

# GridFlow: Structured Latent Flow for Seamless City-Scale 3D Point Cloud Generation

Xinyu Wang<sup>✉</sup>, Muhammad Ibrahim<sup>✉</sup>, Atif Mansoor<sup>✉</sup>, and Ajmal Mian<sup>✉</sup>

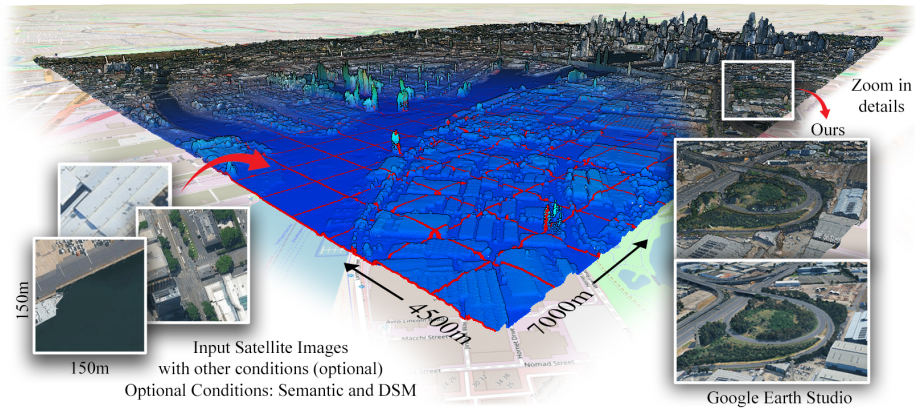
The University of Western Australia, Perth, WA 6009, Australia  
ajmal.mian@uwa.edu.au

**Abstract.** Generating realistic 3D city environments from remote sensing data is important for simulation, urban planning, and mixed reality, yet existing point cloud generation methods are limited to single objects or bounded indoor scenes and cannot handle the scale, seamless tiling, and partial observability challenges of city-scale generation. We present GridFlow, a multi-stage framework that generates dense, colored point clouds ( $10^5$  points per  $150\text{m} \times 150\text{m}$  tile) at city scale, conditioned on satellite imagery, semantic segmentation maps, and digital surface models (DSM). A *Grid-Aligned VAE* encodes each tile into a topology-preserving latent grid where tokens correspond to fixed spatial regions, enabling spatially coherent multi-modal conditioning and compact latent-space edge consistency that implicitly aligns thousands of boundary points for seamless cross-tile generation. A conditional rectified flow model synthesizes geometry latents from the fused multi-modal conditions, and an orientation-aware diffusion colorizer separately handles satellite-visible horizontal surfaces and occluded vertical façades. To support standardized evaluation, we build on public 3D data sources to introduce *City3D-MultiGen*, a benchmark of 163K densely annotated tiles from Melbourne and London with aligned point clouds, satellite images, semantic maps, and elevation data. Experiments show that GridFlow outperforms adapted point cloud generation baselines across all geometry metrics and produces visually coherent colored point clouds with seamless boundaries over arbitrarily large urban extents. Our benchmark details are available at <https://huggingface.co/datasets/e32/City3D-MultiGen>.

**Keywords:** Point Cloud Generation · City-Scale Scene · Flow Matching · Diffusion

## 1 Introduction

Realistic 3D city environments are central to autonomous driving simulation, urban planning, and virtual content creation. While photogrammetric reconstruction yields accurate geometry from dense multi-view images, its dependence on extensive aerial capture limits scalability and prevents content creation for hypothetical or data-scarce regions. Generative models offer a compelling alternative: synthesizing plausible 3D city geometry directly from widely available remote sensing data such as satellite images (see Fig. 1).



**Fig. 1:** Overview of GridFlow visualized in Potree [37]. Given a satellite image and optional semantic/DSM conditions, our three-stage pipeline generates dense, colored city-scale point clouds with seamless cross-tile boundaries.

