SEM-ROVER: Semantic Voxel-Guided Diffusion for Large-Scale Driving Scene Generation

Hiba Dahmani[1,2], Nathan Piasco[1], Moussab Bennehar[1], Luis Roldão[1], Dzmitry Tsishkou[1], Laurent Caraffa[3], Jean-Philippe Tarel[2], and Roland Brémond[2]

[1] Noah's Ark, Huawei Paris Research Center, France
[2] COSYS, Gustave Eiffel University, France
[3] LASTIG, IGN-ENSG, Gustave Eiffel University, France

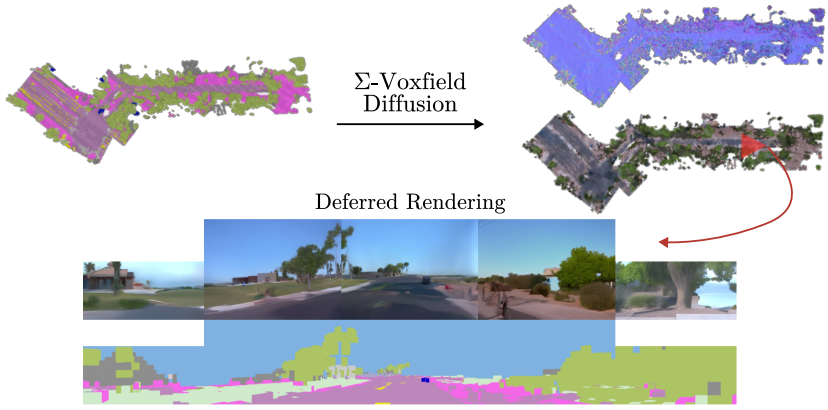


Fig. 1: Our model generates large-scale 3D driving scenes given a coarse semantic voxel grid. Here we show a generated large-scale driving scene spanning ≈100 000 m2

Abstract. Scalable generation of outdoor driving scenes requires 3D representations that remain consistent across multiple viewpoints and scale to large areas. Existing solutions either rely on image or video generative models distilled to 3D space, harming the geometric coherence and restricting the rendering to training views, or are limited to small-scale 3D scene or object-centric generation. In this work, we propose a 3D generative framework based on Σ-Voxfield grid, a discrete representation where each occupied voxel stores a fixed number of colorized surface samples. To generate this representation, we train a semantic-conditioned diffusion model that operates on local voxel neighborhoods and uses 3D positional encodings to capture spatial structure. We scale to large scenes via progressive spatial outpainting over overlapping regions. Finally, we render the generated Σ-Voxfield grid with a deferred rendering module

to obtain photorealistic images, enabling large-scale multiview-consistent 3D scene generation without per-scene optimization. Extensive experiments show that our approach can generate diverse large-scale urban outdoor scenes, renderable into photorealistic images with various sensor configurations and camera trajectories while maintaining moderate computation cost compared to existing approaches.

Keywords: 3D Diffusion · Semantic Voxel Grid · Surface Tokens · Large-Scale Scene Generation

1 Introduction

Generation of 3D outdoor driving scenes is central to simulation, data synthesis, and controllable scene editing. These applications require a scene representation that remain consistent across viewpoints, scale to large spatial extents, and enable free rendering rather than fixed views tied to a predefined camera trajectory.

Existing approaches only partially satisfy these requirements. Occupancy- or layout-based methods capture coarse structures but often miss surface details and realistic appearance. 3D convolutional backbones scale poorly to large scenes because they require processing dense (or near-dense) voxel volumes, and their compute and memory grow rapidly with 3D grid size, making large-extent, high-resolution generation impractical.

Another solution consists of distilling image or video generative models into a common 3D representation, bypassing the scalability limitation of aforementioned approaches. Image- and video-based diffusion models [7, 13–15] can generate photorealistic observations, but their outputs do not form a persistent 3D scene and, therefore, provide limited viewpoint consistency and poor editability. Large-scale world models [19] improve coverage, yet commonly rely on implicit or highly compressed representations that are difficult to render efficiently and to manipulate in a structured way. Consequently, jointly achieving 3D consistency, scalability, and photorealistic rendering in one framework remains challenging.

We address this problem by generating scenes *directly in 3D* using a discrete surface representation, the *Σ-Voxfield grid*. Each occupied voxel stores a fixed number of colorized surface samples, yielding discrete tokens that jointly encode local geometry and appearance while remaining aligned in world coordinates. To generate this representation, we train a *semantic-conditioned diffusion model* that operates on spatially localized neighborhoods of Σ-Voxfield tokens and uses 3D positional encoding to capture spatial structure. Since the model is applied to local neighborhoods, computation stays bounded. To synthesize large scenes, we progressively expand the grid via *spatial outpainting* over overlapping regions, enforcing continuity across neighborhood boundaries.

Finally, the generated Σ-Voxfield grid provides a persistent 3D scene buffer that can be rendered from arbitrary viewpoints without per-scene optimization. We render this buffer efficiently and produce photorealistic images using a deferred rendering module conditioned on the rendered Σ-Voxfield output, which

compensates for surface discretization and missing content such as sky and distant background. Extensive experiments show that our approach scales to large scenes with moderate computation cost compare to competitors.
In summary, our contributions are as follows:

- We introduce a novel 3D representation tailored for 3D generative modeling, **Σ-Voxfield**, a fixed-cardinality discrete surface approximation representing the geometric and photometric field within local voxel.
- To jointly generate the photometric and geometric characteristics of urban scenes, we convert Σ-Voxfield grids to unordered tokens with additional 3D positional encoding for training a semantically conditioned transformer-based diffusion model.
- We scale synthesis via **progressive spatial outpainting** in Σ-Voxfield space, enabling large-scene generation while maintaining a constant computation budget.
- We couple our 3D generation with **deferred rendering** to obtain photo-realistic images conditioned on a persistent 3D buffer, without per-scene optimization.

