# 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, [4] 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. [1] 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 [8], 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 [12] 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 [13]. The diffusion settings (*e.g.*, the variance schedule $\beta_t$) are used as DDPM [4].

### 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 [6] 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 [16] 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 [11], Textual inversion [3], and Lora [5]). 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 [9, 17, 18]. For our experiments, we employ the diffusion model [10] weights[1] 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 |

### C.1. Triplane Visualization

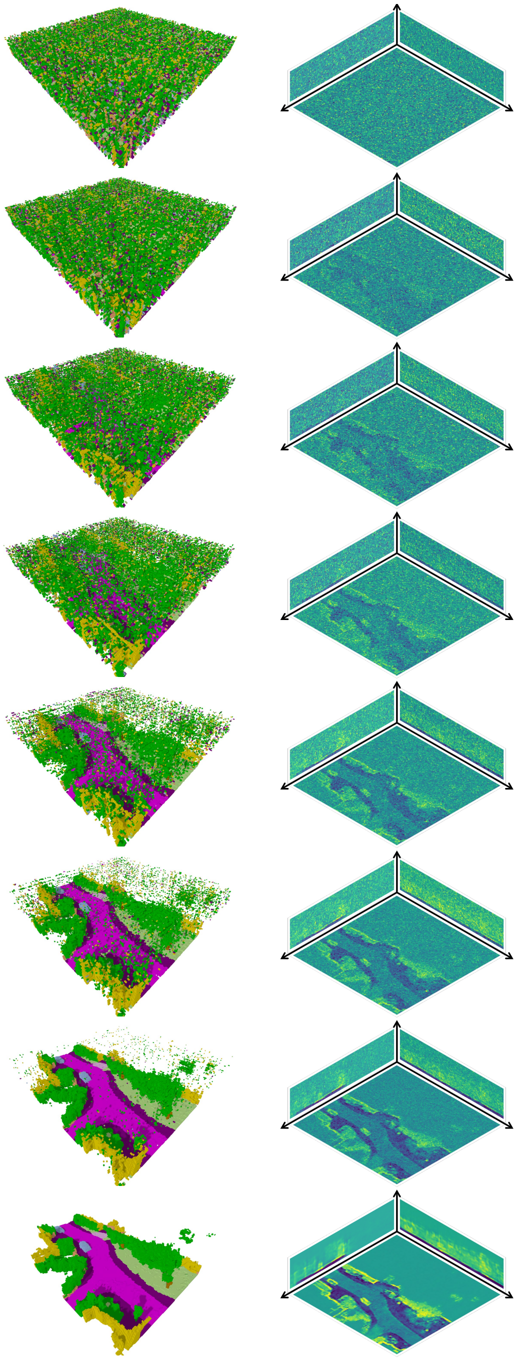


Figure S1. **Triplane visualization during our generation process.** We visualize triplanes (right) and their corresponding scenes (left) according to diffusion steps. We observe distinct denoising patterns where our diffusion model initially constructs low-frequency structures (*e.g.*, roads) in the 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 [4]; we expect this property to be exploited for elastic scene editing in future work.