Generating 3D environments has seen growing interest across multiple representation paradigms. Point cloud generation methods, including normalizing flows [47], denoising diffusion [28, 53], score-based models [3], latent diffusion [50], diffusion transformers [30], and text-conditioned systems [32], have demonstrated impressive fidelity on object benchmarks such as ShapeNet [4, 8] and have been extended to indoor scene synthesis [14, 40, 41, 51]. However, these approaches typically operate on bounded scenes and do not address city-scale tiling, multi-modal remote sensing conditioning, or seamless boundary consistency.

Scaling to outdoor scenes, recent works have explored semantic voxel diffusion for driving scenarios [20, 26], sparse voxel hierarchies for 100m-scale generation [36], LiDAR point cloud diffusion for scene completion [33, 35], and semantic-aware range-view generation [55]. In the city-generation domain, neural-field-based approaches [6, 44, 45] and Gaussian splatting methods [19, 46] produce visually compelling renderings but lack explicit geometry. Despite this progress, generating dense, colored point clouds at city scale from remote sensing data remains open, with three core challenges: (i) *Scale*—a single  $150\text{m} \times 150\text{m}$  tile contains  $\sim 10^5$  points, making direct generation intractable; voxel methods [20, 36] sacrifice detail, and range-view approaches [35, 55] couple density to sensor resolution. (ii) *Seamless tiling*—independently generated tiles must join without seams, yet matching millions of boundary points is prohibitively expensive; existing methods either lack tiling [20, 26] or resort to post-hoc blending [23]. (iii) *Partial observability*—satellite sensors see horizontal surfaces but not vertical façades, requiring complex and separate colorization strategies.

We address these challenges with GridFlow, a multi-stage framework built around a structured latent representation (Fig. 3). A *Grid-Aligned VAE* (GA-VAE) maps each city tile to a structured latent grid  $\mathbf{z} \in \mathbb{R}^{64 \times 25 \times 25}$  where each token corresponds to a fixed  $6\text{m} \times 6\text{m}$  physical region. This spatial grounding



serves two purposes: it aligns the latent space with the condition modalities for spatially coherent conditioning, and reduces generation from a  $10^5$ -point output space to a compact 625-token latent space. Since the latent grid is topology-preserving, a simple MSE loss on the boundary columns of adjacent tile latents implicitly aligns thousands of decoded boundary points, achieving seamless city-scale tiling at negligible additional cost. A conditional rectified flow model then synthesizes latent codes from fused satellite, semantic, and elevation features injected at spatially matching grid positions, while an orientation-aware diffusion colorizer handles the observability asymmetry by applying satellite-guided attention to horizontal surfaces and synthesizing vertical textures from a learned façade pattern bank. To support standardized evaluation, we construct *City3D-MultiGen*, a benchmark of 163K densely annotated tiles built on top of two publicly available 3D data sources, the City of Melbourne 3D Point Cloud [7] and HoliCity [54], covering Melbourne (Australia) and London (UK) respectively, with aligned multi-modal data. Our contributions include:

1. A Grid-Aligned VAE that maps city scenes to a spatially grounded 2D latent grid with explicit token-to-region correspondence, enabling cross-modal conditioning and seamless boundary-aware generation.
2. A latent-space edge consistency mechanism that constrains adjacent tile latents rather than decoded point clouds, reducing the matching problem by two orders of magnitude in dimensionality and enabling seamless infinite-extent city-scale generation.
3. An orientation-aware diffusion colorization model that addresses the partial observability of satellite imagery through dual-path conditioning, separately handling directly visible horizontal surfaces and occluded vertical façades.
4. A large-scale benchmark of 163K tiles from two cities with aligned point clouds, satellite images, semantic segmentation, and DSM, together with a fully automated processing pipeline for reproducible evaluation of conditional city-scale 3D generation.