2 Related Work

2.1 Diffusion Models for Driving Scene Generation

Recent works in driving scene generation largely rely on diffusion models in image or video space. While methods like DreamDrive [17] and MagicDrive [7] synthesize realistic, temporally coherent videos conditioned on text prompts, trajectories, or layouts, their outputs remain tied to specific inference trajectories and do not provide a persistent scene representation in world coordinates.

Several approaches extend this paradigm by reconstructing the underlying 3D structure from intermediate view generation. MagicDrive3D [6] combines multi-view video diffusion with 3D Gaussian Splatting (3DGS) reconstruction, while GEN3C [13] performs video diffusion conditioned on a 3D cache and decodes latent videos into RGB frames, relying on precomputed geometry. InfiniCube [15] scales this design through a pipeline that integrates voxel-level generation, video synthesis, and feed-forward 3DGS reconstruction. More recently, ScenDi [8] proposes a 3D-to-2D diffusion cascade where coarse latent 3D Gaussians are generated and subsequently refined through view-conditioned 2D diffusion. Despite strong visual quality, these methods typically obtain 3D structure through intermediate rendering, reconstruction, or cascaded refinement, rather than directly generating a persistent surface representation in 3D space.

Complementary work explores structured spatial priors for large-scale environments. LSD-3D [20] leverages layout and point cloud conditioning but couples generation with optimization of large Gaussian sets, leading to high memory usage and long per-scene runtimes. Similarly, Urban Architect [14] generates scenes from semantic layouts and reconstructs geometry via optimization of implicit fields guided by 2D diffusion model SDS distillation [21]. Overall, these

Method	Prior	Pipeline	Feed Forward
Urban Architect	3D Layout	2D Diff + NeRF	×
LSD-3D	PC + Boxes	2D Diff + 3DGS	×
GEN3C	Text/Image + PC	Video Diff	✓
MagicDrive3D	HDMap + Text + Traj	Video Diff. + 3DGS	×
InfiniCube	HDMap + Text + Traj.	Vox Diff + Video Diff + FF 3DGS	✓
Ours	Semantic Voxels	Vox Diff. + Deferred rendering	✓

Table 1: Driving-scene generation methods. We review relevant works in terms of required priors and computation pipelines. We explicitly distinguish per-scene optimization from feed-forward diffusion pipelines.

approaches tightly couple view synthesis with reconstruction or optimization, which can limit scalability, editability, and consistent novel-view rendering. In contrast, our method generates a Σ-Voxfield grid directly in 3D with a semantic-conditioned diffusion model, and scales to large scenes via voxel-space outpainting without per-scene optimization. In the Table 1, we provide a comparison of major characteristics of relevant methods for large-scale driving scenes generation.

2.2 Diffusion over Structured 3D Representations

Extending diffusion models to 3D remains challenging due to the high dimensionality and sparsity of 3D spatial data, especially in urban outdoor scenes. Prior works explore diffusion over point clouds, voxels, and sparse grids [18,22,30], but computation often scales with spatial resolution, making large-scale scene synthesis expensive. Primitive-based formulations, such as GaussianCube [31] and DiffusionGS [2], provide explicit and renderable representations, yet are often demonstrated on object-centric or spatially bounded settings.

Latent 3D diffusion improves efficiency by operating in compressed spaces. For instance, L3DG [23] models vector-quantized 3D Gaussian representations using latent diffusion with sparse convolutional encoders, enabling efficient room-scale generation. Relevant to our method, TRELLIS [28] uses a 1-D transformer to diffuse object-centric sparse 3D latent space that can be decoded into various 3D representations. Notably, the 3D conditioning is provided through positional embedding rather than using an explicit 3D operator, such as a 3D convolution block. Their method is limited to small-scale scene or object generation.

Our work differs by diffusing discrete surface tokens directly in 3D space and scaling generation through progressive outpainting, keeping per-step computation bounded while producing a persistent, renderable scene representation.

3 Method

We design our generative framework to meet three key criteria: 3D consistency, scalability, and photorealistic rendering. To ensure the 3D coherence and consistency of our generation, we perform the generation process in the 3D space,

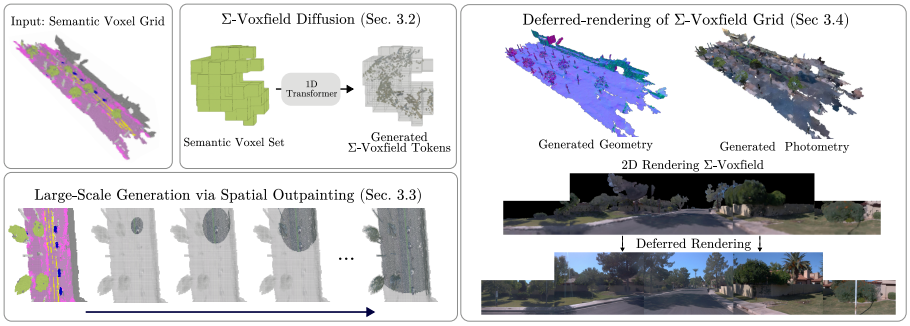


Fig. 2: Overview of SEM-ROVER. Our framework performs generation directly in 3D to ensure consistent geometry and appearance. It represents the scene with a Σ-Voxfield grid and applies transformer-based diffusion over 3D tokens. To scale to large environments efficiently, we use an iterative outpainting strategy. Finally, a deferred rendering engine converts the generated 3D scene into photorealistic views.

rather than distilling 2D generated information into a 3D model. We introduce in Section 3.1 our 3D representation, Σ-Voxfield grid, a local and discrete representation of a colorized surface field designed to be diffused as 3D tokens with a transformer, as explained in Section 3.2. To enable large-scale synthesis while maintaining a reasonable computational budget, we introduce an iterative outpainting method in Section 3.3. Finally, we couple our 3D generation pipeline with a deferred rendering engine to produce photorealistic images as explained in Section 3.4. The overall architecture of our framework is illustrated in Figure 2.

