. To further demonstrate the versatile generation capabilities of our proposed UniScene, we apply it across various scenarios. Figure 8 illustrates UniScene's capacity to generate large-scale, coherent scenes, underscoring the model's ability to produce outputs that are temporally and spatially consistent in extensive environments. The controllable generation of attribute-diverse videos is showcased in Figure 9, highlighting realistic video outputs with desired attribute variations for different input configurations. Additionally, the application of UniScene to scene editing is depicted in Figure 11, demonstrating the model's flexibility in generating consistent and realistic outcomes based on edited scene geometries. The results presented in Figures 12, 13, and 14 further exemplify UniScene's capability to jointly produce high-quality semantic occupancy, video, and LiDAR data, all while maintaining temporal consistency.

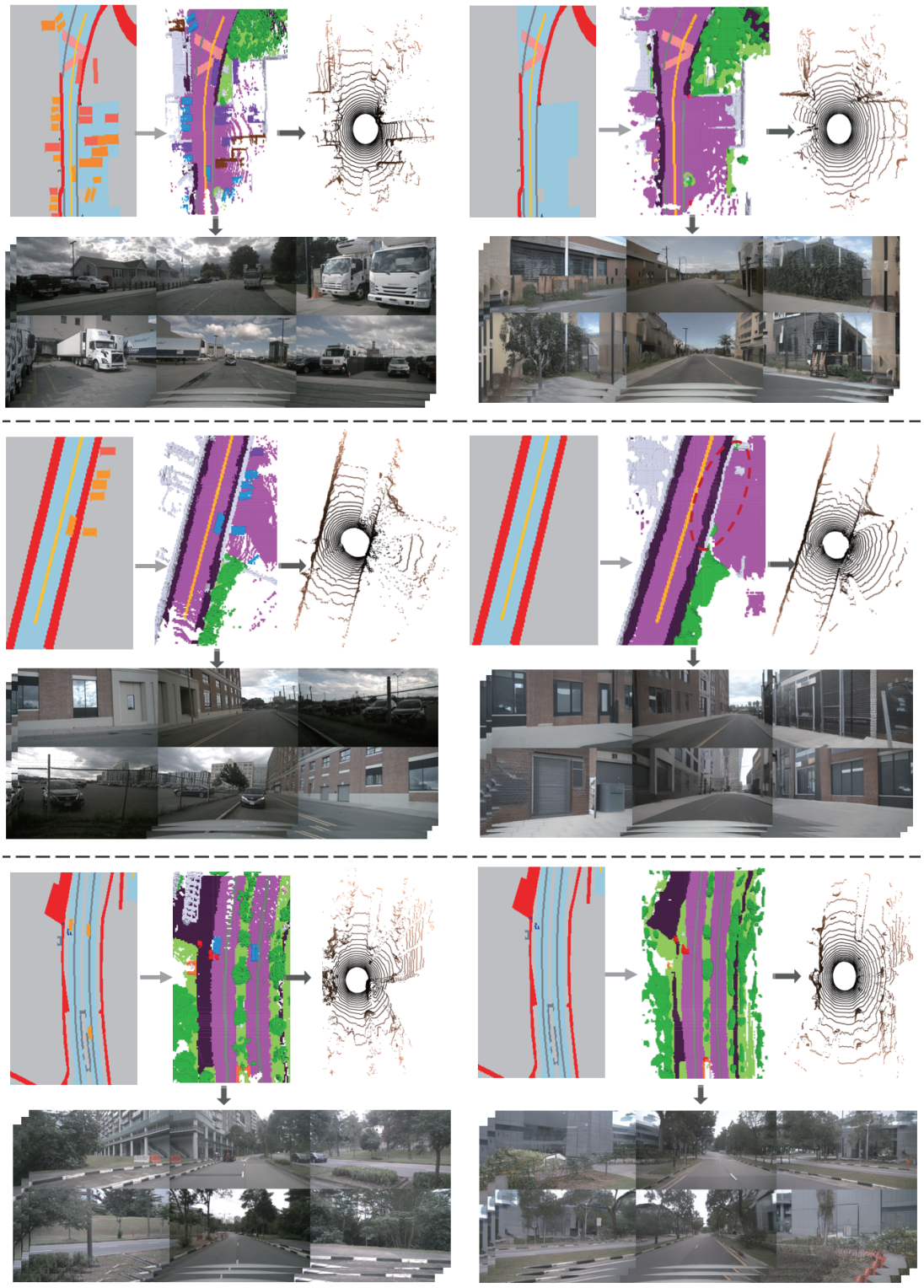

Figure 11. Visualization of controllable generation through geometry editing, demonstrating the ability to manipulate BEV layouts to alter scene geometry. The process begins with modified BEV layouts, followed by the generation of updated semantic occupancy and LiDAR point clouds. These results in multi-view video outputs that reflect the geometric changes, showcasing the model's flexibility in producing consistent and realistic outputs based on edited scene geometries.