[1] https://civitai.com/models/119169/urban-streetview

## C.2. Scene Generation

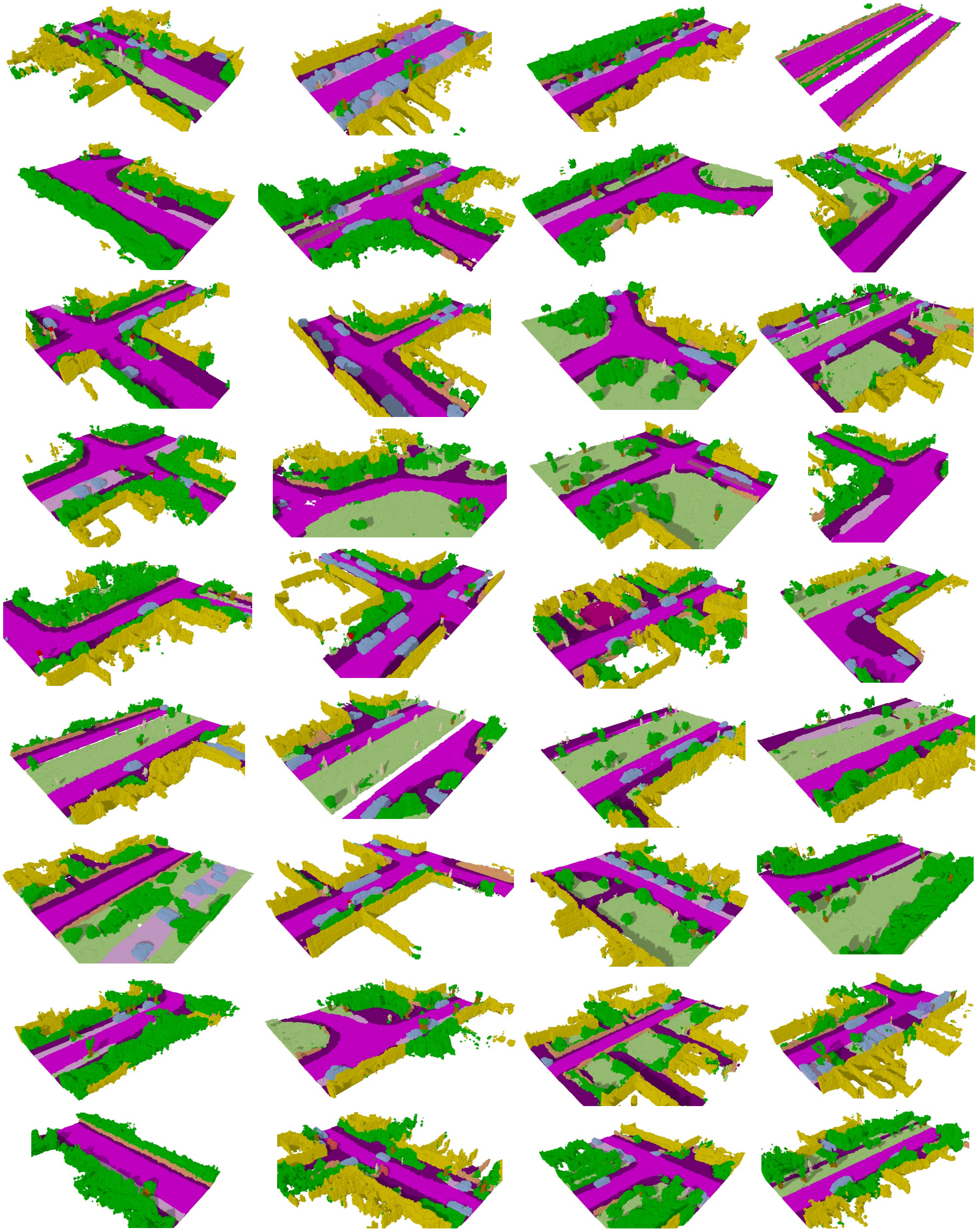


Figure S2. **Scene generation results of our method.** The generated scenes demonstrate various road shapes, including L, T, Y, straight, and crossroads, which show that our method generates diverse samples.

