EditSSC: Toward Editable Semantic Occupancy Scenes with Unconditional Diffusion Models

Fatima Baldé[1] Raoul de Charette[1] Alexandre Boulch[1,2]
[1]Inria, [2]Valeo.ai

https://astra-vision.github.io/EditSSC

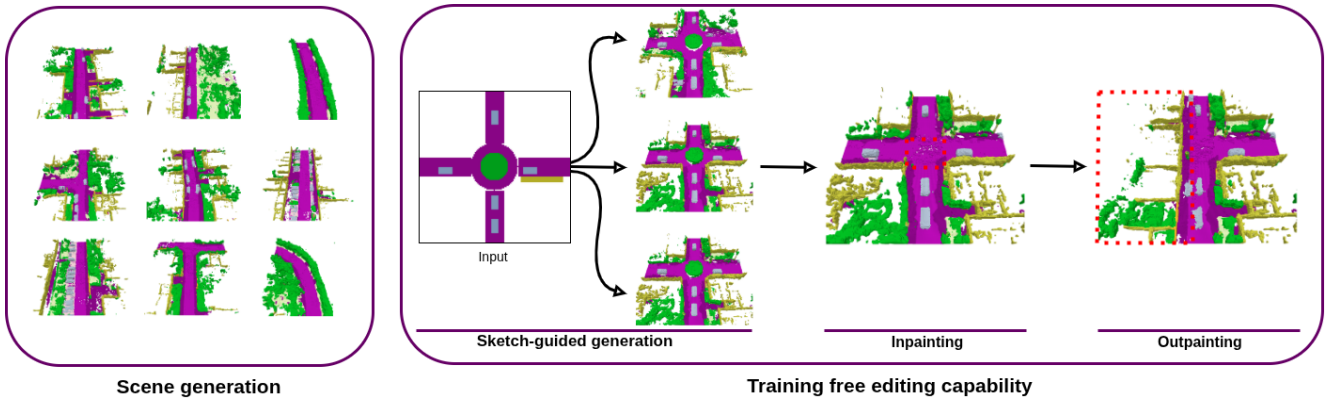

Figure 1. **EditSSC capabilities.** Our scene generation method relies on a latent diffusion model with carefully designed components to enable *training-free* editing capability. While unconditional scene generation (left) can produce multiple diverse samples, all generated scenes faithfully follow the training distribution. EditSSC further enables editing via sketch guidance (using a user-provided layout), inpainting, and outpainting. We illustrate these capabilities (right) through a sequential editing process starting from the generation of a rare heavy-traffic roundabout scene, which is then edited by removing the roundabout region and inpainting it (*cf*., red highlight). The scene is subsequently outpainted to extend the left portion of the layout. These editing capabilities enable more diverse and controllable scene generation, going beyond what the training data alone provides.

Abstract

3D semantic scene generation is crucial for autonomous driving applications, yet most methods rely on complex 3D-specific architectures such as triplane encoders and adapted diffusion networks, limiting both their simplicity and their editing capabilities. We propose EditSSC, an editing-ready method for 3D semantic scene generation using 2D Bird's Eye View (BEV) representations and off-the-shelf latent diffusion network. Our approach reshapes 3D semantic occupancy grids into multi-channel BEV images and leverages the quantized autoencoder and UNet from Stable Diffusion with minimal modifications. We perform diffusion on the latents after quantization, which enables training-free editing capabilities. By exploiting class-to-code correspondences in the codebook, our method supports sketch-guided generation, inpainting, and outpainting without any retraining. On SemanticKITTI, EditSSC outperforms existing 3D-specific baselines on unconditional generation, demonstrating that well-established 2D architectures can be effectively repurposed for 3D scene generation and editing.

1. Introduction

3D scene generation aims to create geometrically and semantically consistent scenes, with applications in data enrichment and on-the-fly simulation for gaming or navigation. With conditioning and editability, the generated distribution can be steered toward desired targets beyond the training data.

Recently, latent diffusion models have become the dominant framework for high-quality generation, especially in image synthesis [6, 31], and have naturally been extended to 3D across modalities such as point clouds [37], meshes [35], and voxel grids [16, 39]. In the context of scene generation, this entails producing multiple objects and surfaces under realistic spatial constraints. While indoor settings involve restricted layouts with complex object arrangements [36], outdoor generation focuses on large-scale scenes with objects placed on a ground plane. In this paper, we focus on the latter.

Such large-scale generation typically relies on 3D-specific architectures that explicitly model geometric structure, achieving strong results at the cost of increased com-

plexity and design effort. For example, in SemCity [16] and BlockFusion [39], scenes are encoded as latent triplanes requiring diffusion networks adapted to this representation and resulting in a complex plane-sharing UNet architecture.

In this paper, we study semantic scene occupancy generation relying on a carefully designed diffusion-based architecture which unlocks novel scene editing capabilities. Such a property is particularly useful for domains where data diversity is limited, such as autonomous driving, as it allows generating out-of-training-distribution samples, for instance for training more robust models.

In Sec. 3, we conduct a pilot study to determine the desirable features of an editing-ready pipeline. In line with recent findings advocating for structured latent space with diffusion models [17], our observations suggest that the organization of the latent space is more crucial for better diffusion than reconstruction performance alone, and that BEV representations are better suited for editing tasks.