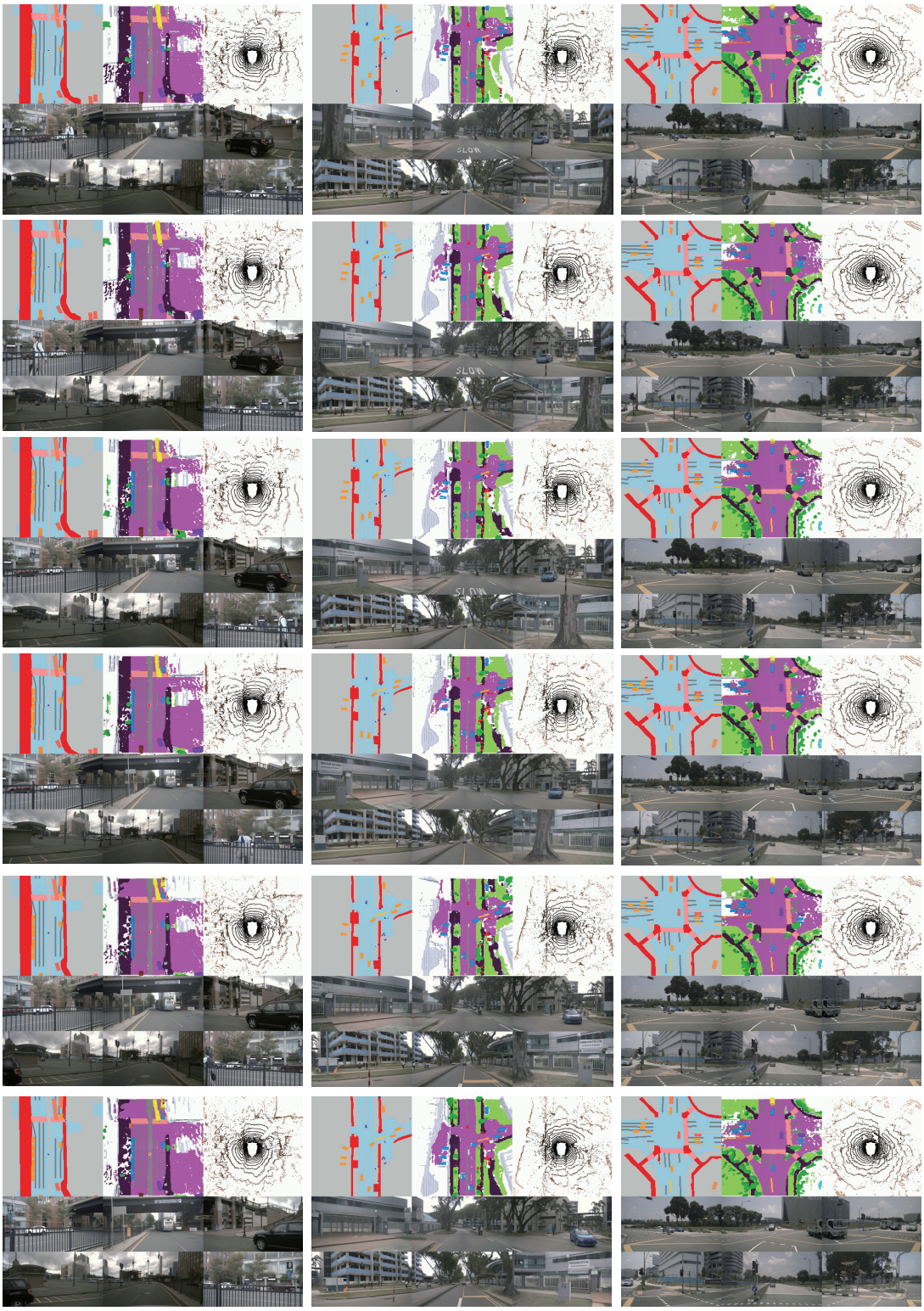

Figure 12. More visualization of unified generation of semantic occupancy, LiDAR point clouds, and multi-view videos.