3.1 Σ-Voxfield: a joint geometric and photometric representation

Definition. To describe a large outdoor driving scene, we propose to use Σ-Voxfield grid. A Σ-Voxfield is a local and discrete representation of a colorized surface field. It is defined at a voxel level and is parametrized by the voxel size v_s and the Σ-Voxfield cardinality n. Each Σ-Voxfield is composed of n 3D points with associated RGB color, sampled on the surface of the scene lying within the boundary of the voxel. Formally, a Σ-Voxfield v_Σ is defined as:

$$v_\Sigma = \left\{(x^i, y^i, z^i), (r^i, g^i, b^i)\right\}_{i\in[1,n]}, \tag{1}$$

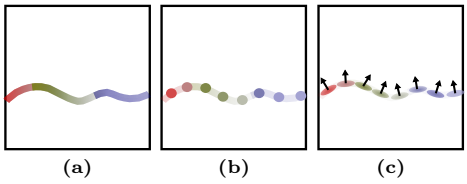


Fig. 3: Simplified illustration of 2D Σ Voxfield. a) A 2D voxel containing a continuous colorized surface field. b) 2D Σ Voxfield points, uniformly sampled on the surface. c) Rendering method of the Σ Voxfield, where each point is replaced by a 2D Gaussians aligned with the implicit surface.

with (x^i, y^i, z^i) the 3D position of the sampled point i, defined relative to the voxel center, and (r^i, g^i, b^i) its color.

Properties. A Σ-Voxfield represents both the local geometry and the appearance of the scene with a fixed number of points, making this representation an ideal choice for generative 3D modeling. Indeed, geometry and photometry are entangled properties of a scene; it is fundamental to generate them jointly to capture the complexity of outdoor scenes. Moreover, given the fixed cardinality of Σ-Voxfield, it is straightforward to consider each Σ-Voxfield as a token within a transformer architecture.

2D Rendering. Σ-Voxfield is a point-based representation that can be easily rendered into 2D images. However, like any point cloud rendering, the completeness of the 2D rendering will depend a lot on the density of the point cloud. In order to maintain the cardinality of Σ-Voxfield sufficiently low to be tractable for a transformer model, we propose a conversion method that will increase the completeness of 2D rendering of Σ-Voxfield without increasing the number of sampled points. For each point in the Σ-Voxfield, we create a 2D Gaussian aligned with the surface implicitly present in our representation. Formally, we compute a local normal (n_x^i, n_y^i, n_z^i) via PCA over spatial neighbors and use this normal to initialize rotation matrices $R^i \in SO(3)$ that align each Gaussian with the local tangent plane to the surface. We fix the scale factor along the plane axis to ensure optimal coverage for the 2D rendering. A simplified illustration of Σ-Voxfield definition and properties is shown in Figure 3.

Σ-Voxfield grid conversion. Given a textured mesh of a scene, we can easily obtain the counterpart Σ-Voxfield grid representation. We first voxelize the 3D scene with voxel size v_s, then discard all the empty voxels. For each remaining voxel, we uniformly sample n points on the textured mesh surface lying within the voxel to obtain the Σ-Voxfield grid.

3.2 Σ-Voxfields Diffusion

We perform diffusion over *local sets* of Σ-Voxfields. We denote by $\mathcal{G}$ the set of all non-empty Σ-Voxfields in a scene. We consider a local subsets of adjacent voxels $\mathcal{X}_\xi \subset \mathcal{G}$ containing at most N_ξ Σ-Voxfields.

Each Σ-Voxfield $v_\Sigma \in \mathcal{X}_\xi$ is represented by channel-wise stacking of its n surface samples:

$$\psi(v_\Sigma) = \left[x^1, y^1, z^1, r^1, g^1, b^1, \ldots, x^n, y^n, z^n, r^n, g^n, b^n\right] \in \mathbb{R}^{6n}. \tag{2}$$

We order the points stacked in v_Σ by increasing distance to the Σ-Voxfield center, so that $\psi(v_\Sigma)$ is defined deterministically.

Semantic conditioning. We associate for each Σ-Voxfield v_Σ a semantic label s_{v_Σ} (e.g., road, sidewalk, building, vegetation, etc.). The model is conditioned on the semantic labels $\{s_{v_\Sigma}\}_{v_\Sigma \in \mathcal{X}_\xi}$.

3D positional embedding. We also denote by $\mathbf{x}_{v_\Sigma}$ the center location of each v_Σ. The center locations are used through 3D positional encodings to expose the 3D structure of the scene to the transformer. Formally, we compute a sinusoidal positional encoding from the 3D coordinates of each $v_\Sigma \in \mathcal{X}_\xi$, project it through a learnable layer and then sum it with the corresponding noisy token.

We diffuse the set $\{\psi(v_\Sigma) \mid v_\Sigma \in \mathcal{X}_\xi\}$ with a 1D Diffusion Transformer [5] architecture by applying the standard forward diffusion process:

$$q(\psi(\mathcal{X}_{\xi,t}) \mid \psi(\mathcal{X}_{\xi,0})) = \sqrt{\bar{\alpha}_t}\,\psi(\mathcal{X}_{\xi,0}) + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

where t denotes the time step and $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ is the cumulative product of the noise schedule. We train our model f_θ with *sample prediction*:

$$\widehat{\psi}(\mathcal{X}_{\xi,0}) = f_\theta(\psi(\mathcal{X}_{\xi,t}), t, \mathbf{S}_\xi), \qquad \mathbf{S}_\xi = \{(s_{v_\Sigma}, \mathbf{x}_{v_\Sigma}) \mid v_\Sigma \in \mathcal{X}_\xi\}. \tag{4}$$

We minimize an ℓ_2 loss between $\widehat{\psi}(\mathcal{X}_{\xi,0})$ and $\psi(\mathcal{X}_{\xi,0})$. At inference, we run the reverse process to sample $\mathcal{X}_{\xi,0}$ from noise.