Building on these observations, we propose a simple and efficient, editing-ready method, coined **EditSSC**. It uses a 2D representation, allowing us to leverage diffusion pipelines developed for image processing. We first adapt with minimal changes the existing quantized autoencoders for images to ingest voxel grids, applying diffusion to the quantized latents, and train a lightweight version of the Stable Diffusion UNet for generation in the latent space. While this choice yields comparable results on SemanticKITTI for unconditional scene generation, it enables exploiting an overlooked property of vector quantization for 3D generation. Specifically, the discrete codebook allows us to retrieve class prototypes which can be used at inference to condition the model without any retraining nor test-time adaptation. As a result, as highlighted in Fig. 1, EditSSC supports sketch-guided scene generation as well as scene editing via inpainting and outpainting.

To summarize, our contributions are the following:

- We repurpose standard 2D diffusion pipelines for 3D semantic occupancy scene generation, showing that well-established 2D architectures can be effectively repurposed for this task;
- We show that our simple pipeline achieves competitive performance on unconditional scene generation;
- We demonstrate training-free editing capabilities based on our architecture, including sketch-guided generation, inpainting, and outpainting.

2. Related work

Diffusion models. Diffusion models [10] aim at fitting a distribution through an iterative denoising process. Given their tremendous performance, diffusion models have been applied to various fields ranging from image generation [31] and texture synthesis [23] to autonomous driving [19].

In image processing in particular, diffusion models are employed in various settings for inpainting or outpainting [24, 32], as well as for text-to-image conditional generation [31, 33]. Performance for both conditional and unconditional generation has drastically improved via representation alignment with self-supervised models such as DinoV2 [28, 42], the enforcement of invariance to data transformation [14] or the introduction of more efficient guidance processes, *e.g.*, classifier-free guidance [9].

3D scene generation. The success of diffusion models for image generation has naturally led to the exploration of 3D object generation with various data representations such as voxel grids [18, 26, 44], meshes [22], implicit functions [12, 34, 35] or point clouds [25, 37]. More closely related to our work, scene generation aims at producing a 3D arrangement of multiple shapes. Owing to its complexity, it is only recently that diffusion models have been employed for such a task. As for objects, various scene representations have been explored using diffusion models for indoor [1, 11, 27], outdoor [13, 15, 16, 21] or both [29, 39]. In particular, SSD [15] and SemCity [16] tackle 3D outdoor generation with triplane representation. At the cost of a 3D-specific design, they propose a triplane autoencoder and a UNet-based diffusion model adapted to process triplanes. These generative processes have also been used to improve performance on Semantic Scene Completion (SSC), by directly conditioning the generation on a LiDAR input [5] or by using diffusion models to refine the output of existing SSC methods [16]. Such conditioning is however more complex for methods employing triplane representations.

We discuss more closely the choice of representation and general design choices for the diffusion models.

3. Pilot study

From Sec. 2 it follows that recent generative SSC methods typically use two-stage diffusion models with scenes encoded as triplane [5, 15, 16, 40] and diffusion in latent space [15, 16, 40] learned with a (variational) autoencoder. As our goal is to generate and edit 3D semantic scenes, it raises two questions, which we investigate below.

What makes a good editing space? The ability to control or edit the generated scenes is particularly desirable for applications like autonomous driving where the available scenes come (i) in limited number and (ii) with a limited diversity. SemanticKITTI [2], the most popular SSC dataset, is no exception since its training set consists of only 10 sequences from the same German neighborhoods.

Editing generated scenes is not new and was already addressed by SSEditor [43] which relies on categorical masks for scene editing. However, their use of triplane masks is not intuitive for editing as the user needs to draw category profiles in all three directions. Instead, Bird's Eye View (BEV) appears to be a natural choice, since driving scenes are mainly spread along two axes as objects are

Representation	Autoencoder IoU↑	Autoencoder mIoU↑	Diffusion FID↓	Diffusion CKL↓	Diffusion Prec↑
SemCity w/ Autoencoder					
Triplane [16]	84.84	84.65	104.1	**0.0936**	0.0329
BEV	80.30	77.84	120.1	0.1310	0.0453
SemCity w/ VQ-VAE					
BEV	80.10	68.35	**97.5**	0.0968	**0.0249**
MLP					
BEV	**98.90**	**98.50**	156.9	0.0312	0.0115

Table 1. **Architectures for diffusion**. We report the original SemCity [16] (triplane) along with two BEV variants using the original SemCity autoencoder (top), a VQ-VAE (middle) and a simple MLP (bottom). Despite achieving near-perfect reconstruction, the MLP yields the worst diffusion performance, while the VQ-VAE achieves the best despite lower reconstruction scores. This confirms that latent space structure matters more than reconstruction fidelity for diffusion quality. *Cf*. Sec. 5 for metrics details.

rarely stacked. This observation suggests that 2D conditioning should be sufficient for good generation. In addition, BEV representations are practically easier and more intuitive to edit than a full 3D scene.

We observe that existing triplane diffusion can be easily adapted to BEV diffusion by simply pooling the feature volume of the encoder along the vertical direction only. We experiment with this using the SemCity [16] architecture. Specifically, we observe that the autoencoder reconstruction performance drops by 4 IoU and 7 mIoU when switching from triplane to BEV representation. Subsequently, one might be tempted to assess the superiority of triplane representation. Instead, we question the relationship between reconstruction and diffusion performance.