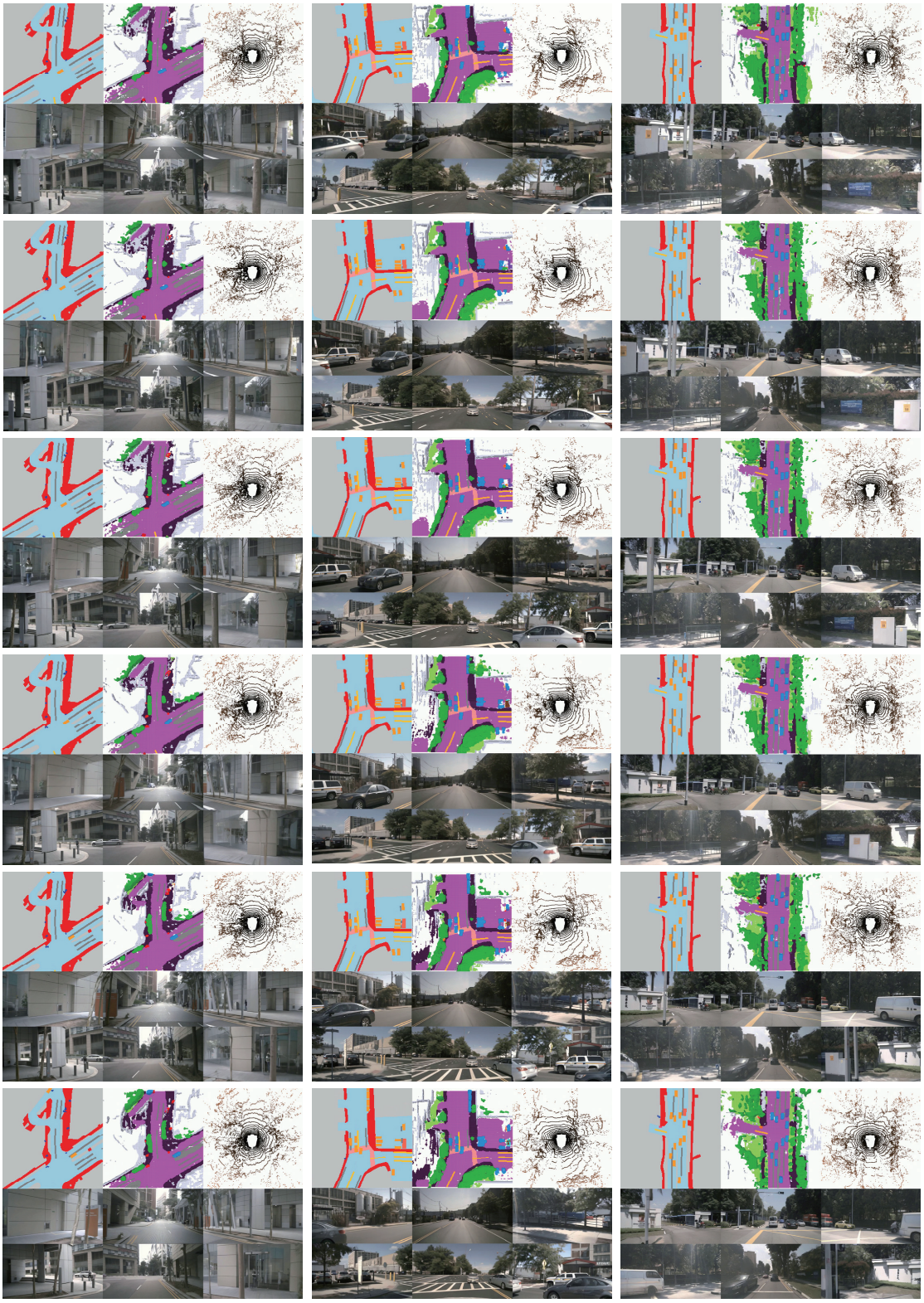

Figure 13. More visualization of unified generation of semantic occupancy, LiDAR point clouds, and multi-view videos.