## 2 Related Work

**Object-Level Point Cloud Generation.** Early deep generative models for point clouds operate on single objects from benchmarks like ShapeNet [4]. Achliopotas *et al.* [1] introduced auto-encoder-based latent-space GANs, followed by PointFlow [47] which models continuous normalizing flows, DPM [28] which applies denoising diffusion directly in point space, and LION [50] which performs latent diffusion with a hierarchical VAE. These methods produce high-quality shapes of 2–4K points but assume centered, single-object inputs and offer no mechanism for spatial conditioning, cross-tile consistency, or scaling beyond  $\sim 10^4$  points.

**Indoor Scene Generation.** Beyond single objects, several methods tackle room-scale 3D scene synthesis. Text2Room [14] generates textured room meshes

from text prompts by iteratively inpainting and fusing depth maps. CommonScenes [51] produces layout-conditioned 3D indoor scenes guided by scene graphs, while DiffuScene [40] formulates indoor layout generation as a denoising diffusion process over object arrangements. Wu *et al.* [43] introduce sketch-and-text guided diffusion for colored point cloud generation, later extended to leverage external knowledge for richer 3D scene synthesis from sketches [42]. While these methods advance scene-level generation, they target bounded indoor environments and do not address the tiling, scale, or remote sensing conditioning requirements of city-scale generation.

**Outdoor Scene-Level Generation.** Recent work extends diffusion models to outdoor scenes at larger scales. XCube [36] generates sparse voxel hierarchies up to  $1024^3$  resolution for  $100\text{m} \times 100\text{m}$  outdoor scenes, while SemCity [20] and PDD [26] produce semantic occupancy grids via triplane and coarse-to-fine discrete diffusion, respectively. LiDiff [33] and LiDM [35] further apply point- and range-image-level diffusion for scene completion and generation, Spiral [55] jointly produces depth, reflectance, and semantic labels via a unified range-view diffusion model, Skip Mamba Diffusion [21] combines state-space models with diffusion for monocular 3D semantic scene completion, UrbanDiff [52] and DynamicCity [2] generate semantic occupancy from driving data, and LT3SD [29] enables patch-by-patch infinite 3D scene generation via latent tree diffusion. These methods operate primarily on driving-centric data (SemanticKITTI, nuScenes) and are either unconditional or conditioned on LiDAR scans rather than satellite imagery.

**City-Scale 3D Generation.** SceneDreamer [6], CityDreamer [44], and CityDreamer4D [45] produce unbounded city scenes via neural hash-grid fields but store geometry only implicitly. InfiniCity [23] tiles octree voxels with neural rendering. GaussianCity [46], Skyfall-GS [19], and Horizon-GS [16] adopt Gaussian splatting for city scenes but produce Gaussian primitives rather than geometric point clouds. ImphiCity [39] learns implicit occupancy fields from satellite images for city-scale surface reconstruction, but targets reconstruction rather than generation. Closer to our setting, Sat2City [15], MagicCity [48], and Sat2RealCity [17] generate 3D city content from satellite imagery via diffusion or multi-view synthesis, and CymbaDiff [22] generates semantic city scenes from sketches.

A parallel line of work targets unbounded scene generation by tiling object- or chunk-level generators. NuiScene [18] trains an explicit outpainting model, while SynCity [9], TRELLISWorld [5], and Extend3D [49] compose overlapping tiles via 2D inpainting, cosine-weighted latent blending, or coupled denoising, respectively. These methods, along with the block-wise extrapolation of BlockFusion [41], achieve cross-tile coherence at inference time through overlapping regions, blending, or joint denoising, which couples generation cost to the overlap and can leave residual seams that require post-hoc registration. In contrast, our latent-space edge consistency is a learned constraint imposed during training,

allowing tiles to be generated fully independently at inference without overlap or blending.