Which design for the autoencoder? For generative SSC, it is rather common [15, 40] to justify architectural choices based on autoencoder reconstruction performance, as this is significantly easier to obtain than diffusion results. This follows a somewhat intuitive belief that better reconstruction performance leads to better generative capability. However, recent work for images [17] has shown that the structure of the latent space, in particular its smoothness and regularity, plays a critical role in diffusion quality, sometimes more than reconstruction fidelity. To investigate whether similar observations hold for 3D scene generation, we train the diffusion stage on the SemCity [16] architecture and report unconditional scene generation performance.

In Tab. 1 ('SemCity w/ Autoencoder'), the diffusion performance of the above-mentioned variants correlates with the autoencoder performance, *i.e*., BEV representation performs worse than triplane. We then replace the original SemCity autoencoder with a vector-quantized VAE (VQ-VAE), using BEV representation. Results in Tab. 1 ('SemCity w/ VQ-VAE') reveal a different picture. While the VQ-VAE reconstruction performance is significantly lower than its AE counterpart (-9.5 mIoU↑), the resulting diffusion performance is notably better (-23 FID↓).

For further investigation, we additionally train a simple MLP-based autoencoder that flattens each pillar of the voxel grid into a BEV feature vector. The latter achieves near-perfect reconstruction (IoU: 98.9, mIoU: 98.5), yet yields the worst diffusion performance by a large margin (FID: 156.9), as the latent manifold is sparse and irregular. This confirms that high reconstruction fidelity with an unstructured latent space is detrimental to diffusion [17].

Altogether, these observations suggest that reconstruction quality alone is not a reliable proxy for generation capability. Instead, the discrete and compact latent space imposed by vector quantization provides a structured and regular representation, which is more amenable to diffusion modeling. This is consistent with the findings of [17], showing that latent smoothness and regularity are key properties for effective latent diffusion. It follows that the design of the autoencoder should not only target high reconstruction scores, but also consider the structure of the resulting latent space and its amenability to diffusion modeling.

4. Method

Our method, coined EditSSC, is designed for semantic 3D scene generation with editing capability. As illustrated in Fig. 2, it relies on a classical two-stage latent diffusion scheme but builds on key observations from Sec. 3. First, we use a BEV encoding as it is easier to manipulate. Second, we rely on a vector-quantized autoencoder (VQ-VAE) which has two benefits: (i) it was shown to perform well with BEV representation (*cf*. Tab. 1), (ii) it has a unique property that enables training-free in/out-painting. The application of these careful design choices enables editing-ready 3D semantic scene generation.

In detail, as shown in Fig. 2, in the first stage (Sec. 4.1) we train the VQ-VAE from Stable Diffusion [31] to compress 3D semantic occupancy scenes into compact 2D latent representations. The key idea is to reshape the 3D voxel grid into a Bird's Eye View (BEV) image by folding the height dimension into the channel dimension, allowing us to directly leverage a proven image autoencoder without any 3D-specific module. In the second stage (Sec. 4.2) we train a lightweight version of the Stable Diffusion UNet on the quantized BEV latents to generate new scenes. Finally, we show in Sec. 4.3 that the discrete structure of the VQ-VAE enables training-free editing capabilities.

4.1. Autoencoder

We consider a 3D semantic occupancy scene represented as a voxel grid of shape $X \times Y \times Z$, where each voxel contains a semantic class label. To process this 3D input with a 2D image autoencoder, we first map each class label to a learned embedding vector of dimension D using an em-

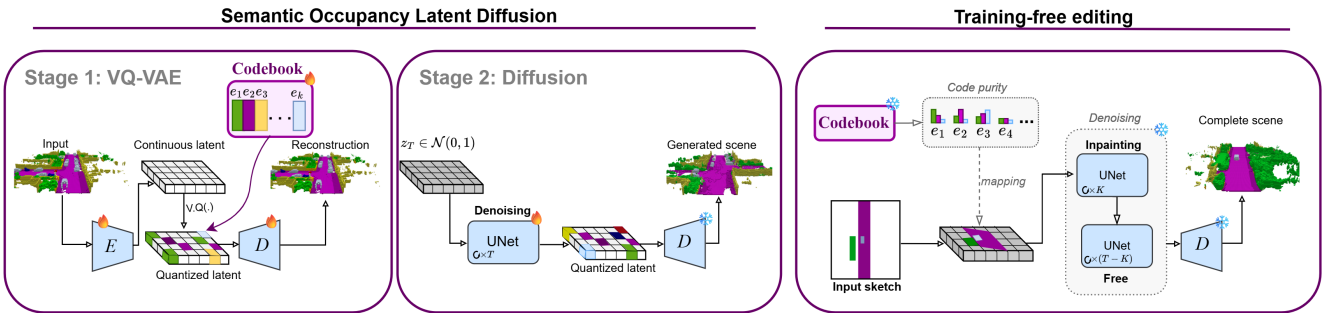

Figure 2. **Overview of EditSSC.** The training consists of two stages. In the first stage (Sec. 4.1), a 3D semantic occupancy scene is passed through a VQ-VAE that compresses it into discrete latent codes. In the second stage (Sec. 4.2), a lightweight U-Net performs diffusion on the quantized latents to generate new scenes. The discrete codebook further enables training-free editing via class-to-code correspondences (Sec. 4.3).

bedding layer, yielding a tensor of shape $X \times Y \times Z \times D$. We then reshape this tensor by folding the height and embedding dimensions into the channel axis, obtaining a 2D feature map of shape $X \times Y \times (Z \cdot D)$, which can be interpreted as a multi-channel BEV image encoding the full vertical structure of the scene at each spatial position.

