# SemCity: Semantic Scene Generation with Triplane Diffusion

Jumin Lee[1*] Sebin Lee[1*] Changho Jo[2] Woobin Im[1] Juhyeong Seon[1] Sung-Eui Yoon[1]

[1]KAIST [2]Neosapience, Inc.

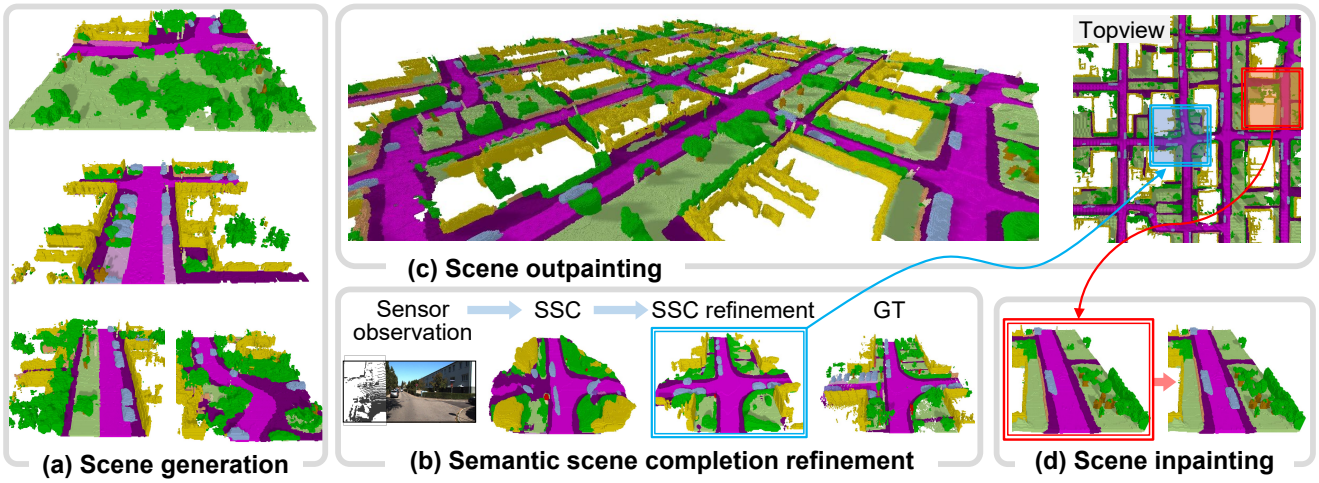



Figure 1. We introduce a diffusion framework, *SemCity*, designed for generating semantic scenes in real-world outdoor environments as shown in (a). We extend our diffusion model to various practical tasks: semantic scene completion refinement, scene outpainting, and scene inpainting. For instance, the comprehensive scenario is displayed in (b) → (c) → (d): the refined scene (SSC refinement) (b) is outpainted to a broader scene (c); then, an object (in this case, a car) is seamlessly integrated into the scene via our inpainting process (d).

## Abstract

*We present "SemCity," a 3D diffusion model for semantic scene generation in real-world outdoor environments. Most 3D diffusion models focus on generating a single object, synthetic indoor scenes, or synthetic outdoor scenes, while the generation of real-world outdoor scenes is rarely addressed. In this paper, we concentrate on generating a real-outdoor scene through learning a diffusion model on a real-world outdoor dataset. In contrast to synthetic data, real-outdoor datasets often contain more empty spaces due to sensor limitations, causing challenges in learning real-outdoor distributions. To address this issue, we exploit a triplane representation as a proxy form of scene distributions to be learned by our diffusion model. Furthermore, we propose a triplane manipulation that integrates seamlessly with our triplane diffusion model. The manipulation improves our diffusion model's applicability in a variety of downstream tasks related to outdoor scene generation such as scene inpainting, scene outpainting, and semantic scene completion refinements. In experimental results, we demonstrate that our triplane diffusion model shows meaningful generation results compared with existing work in a real-outdoor dataset, SemanticKITTI. We also show our triplane manipulation facilitates seamlessly adding, removing, or modifying objects within a scene. Further, it also enables the expansion of scenes toward a city-level scale. Finally, we evaluate our method on semantic scene completion refinements where our diffusion model enhances predictions of semantic scene completion networks by learning scene distribution. Our code is available at* https://github.com/zoomin-lee/SemCity.

## 1. Introduction

Diffusion models [17] have emerged as a promising generation tool owing to its state-of-the-art generation results in image domain [41, 42]. This advance has led to active exploration in extending diffusion models to 3D data generation across both academic and industrial groups. In the 3D domain, diffusion models have shown remarkable

*Both authors contributed equally to this work as co-first authors.

capabilities in generating diverse 3D forms (*e.g.*, voxels, meshes) [26, 28]. While those 3D diffusion models primarily aim to craft a single object, generating scenes consisting of multiple objects remains a relatively unexplored area in the 3D diffusion domain.

Scene generative diffusion models focus on crafting both geometrically and semantically coherent environments. Compared with a single object generation, generating a scene with multiple objects requires an understanding of more complex geometric and semantic structure due to a larger spatial extent [48]. There are primarily two streams of scene generative diffusion models, each tailored to either indoor or outdoor settings. In particular, the outdoor environments have inherent challenges caused by a broader landscape than indoor ones.

We propose to leverage a triplane representation [10] for broader outdoor scenes, a method of factorizing 3D data onto three orthogonal 2D planes, as utilized in 3D object reconstruction and NeRF models [2, 34, 53]. We excavate its advantages in addressing the data sparsity problem typically found in outdoor datasets due to the sensor limitations (*e.g.*, occlusions, range constraints) in capturing outdoor scenes. Triplane representation helps to reduce the inclusion of unnecessary empty information through the factorization of 3D data to 2D planes [10]. This efficiency in capturing relevant spatial detail makes it an effective tool for representing the many objects typically found in outdoor environments.

In this paper, we design our diffusion framework based on triplane representations. Our triplane autoencoder learns to compress a voxelized scene into a triplane representation by reconstructing semantic labels of the scene. Following this, the triplane diffusion model is trained and used to generate new scenes, as shown in Fig. 1(a), by creating novel triplanes based on the efficient representation. Further, we propose a triplane manipulation method, which extends our triplane diffusion model toward several practical tasks (*i.e.*, scene inpainting, scene outpainting, and semantic scene completion refinements) as shown in Fig. 1(b-d). Our method can seamlessly add, remove, and modify objects in real-outdoor scenes while maintaining the semantic coherence of environments.

Our contributions are summarized as follows:

- We disclose the applicability of the triplane representation through generating semantic scenes for real-outdoor environments and extend its views in practical downstream tasks: scene inpainting, scene outpainting, and semantic scene completion refinement.
- We propose to manipulate triplane features during our diffusion process, facilitating seamlessly extending our method toward the downstream tasks.
- We demonstrate that the proposed method significantly enhances the quality of generated scenes in real-world outdoor environments.

## 2. Related Work

**Diffusion Models.** Diffusion models [17] learn data distributions via iterative denoising processes based on score functions [47]. Its generated results have shown remarkably realistic appearances with high fidelity and diversity in a variety of 2D image synthesis such as outpainting [41, 64], inpainting [30, 41] and text-to-image generation [35, 42]. Built upon these achievements, diffusion models have also been extended into the 3D domain, generating impressive results in various 3D shapes, including voxel grids [26, 67], point clouds [31, 60, 61], meshes [28], and implicit functions [22, 44, 45, 55]. While these models can craft a single 3D object, our model focuses on generating a 3D scene composed of multiple objects using a categorical voxel data structure, which is a relatively under-explored area in the 3D diffusion domain.

**Diffusion Models for Scene Generation.** In contrast to a single object generation, scene generation involves an understanding of the larger 3D space, causing more semantic and geometric complexities [48]. Diffusion models for scene generation have been studied in both indoor and outdoor environments. In indoor settings, diffusion models aim to learn distributions of relations among objects by representing them as scene graphs [21]. The scene graphs contain object attributes (*e.g.*, location, orientation, and size), capturing the intricate inter-object relationships within bounded spaces [48, 62]. For outdoor scenes, the challenges are distinct, frequently including a lot of empty areas (*e.g.*, sky, open areas) resulting from the broader landscapes. Traditional approach [24] has relied on discrete diffusion methods [18] on voxel space, necessitating a detailed representation of every air volume.

In this paper, we demonstrate that triplane diffusion is highly effective for generating real-outdoor scenes. By abstracting 3D spaces into three orthogonal 2D planes, triplane representation [10] effectively captures the vastness of outdoor environments, predominantly composed of air. Beyond its data efficiency, the triplane excels in focusing on other significant objects (*e.g.*, vehicles, buildings) by allocating lesser attention to less informative elements like air. Our approach stands in stark contrast to prior work [24], which were constrained to synthetic datasets with considerably less empty space. In real datasets, inherent sensor limitations, such as a limited field-of-view, a limited ranges, and the inability to capture occluded areas like the rears of buildings, lead to a prevalence of empty space. Furthermore, we emphasize the versatility of our framework by demonstrating its extension to various downstream tasks, including scene inpainting, outpainting, and semantic scene completion refinement.

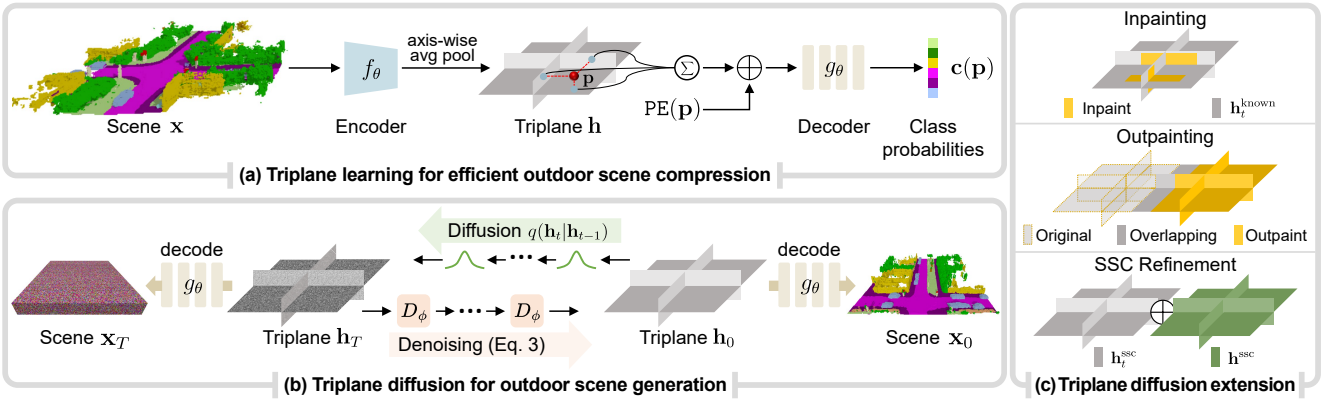



Figure 2. **Overview of ours.** (a) A 3D semantic map $\mathbf{x}$ is encoded by a triplane encoder $f_\theta$ and factorized to a triplane $\mathbf{h}$. The triplane coupled with a positional encoding $\texttt{PE}(\mathbf{p})$ is decoded by an implicit decoder $g_\theta$, resulting in class probabilities for each coordinate $\mathbf{p}$. (b) Our triplane diffusion model $D_\phi$ learns to generate a novel triplane for semantic scene generation via denoising diffusion process. (c) We further extend our triplane diffusion beyond a simple scene generation toward various practical scenarios by manipulating triplanes in (b).