3.3 Large-Scale Scene Generation via Spatial outpainting

Our diffusion model operates on local sets, keeping computational cost constant but constraining the spatial extent generated per denoising pass. To synthesize larger scenes, we progressively generate the full Σ-Voxfield grid by *outpainting* new regions while conditioning on the existing neighborhood using the Repaint [16] diffusion scheduler.

Given a local set $\mathcal{X}_\xi$ containing denoised and noisy Σ-Voxfield tokens, we partition it into a *known* part and a *target* part:

$$\mathcal{X}_\xi = \{\mathcal{X}_\xi^{\text{known}}, \mathcal{X}_\xi^{\text{target}}\}.$$

We keep $\mathcal{X}_\xi^{\text{known}}$ fixed and diffuse only $\mathcal{X}_\xi^{\text{target}}$. During reverse diffusion, each denoising step overwrites $\mathcal{X}_\xi^{\text{known}}$ with its fixed values and updates only $\mathcal{X}_\xi^{\text{target}}$, propagating geometry and appearance consistently across overlapping local sets. Details of partitioning an initial scene $\mathcal{G}$ into overlapping subsets $\mathcal{X}_\xi$ are provided in the supplementary materials.

Scalability. This progressive formulation decouples generation cost from scene size: the model is always applied to local sets of maximum size N_ξ Σ-Voxfields, yet can be iterated to expand to arbitrarily large extents. Thus, we synthesize scenes with tens of thousands of Σ-Voxfields with linearly growing inference-time while keeping memory and compute comparable to a single denoising process.

3.4 Deferred-rendering of Σ-Voxfield grid

The model introduced in Section 3.2, coupled with our outpainting strategy described in Section 3.3 generates a Σ-Voxfield grid representing a large outdoor scene. The representation can be rendered efficiently into 2D images using

2DGS [9], as explained in Section 3.1. The rendered frames are by nature 3D consistent because Σ-Voxfield represents a discretized version of the scene surfaces, it cannot be used as is for most downstream applications requiring high-fidelity rendering. We propose using the rendered images from the Σ-Voxfield grid as the 3D-buffer input of a deferred-rendering module.

Rendering engine. We denote $I_\Sigma(w)$ the rendered image from the Σ-Voxfield grid at pose w and $I(w)$ the real image of the scene at the same pose. Our rendering module can be defined as a function R that outputs an image $I_{DR}(w)$ conditioned on $I_\Sigma(w)$: $I_{DR}(w) = R(I_\Sigma(w))$. Notice that I_Σ does not necessarily contain all the necessary information to be decoded into I_{DR}, because: 1. $I_\Sigma(w)$ is a simplification through decimation of the real geometry and photometry of the scene and 2. some parts of the scene may not be covered by the Σ-Voxfield grid, such as distant background and sky region as shows Figure 2 (2.4). For these reason, we propose to use a diffusion model with generative capability to implement our rendering engine R.

Diffusion rendering. We use a modified version of Stable Diffusion as our generative 2D deferred rendering engine. Stable Diffusion is a latent diffusion UNet model [24] that iteratively denoises a Gaussian variable x_T to reconstruct the original data sample x_0. x_0 is obtained by computing the latent representation of the original image I with a Variational Autoencoder (VAE) [10]. The denoising network R_ϕ, is trained to predict noise conditioned on input signals x_Σ (the latent representation of I_Σ) by minimizing:

$$\mathcal{L}(\phi) = \mathbb{E}_{t,\epsilon} \| R_\phi(x_t, x_\Sigma) - \epsilon \|_2^2, \tag{5}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is the additive Gaussian noise, $t \sim \mathcal{U}(0, T)$ is the time step, x_t is the noisy latent at t.

Sky and background modeling. To avoid hallucinating geometry in areas not covered by our 3D buffer, we use an additional visibility mask as conditioning to indicate sky and background regions. During training, this mask is computed by segmenting the sky region in the real images, while at inference, we use a binary mask derived from the 3D buffer I_Σ, indicating area without any 2D Gaussians.

Temporal consistency. Area covered by our 3D buffer can be decoded almost deterministically by our diffusion model, as I_Σ contain the coarse shape and colors of objects in the scene. However, high frequency details and area not covered by our 3D buffer are generated stochastically and can change subtly from one pose to another. To ensure temporal consistency of a generated sequence of images, we implement two variants for our diffusion renderer:

- Autoregressive Stable Diffusion (ASD): inspired by GameNGen [26], we use as an additional conditioning to our diffusion model the previously predicted frame to ensure temporal consistency.
- Video Stable Diffusion (VSD): we train a VSD [1] to generate 12 frames at each rendering step, ensuring a better temporal coherence compared to ASD at a cost of higher memory consumption.

More details about these models can be found in our supplementary materials.

3.5 Data processing

Large scale textured mesh computation. We build the Σ-Voxfield grid from multi-view driving sequences using a geometry-based preprocessing pipeline. We reconstruct each scene with OmniRe [4], a 3DGS method for urban dynamic scenes with an additional normal supervision regularizer obtained via DepthAnything [12] monocular prior. From this reconstruction, we extract optimized poses and depth maps of the static background, fuse depths into an SDF, and extract a surface mesh $\mathcal{M}$ via Marching Cubes. We texture $\mathcal{M}$ by aggregating multi-view RGB observations using OpenMVS [3]. Example of computed mesh and additional details about our data pre-processing pipeline can be found in the supplementary.

Semantic Σ-Voxfield tokens computation. Σ-Voxfield grids are obtained from the textured mesh as explained in Section 3.1. We also compute an aligned semantic voxel grid for our diffusion model conditioning by back-projecting and aggregating per-frame semantic segmentation with a modified TSDF fusion method.