This BEV image is passed through the VQ-VAE of Stable Diffusion [31], by modifying the number of input channels from 3 (RGB) to $Z \cdot D$ and adapting the latent space dimensionality accordingly. The encoder compresses the BEV image into a spatial grid of discrete latent codes via vector quantization. At decoding time, the VQ-VAE decoder reconstructs a feature map of shape $X \times Y \times (Z \cdot D)$, which is reshaped back into a 3D volume of shape $X \times Y \times Z \times D$. A classification head then maps each voxel feature to semantic class logits.

Subsequently, the autoencoder jointly trains the embedding layers and classification head in an end-to-end fashion, with the following combined loss:

$$\mathcal{L}_{\text{VQ-VAE}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{Lovász}} + \lambda \mathcal{L}_{\text{quant}} \tag{1}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss providing per-voxel supervision, $\mathcal{L}_{\text{Lovász}}$ is the Lovász-Softmax loss [3] which directly optimizes the intersection-over-union (IoU) metric, improving the handling of underrepresented classes, and $\mathcal{L}_{\text{quant}}$ is the VQ-VAE quantization loss [38] defined as:

$$\mathcal{L}_{\text{quant}} = \|\,\text{sg}[\mathbf{z}_e(\mathbf{x})] - \mathbf{e}\|_2^2 + \beta \|\mathbf{z}_e(\mathbf{x}) - \text{sg}[\mathbf{e}]\|_2^2 \tag{2}$$

where $\mathbf{z}_e(\mathbf{x})$ is the encoder output, $\mathbf{e}$ is the nearest codebook entry, $\text{sg}[\cdot]$ denotes the stop-gradient operator, and β controls the commitment weight. λ balances the quantization loss with the reconstruction losses.

4.2. Diffusion

Having trained the VQ-VAE, we extract the latent representations of all samples in our training set. We then train a denoising diffusion probabilistic model (DDPM) [10] directly on the latents obtained after quantization, rather than on the continuous latents before quantization, which we found to yield more stable results (Sec 5.5).

Forward process. Given a clean BEV latent $\mathbf{z}_0$, the forward process gradually adds Gaussian noise over $T = 1000$ steps, producing a sequence of increasingly noisy latents $\mathbf{z}_1, \ldots, \mathbf{z}_T$. Each step follows $q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\,\mathbf{z}_{t-1},\ \beta_t \mathbf{I})$, where β_t is a variance schedule. A useful property of this formulation is that one can directly sample $\mathbf{z}_t$ at any timestep as $q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\,\mathbf{z}_0,\ (1-\bar{\alpha}_t)\,\mathbf{I})$, with $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$. At $t = T$, the latent is approximately distributed as pure Gaussian noise.

Reverse process. A denoising network D_ϕ is trained to reverse this corruption. Following x_0-parameterization [10], the network directly predicts the clean sample $\mathbf{z}_0$ from a noisy input $\mathbf{z}_t$ and the timestep t, by minimizing:

$$\mathcal{L}_D = \mathbb{E}_{t\sim\mathcal{U}(1,T)} \|\mathbf{z}_0 - D_\phi(\mathbf{z}_t, t)\|_2^2 \tag{3}$$

For D_ϕ, we use the UNet architecture from Stable Diffusion [31] retaining attention layers only at the lowest resolution level before the bottleneck. This results in a significantly lighter model, suited to the lower complexity of BEV latent maps compared to natural images.

At inference, we start from $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively denoise following the DDPM reverse process. The generated latent $\mathbf{z}_0$ is then decoded through the VQ-VAE decoder and classification head to produce a 3D semantic occupancy scene.

4.3. Training-free editing

An interesting property of our quantized latent space is that the codebook entries exhibit strong correspondences with semantic classes, as illustrated in Fig. 2 (right). To quantify this, we use the purity of a codebook entry, defined as the proportion of voxels assigned to that entry that belong to its

most frequent class. Based on the training set, we observe that most codebook entries achieve high purity, *i.e.*, they tend to be mostly associated with a single semantic class. Subsequently, we build a class-to-code mapping by selecting for each class the codebook entry that is both most frequently used by that class and has high purity. In practice, the two criteria are highly correlated. This mapping enables training-free editing capabilities without any retraining nor test-time adaptation.

Sketch guidance. Assuming a user-provided BEV sketch, we convert it into latent codes by leveraging our class-to-code mapping and then perform T denoising steps with layout guidance to complete unknown areas. Inspired by RePaint [24], we replace the known regions with the layout codes at each denoising step, but only for the first K steps. For the last $T - K$ steps, we release the constraint and let the model denoise the full latent freely, allowing it to refine the shape and boundaries of the user-specified objects for better coherence with the surrounding scene.

Inpainting/Outpainting. The above sketch guidance principles extend to inpainting and outpainting, where the known fraction of the scene is preserved at every denoising step, and the model generates coherent content in the missing (or outer) regions throughout the full T denoising steps.

5. Experiments

In the following, we evaluate EditSSC performance along several axes. We first evaluate the quantitative and qualitative performance of our autoencoder in Sec. 5.1 and unconditional scene generation in Sec. 5.2, while comparing against variants of SemCity [16]. We then evaluate downstream tasks such as LiDAR-conditioned generation in Sec. 5.3, comparing to various SSC baselines. We also present qualitative results of the training-free editing capabilities, including sketch-guided generation, inpainting, and outpainting in Sec. 5.4. Last, we ablate our design and architecture choices in Sec. 5.5.