**3D Inpainting and Outpainting.** In 3D inpainting, the primary objective is to fill in missing portions or modify existing elements of 3D data while maintaining geometric consistency. Most existing works concentrate on single-object inpainting [3, 25, 51]; for instance, they seamlessly transit a 3D chair's leg count from three to four. Contrary to inpainting, 3D outpainting is to extrapolate a given scene over an unobserved space. The existing work [1] focuses on scene outpainting within bounded indoor environments such as rooms. While in 2D images, inpainting and outpainting are not restricted to a single object [12, 20, 59]. Likewise, in 3D space, we focus on scene-level inpainting, which seamlessly adds, removes, or modifies objects in a scene. Further, our scene-level outpainting is not constrained to the bounded scenes; we extrapolate the outdoor scene from sensor range (*e.g.*, LiDAR) to city-scale.

**Semantic Scene Completion.** Semantic scene completion (SSC) [46] is pivotal for 3D scene understanding, where it jointly infers completion and semantic segmentation of the 3D scene from sensor observations such as RGB images [9, 27, 32] or point clouds [56–58]. In addition, SSC plays a crucial role in supporting comprehensive autonomous navigation systems, notably in essential downstream tasks like path planning [5, 11, 50] and map construction [13, 49]. Despite significant progress in the field, a persistent challenge is the semantic and geometric discrepancies between the SSC-estimated scenes and their real counterparts, as illustrated in Fig. 1(c). These discrepancies can undermine the performance of downstream tasks. Our triplane diffusion model can help to bridge this gap by exploiting 3D scene priors. This approach enhances the reliability and effectiveness of SSC, which is expected to improve its application in autonomous navigation systems.

## 3. Method

In this section, we elucidate our triplane diffusion model and its extensions. Our triplane diffusion model aims to synthesize novel real-outdoor scenes by generating a triplane, a proxy representation, which effectively addresses the inherent challenges of real-outdoor scene synthesis. This triplane representation is learned by our triplane autoencoder, which abstracts the geometric and semantic intricacies of a scene into three orthogonal 2D feature planes, namely, the $xy$, $xz$, and $yz$ planes (Sec. 3.1). Then, our diffusion model learns triplane distributions of scenes, generating novel triplanes (Sec. 3.2). We extend our triplane diffusion model toward various practical scenarios: scene inpainting, scene outpainting, and semantic scene completion refinements (Sec. 3.3).

### 3.1. Representing a Semantic Scene with Triplane

To represent a 3D scene as a triplane, our triplane autoencoder learns to compress a 3D scene into a triplane representation as shown in Fig. S3(a). The autoencoder consists of two modules: (1) an encoder $f_\theta$ yielding a triplane, and (2) an implicit multi-layer perceptron (MLP) decoder $g_\theta$ for reconstruction from the triplane.

The encoder $f_\theta$ takes a voxelized scene $\mathbf{x} \in \mathbb{R}^{X \times Y \times Z}$ containing $N$ classes within a spatial grid of resolution $X \times Y \times Z$. It then yields an axis-aligned triplane representation $\mathbf{h} = [\mathbf{h}^{xy}, \mathbf{h}^{xz}, \mathbf{h}^{yz}]$. The triplane consists of three planes, each characterized by distinct dimensional properties: $\mathbf{h}^{xy} \in \mathbb{R}^{C_h \times X_h \times Y_h}$, $\mathbf{h}^{xz} \in \mathbb{R}^{C_h \times X_h \times Z_h}$, and $\mathbf{h}^{yz} \in \mathbb{R}^{C_h \times Y_h \times Z_h}$, where $C_h$ stands for a feature dimension, and $X_h$, $Y_h$, and $Z_h$ denote spatial dimension of the triplane. During the encoding phase, a 3D feature volume is extracted by 3D convolutional layers from the scene $\mathbf{x}$, re-

sulting in the triplane via axis-wise average pooling. Given a 3D coordinate $\mathbf{p}=(x,y,z)$, the triplane is interpreted as a summation of vectors bilinearly interpolated from each plane: $\mathbf{h}(\mathbf{p})=\mathbf{h}^{xy}(x,y)+\mathbf{h}^{xz}(x,z)+\mathbf{h}^{yz}(y,z)$.

To reconstruct the 3D scene $\mathbf{x}$, we decode the encoded triplane $\mathbf{h}$ with an implicit MLP decoder $g_\theta$ that predicts semantic class probabilities. The decoder takes the triplane vector $\mathbf{h}(\mathbf{p})$ with its sinusoidal positional embedding $\texttt{PE}(\mathbf{p})$ [33], resulting in class probabilities $\mathbf{c}(\mathbf{p}) = g_\theta(\mathbf{h}(\mathbf{p}), \texttt{PE}(\mathbf{p})) \in [0,1]^N$. The positional embedding produces high-frequency features according to the coordinates $\mathbf{p}$, which helps the implicit decoder $g_\theta$ represent high-frequency scene contents [52].

The encoder $f_\theta$ and the MLP decoder $g_\theta$ are trained with the autoencoder loss $\mathcal{L}_{\text{AE}}$ and scene label $\mathbf{x}(\mathbf{p})$ as:

$$\mathcal{L}_{\text{AE}} = \mathbb{E}_{\mathbf{p}\sim\mathcal{P}}[\ell_{\text{CE}}(\mathbf{c}(\mathbf{p}),\mathbf{x}(\mathbf{p})) + \lambda\ell_{\text{LZ}}(\mathbf{c}(\mathbf{p}),\mathbf{x}(\mathbf{p}))], \quad (1)$$

where $\lambda$ is a loss weight, and $\mathcal{P}$ is the set of grid coordinates of the scene. We use the weighted cross-entropy loss $\ell_{\text{CE}}$ [37] and the Lovász-softmax loss $\ell_{\text{LZ}}$ [7] to learn imbalanced semantic distributions of the scene.

### 3.2. Triplane Diffusion

Based on the triplane representation of the 3D semantic scene, our triplane diffusion model $D_\phi$ learns to generate a novel triplane through denoising diffusion probabilistic models [17] as shown in Fig. S3(b). This triplane generation leads to generation of 3D scene through decoding the generated triplane with the implicit MLP decoder $g_\theta$. Through the $\boldsymbol{x}_0$-parameterization [4], the diffusion model $D_\phi$ is trained to reconstruct the triplane $\mathbf{h}$ given its corrupted triplane $\mathbf{h}_t$ sampled from a diffusion process $q(\mathbf{h}_t|\mathbf{h}) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\,\mathbf{h}, (1-\bar{\alpha}_t)\,\mathbf{I})$, where $\mathcal{N}$ is the Gaussian distribution, $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$, and $\alpha_t = 1-\beta_t$ with a variance schedule $\beta_t$. The diffusion process $q(\mathbf{h}_t|\mathbf{h})$ is derived from the Markovian chain rule with a single step's diffusion process $q(\mathbf{h}_t|\mathbf{h}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{h}_{t-1}, \beta_t\mathbf{I})$. Thus, the triplane diffusion loss is defined as:

$$\mathcal{L}_{\text{D}} = \mathbb{E}_{t\sim\mathcal{U}(1,T)}||\mathbf{h} - D_\phi(\mathbf{h}_t, t)||_p, \quad (2)$$

where $T$ is the number of denoising steps, and $p$ represents the order of the norm. The timestep $t$ is sampled from the discrete uniform distribution $\mathcal{U}$.

After training, the diffusion model $D_\phi$ generates a novel triplane $\mathbf{h}_0$ via the iterative DDPM generation process [17] starting from $\mathbf{h}_T \sim \mathcal{N}(\mathbf{0},\mathbf{I})$:

$$\mathbf{h}_{t-1} \sim \mathcal{N}\left(\gamma_t\mathbf{h}_t + \delta_t D_\phi(\mathbf{h}_t, t), \beta_t^2\,\mathbf{I}\right), \quad (3)$$

with $\gamma_t := \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})/(1-\bar{\alpha}_t)$ and $\delta_t := \sqrt{\bar{\alpha}_{t-1}}\beta_t/(1-\bar{\alpha}_t)$. From the generated triplane $\mathbf{h}_0$, we generate a novel 3D semantic scene $\mathbf{x}_0$ by querying coordinates $\mathbf{p}$ to the implicit decoder, *i.e.*, $g_\theta(\mathbf{h}_0(\mathbf{p}), \texttt{PE}(\mathbf{p}))$.

### 3.3. Applications with Triplane Manipulation

Building on the triplane diffusion process (Sec. 3.2), we propose a triplane manipulation that allows our model to facilitate a variety of practical downstream tasks with few modifications, as illustrated in Fig. S3(c).