### C.3. Semantic Scene Completion Refinement

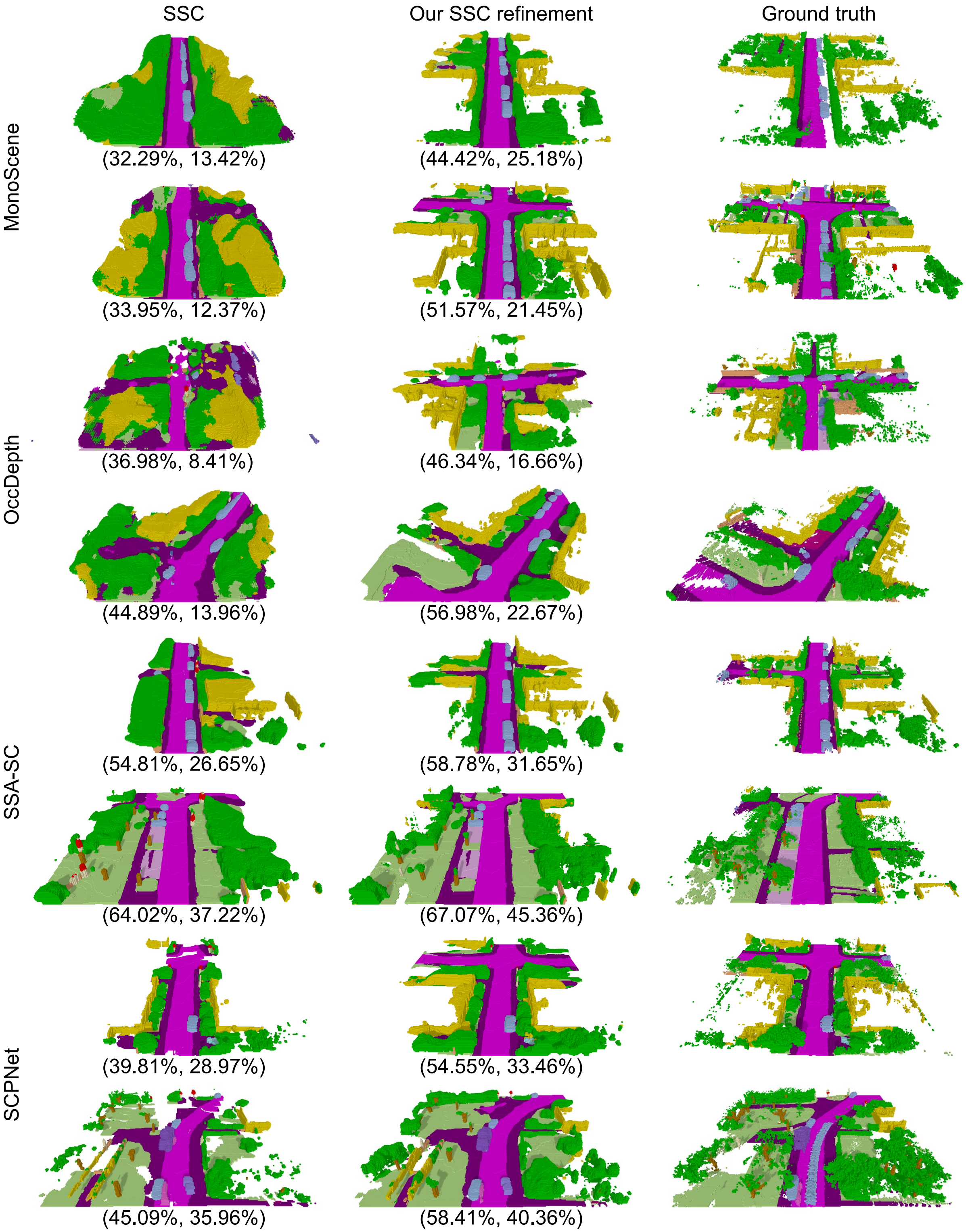


Figure S3. **Results of semantic scene completion refinement of our method.** The parentheses report the SSC metrics as (IoU, mIoU). Our method refines the results of state-of-the-art SSC methods. The MonoScene [2] and OccDepth [7] methods use a RGB input. The SSA-SC [15] and SCPNet [14] employ LiDAR point clouds as an input.