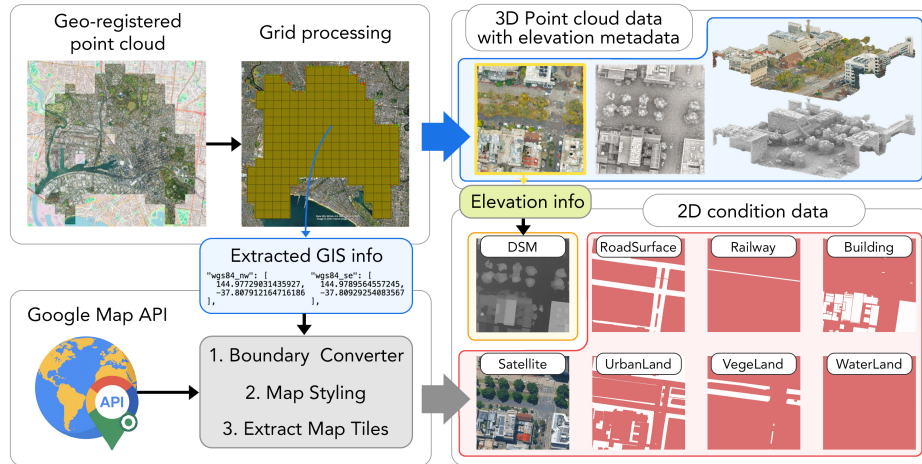
Our work differs in three respects: (i) we condition on widely available satellite images and semantic maps rather than requiring 3D sensor inputs, (ii) we generate explicit colored point clouds instead of voxel occupancy, and (iii) we enforce latent-space edge consistency for seamless tiling, a requirement absent in prior city-scale or driving-scenario methods.

### 3 City3D-MultiGen Benchmark

Existing city-scale 3D generation methods typically evaluate on small-scale or synthetic scenes, lacking large-scale benchmarking with aligned multi-modal conditions. To address this gap, we construct *City3D-MultiGen*, a benchmark comprising 163K densely annotated tiles from two cities on different continents, each with accurately aligned point cloud geometry, satellite images, semantic segmentation, and elevation data. We release the full processing pipeline code, tile metadata, and ready-to-run data retrieval scripts that allow users to reproduce the complete dataset at <https://huggingface.co/datasets/e32/City3D-MultiGen>. The point clouds are obtained from their original public providers, and the satellite imagery and semantic maps through these scripts, rather than redistributed directly, as the source-data licenses and the map provider’s terms of service do not permit redistribution. An overview of the data processing pipeline is shown in Fig. 2.

**Source Data.** We draw on two complementary 3D sources: (i) The City of Melbourne 3D Point Cloud [7] covers  $\sim 37.7 \text{ km}^2$  of metropolitan Melbourne, Australia, captured via aerial photogrammetry at 7.5 cm ground sample distance with 25 cm spatial accuracy. The dense, colorized point cloud is distributed as georeferenced LAS tiles with RGB information. (ii) HoliCity [54] provides high-fidelity publicly accessible FBX models covering  $2 \text{ km}^2$  of central London, United Kingdom. Point clouds were extracted by uniformly sampling the mesh surfaces of the FBX model files. The two city datasets differ substantially: Melbourne is largely suburban with low-rise buildings, while London is dense and structurally complex, enabling cross-domain evaluation.

**Automated Multi-Modal Alignment Pipeline.** Both datasets are processed through an identical, fully automated pipeline with no manual annotation. The geographic extent is partitioned into  $150\text{m} \times 150\text{m}$  tiles on a regular UTM grid with 20 m center spacing. Each tile is paired with three spatially aligned condition modalities at  $256 \times 256$  resolution: (i) **satellite imagery** (RGB) retrieved via the Google Maps Static API, (ii) **semantic segmentation** (6 binary class masks: *Building*, *RoadSurface*, *Railway*, *VegetationLand*, *UrbanLand*, *WaterSurface*) obtained by parsing custom-styled Google Maps renders, and (iii) a **DSM** rasterized from point cloud elevations and normalized to  $[0, 1]$ . Point coordinates