Dataset computation for deferred rendering. To obtain pairs of images $\{I_\Sigma, I\}$ necessary to train our deferred rendering diffusion module, we render for each camera and at each pose w the Σ-Voxfield image $I_\Sigma(w)$. Because the original corresponding image $I(w)$ may contain dynamic objects, we render using the 3DGS static field of the trained model a static image $\hat{I}_s(w)$.

4 Experiments

4.1 Experimental setup

Datasets. We evaluate on two large-scale autonomous driving datasets: Waymo Open Dataset (WOD) [25] and PandaSet [29]. Training scene splits are detailed in the supplementary materials.

Σ-Voxfield parameters. In all our experiment We use Σ-Voxfield with voxel size $v_s = 0.6m$, $n = 20$ sampled points and local sets of Σ-Voxfields $\mathcal{X}_\xi$ composed of $N_\xi \in [50, 150]$ Σ-Voxfields. $\mathcal{X}_\xi$ typically cover a scene of $\sim 4{\times}4m^2$, a good tradeoff between number of 3D points and scene context. Given these parameter, we choose $r = 0.04m$ as fixed splat radius for 2DGS Σ-Voxfield rendering.

Model architecture and training details. For our Σ-Voxfield diffusion backbone, we use a 1-Dimensional DiT diffusion backbone with masked attention over voxel tokens, restricting each token to attend only to voxels within a 3-meter neighborhood. The model is trained with 1,000 denoising steps, ADAM optimizer and learning rate of $5e-4$. During training, we randomly drop the semantic conditioning with a probability of 10% to enable classifier-free guidance. At inference time, we apply classifier-free guidance with a scale of 4.0. All models are trained for 4 days on 2×24GB GPUs with performance comparable to an RTX 4090. ASD deferred renderer is finetuned from SD 1.5 with ADAM

optimizer and a learning rate of $5e-5$ on 1 GPU for approximately 4 days. More training hyperparameters can be found in our supplementary materials.

Competitors. We compare our proposal to two SOTA methods for large scale scene generation with different generation paradigm. GEN3C [13] is a video diffusion based on the powerful COSMOS diffusion backbone [19] with additional point cloud conditioning to guide the generation. Similar to LSD-3D [20], we condition GEN3C on our initial rendering. InfiniCube [15] is a multi-step pipeline that start to generate a fine voxel grid from an HDMap, followed by a video diffusion model conditioned on the generated geometry. The generated frames are used by a feedforward 3DGS network to obtain the final reconstruction.

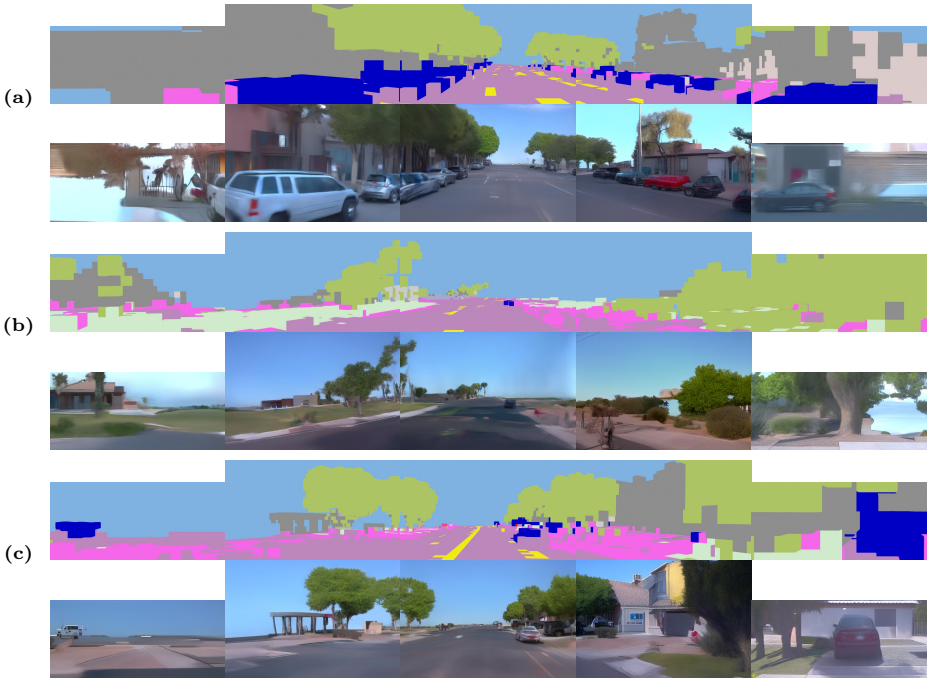


Fig. 4: Qualitative results on WOD. We show three WOD scenes (a–c). For each scene, the **top** strip visualizes the semantic voxel rendering used for conditioning, and the **bottom** strip shows the corresponding generated scene from 5 camera views.

4.2 Qualitative Results

We evaluate on **WOD** (Figure 4) and **PandaSet** (Figure 5) datasets and present qualitative evidence that Σ-Voxfields Diffusion generates scenes that first adhere to the semantic voxel prior in global structure, secondly remain locally coherent near object regions and semantic boundaries, and finally are multi-view consistent under camera motion and multi-camera setup. On WOD, generations capture road topology and surrounding layout while maintaining stable appearance

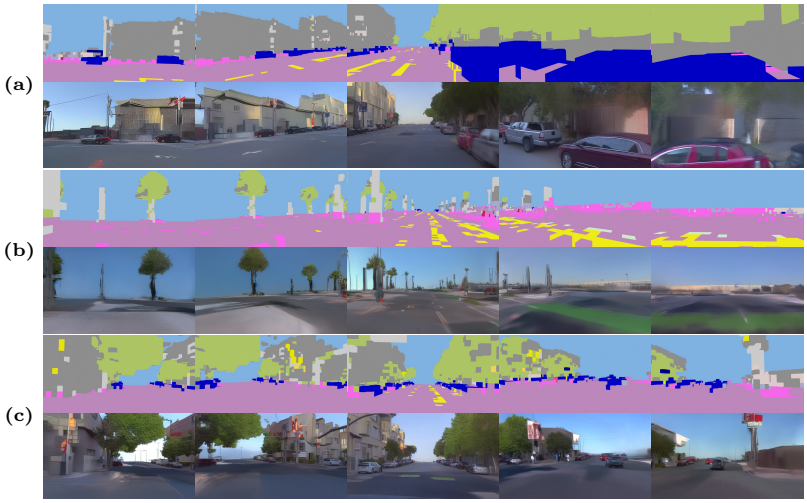


Fig. 5: Qualitative results on PandaSet. We show three PandaSet scenes (a–c). For each scene, the **top** strip visualizes the semantic voxel rendering used for conditioning, and the **bottom** strip shows the corresponding generated scene from 5 camera views.