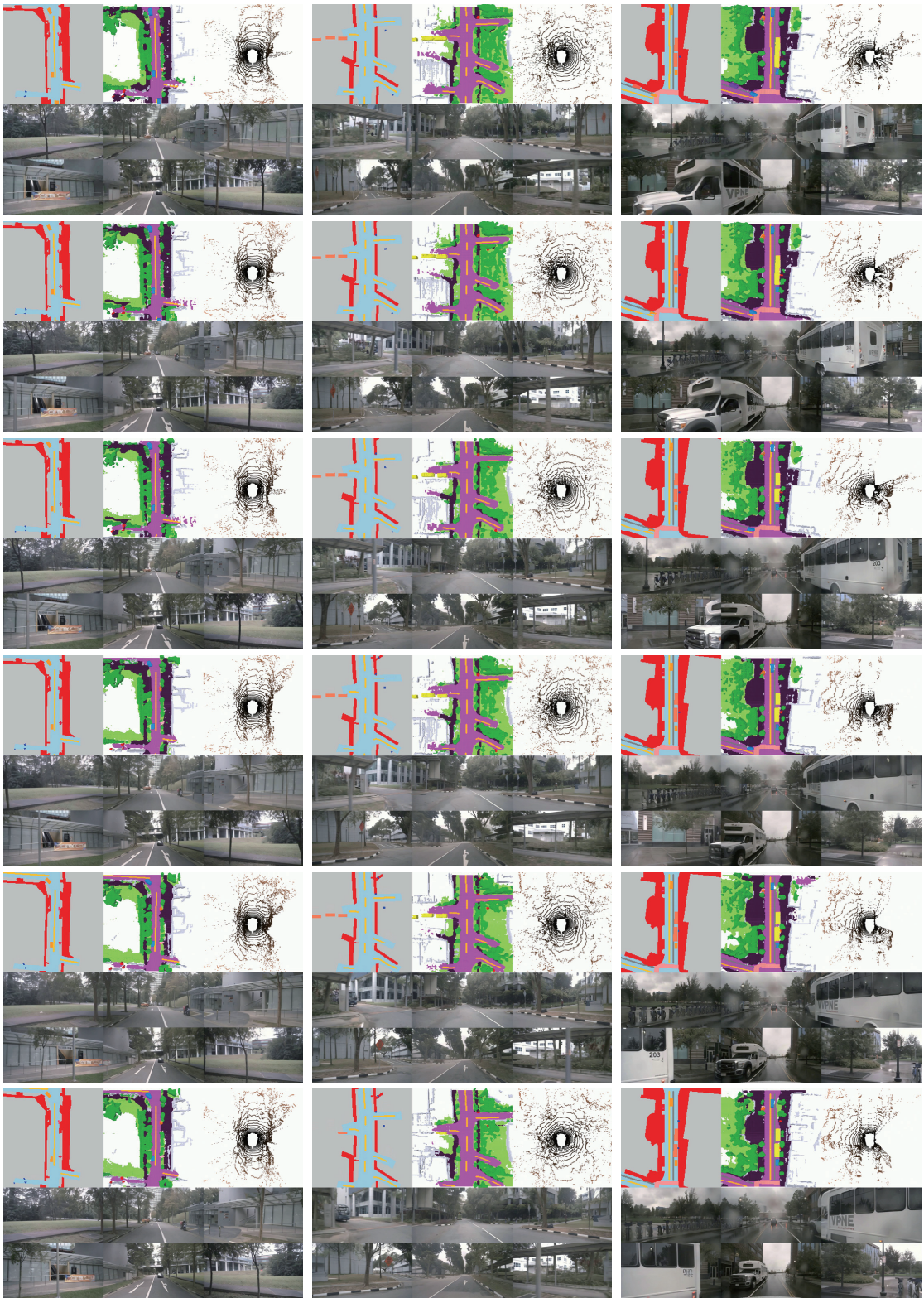

Figure 14. More visualization of unified generation of semantic occupancy, LiDAR point clouds, and multi-view videos.