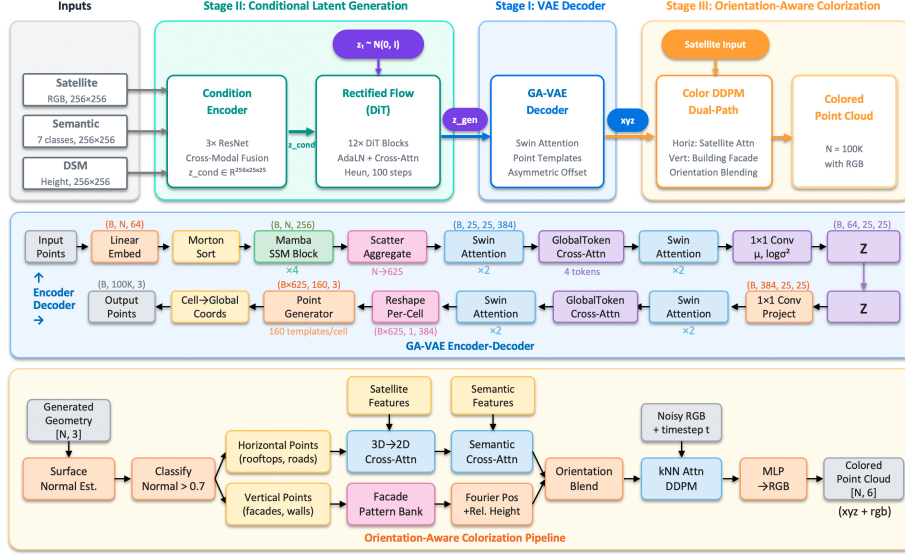
**Fig. 2:** Overview of the City3D-MultiGen data processing pipeline. Geo-registered point clouds are partitioned into  $150\text{m} \times 150\text{m}$  tiles on a regular grid. Geographic coordinates extracted from each tile are used to query the Google Maps API, which retrieves spatially aligned satellite imagery, styled semantic segmentation maps (via custom map styling), and DSM rasterized from the point cloud elevation. The resulting multi-modal tile set provides paired 3D geometry and multi-modal condition data for training and evaluation.

are centered per tile and normalized to  $[-1, 1]^3$ , with the vertical axis scaled independently because building heights (typically 10–50 m) are much smaller than the 150m horizontal tile extent; uniform scaling would compress height variations into a narrow range, losing fine vertical detail. Each raw tile contains approximately 150,000 points. The dataset is split 80/10/10 into training, validation, and test sets based on contiguous spatial regions along the grid coordinate axes, with a minimum separation of  $\geq 150\text{m}$  between splits, ensuring no geographic overlap between training and test data.

## 4 Method

The primary input to our method is a single satellite image of a  $150\text{m} \times 150\text{m}$  city tile at  $256 \times 256$  resolution. Semantic segmentation maps (six land-cover classes), which can be extracted automatically from the satellite image, and DSM, when available, serve as auxiliary conditions that further refine spatial layout and elevation. Structured condition dropout during training (Sec. 4.2) ensures that the model remains effective under any subset of available modalities. The output of our method is a colored point cloud of  $N=100,000$  points. To the best of our knowledge, this is the first method capable of generating a seamless city-scale point cloud of this magnitude.

Our method comprises three stages as shown in Fig. 3: (1) a *Grid-Aligned Variational Autoencoder* (GA-VAE) that learns a topology-preserving latent rep-



**Fig. 3:** Overview of the proposed GridFlow framework. **Stage I:** A Grid-Aligned VAE compresses 100K-point city clouds into a topology-preserving  $64 \times 25 \times 25$  latent grid. **Stage II:** A conditional latent generator synthesizes geometry latents with boundary consistency constraints from fused satellite/semantic/DSM features. **Stage III:** An orientation-aware colorization model produces realistic appearance via dual-path surface conditioning.

resentation of city-scale point clouds (Sec. 4.1), (2) a *conditional latent generator* that synthesizes geometry in the learned latent space from the fused input modalities (Sec. 4.2), and (3) an *orientation-aware colorization* model that produces realistic appearance for the generated geometry (Sec. 4.3). Our design avoids the training instability of joint geometry-color optimization and allows each stage to employ the most suitable generative framework.