**Scene Inpainting.** Our scene inpainting randomly edits a 3D scene, seamlessly adding, modifying, or removing objects while maintaining the consistency and realism of the scene. For instance, the inpainting includes scenarios where cars or sidewalks appear and then disappear, or vice versa, as shown in Fig. 1(d). Inspired by the RePaint sampling strategy [30], we propose a 3D-aware inpainting approach with semantic coherence. RePaint focuses on the image domain without explicitly considering the fidelity of the underlying 3D scene. To facilitate 3D-aware inpainting, we inpaint the triplane, serving as a compact proxy representation for the scene. We define a binary spatial *trimask* $\mathbf{m} = [\mathbf{m}^{xy}, \mathbf{m}^{xz}, \mathbf{m}^{yz}]$ covering inpainting regions on triplane space, allowing us to control the generation process on the masked region. The trimask $\mathbf{m}$ is set to have ones for inpainting regions and zeros for others. We override the $t$-th triplane $\mathbf{h}_t = [\mathbf{h}_t^{xy}, \mathbf{h}_t^{xz}, \mathbf{h}_t^{yz}]$ of the generation process (Eq. 3) as follows:

$$\mathbf{h}_t \leftarrow \mathbf{m}\otimes\mathbf{h}_t + (\mathbf{1}-\mathbf{m})\otimes\mathbf{h}_t^{\text{known}}, \quad (4)$$

where $\otimes$ is the element-wise product, and the known triplane $\mathbf{h}_t^{\text{known}}$ for intact regions is sampled from the diffusion process, *i.e.*, $q(\mathbf{h}_t^{\text{known}}|\mathbf{h}) := q(\mathbf{h}_t|\mathbf{h})$, which adheres to a known Gaussian distribution.

**Scene Outpainting.** Our scene outpainting extends the boundaries of the 3D scene without additional training as with inpainting. To seamlessly outpaint a scene, the regions to be extended should be conditioned on the original scene. We propose to inject this intuition into triplane representation as shown in Fig. S3(c). For covering regions to be outpainted, our diffusion model generates a novel triplane that is partially overlapped with the original triplane. We implement this with the concept of the trimask $\mathbf{m}$ and the known triplane $\mathbf{h}_t^{\text{known}}$ as in our scene inpainting. The trimask $\mathbf{m}$ covers regions to be outpainted, and the $\mathbf{h}_t^{\text{known}}$ is obtained from the intersections of triplanes between original and outpainted regions. Based on the outpainting strategy, we extend a given scene toward cardinal and intercardinal directions, facilitating the creation of an unbounded scene. While our triplane diffusion model is trained using triplanes of a fixed size, it demonstrates the capability to outpaint scenes to be several times bigger than the original scene, as illustrated in Fig. 1(c).

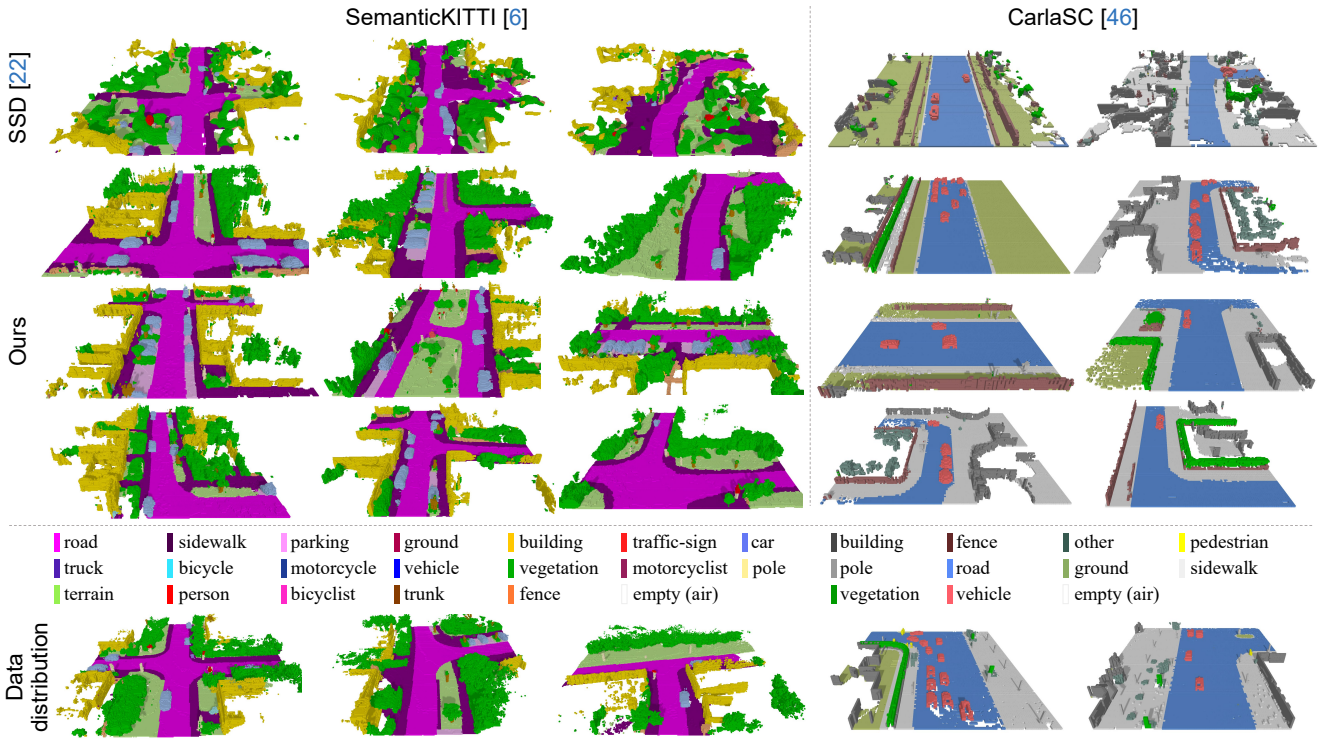



Figure 3. **Scene generation results** using both real and synthetic outdoor datasets – SemanticKITTI [6] and CarlaSC [54]. Our results showcase the effective generation of overall structures, including roads and buildings, along with detailed objects such as cars.

**Semantic Scene Completion Refinement.** SSC models complete and segment a 3D scene from sensor observations such as images or point clouds. We observe that the SSC results show a geometric and semantic discrepancy compared with data distributions, as shown in Fig. 1(b). We extend our triplane diffusion model to refine predictions of SSC models toward reducing the discrepancy. To effectively condition our triplane diffusion scheme to SSC model's prediction $\mathbf{x}^{\text{ssc}}$, we utilize its triplane representation $\mathbf{h}^{\text{ssc}} = f_\theta(\mathbf{x}^{\text{ssc}})$ derived by our triplane encoder $f_\theta$. We extend our triplane diffusion scheme with a simple modification of the triplane $\mathbf{h}_t$ in the diffusion loss (Eq. 2) and the generation process (Eq. 3) as follows:

$$\mathbf{h}_t = \mathbf{h}_t^{\text{ssc}} \oplus \mathbf{h}^{\text{ssc}}, \tag{5}$$

where $\oplus$ is concatenation, and $\mathbf{h}_t^{\text{ssc}}$ is a $t$-th diffused triplane sampled by the DDPS diffusion process [23] with the SSC prediction's triplane $\mathbf{h}^{\text{ssc}}$.

## 4. Experiments

### 4.1. Experimental Details

**Training Dataset.** We validate our method on the SemanticKITTI [6] and CarlaSC [54] datasets. SemanticKITTI provides 3D semantic scenes of real-outdoor environments with labels for 20 semantic classes. Each scene is represented by a voxel grid of $256 \times 256 \times 32$, covering an area of $51.2\,\text{m}$ in front of the car, extending $51.2\,\text{m}$ on each side, and reaching up to a height of $6.4\,\text{m}$. The dataset retains object motion traces as a result of sensor frame integration, which is employed to establish a dense ground truth. In contrast, CarlaSC is a synthetic dataset that provides 3D semantic outdoor scenes without the trace of moving objects. The dataset contains annotated 11 semantic classes with a voxel grid of $128 \times 128 \times 8$ and covers a distance of $25.6\,\text{m}$ in front and behind the car, $25.6\,\text{m}$ laterally on each side, and $3\,\text{m}$ in height.

**Implementation Details.** Our experiments are deployed on a single NVIDIA RTX 3090 GPU with a batch size of 4 for the triplane autoencoder and 18 for the triplane diffusion model. For the triplane autoencoder, the input scene is encoded to triplane with a spatial resolution $(X_h, Y_h, Z_h) = (128, 128, 32)$, and the feature dimension $C_h$ is 16. The loss weight $\lambda$ in Eq. 1 is set to 1.0. The order of the norm $p$ in Eq. 2 is set to 1 for SSC refinements and 2 for other cases. For the diffusion model, the learning rate is initialized to 1e-4 and then decreases linearly. During the diffusion process, we use the default settings [39] with 100 time steps ($T$). For our triplane inpainting and outpainting, we employ the RePaint sampling strategy [30] as a reference and perform a repaint with 5 resampling and a jump size of 20.

| Model | FID ↓ | KID ↓ | IS ↑ | Prec ↑ | Rec ↑ |
|---|---|---|---|---|---|
| SemanticKITTI [6] | | | | | |
| SSD [24] | 112.82 | 0.12 | 2.23 | 0.01 | 0.08 |
| SemCity (Ours) | 56.55 | 0.04 | 3.25 | 0.39 | 0.32 |
| CarlaSC [54] | | | | | |
| SSD [24] | 87.39 | 0.09 | 2.44 | 0.14 | 0.07 |
| SemCity (Ours) | 40.63 | 0.02 | 3.51 | 0.31 | 0.09 |

Table 1. **Results of semantic scene generation.** Each metric is computed between the rendered image of the generated scene and the ones of the actual scene at a resolution of $1440 \times 2048$.

**Evaluation Metrics.** Following the scene generation works [48, 62], we evaluate the performance of semantic scene generation by examining both the diversity and fidelity of 3D semantic scenes within the rendered images. We use recall to evaluate diversity, while precision and inception score (IS) are used to evaluate fidelity. The Fréchet Inception Distance (FID) [16] and Kernel Inception Distance (KID) [8] metrics are also utilized, as they reflect the combined effect of both diversity and fidelity on scene quality [14]. In terms of semantic scene completion (SSC) refinement performance, we follow the protocols defined in SSC works [9, 32, 56, 57]. The Intersection-over-Union (IoU) metric is used to quantify scene completeness, while the mean IoU (mIoU) provides a measure for the quality of semantic segmentation. These metrics together enable a comprehensive evaluation of how well the semantic scene completion methods perform in terms of accurately filling in and labeling the scene components.