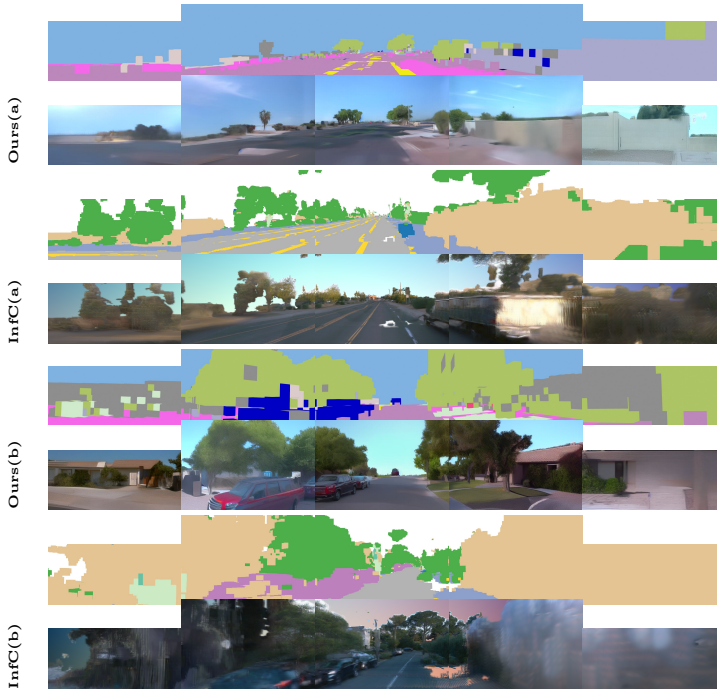


Fig. 6: Qualitative comparison to InfiniCube [15]. For two waymo scenes (a) and (b), we show five camera of the semantic conditioning and below the rendered generated scenes for Our method (Ours) and Infinicube (InfC).



Fig. 7: Qualitative results: generative capabilities., the **top** row visualizes the conditioning signal, and the two rows below show **different generations** sampled from the same conditioning, highlighting output diversity under fixed structure.

across consecutive views. We observe the same behavior on PandaSet, indicating that the method transfers well across datasets. Figure 6 further compares our method to InfiniCube [15]. Across both scenes and multiple viewpoints, our generations better preserve scene geometry and layout consistency in a multi view setting. In particular, for side-camera viewpoints, our method produces more plausible and stable results, whereas InfiniCube degrades noticeably, consistent with its training primarily done with front-facing camera trajectories. As illustrated by the semantic voxel grids in Figure 6, InfiniCube uses a fine voxel conditioning (voxel size of 0.1m), whereas our coarser grid (voxel size of 0.6m) intentionally leaves more room for generative geometry, allowing greater shape variability.

Finally, Figure 7 illustrates the **generative capability** of our model by producing multiple plausible scenes from the same conditioning signal. While the semantic scaffold constrains the global layout, different samples vary in local geometry and appearance (e.g., surface details and textures) while remaining consistent with the conditioning and preserving multi-view coherence.

4.3 Quantitative Results

We report **inference characteristics** and **rendered-view image quality** measured by FID/KID in Table 2. To compute these metrics, we render images from generated scenes and compute FID/KID against real frames from the corresponding ground-truth sequences, evaluated on *seen* views (ground-truth poses) and *novel* views (poses shifted from the original trajectory).

Rendered-view image quality. Table 2 shows that our method is competitive on seen views and improves robustness under viewpoint shifts. Compared to GEN3C [13], **ours** yields substantially better FID/KID on both splits, consistent with more coherent 3D structure from our explicit 3D generation. Notably, our gains are most pronounced on *novel* views, where viewpoint shifts expose geometric inconsistencies in methods that are trained with restricted view coverage



Fig. 8: Comparaison of Σ-Voxfield (bottom) vs semantic field (top) conditioning for ASD-based rendering

(e.g., front views), such as InfiniCube. In contrast, our Σ-Voxfield representation better preserves geometry across pose perturbations, leading to improved shifted-view rendering.

Inference cost and scalability. Table 2 summarizes inference memory footprints. Our method uses 8 GB VRAM, versus 75 GB for InfiniCube [15] and 43 GB for GEN3C [13]. Runtime remains practical: we generate a scene in ∼20 minutes, comparable to InfiniCube at similar scale.

Method	FID (seen)↓	FID (novel)↓	KID (seen)↓	KID (novel)↓	Min. VRAM
InfiniCube [15]	84.14	99.13	**0.03**	**0.06**	75 GB
GEN3C [13]	113.27	117.63	0.08	0.09	43 GB
Ours (ASD)	**81.98**	**89.20**	0.05	**0.06**	**8 GB**

Table 2: Quantitative Evaluation of Generation Quality for 3D Scene Generation with our proposed method and existing approaches.

4.4 Ablation Studies

3D diffusion evaluation We evaluate the Σ-Voxfield diffusion model in a learned 3D feature space, since image metrics (e.g., FID) do not apply to 3D generation. We train a PointNet++ semantic segmentation network on labeled point clouds used as a feature extractor to evaluated using F3D and MMD. Tab. 3 shows the full model performs best: removing semantic conditioning degrades F3D/MMD, and disabling point ordering also hurts performance, indicating both are important.