Dataset. We conduct all our experiments on SemanticKITTI [2], the reference for SSC. The dataset includes a collection of 11 large-scale outdoor sequences of LiDAR scans with dense semantic occupancy annotations. Scenes are encoded as voxel grids of size $256 \times 256 \times 32$ with a resolution of $0.2\,\text{m}$ per voxel, covering a spatial extent of $51.2 \times 51.2 \times 6.4$ meters, with 20 semantic classes. We train our models on the training split and evaluate the VQ-VAE reconstruction on the validation split.

Metrics. We evaluate the quality of generated scenes using five metrics. We compute the Fréchet Inception Distance (FID) [8] and Kernel Inception Distance (KID) [4] on BEV images obtained by projecting the top semantic class of each scene onto the ground plane, where each class is mapped to its corresponding color. We also compute Precision and Recall to evaluate the fidelity and diversity of the generated scenes, respectively. Additionally, we report the Categorical KL divergence (CKL), which measures the KL divergence between the per-class frequency distributions of the training set and the generated set, computed on the full 3D voxel grids. This metric captures whether the generated scenes preserve the overall class distribution of the training data. All metrics are computed over 5000 generated samples, ensuring reliable FID estimation.

Unlike SemCity [16], which computes these metrics on frontal view renderings, we evaluate on BEV images, which better capture the spatial layout and overall structure of outdoor scenes. For conditional generation tasks, where ground truth is available, we additionally report the mean Intersection-over-Union (mIoU) and per-class Intersection-over-Union (IoU) to evaluate the semantic accuracy of the generated scenes.

5.1. Autoencoder evaluation.

We first evaluate the reconstruction quality of our VQ-VAE on the SemanticKITTI validation set, and compare it against the SemCity autoencoder in its triplane and BEV variants, as well as the SemCity BEV autoencoder augmented with vector quantization (BEV VQ-VAE).

	Autoencoder		Diffusion		
Method	IoU↑	mIoU↑	KID↓	CKL↓	Prec↑
SemCity (triplane) [16]	84.84	84.65	104.1	0.0936	0.0329
SemCity (BEV) [16]	80.30	77.84	120.1	0.1310	0.0453
SemCity (BEV VQ-VAE)	80.10	68.35	97.5	0.0968	**0.0249**
EditSSC (ours)	81.90	72.20	**84.9**	**0.0818**	0.0362

Table 2. **Reconstruction and diffusion performance.** Extending Tab. 1, we report the performance of EditSSC alongside variants of SemCity. As discussed in Sec. 3, we purposely do not highlight best autoencoder performance, as the structure of the latent space plays a more critical role than reconstruction fidelity for generation quality.

Autoencoder metrics are reported in Tab. 2, extending the prior results in Tab. 1. As observed in Sec. 3, the SemCity triplane autoencoder achieves the highest reconstruction scores, benefiting from its rich three-plane representation. However, as detailed in 3, this does not translate into better generation performance. Among quantized architectures, EditSSC, which employs a VQ-VAE based on Stable Diffusion, outperforms the SemCity BEV VQ-VAE in both IoU and mIoU, confirming the effectiveness of repurposing a standard 2D autoencoder for 3D semantic occupancy.

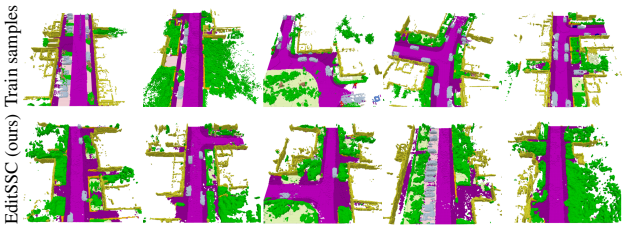

Figure 3. **Unconditional generation**. Our method generates plausible scenes (bottom) which follow the class distribution and general structure of the training set (top).

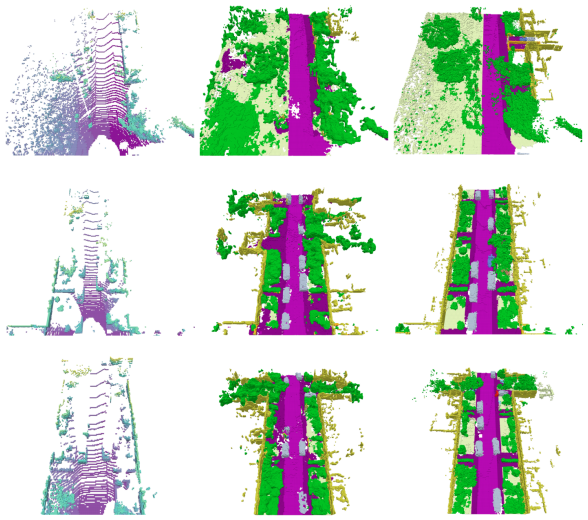

Figure 4. **LiDAR-conditioned generation.** Given a LiDAR scan (left), our model generates a semantic occupancy scene (middle) which resembles the ground truth (right).

5.2. Unconditional generation.

For unconditional generation, we sample a random BEV latent $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively denoise it using the trained UNet following the DDPM [10] reverse process. The resulting latent $\mathbf{z}_0$ is then passed through the VQ-VAE decoder and classification head to reconstruct a full 3D semantic occupancy scene.