### 4.2. Semantic Scene Generation

Fig. 3 illustrates a qualitative comparison on the SemanticKITTI [6] and CarlaSC [54] datasets. SSD [24] shows impressive results on the CarlaSC dataset. However, its performance on the SemanticKITTI dataset, which is a real-world dataset, is notably constrained. This limitation primarily arises from the SSD's voxel-based representation, which struggles with the more prevalent empty spaces in real-outdoor datasets compared to synthetic ones. This issue is especially evident in the generation of buildings and roads, where SSD often fails to define boundaries accurately. Furthermore, the model's limitations extend to representing finer structures, such as trunks and leaves, as well as traffic light poles and signals. It also struggles with generating uniform shapes of vehicles, often resulting in irregular shapes. In contrast, our method demonstrates the ability to effectively synthesize detailed scenes even on the real dataset, as illustrated in Fig. 3. It performs better than SSD [24] in accurately capturing complex building shapes on the CarlaSC dataset. In addition, our method exhibits remarkable proficiency in generating the overall contours of roads and buildings, along with intricate details on the SemanticKITTI dataset. Tab. 1 provides a detailed comparative evaluation using various metrics. Our model shows significant improvements in both the fidelity and diversity of the generated scenes. Moreover, our generated result is not tied to fixed resolution by means of the implicit neural representation, as depicted in Fig. 4. For additional results, please refer to the Supplementary Material.

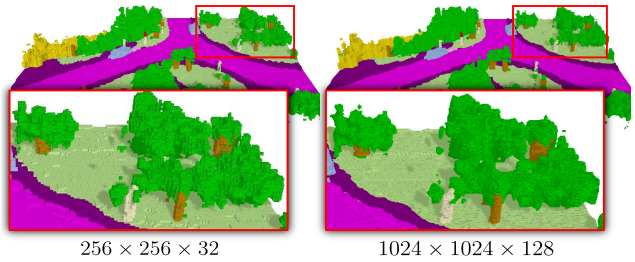



Figure 4. **Higher-resolution scene generation.** Building upon our implicit decoder, higher-resolution scene ($1024 \times 1024 \times 128$) can be generated compared with a resolution of training dataset ($256 \times 256 \times 32$).

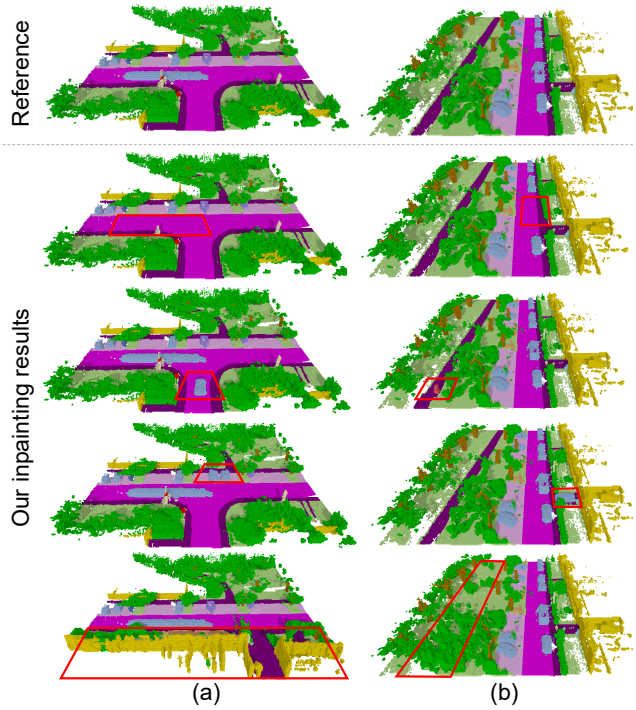



Figure 5. **Scene inpainting of our method.** The red boxes denote inpainting regions. (a) and (b) show our inpainting examples from reference images.

### 4.3. Applications of Triplane Diffusion

**Scene Inpainting.** Fig. 5 presents the qualitative results of our inpainting, demonstrating its effectiveness in inpainting both small and large regions within a scene, while maintaining the coherence of 3D contexts. In detail, the second row of **(a)** and **(b)** illustrates the model's seamless removal

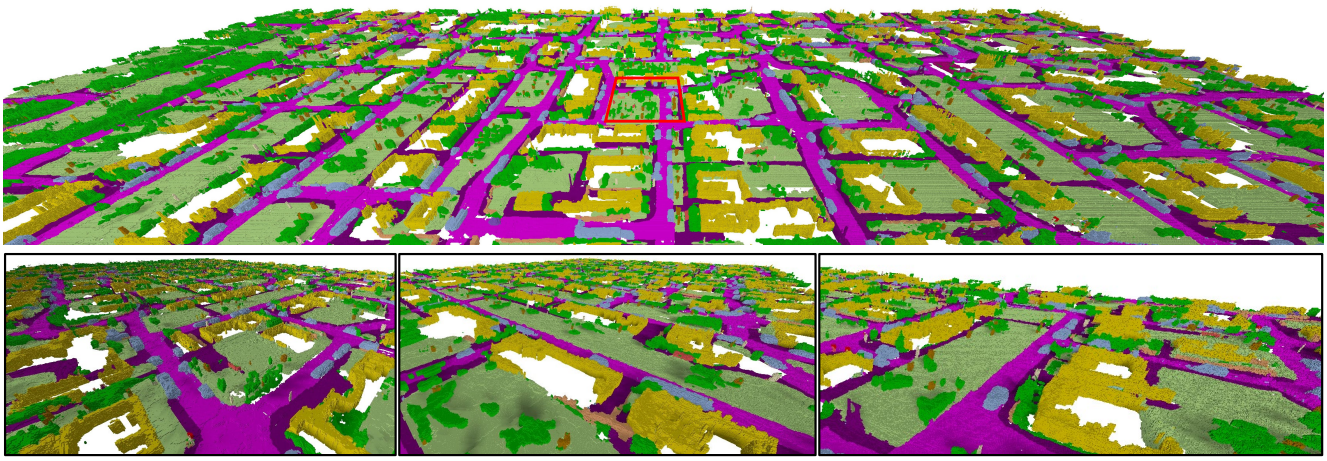


Figure 6. **Our scene outpainting results.** The red box at the center of the figure in the first row represents the given scene for outpainting. The zoomed views of the outpainted scene are depicted in the second row. The outpainted scene is expanded from the given size of $256 \times 256 \times 32$ to $1792 \times 3328 \times 32$.

| SSC Input | Method | IoU ↑ | mIoU ↑ |
|---|---|---|---|
| RGB | MonoScene [9] | 37.12 | 11.50 |
| | MonoScene + Ours | 50.44 | 17.08 |
| | OccDepth [32] | 41.60 | 12.84 |
| | OccDepth + Ours | 50.20 | 16.79 |
| Point Cloud | SSA-SC [58] | 58.25 | 24.54 |
| | SSA-SC + Ours | 60.71 | 25.58 |
| | SCPNet [56] | 50.24 | 37.55 |
| | SCPNet + Ours | 59.25 | 38.19 |

Table 2. **Quantitative results of refining SSC on SemanticKITTI validation set [6].** The results are based on the weights released by the authors on GitHub.

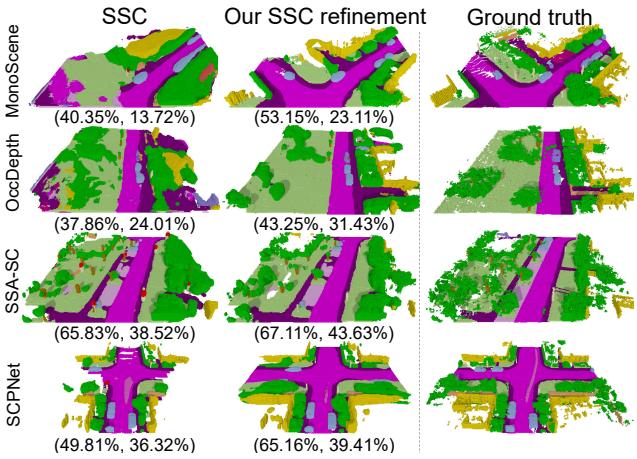
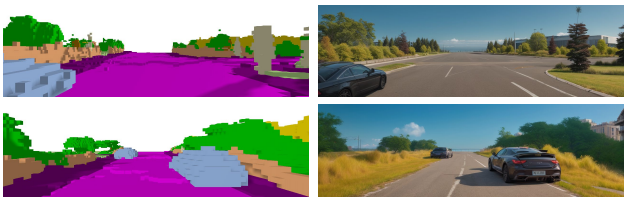



Figure 7. **Semantic scene completion refinement.** The SSC metrics are reported in the parentheses as (IoU, mIoU).

of vehicles, which harmonizes with the adjacent road. The third row demonstrates the insertion of new entities — a vehicle in **(a)** and a person in **(b)** — that are contextually congruent with the reference scene. The fourth row in **(a)** exemplifies the model's dual functionality in both modifying and adding vehicles within the scene. The fifth row in both columns underscores the model's proficiency in modifying scenes. Here, the model alters existing scene components, showcasing its ability to transform the overall ambiance of the scene. These results show our model's adeptness not just in object-level inpainting but also in scene-level inpainting.

**Scene Outpainting.** Fig. 6 illustrates a generated outpainting city-level scene, extending a $256 \times 256 \times 32$ scene to a substantial $1792 \times 3328 \times 32$ landscape. Although our model was not designed to generate cityscapes, it demonstrates the capability of maintaining coherence over a large area. The roads are connected in a meaningful and varied way, and various objects such as buildings, cars, and people have been created around them. Please check the details of the scene in Fig. 6.

**Semantic Scene Completion Refinement.** In Fig. 7, there is a notable semantic and geometric discrepancy between scenes predicted by existing semantic scene completion (SSC) methods and their real scene counterparts. Our model helps to bridge this gap by employing a 3D scene prior, which is effectively modeled through our diffusion model. SCPNet [56] also attempts to learn geometric priors from a teacher model trained with merged sequential frames. Yet, its completion network is inclined toward con-



(a) (b)

Figure 8. **Semantic scene to RGB.** (a) Semantic maps are rendered into a driving view from our generated scenes. (b) We generate images through ControlNet [63], an easily accessible image-to-image model, with generated semantic maps without shadows.

| Method | FID ↓ | KID ↓ | IS ↑ | Prec ↑ | Rec ↑ |
|---|---|---|---|---|---|
| Ours | 56.55 | 0.04 | 3.25 | 0.39 | 0.32 |
| w/o PE | 59.14 | 0.04 | 3.21 | 0.30 | 0.31 |
| $xy$-plane (w/o triplane) | 75.56 | 0.06 | 3.08 | 0.28 | 0.27 |
| 3D-volume (w/o triplane) | 136.50 | 0.15 | 1.96 | 0.14 | 0.12 |

Table 3. **Ablation studies on scene generation.** We ablate our model variants on SemanticKITTI [6].

servative estimations, which frequently result in partially filled spaces. In the case of SSA-SC [58], which utilizes bird's-eye view features, it is sometimes inappropriate due to the label's inherent limitations of the bird's-eye perspective in capturing certain semantic details. This discrepancy issue is even more evident in RGB-based methods, such as MonoScene [9] and OccDepth [32], which often exhibit diminished sharpness when image features are projected into 3D space. While SSC models show variations from real-world data distributions, our model shows potential in aligning these more closely with reality, as shown in Fig. 7.