Method	**F3D↓**	**MMD↓**
w/o sem. cond.	3.927	0.105
w/o ordering	3.585	0.093
Ours	3.523	0.091

Table 3: Feature-space ablation of our 3D diffusion model.

Surface point and voxel size: We analyze the influence of the number of sampled surface points and the voxel size used for conditioning. Sampling 20 points offers the best balance between quality and efficiency, as further increases

	Semantic	FreeVS	**Σ-Voxfield**
FVD↓	143.62	97.51	**53.01**

Table 4: Conditioning for deferred rendering.

# Points	Speed↑	F3D↓	MMD↓
5	1.95×	4.11	0.08
20 (ours)	1.00×	**3.52**	0.09
70	0.51×	3.50	0.09

Table 5: Effect of the number of sampled surface points on 3D generation quality and inference speed.

Voxel size (m)	0.3	0.6	1.2
PSNR↑	15.20	**15.50**	14.83
SSIM↑	**0.60**	0.58	0.54

Table 6: Influence of voxel size on deferred rendering quality.

provide only marginal gains at a substantial computational cost (Table 5). For voxelization, a resolution of 0.6 m yields the best overall performance (Table 6); finer grids over-constrain the conditioning signal and reduce geometric diversity, while coarser grids degrade surface fidelity.

Conditioning for deferred-rendering We compare three conditioning strategies for the deferred renderer: *semantic maps* (UniScene [11]-style, our implementation with ASD), *projected LiDAR* (FreeVS [27], authors' implementation), and our *Σ-Voxfield* projection. The Σ-Voxfield representation substantially outperforms the alternative conditioning signals, indicating that jointly encoding geometry and appearance is critical for rendering quality and temporal consistency as shown in Table 4. As illustrated in Figure 8, semantic conditioning leads to noticeable temporal degradation, whereas our conditioning strategy maintains coherent results over time.

4.5 Applications



Fig. 9: Semantic editing: We remove existing cars by editing the semantic grid (red boxes), then regenerate the content conditioned on the updated semantics. The generated results remain coherent and consistent with the surrounding scene.

Semantic editing. We enable object-level editing directly on the semantic grid: vehicles can be removed or newly inserted, after which the model regenerates the scene content accordingly. Guided by the edited semantic grid, the resulting generations remain spatially consistent and coherent, as shown in Figure 9.

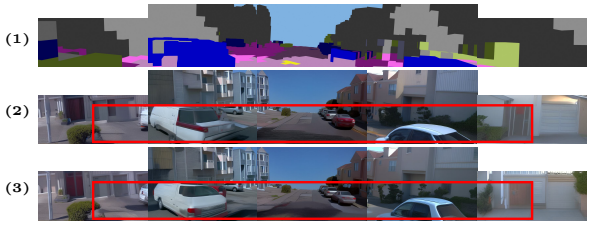


Fig. 10: Scene inpainting We mask a voxel region as shows (1) (darker voxel colors for unmasked) and re-generate only this region while keeping the rest of the Σ-Voxfield grid fixed. (2) and (3) show two inpainted results (red box) that remain coherent with the background and vary in both structure and appearance.

Scene inpainting. Figure 10 demonstrates **local editing** enabled by voxel-space inpainting. Starting from a generated Σ-Voxfield grid, we define a foreground region as a subset of Σ-Voxfields and re-run diffusion only on this target subset while keeping the rest of the grid fixed. While different samples produce diverse foreground realizations that remain coherent with the surrounding context across viewpoints.

Infinite scene generation. Despite being trained on local voxel neighborhoods, our method produces continuous scenes over large spatial extents as shown in Figure 1.

4.6 Limitations

While our method enables scalable and semantically structured 3D scene generation, several limitations remain. First, our model provides limited control over the generated appearance: conditioning is primarily semantic and geometric, so attributes such as texture style, lighting, and material properties cannot be specified explicitly. Second, our formulation focuses on static scene generation and does not model dynamic elements such as moving vehicles, pedestrians, or temporal scene evolution. Extending the representation and diffusion process to account for the dynamics and time-varying content remains an open challenge.

5 Conclusion

We introduce a scalable framework for semantically structured 3D driving-scene generation that operates directly in 3D. Our method generates a Σ-Voxfield grid with a semantic-conditioned diffusion model over local neighborhoods, and scales to large environments via voxel-space spatial outpainting. The resulting 3D buffer is rendered with a deferred diffusion engine to produce photorealistic images without per-scene optimization. Experiments on Waymo and PandaSet demonstrate competitive rendered-view quality, strong semantic coherence, and scalability to large scenes with moderate computation cost. This work points toward structured 3D scene generation and motivates future extensions such as richer appearance control and dynamic scene modeling.