### C.4. Scene Outpainting

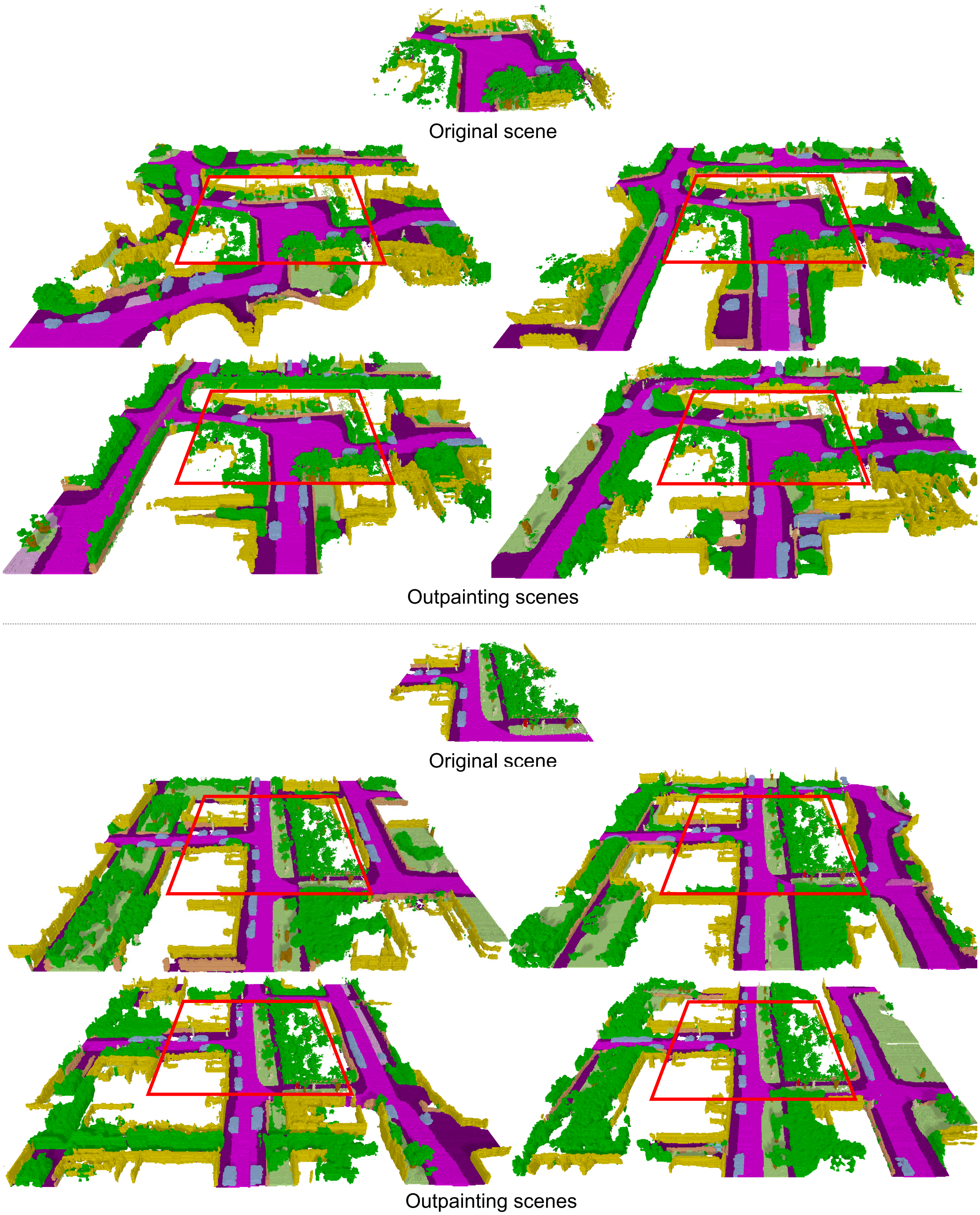


Figure S4. **Scene outpainting results of our method.** We visualize various outpainting results generated from two scenes. The outpainted scene is expanded from the given size of $256 \times 256 \times 32$ to $512 \times 512 \times 32$ without any guidance. The red boxes mean an original scene for outpainting.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