As indicated in Tab. 2, our SSC refinement process appears to offer improvements to all state-of-the-art SSC models. These preliminary results suggest our model's effectiveness in providing not just more accurate semantic segmentation but also a more complete scene.

**Semantic Scene to RGB Image.** As illustrated in Fig. 8, we conducted an image-to-image generation experiment. The semantic maps are rendered into a driving view without shadows; then, they are utilized as inputs for ControlNet [63]. The generated RGB images are geometrically and semantically plausible but display a synthetic quality since the pretrained ControlNet was not trained on actual autonomous driving datasets.

### 4.4. Ablation

As shown in Tab. 3, we conducted ablation studies on our triplane diffusion model for scene generation, focusing on two key design elements:

**Positional Embedding.** Our variant excluded positional embedding `PE`, which generates high-frequency features critical for detailed scene reconstruction. Its absence resulted in lower performance across all metrics.

**Triplane Representation.** We evaluated the effectiveness of the triplane representation for real-outdoor scene generation. The triplane and $xy$-plane enhance the generation quality compared with 3D features, while the $xy$-plane shows lower performance than triplanes. We suspect that the excessive factorization limits the representation capability of the $xy$-plane compared to the triplane.

### 4.5. Limitation

While our model demonstrates significant progress in the generation of 3D real-outdoor scenes, it inherently reflects the characteristics of its training data, as detailed in Sec. 4.1. This reliance introduces several limitations. One notable challenge is the model's difficulty in accurately depicting areas occluded from the sensor's viewpoint, such as the rear sides of buildings, often leading to their incomplete representation in our generated scenes. Moreover, since the dataset is captured from a driving view, there is an inherent shortfall in capturing the full height of buildings. This brings a partial representation of the vertical structure of buildings and other tall elements in the scenes. Another issue is the model tends to produce traces of moving objects stemming from dataset pre-processing that merges sequential frames. For future work, incorporating prior knowledge of the city into the model could yield better results, such as addressing occluded areas or building heights.

## 5. Conclusion

We have proposed a diffusion framework called *SemCity* for real-world outdoor scene generation. The seminal idea is to generate a scene by factorizing real-outdoor scenes into triplane representations. Our triplane representation outperforms traditional voxel-based approaches, producing scenes that are not only visually more appealing but also rich in semantic detail, effectively capturing the complexity of various objects within the scene. Ours is not constrained by fixed resolutions thanks to the incorporation of an implicit neural representation. We have further expanded the capabilities of our triplane diffusion model to several practical applications, including scene inpainting, scene outpainting, and semantic scene completion refinement. Specifically, by manipulating triplanes during the diffusion process, we achieve seamless inpainting and outpainting at both the object and scene levels. Ours is used to more closely align the scenes predicted by existing semantic scene completion methods with the actual data distribution using our learned 3D prior. We believe that our work provides a road map of real-outdoor scene generation to research communities.

# SemCity: Semantic Scene Generation with Triplane Diffusion - Supplementary Material -

In this supplementary material, we report additional contents for an in-depth understanding of our method: backgrounds for diffusion models (Sec. A), implementation details of our method (Sec. B), and our additional experimental results (Sec. C). Specifically, we visualize our generation results across scene generation, scene inpainting, scene outpainting, and semantic scene completion refinement. We further demonstrate RGB images generated from our scene samples.

## A. Backgrounds of Diffusion Models

Diffusion models synthesize data (*e.g.*, images) by gradually transforming a random noise distribution into a data distribution through a reverse Markov process. This process involves two main phases: the forward process (*i.e.*, diffusion process) and the reverse process (*i.e.*, denoising process).

### A.1. Forward Process

In the forward process, a given data $\mathbf{x}_0 \sim p(\mathbf{x}_0)$ is gradually corrupted by adding noise over a series of steps. This process transforms the original data distribution into a Gaussian distribution. The forward process is modeled as a Markov chain, where each step adds a small amount of noise, making it easy to compute and invert:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \tag{S1}$$

Here, $\mathbf{x}_t$ is a noised data at step $t$, $\beta_t$ is a variance schedule, and $\mathcal{N}$ denotes the Gaussian distribution. $t$ is defined within $1 \leq t \leq T$ with the maximum denoising steps $T$.

The $t$-th noised data $\mathbf{x}_t$ is sampled via iteration of the forward process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ in Eq. S1; however, $\mathbf{x}_t$ can be simply obtained as a closed form with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \Pi_{s=0}^{t}\alpha_s$:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{S2}$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \boldsymbol{\epsilon}\sqrt{(1-\bar{\alpha}_t)}, \tag{S3}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $1-\bar{\alpha}_t$ is a variance of the noise for an arbitrary timestep $t$.

### A.2. Reverse Process

The reverse process iteratively removes noises from the sample to generate a coherent structure resembling the original data $\mathbf{x}_0$ distribution. Each denoising step can be expressed as a reverse Markov chain:

$$p_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\boldsymbol{\mu}_\phi(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\phi(\mathbf{x}_t, t)), \tag{S4}$$

where $\boldsymbol{\mu}_\phi$ and $\boldsymbol{\Sigma}_\phi$ are the mean and covariance of the reverse process at step $t$, parameterized by learnable parameters $\phi$. In particular, [17] proposes that a model $\boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t)$ can simply be trained to predict the noise $\boldsymbol{\epsilon}$ instead of directly parameterizing the mean $\boldsymbol{\mu}_\phi(\mathbf{x}_t, t)$. They assume the covariance $\boldsymbol{\Sigma}_\phi(\mathbf{x}_t, t)$ is constant. Thus, we can define a diffusion loss as:

$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}(1,T),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t)||_2, \tag{S5}$$

where $\mathcal{U}$ is the discrete uniform distribution. [4] suggests the $\mathbf{x}_0$-parameterization where a model $\mathbf{x}_\phi$ predicts the input data $\mathbf{x}_0$ directly, rather than predicting the added noise $\boldsymbol{\epsilon}$. The diffusion loss for the $\mathbf{x}_0$-parameterization is defined as:

$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}(1,T)}||\mathbf{x}_0 - \mathbf{x}_\phi(\mathbf{x}_t, t)||_2. \tag{S6}$$

This loss function is the basis of our triplane diffusion loss in Eq. 2 of the main paper.

## B. Implementation Details

### B.1. Training Setting

**Triplane Autoencoder.** As described in Sec. 3.1 of the main paper, our triplane autoencoder consists of two modules: the triplane encoder $f_\theta$ and the implicit MLP decoder $g_\theta$. We configure the encoder $f_\theta$ with six 3D convolutional layers with a skip connection and design our MLP decoder $g_\theta$ to be light to mitigate the training burden. The MLP decoder consists of four 128-dimensional fully-connected layers with a skip connection. Following [33], the positional encoding $\texttt{PE}(\mathbf{p})$ at coordinates $\mathbf{p}$ is used as sinusoidal functions defined as: $\texttt{PE}(\mathbf{p}) = \big[\sin(2^0\pi\mathbf{p}), \cos(2^0\pi\mathbf{p}), \ldots, \sin(2^5\pi\mathbf{p}), \cos(2^5\pi\mathbf{p})\big]$.

**Triplane Diffusion Model.** Based on the observation [41] where the sample diversity depends on $L_1$ or $L_2$ diffusion

loss, the norm factor $p$ of the triplane diffusion loss (Eq. 2 of the main paper) is set to 1 or 2. For more diversity of generation results, we set $p = 2$ (*i.e.*, $L_2$) in scene generation, scene inpainting, and scene outpainting. In contrast, we use $p = 1$ (*i.e.*, $L_1$) for semantic scene completion refinement following [43]. The diffusion settings (*e.g.*, the variance schedule $\beta_t$) are used as DDPM [17].

### B.2. Generation Setting

**Scene Outpainting.** Our model extrapolates a given scene, resulting in a larger scale scene as depicted in Fig. S4, Fig. S5 and Fig. 6 of the main paper. As shown in Fig. S4, our model is capable of generating a variety of extended scenes. To enhance its effectiveness, we have incorporated an interactive outpainting system [29] that allows users to guide the scene generation process. This interaction is a demonstration of the model's flexibility and responsiveness to user preferences. Users may keep the original outpainting or regenerate it to correspond more closely to their visual objectives. This capability enables users to create finely-tuned urban scenes on a city-scale, as shown in Fig. S5 and Fig. 6 of the main paper.

**Semantic Scene to RGB Image.** We exploit ControlNet [63] to generate RGB images from our semantic scenes. ControlNet supports various conditional inputs (*e.g.*, segmentation or depth maps) and can be easily integrated with other fine-tuned models (*e.g.*, Dreambooth [40], Textual inversion [15], and Lora [19]). We manipulate a semantic map rendered from our generated scenes and generate an RGB image through the following process. An initial RGB image is obtained by conditioning semantic and depth maps rendered from our generated scene. Afterward, we generate a final image from the initial RGB map with conditional segmentation and depth maps obtained from ControlNet preprocessors [36, 65, 66]. For our experiments, we employ the diffusion model [38] weights* fine-tuned on urban street views to generate images analogous to driving scenes.