We report results of both our method and all three variants of SemCity, using 100 denoising steps at inference, with the training set as reference distribution. Diffusion performance in Tab. 2 shows that our fully 2D pipeline outperforms baselines on two out of three metrics, with a large gap in KID (-6.6) and CKL (-0.015) showing that it produces not only more realistic scenes but also more faithfully reproduces the training class distribution. Notably, even the SemCity BEV VQ-VAE variant, which uses 3D convolutions in its encoder, falls behind our approach based on a standard 2D VQ-VAE. This further reinforces that 3D-specific modules such as 3D convolutions are not necessary for high-quality outdoor semantic occupancy generation, and that well-established 2D architectures can be effectively repurposed for this task.

Method	IoU↑	mIoU↑
SSC		
TS3D [7]	50.6	17.7
LMSCNet [30]	55.7	17.0
JS3C-Net [41]	57.0	24.0
DPS2CNet [20]	60.8	26.7
DiffSSC [5]	60.3	26.7
Editable SSC		
EditSSC (ours)	42.1	12.5

Table 3. **LiDAR-conditioned generation on SemanticKITTI.** We report the performance of general SSC methods as well as editable SSC methods on SemanticKITTI (val. set). We intentionally omit SSEditor, which does not report LiDAR-conditioned performance.

In Fig. 3, we also report examples of scenes generated by our method, alongside reference samples from the training set. Visuals confirm the ability of EditSSC to produce realistic scenes, which visually resemble those seen in the training set.

5.3. Conditional generation.

We also evaluate the ability of EditSSC to perform LiDAR-conditioned scene generation, a well-established task. To do so, we voxelize the input LiDAR scan, which is then passed to an encoder composed of 3D convolutional layers. The resulting features are then pooled along the height axis to obtain a BEV feature map, which is concatenated with the noisy latent as input to the diffusion model. The model is trained end-to-end with this conditioning signal.

We report results in Tab. 3 alongside existing SSC methods, highlighting that EditSSC is the only truly editable SSC method that supports LiDAR-conditioned generation. Notably, SSEditor [43] does not support LiDAR-conditioned generation due to its complex architecture. Compared to other SSC techniques, we observe a gap in IoU and mIoU, as existing SSC methods largely outperform EditSSC, highlighting the need for further research on editable SSC. Yet, qualitative results in Fig. 4 demonstrate that our method faithfully follows the input LiDAR scan geometry (left) to generate plausible reconstructions (middle). Besides road layout, note for example that vehicles are accurately positioned in the generations w.r.t. the input scan.

5.4. Training-free editing capabilities.

We present qualitative results of the training-free editing capabilities introduced in Sec. 4.3.

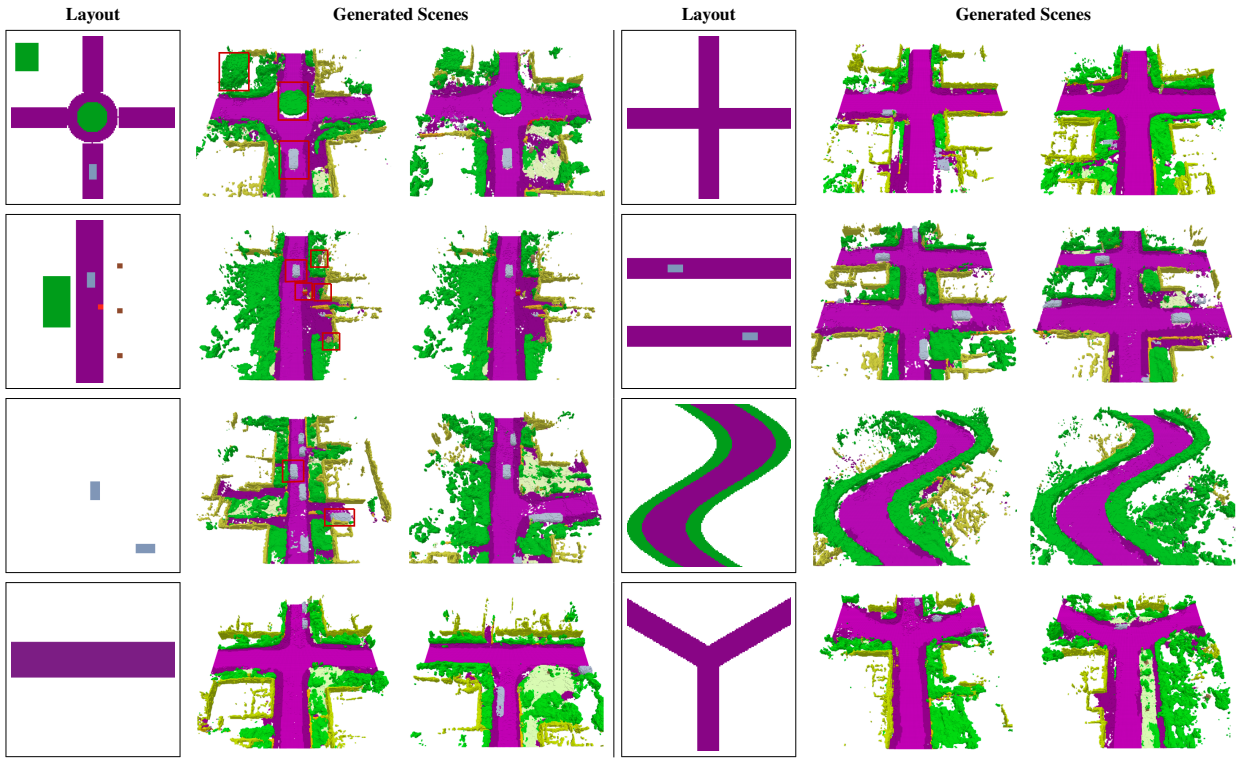

Figure 5. **Training-free sketch-guided generation.** Given a user-drawn BEV layout, our method generates diverse and coherent 3D semantic scenes without any retraining. Each layout is shown with two independently generated scenes, demonstrating both fidelity to the user-specified structure and diversity in the completions.