### C.5. City-level Generation

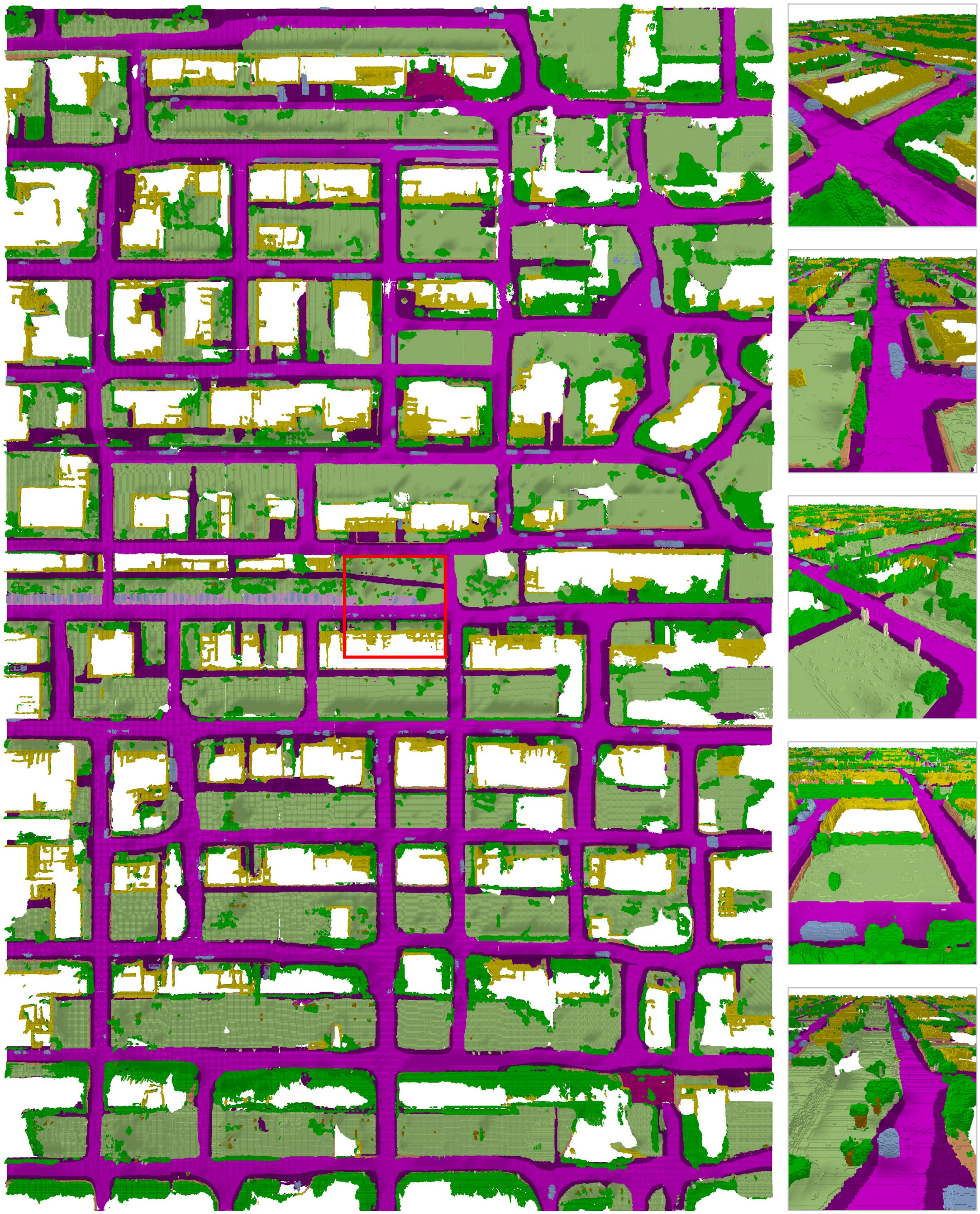
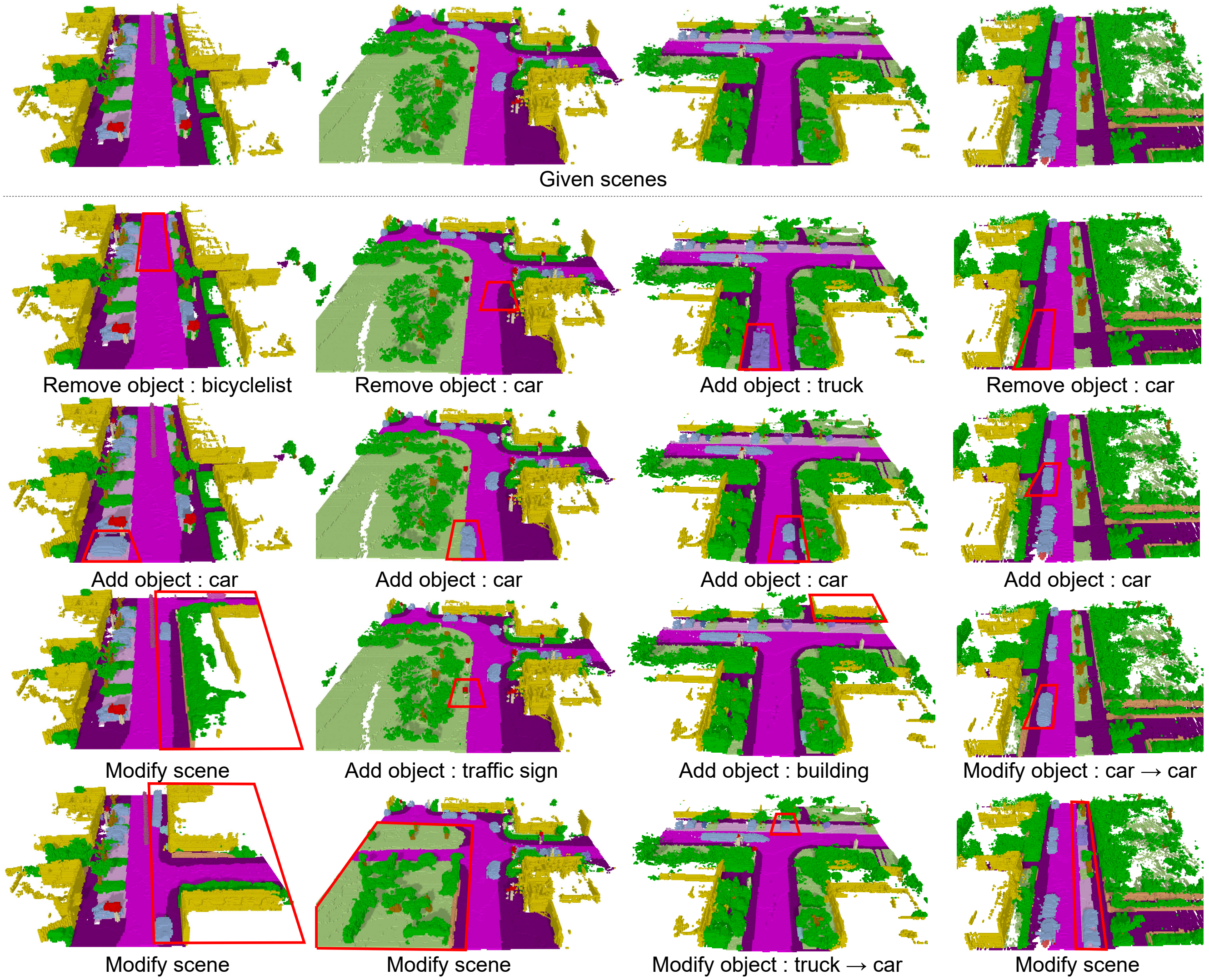