## C. Additional Experimental Results

In this section, we visualize additional generated scenes of our method in the various applications, including 1) scene generation, 2) scene inpainting, 3) scene outpainting, 4) semantic scene completion refinement, and 5) semantic scene to RGB image. For visualizations, colors are used as below.

| | | | | |
|---|---|---|---|---|
| road | sidewalk | parking | ground | car |
| truck | bicycle | motorcycle | vehicle | pole |
| terrain | motorcyclist | bicyclist | trunk | fence |
| building | traffic-sign | vegetation | person | empty |

*https://civitai.com/models/119169/urban-streetview

### C.1. Triplane Visualization

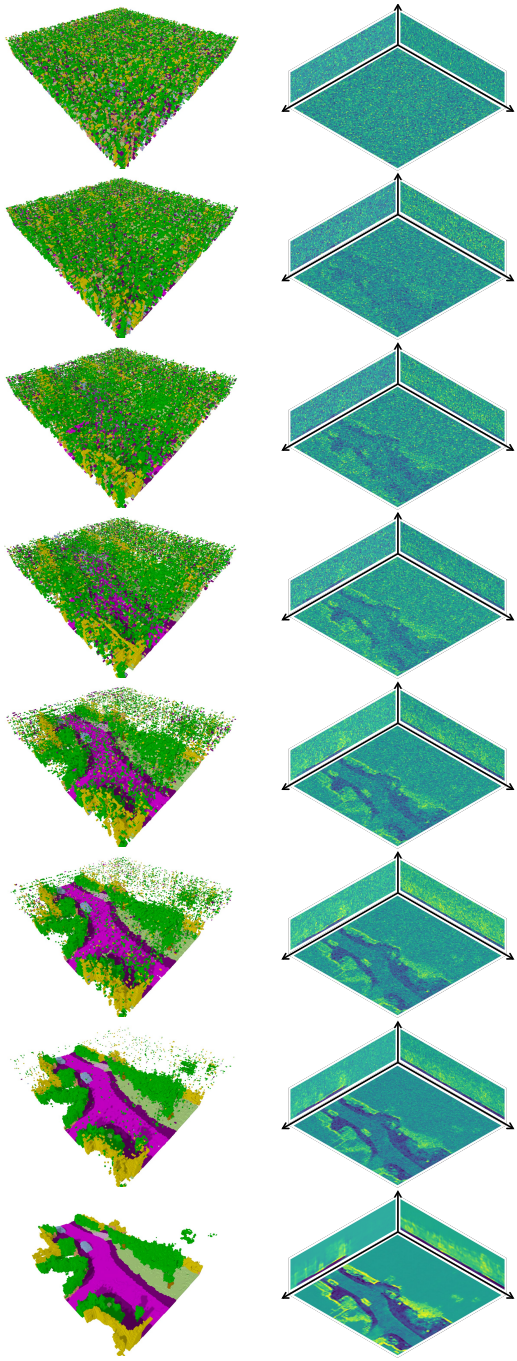


Figure S1. **Triplane visualization during our generation process.** We visualize triplanes (right) and their corresponding scenes (left) according to diffusion steps. We observe distinct denoising patterns where our diffusion model initially constructs low-frequency structures (*e.g.*, roads) in the early stages of denoising. In contrast, high-frequency details (*e.g.*, edges) are progressively refined in the later stages of the process. This phenomenon can also be found in image diffusion models [17]; we expect this property to be exploited for elastic scene editing in future work.

## C.2. Scene Generation

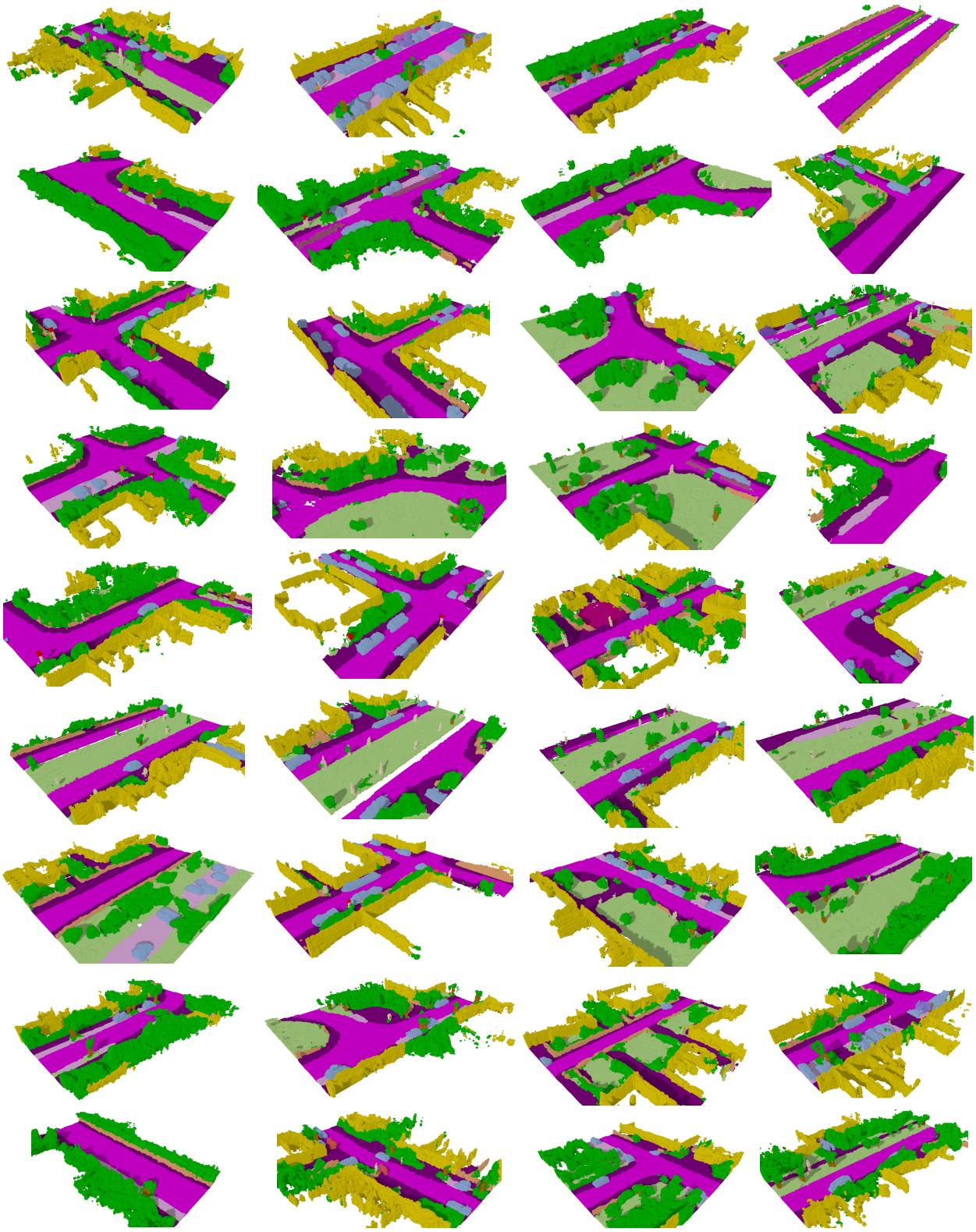


Figure S2. **Scene generation results of our method.** The generated scenes demonstrate various road shapes, including L, T, Y, straight, and crossroads, which show that our method generates diverse samples.

### C.3. Semantic Scene Completion Refinement

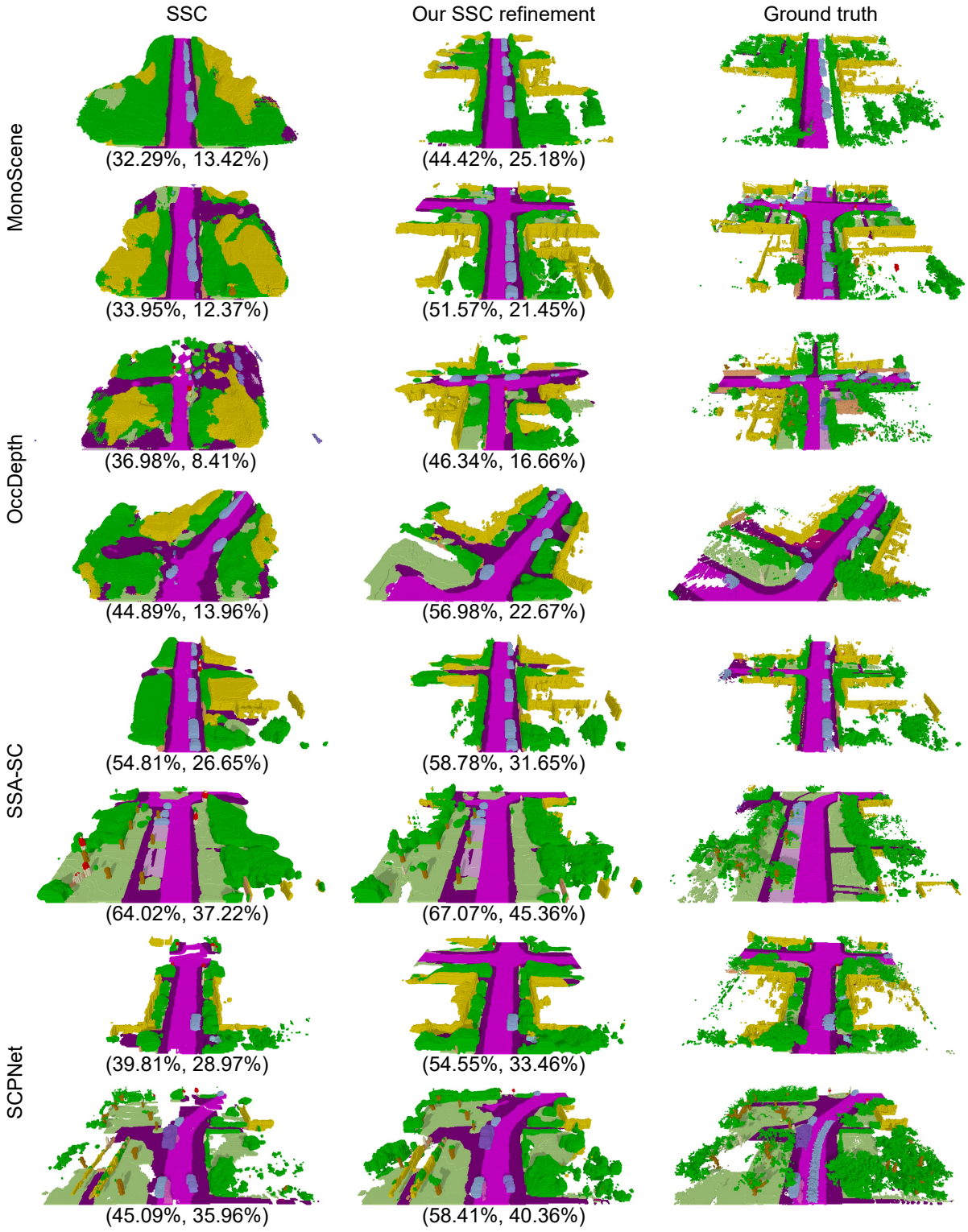



Figure S3. **Results of semantic scene completion refinement of our method.** The parentheses report the SSC metrics as (IoU, mIoU). Our method refines the results of state-of-the-art SSC methods. The MonoScene [9] and OccDepth [32] methods use a RGB input. The SSA-SC [58] and SCPNet [56] employ LiDAR point clouds as an input.