#### 4.1 Grid-Aligned Variational Autoencoder

City environments exhibit natural spatial regularity. Buildings, roads, and vegetation form locally coherent clusters throughout the horizontal plane. The GA-VAE uses this regularity by dividing the horizontal plane into a  $25 \times 25$  grid of  $6m \times 6m$  cells. Each cell is encoded separately, and the model then combines information across cells in a hierarchical way. The latent  $\mathbf{z} \in \mathbb{R}^{64 \times 25 \times 25}$  keeps a one-to-one link between tokens and physical regions to facilitate spatially aligned conditioning (Sec. 4.2) and latent-space edge consistency (Sec. 4.2).

**Encoding.** Each cell contains  $\sim 160$  points on average. Points are sorted along a Morton curve [31] and encoded by stacked selective state-space blocks (SSM) [11] in  $\mathcal{O}(N)$  time into a fixed-dimensional cell descriptor. The 625 cell features are



then arranged on the  $25 \times 25$  grid and refined by Swin [27] windowed self-attention layers for local communication, augmented with a bottleneck cross-attention module at the mid-layer: a small set of learnable global tokens (4 tokens) aggregates scene-level information from all cells and broadcasts it back, enabling global context without full attention. The same module is applied symmetrically in the decoder.

**Decoding.** A symmetric windowed-attention decoder maps the latent  $\mathbf{z}$  back to per-cell features. For each cell, *learnable point templates* (160 points initialized from  $\mathcal{N}(0, 0.09)$ ) are refined via cross-attention to the decoded cell feature, followed by self-attention among the template points. Separate MLP heads predict horizontal and vertical offsets with *asymmetric scaling*: horizontal offsets are scaled by 0.5 (matching the 6m cell extent), while vertical offsets are scaled by 4.0 (accommodating the much larger range of building heights), with a learned per-cell height bias anchoring the generated points to the appropriate elevation. The final coordinates are obtained by adding offsets to cell centers, yielding  $25^2 \times 160 = 100,000$  points.

**Training Objective.** The training objective is defined as

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{CD}} + \lambda_h \mathcal{L}_{\text{height}} + \lambda_b \mathcal{L}_{\text{bnd}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}, \quad (1)$$

where  $\mathcal{L}_{\text{CD}}$  is the bidirectional Chamfer distance,  $\mathcal{L}_{\text{height}}$  is a KL divergence between soft height histograms that regularizes vertical structure,  $\mathcal{L}_{\text{bnd}}$  is Chamfer distance on tile-boundary points to preserve reconstruction fidelity near tile edges, and  $\mathcal{L}_{\text{KL}}$  is the standard VAE regularization with a very low weight ( $\lambda_{\text{KL}}=10^{-5}$ ) to prioritize reconstruction.

## 4.2 Conditional Latent Generation

**Multi-Modal Condition Fusion.** Three ResNet-style encoders extract features from satellite images, semantic segmentation maps, and DSM respectively, with capacity proportional to each modality’s information density. The features are projected to a common dimensionality and fused via bidirectional cross-modal attention, where each modality queries the other two. The fused representation  $\mathbf{z}_{\text{cond}} \in \mathbb{R}^{256 \times 25 \times 25}$  is spatially aligned with the VAE latent grid. Structured condition dropout randomly zeros subsets of modalities during training, so the model learns to generate both conditionally and unconditionally; at inference, classifier-free guidance [13] extrapolates beyond the conditional prediction by contrasting it with the unconditional one, sharpening output fidelity. This also ensures robustness when some modalities are unavailable.