Sketch-guided generation. We illustrate the sketch-guided generation capability in Fig. 5, using simple hand-drawn BEV layouts depicting various road configurations. For all tested scenarios, our model generates diverse and plausible 3D semantic scenes. The red boxes highlight the layout regions, confirming that the model faithfully respects the user-specified elements. Notably, for a given layout, each generation produces a different yet coherent completion, demonstrating the diversity of the generation process.

Inpainting. As shown in Fig. 6 (top), given a scene with a masked region, the generated content blends seamlessly with the preserved regions, producing realistic transitions without visible artifacts.

Outpainting. Fig. 6 (bottom) shows outpainting results, where the model produces spatially coherent generation that naturally extends the road layout, vegetation, and surrounding structures of the input scene.

Beyond sketch-guided generation, inpainting and outpainting further demonstrate the flexibility offered by the discrete latent space of our pipeline, enabling diverse scene editing scenarios *without any retraining*.

5.5. Ablation Study

VQ-VAE configuration. In Tab. 4, we ablate the VQ-VAE configuration by varying the codebook size (512, 1024, 2048) and the latent channel dimension (4, 8, 16). We observe that larger codebooks and higher dimensions generally improve reconstruction, with the 2048 codes / dim 8 configuration achieving the best scores; although we highlight that higher reconstruction does not always lead to higher diffusion capability, as discussed in Sec. 3. Besides, we find that codebook utilization significantly drops with larger codebooks, from 100% with 512 codes to 43.7% with 2048 codes. A low utilization implies that a large portion of the codebook entries are never used, resulting in wasted capacity. Therefore, for efficiency reasons, we choose to use 512 codes with dim 8 in our final pipeline, as this configuration achieves competitive diffusion performance *while ensuring full codebook utilization*.

Pre vs post-quantization diffusion. Our training-free capability directly emerges from our choice of diffusion on discrete latents, which enables code-to-class mapping (*cf*. Sec. 4.3). We assess the adequacy of such a choice by comparing our discrete latent diffusion (*i.e*., after quantization) with diffusion on continuous latents (before quantization).

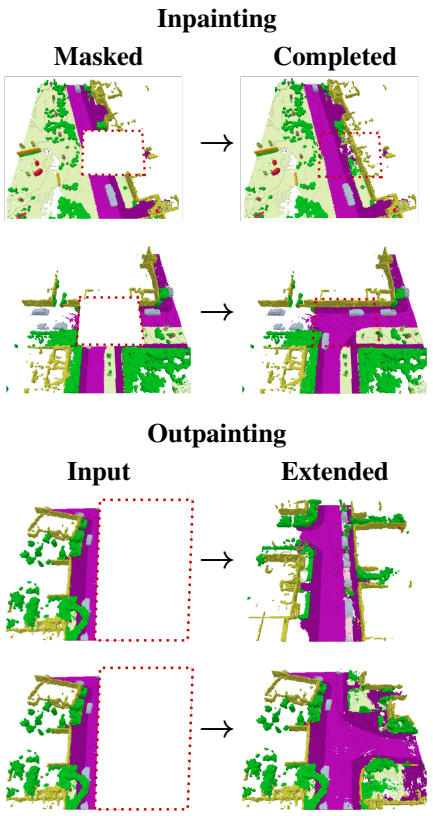

Figure 6. **Scene inpainting and outpainting.** Inpainting (top): given a scene with a masked region, our model generates coherent content to fill the missing area while preserving the known regions. Outpainting (bottom): given half of an existing scene, our model extends it by generating the missing half, producing a spatially coherent continuation.

			Autoencoder		Diffusion		
Codes	Dim.	Codebook utilization ↑	IoU↑	mIoU↑	FID↓	KID↓	CKL↓
512	4		84.00	71.08	-	-	-
512	8	**100%**	81.90	72.20	**84.9**	0.0818	0.0362
512	16		83.90	72.80	-	-	-
1024	8	59.1%	84.10	72.50	85.70	**0.0812**	**0.0326**
2048	8	43.7%	**85.09**	**74.04**	93.77	0.0914	0.0361

Table 4. **VQ-VAE ablation.** Reconstruction performance on the SemanticKITTI validation set for varying codebook sizes and latent dimensions. We select 512 codes with dimension 8, which produces reasonably good performance while ensuring full codebook utilization.

Results in Tab. 5 show that while the continuous variant achieves lower FID and KID, suggesting marginally better per-sample quality, discrete latent diffusion obtains better CKL and Recall, indicating a more faithful class distribution and greater diversity. Overall, we consider both variants to produce on-par performance, which validates our choice of discrete latent diffusion given its unique property.

Diffusion	Editable	FID↓	KID↓	CKL↓	Prec↑	Rec↑
Pre-quant. latent	✗	**81.60**	**0.0777**	0.0435	**0.094**	0.1276
Post-quant. latent	✓	84.90	0.0818	**0.0362**	0.086	**0.1374**

Table 5. **Pre- vs post-quantization diffusion.** While both variants achieve comparable overall performance, only post-quantization latent diffusion enables training-free, editable scene generation.