### C.4. Scene Outpainting

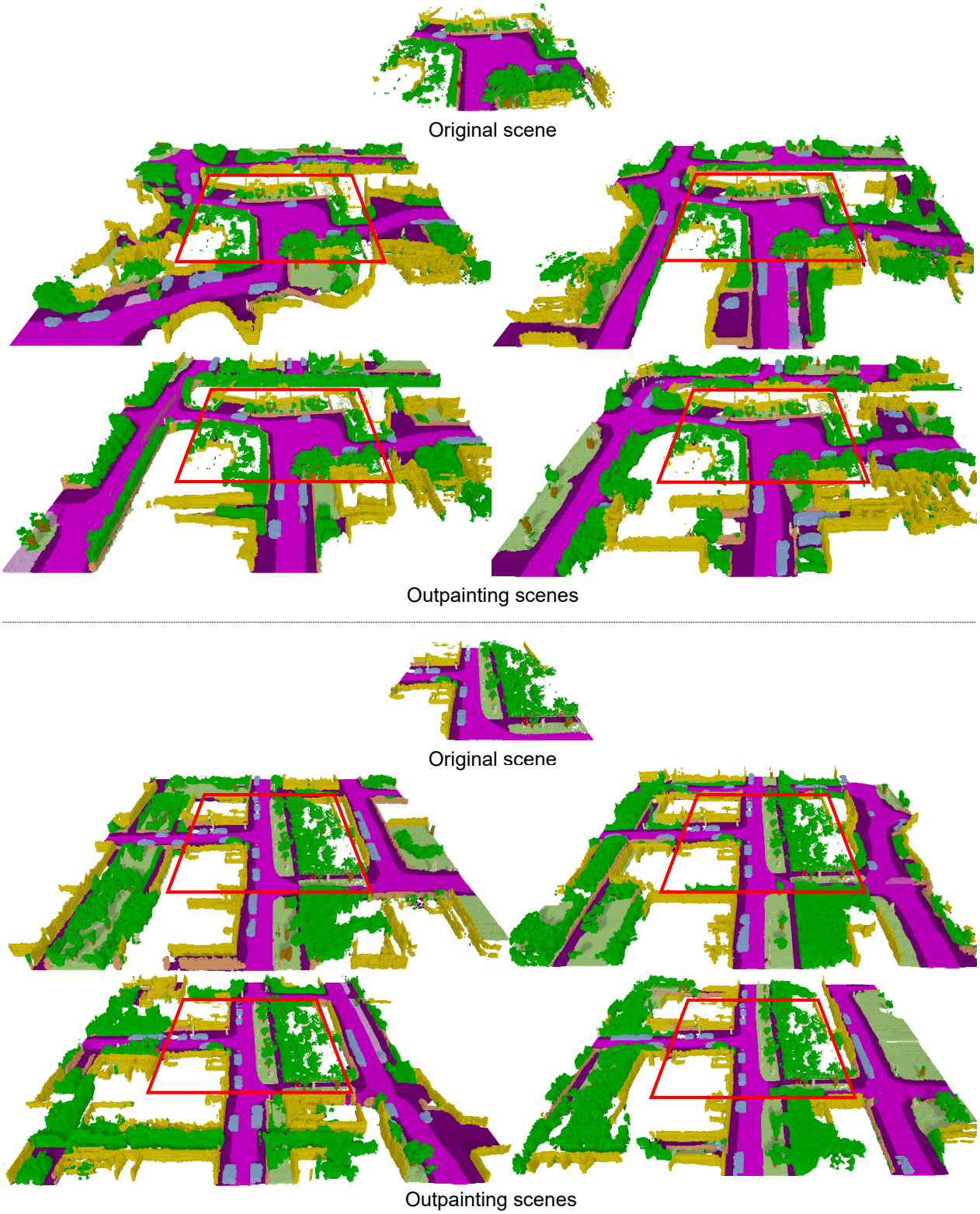



Figure S4. **Scene outpainting results of our method.** We visualize various outpainting results generated from two scenes. The outpainted scene is expanded from the given size of $256 \times 256 \times 32$ to $512 \times 512 \times 32$ without any guidance. The red boxes mean an original scene for outpainting. Our method produces various outpainted scenes from an identical original scene.

## C.5. City-level Generation

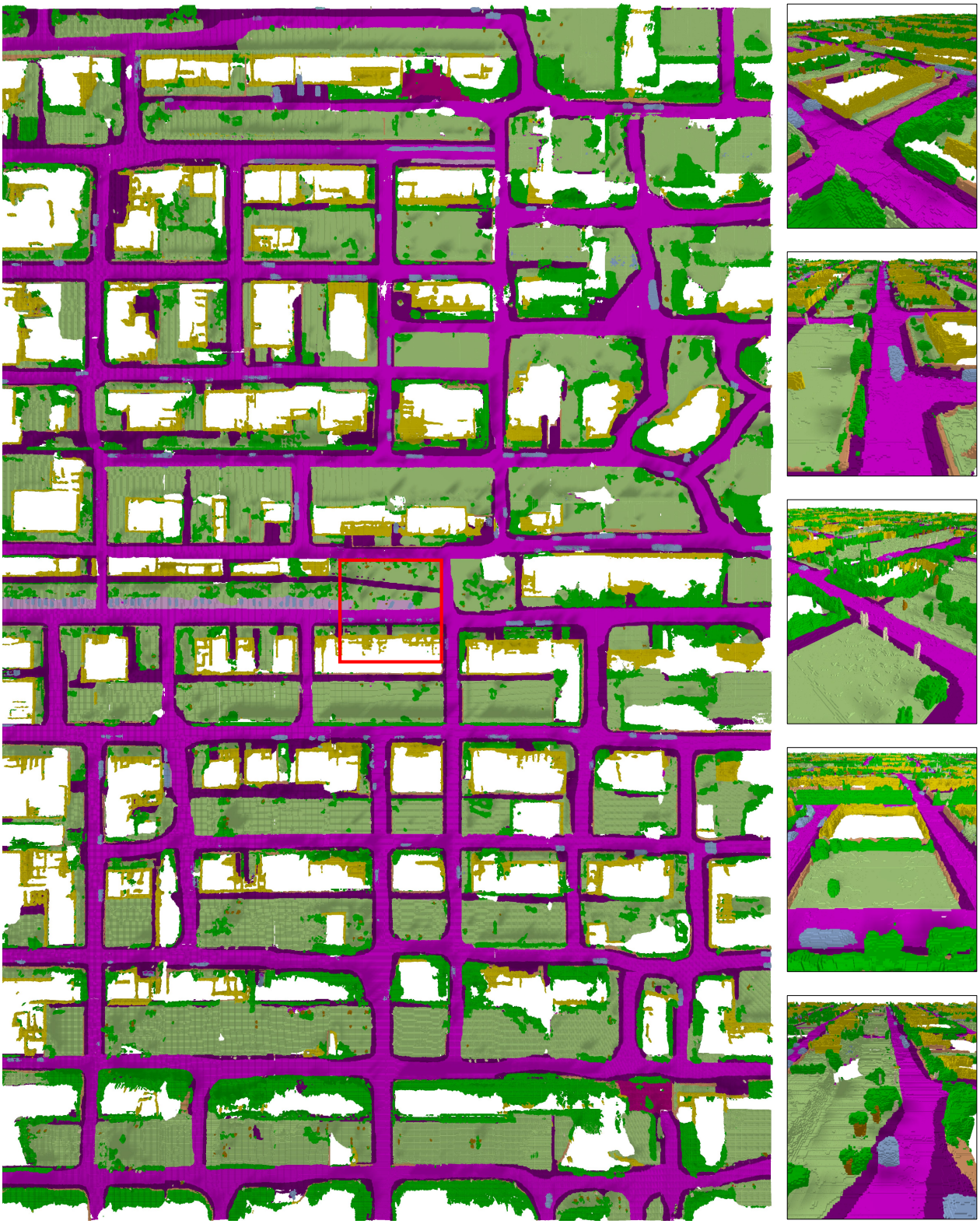


Figure S5. **City-scale outpainted scene.** The first column displays a city-scale scene, showcasing an expansive urban landscape. The city-scale scene is expanded from the original size of $256 \times 256 \times 32$ to $1792 \times 2816 \times 32$. The second column figures provide close-up views of specific areas within the city-scale scene. The red box means an original scene for outpainting.