Figure S5. **City-scale outpainted scene.** The first column displays a city-scale scene, showcasing an expansive urban landscape. The city-scale scene is expanded from the original size of 256 × 256 × 32 to 1792 × 2816 × 32. The second column figures provide close-up views of specific areas within the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

### C.6. Scene Inpainting



Given scenes

Remove object : bicyclelist

Remove object : car

Add object : truck

Remove object : car

Add object : car

Add object : car

Add object : car

Add object : car

Modify scene

Add object : traffic sign

Add object : building

Modify object : car → car

Modify scene

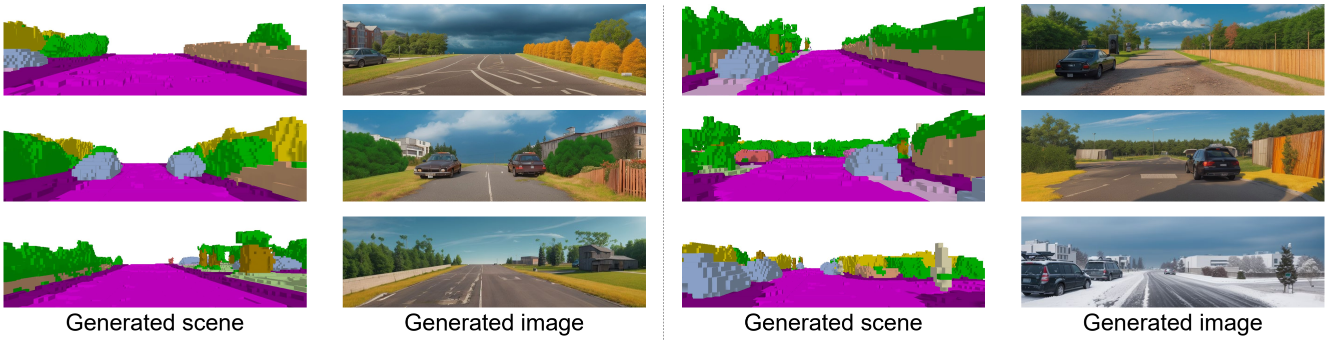
Modify scene

Modify object : truck → car

Modify scene

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

### C.7. Semantic Scene to RGB Image



Generated scene

Generated image

Generated scene

Generated image

Figure S7. **RGB images generated from our generated scenes.** ControlNet [16] is utilized to generate images from our generated scenes. In the last figure illustrating a snowy scene, we added a text prompt 'snow'.

## References

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

[2] 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. 4

[3] 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. 2

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

[5] 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. 2

[6] Lnyan. Outpainting with stable diffusion on an infinite canvas. https://github.com/lkwq007/stablediffusion-infinity, 2022. 2

[7] 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. 4

[8] 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. 1

[9] 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. 2

[10] 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. 2

[11] 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. 2

[12] 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

[13] 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. 2

[14] 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. 4

[15] 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. 4

[16] 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. 2, 7

[17] 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. 2

[18] 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. 2