References

[1] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021. 1

[2] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018. 2

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1, 3, 6

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 5

[5] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022. 5

[6] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *ICLR*, 2024. 1, 5, 6

[7] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *arXiv preprint arXiv:2405.17398*, 2024. 1, 6

[8] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. Lidar snowfall simulation for robust 3d object detection. In *CVPR*, 2022. 1, 2

[9] 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*, 2017. 5

[10] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023. 5, 6

[11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 4

[12] Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation. In *CVPR*, 2021. 6

[13] Jumin Lee, Woobin Im, Sebin Lee, and Sung-Eui Yoon. Diffusion probabilistic models for scene-scale 3d categorical data. *Workshop on Image Processing and Image Understanding*, 2023. 1

[14] Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Juhyeong Seon, and Sung-Eui Yoon. Semcity: Semantic scene generation with triplane diffusion. In *CVPR*, 2024. 1

[15] Bohan Li, Yasheng Sun, Jingxin Dong, Zheng Zhu, Jinming Liu, Xin Jin, and Wenjun Zeng. One at a time: Progressive multi-step volumetric probability learning for reliable 3d scene perception. In *AAAI*, 2024. 5

[16] Leheng Li, Weichao Qiu, Yingjie Cai, Xu Yan, Qing Lian, Bingbing Liu, and Ying-Cong Chen. Syntheocc: Synthesize geometric-controlled street view images through 3d semantic mpis. *arXiv preprint arXiv:2410.00337*, 2024. 1

[17] Yuheng Liu, Xinke Li, Xueting Li, Lu Qi, Chongshou Li, and Ming-Hsuan Yang. Pyramid diffusion for fine 3d large scene generation. *ECCV*, 2024. 1, 5

[18] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*. IEEE, 2023. 5

[19] Jiachen Lu, Ze Huang, Jiahui Zhang, Zeyu Yang, and Li Zhang. Wovogen: World volume-aware diffusion for controllable multi-camera driving scene generation. *arXiv preprint arXiv:2312.02934*, 2023. 1, 6

[20] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 2

[21] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 3

[22] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 1, 5

[23] 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*, 2021. 1

[24] Haoxi Ran, Vitor Guizilini, and Yue Wang. Towards realistic scene generation with lidar diffusion models. In *CVPR*, 2024. 1

[25] Alexander Swerdlow, Runsheng Xu, and Bolei Zhou. Street-view image generation from a bird's-eye view layout. *IEEE Robotics and Automation Letters*, 2024. 5, 6

[26] 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. 5

[27] Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *arXiv preprint arXiv:2405.20337*, 2024. 1, 2, 6

[28] Xiaofeng Wang, Zheng Zhu, Yunpeng Zhang, Guan Huang, Yun Ye, Wenbo Xu, Ziwei Chen, and Xingang Wang. Are we ready for vision-centric driving streaming perception? the asap benchmark. *arXiv preprint arXiv:2212.08914*, 2022. 5

[29] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. *ICCV*, 2023. 5

[30] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *ECCV*, 2024. 1

[31] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *CVPR*, 2024. 1, 5, 6

[32] Julong Wei, Shanshuai Yuan, Pengfei Li, Qingda Hu, Zhongxue Gan, and Wenchao Ding. Occllama: An occupancy-language-action generative world model for autonomous driving. *arXiv preprint arXiv:2409.03272*, 2024. 2, 3, 6

[33] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, 2023. 5

[34] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving, 2023. 6

[35] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *CVPR*, 2024. 1, 5

[36] 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*, 2023. 6

[37] Yuwen Xiong, Wei-Chiu Ma, Jingkang Wang, and Raquel Urtasun. Learning compact representations for lidar completion and generation. In *CVPR*, 2023. 1

[38] Yuwen Xiong, Wei-Chiu Ma, Jingkang Wang, and Raquel Urtasun. Ultralidar: Learning compact representations for lidar completion and generation. *arXiv preprint arXiv:2311.01448*, 2023. 1, 2

[39] 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. 6

[40] Junge Zhang, Qihang Zhang, Li Zhang, Ramana Rao Kompella, Gaowen Liu, and Bolei Zhou. Urban scene diffusion through semantic occupancy map. *arXiv preprint arXiv:2403.11697*, 2024. 1

[41] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 4

[42] Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. *arXiv preprint arXiv:2403.06845*, 2024. 1, 5, 6

[43] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. *arXiv preprint arXiv:2311.16038*, 2023. 2, 3, 5, 6

[44] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, 2022. 5

[45] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 6

[46] Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Learning to generate realistic lidar point cloud. In *ECCV*, 2022. 1, 2

[47] Vlas Zyrianov, Henry Che, Zhijian Liu, and Shenlong Wang. Lidardm: Generative lidar simulation in a generated world. *arXiv preprint arXiv:2404.02903*, 2024. 1, 5, 6