## C.6. Scene Inpainting

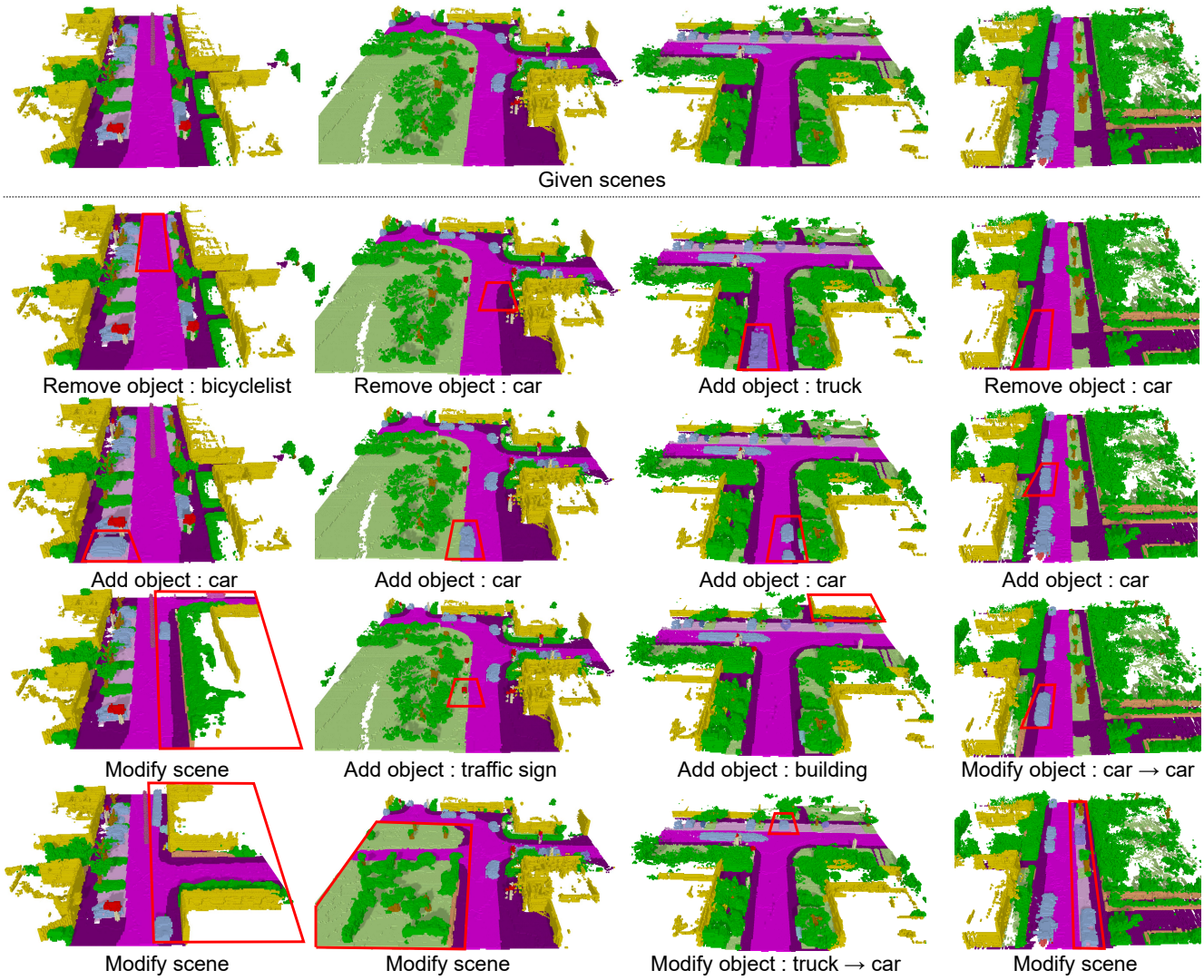



Figure S6. **Scene inpainting results of our method.** The red boxes refer to inpainting regions.

## C.7. Semantic Scene to RGB Image

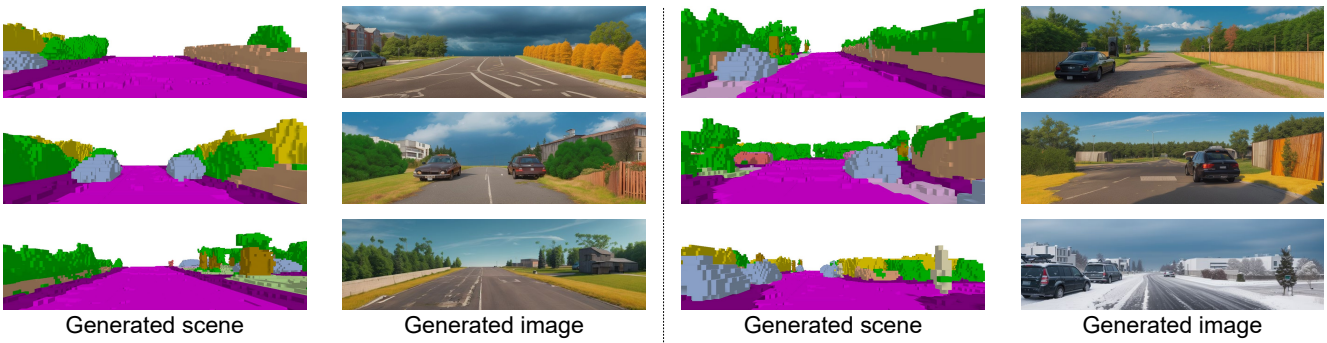



Figure S7. **RGB images generated from our generated scenes.** ControlNet [63] is utilized to generate images from our generated scenes. In the last figure illustrating a snowy scene, we added a text prompt 'snow'.

**Acknowledgement.** This work was supported by IITP (Institute of Information & Communications Technology Planning & Evaluation) and ITRC (Information Technology Research Center), funded by Korea government (MSIT) (RS-2023-00237965(2024) and IITP-2024-2020-0-01460). Prof. Sung-Eui Yoon is a corresponding author.

## References

[1] Ali Abbasi, Sinan Kalkan, and Yusuf Sahillioğlu. Deep 3d semantic scene extrapolation. *The Visual Computer*, 35:271–279, 2019. 3

[2] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12608–12618, 2023. 2

[3] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J. Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12608–12618, 2023. 3

[4] 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. 4, 9

[5] Luca Bartolomei, Lucas Teixeira, and Margarita Chli. Perception-aware path planning for uavs using semantic segmentation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5808–5815, 2020. 3

[6] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 5, 6, 7, 8

[7] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4413–4421, 2018. 4

[8] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018. 6

[9] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3991–4001, 2022. 3, 6, 7, 8, 12

[10] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 2

[11] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020. 3

[12] Yen-Chi Cheng, Chieh Hubert Lin, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, and Ming-Hsuan Yang. Inout: Diverse image outpainting via gan inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11431–11440, 2022. 3

[13] Xu Cui, Chenggang Lu, and Jinxiang Wang. 3d semantic map construction using improved orb-slam2 for mobile robot in edge computing environment. *IEEE Access*, 8:67179–67191, 2020. 3

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 6

[15] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. 10

[16] 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. *Advances in neural information processing systems*, 30, 2017. 6

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 4, 9, 10

[18] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021. 2

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 10

[20] Changho Jo, Woobin Im, and Sung-Eui Yoon. In-n-out: Towards good initialization for inpainting and outpainting. *arXiv preprint arXiv:2106.13953*, 2021. 3

[21] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 2

[22] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2

[23] Zeqiang Lai, Yuchen Duan, Jifeng Dai, Ziheng Li, Ying Fu, Hongsheng Li, Yu Qiao, and Wenhai Wang. Denoising diffusion semantic segmentation with mask prior modeling. *arXiv preprint arXiv:2306.01721*, 2023. 5

[24] Jumin Lee, Woobin Im, Sebin Lee, and Sung-Eui Yoon. Diffusion probabilistic models for scene-scale 3d categorical data. *arXiv preprint arXiv:2301.00527*, 2023. 2, 6

[25] Jiabao Lei, Jiapeng Tang, and Kui Jia. Generative scene synthesis via incremental view inpainting using rgbd diffusion models. *arXiv preprint arXiv:2212.05993*, 2022. 3

[26] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12642–12651, 2023. 2

[27] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9087–9098, 2023. 3

[28] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. *arXiv preprint arXiv:2303.08133*, 2023. 2

[29] Lnyan. Outpainting with stable diffusion on an infinite canvas. https://github.com/lkwq007/stablediffusion-infinity, 2022. 10

[30] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2, 4, 5

[31] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2837–2845, 2021. 2

[32] Ruihang Miao, Weizhou Liu, Mingrui Chen, Zheng Gong, Weixin Xu, Chen Hu, and Shuchang Zhou. Occdepth: A depth-aware method for 3d semantic scene completion. *arXiv preprint arXiv:2302.13540*, 2023. 3, 6, 7, 8, 12

[33] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 4, 9

[34] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2

[36] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 10

[37] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *2020 International Conference on 3D Vision (3DV)*, pages 111–119. IEEE, 2020. 4

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 10

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 5

[40] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 10

[41] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 1, 2, 9

[42] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 2

[43] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022. 10

[44] Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. Diffusion-based signed distance fields for 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20887–20897, 2023. 2

[45] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023. 2

[46] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017. 3

[47] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020. 2

[48] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis. *arXiv preprint arXiv:2303.14207*, 2023. 2, 6

[49] Chaoqun Wang, Wenzheng Chi, Yuxiang Sun, and Max Q-H Meng. Autonomous robotic exploration by incremental road map construction. *IEEE Transactions on Automation Science and Engineering*, 16(4):1720–1731, 2019. 3

[50] Chaoqun Wang, Delong Zhu, Teng Li, Max Q-H Meng, and Clarence W De Silva. Efficient autonomous robotic exploration with semantic road map in indoor environments. *IEEE Robotics and Automation Letters*, 4(3):2989–2996, 2019. 3

[51] Xinying Wang, Dikai Xu, and Fangming Gu. 3d model inpainting based on 3d deep convolutional generative adversarial network. *IEEE Access*, 8:170355–170363, 2020. 3

[52] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. Pet-neus: Positional encoding tri-planes for neural surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12598–12607, 2023. 4

[53] Zhen Wang, Shijie Zhou, Jeong Joon Park, Despoina Paschalidou, Suya You, Gordon Wetzstein, Leonidas Guibas, and Achuta Kadambi. Alto: Alternating latent topologies for implicit 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 259–270, 2023. 2

[54] Joey Wilson, Jingyu Song, Yuewei Fu, Arthur Zhang, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. Motionsc: Data set and network for real-time semantic mapping in dynamic environments. *arXiv preprint arXiv:2203.07060*, 2022. 5, 6

[55] Rundi Wu, Ruoshi Liu, Carl Vondrick, and Changxi Zheng. Sin3dm: Learning a diffusion model from a single 3d textured shape. *arXiv preprint arXiv:2305.15399*, 2023. 2

[56] Zhaoyang Xia, Youquan Liu, Xin Li, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, and Yu Qiao. Scpnet: Semantic scene completion on point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17642–17651, 2023. 3, 6, 7, 12

[57] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3101–3109, 2021. 6

[58] Xuemeng Yang, Hao Zou, Xin Kong, Tianxin Huang, Yong Liu, Wanlong Li, Feng Wen, and Hongbo Zhang. Semantic segmentation-assisted scene completion for lidar point clouds. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3555–3562. IEEE, 2021. 3, 7, 8, 12

[59] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 3

[60] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengzhe Liu, and Xiaojuan Qi. Texture generation on 3d meshes with point-uv diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4206–4216, 2023. 2

[61] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[62] Guangyao Zhai, Evin Pınar Örnek, Shun-Cheng Wu, Yan Di, Federico Tombari, Nassir Navab, and Benjamin Busam. Commonscenes: Generating commonsense 3d indoor scenes with scene graphs. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 6

[63] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 8, 10, 15

[64] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10188–10198, 2023. 2

[65] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 10

[66] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 10

[67] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 2