References

1. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., Jampani, V., Rombach, R.: Stable video diffusion: Scaling latent video diffusion models to large datasets (2023), https://arxiv.org/abs/2311.15127
2. Cai, Y., Zhang, H., Zhang, K., Liang, Y., Ren, M., Luan, F., Liu, Q., Kim, S.Y., Zhang, J., Zhang, Z., Zhou, Y., Zhang, Y., Yang, X., Lin, Z., Yuille, A.: Baking gaussian splatting into diffusion denoiser for fast and scalable single-stage image-to-3D generation and reconstruction. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2025)
3. Cernea, D.: OpenMVS: Multi-view stereo reconstruction library (2020), https://cdcseacave.github.io
4. Chen, Z., Yang, J., Huang, J., de Lutio, R., Esturo, J.M., Ivanovic, B., Litany, O., Gojcic, Z., Fidler, S., Pavone, M., Song, L., Wang, Y.: OmniRe: Omni urban scene reconstruction. In: Proc. of the International Conf. on Learning Representations (ICLR) (2025)
5. Dhariwal, P., Nichol, A.: Scalable Diffusion Models with Transformers. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2023)
6. Gao, R., Chen, K., Li, Z., Hong, L., Li, Z., Xu, Q.: MagicDrive3D: Controllable 3D generation for any-view rendering in street scenes. In: Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV) (2026)
7. Gao, R., Chen, K., Xie, E., Hong, L., Li, Z., Yeung, D.Y., Xu, Q.: MagicDrive: Street view generation with diverse 3D geometry control. In: Proc. of the International Conf. on Learning Representations (ICLR) (2024)
8. Guo, H., Shao, J., Chen, X., Tan, X., Miao, S., Shen, Y., Liao, Y.: ScenDi: 3D-to-2D scene diffusion cascades for urban generation. arXiv:2601.15221 (2026)
9. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In: SIGGRAPH 2024 Conference Papers (2024)
10. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2022), https://arxiv.org/abs/1312.6114
11. Li, B., Guo, J., Liu, H., Zou, Y., Ding, Y., Chen, X., Zhu, H., Tan, F., Zhang, C., Wang, T., Zhou, S., Zhang, L., Qi, X., Zhao, H., Yang, M., Zeng, W., Jin, X.: UniScene: Unified occupancy-centric driving scene generation. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
12. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth Anything 3: Recovering the visual space from any views. arXiv:2511.10647 (2025)
13. Liu, X., Zhang, Y., Chen, Z., Wang, J., Tang, Y., et al.: GEN3C: Generalizable 3d scene generation from video diffusion with 3D cache. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)
14. Lu, F., Lin, K.Y., Xu, Y., Li, H., Chen, G., Jiang, C.: Urban Architect: Steerable 3D urban scene generation with layout prior. ArXiv:2404.06780 (2024)
15. Lu, Y., Ren, X., Yang, J., Shen, T., Wu, Z., Gao, J., Wang, Y., Chen, S., Chen, M., Fidler, S., Huang, J.: InfiniCube: Unbounded and controllable dynamic 3D driving scene generation with world-guided video models. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2025)
16. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Gool, L.V.: RePaint: Inpainting using denoising diffusion probabilistic models. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2022)

17. Mao, J., Li, B., Ivanovic, B., Chen, Y., Wang, Y., You, Y., Xiao, C., Xu, D., Pavone, M., Wang, Y.: DreamDrive: Generative 4D scene modeling from street view images. In: Proc. IEEE International Conf. on Robotics and Automation (ICRA) (2024)
18. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-E: A system for generating 3D point clouds from complex prompts. ArXiv:2212.08751 (2022)
19. NVIDIA, :, Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., Dworakowski, D., Fan, J., Fenzi, M., Ferroni, F., Fidler, S., Fox, D., Ge, S., Ge, Y., Gu, J., Gururani, S., He, E., Huang, J., Huffman, J., Jannaty, P., Jin, J., Kim, S.W., Klár, G., Lam, G., Lan, S., Leal-Taixe, L., Li, A., Li, Z., Lin, C.H., Lin, T.Y., Ling, H., Liu, M.Y., Liu, X., Luo, A., Ma, Q., Mao, H., Mo, K., Mousavian, A., Nah, S., Niverty, S., Page, D., Paschalidou, D., Patel, Z., Pavao, L., Ramezanali, M., Reda, F., Ren, X., Sabavat, V.R.N., Schmerling, E., Shi, S., Stefaniak, B., Tang, S., Tchapmi, L., Tredak, P., Tseng, W.C., Varghese, J., Wang, H., Wang, H., Wang, H., Wang, T.C., Wei, F., Wei, X., Wu, J.Z., Xu, J., Yang, W., Yen-Chen, L., Zeng, X., Zeng, Y., Zhang, J., Zhang, Q., Zhang, Y., Zhao, Q., Zolkowski, A.: Cosmos world foundation model platform for physical ai (2025), https://arxiv.org/abs/2501.03575
20. Ost, J., Ramazzina, A., Joshi, A., Bömer, M., Bijelic, M., Heide, F.: LSD-3D: Large-scale 3D driving scene generation with geometry grounding. In: Proc. of the Conf. on Artificial Intelligence (AAAI) (2026)
21. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion (2022), https://arxiv.org/abs/2209.14988
22. Ren, X., Lu, Y., Liang, H., Wu, Z., Ling, H., Chen, M., Fidler, S., Williams, F., Huang, J.: SCube: Instant large-scale scene reconstruction using voxsplats. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)
23. Roessle, B., Gallo, I., Belongie, S., Geiger, A.: L3DG: Latent 3D gaussian diffusion. In: Proc. of SIGGRAPH Asia (2024)
24. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2022), https://arxiv.org/abs/2112.10752
25. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Wang, W., Anguelov, D.: Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020)
26. Valevski, D., Leviathan, Y., Arar, M., Fruchter, S.: Diffusion models are real-time game engines (2025), https://arxiv.org/abs/2408.14837
27. Wang, Q., Fan, L., Wang, Y., Chen, Y., Zhang, Z.: Freevs: Generative view synthesis on free driving trajectory (2024), https://arxiv.org/abs/2410.18079
28. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation (2025), https://arxiv.org/abs/2412.01506
29. Xiao, J., Owens, A., Torralba, A.: PandaSet: Advanced sensor suite dataset for autonomous driving. In: IEEE/CVF International Conf. on Computer Vision (ICCV) (2021)
30. Xiong, B., Wei, S.T., Zheng, X.Y., Cao, Y.P., Lian, Z., Wang, P.S.: OctFusion: Octree-based diffusion models for 3D shape generation. In: Proc. of the Conference of the European Association for Computer Graphics (Eurographics) (2025)
31. Zhang, B., Cheng, Y., Yang, J., Wang, C., Zhao, F., Tang, Y., Chen, D., Guo, B.: GaussianCube: A structured and explicit radiance representation for 3D generative modeling. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)