6. Conclusion and perspective.

In this paper, we present EditSSC, a diffusion-based pipeline for 3D semantic occupancy scene generation. By leveraging a latent Bird's Eye View (BEV) representation, our method enables intuitive latent editing within an image-like spatial domain. While existing 3D reference methods often suffer performance degradation when adapted to a BEV setup, our approach outperforms them in unconditional generation and remains competitive when conditioned on LiDAR frames.

The integration of a Vector Quantized (VQ) representation allows for direct manipulation of the latent space. By prescribing specific class latents within designated areas, we can generate conditioned scenes out of the box, without requiring any network retraining.

EditSSC demonstrates promising results, opening several avenues for future work: improving conditional performance while preserving the simplicity of the design; expanding training datasets to more effectively utilize the capacity of the VQ codebook or integrating pretrained diffusion models. The BEV representation facilitates the use of standard and pretrained 2D diffusion models. This presents an appealing direction for leveraging large-scale pretraining and incorporating multi-modal conditioning, such as text to 3D scene generation.

Acknowledgments This work was performed using HPC resources from GENCI–IDRIS (Grant AD011017284, AD011012883R4).

References

[1] Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Xingguang Yan, Gordon Wetzstein, Leonidas Guibas, and Andrea Tagliasacchi. Cc3d: Layout-conditioned generation of compositional 3d scenes. In *ICCV*, 2023. 2

[2] 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 *ICCV*, 2019. 2, 5

[3] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018. 4

[4] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *ICLR*, 2018. 5

[5] Helin Cao and Sven Behnke. Diffssc: Semantic lidar scan completion using denoising diffusion probabilistic models. In *IROS*, 2025. 2, 6

[6] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 1

[7] Martin Garbade, Yueh-Tung Chen, Johann Sawatzky, and Juergen Gall. Two stream 3d semantic scene completion. In *CVPR-W*, 2019. 6

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 5

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33, 2020. 2, 4, 6

[11] Xiaoliang Ju, Zhaoyang Huang, Yijin Li, Guofeng Zhang, Yu Qiao, and Hongsheng Li. Diffindscene: Diffusion-based high-quality 3d indoor scene generation. In *CVPR*, 2024. 2

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

[13] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *CVPR*, 2023. 2

[14] Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling. In *ICML*, 2025. 2

[15] 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, 3

[16] Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Juhyeong Seon, and Sung-Eui Yoon. Semcity: Semantic scene generation with triplane diffusion. In *CVPR*, 2024. 1, 2, 3, 5

[17] Junho Lee, Jeongwoo Shin, Hyungwook Choi, and Joonseok Lee. Latent diffusion models with masked autoencoders. In *ICCV*, 2025. 2, 3

[18] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *CVPR*, 2023. 2

[19] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *CVPR*, 2025. 2

[20] Wei Liu, Ziwen Kang, Yongtao Yu, Zheng Gong, Yuchao Zheng, Xiaohui Huang, Haiyan Guan, Lingfei Ma, and Dedong Zhang. A dual-path network for semantic scene completion of single-frame lidar point clouds. *Int. J. Appl. Earth Obs.*, 2026. 6

[21] Yuheng Liu, Xinke Li, Xueting Li, Lu Qi, Chongshou Li, and Ming-Hsuan Yang. Pyramid diffusion for fine 3d large scene generation. In *ECCV*. Springer, 2024. 2

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

[23] Ivan Lopes, Valentin Deschaintre, Yannick Hold-Geoffroy, and Raoul de Charette. Matswap: Light-aware material transfers in images. In *Computer Graphics Forum*. Wiley Online Library, 2025. 2

[24] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 2, 5

[25] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021. 2

[26] Nissim Maruani, Wang Yifan, Matthew Fisher, Pierre Alliez, and Mathieu Desbrun. Shapeshifter: 3d variations using multiscale and sparse point-voxel diffusion. In *CVPR*, 2025. 2

[27] Quan Meng, Lei Li, Matthias Nießner, and Angela Dai. Lt3sd: Latent trees for 3d scene diffusion. In *CVPR*, 2025. 2

[28] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2

[29] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *CVPR*, 2024. 2

[30] Luis Roldao, Raoul De Charette, and Anne Verroust-Blondet. LMSCNet: Lightweight multiscale 3D semantic completion. In *3DV*, 2020. 6

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 3, 4

[32] 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*, 2022. 2

[33] 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. *NeurIPS*, 35, 2022. 2

[34] Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. Diffusion-based signed distance fields for 3d shape generation. In *CVPR*, 2023. 2

[35] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *CVPR*, 2023. 1, 2

[36] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *CVPR*, 2024. 1

[37] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *NeurIPS*, 35, 2022. 1, 2

[38] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 30, 2017. 4

[39] Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, et al. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. *ACM Transactions on Graphics (ToG)*, 43(4), 2024. 1, 2

[40] Zichen Xi, Hao-Xiang Chen, Nan Xue, Hongyu Yan, Qi-Yuan Feng, Levent Burak Kara, Joaquim Jorge, and Qun-Ce Xu. Flowssc: Universal generative monocular semantic scene completion via one-step latent diffusion. *Robotics and Automation Letters*, 2026. 2, 3

[41] 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 *AAAI*, 2021. 6

[42] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2024. 2

[43] Haowen Zheng, Jiahao Pang, Zhiqiang Pu, and Yanyan Liang. Sseditor: Controllable mask-to-scene generation with diffusion model. *Knowledge-Based Systems*, 2026. 2, 6

[44] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 2021. 2