**Rectified Flow.** Geometry latents are generated via rectified flow matching [24, 25], which learns a velocity field transporting Gaussian noise to data along straight paths. Before flow training, VAE latents are normalized to zero mean

and unit variance per channel, stabilizing the velocity targets. Given noise  $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and a normalized VAE latent  $\mathbf{z}_0$ , the interpolant is  $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\mathbf{z}_1$  with target velocity  $\mathbf{v} = \mathbf{z}_1 - \mathbf{z}_0$ . A Transformer-based velocity network [34] operating on the flattened  $25 \times 25$  latent tokens is trained with:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{\mathbf{z}_0, \mathbf{z}_1, t} [\|\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{z}_{\text{cond}}) - (\mathbf{z}_1 - \mathbf{z}_0)\|^2]. \quad (2)$$

Conditions enter via two paths: a *global path* that modulates each transformer layer through AdaLN scale/shift parameters derived from the pooled condition and timestep, and a *local path* where  $25 \times 25$  condition tokens serve as cross-attention keys/values every two blocks, gated by a learned sigmoid. At inference, sampling uses 50-step Heun integration with classifier-free guidance [13].

**Latent-Space Edge Consistency.** City-scale generation requires seamless tiling across independently processed 150m tiles. Rather than matching  $10^5$  boundary points in the output space, consistency is enforced directly in the compact latent space. For adjacent tiles  $T_a, T_b$ , the boundary columns (or rows) of their latents are constrained to agree:

$$\mathcal{L}_{\text{edge}} = \frac{1}{2CW_e} \sum_{c,h} \sum_{w=1}^{W_e} (\mathbf{z}_{c,h,W-w}^{(a)} - \mathbf{z}_{c,h,w-1}^{(b)})^2, \quad (3)$$

where  $W_e=2$  is the boundary margin width ( $\sim 12\text{m}$ ), with an analogous formulation for vertical neighbors. During training, each sample draws an adjacent tile with probability 0.5. Because the latent grid is topology-preserving, matching two columns of 64-dimensional tokens implicitly aligns the geometry of  $\sim 6,400$  boundary points, making this constraint both efficient and effective.

### 4.3 Orientation-Aware Colorization

The observability asymmetry between horizontal and vertical surfaces demands fundamentally different colorization strategies. Rooftops and roads are directly visible in satellite images and can be colored via learned 3D-to-2D attention; building façades, however, are invisible to top-down sensors and must instead be synthesized from learned texture priors. Prior point cloud colorization methods [10, 38] assume full visibility; our satellite setting requires handling partial observability, motivating a dual-path DDPM [12] architecture.

**Dual-Path Conditioning.** A point is labeled horizontal when the vertical component of its normal is greater than 0.7, and vertical otherwise. The *horizontal path* uses cross-attention between 3D point coordinates and satellite/semantic/DSM features, learning an implicit 3D-to-2D correspondence that accommodates varying viewing angles without requiring orthorectification. The *vertical path* employs a learned bank of canonical façade texture patterns, selected and modulated by multi-scale Fourier positional features, relative height within each building, and

semantic class embeddings. An orientation-aware blending network produces per-point weights that smoothly combine both paths, mixing learned and geometric priors.

**Diffusion and Training.** The fused conditioning features and noisy RGB tokens are processed by local  $k$ -NN attention blocks with relative positional encoding. Training proceeds in three phases that progressively introduce loss objectives: basic noise prediction and  $L_1$  reconstruction (with  $2\times$  weight on vertical surfaces), then  $k$ -NN smoothness regularization for local color coherence, and finally a vertical texture gradient loss that enforces floor-level repetition patterns characteristic of building façades. This curriculum prevents the model from collapsing to averaged appearances before learning fine-grained structures.

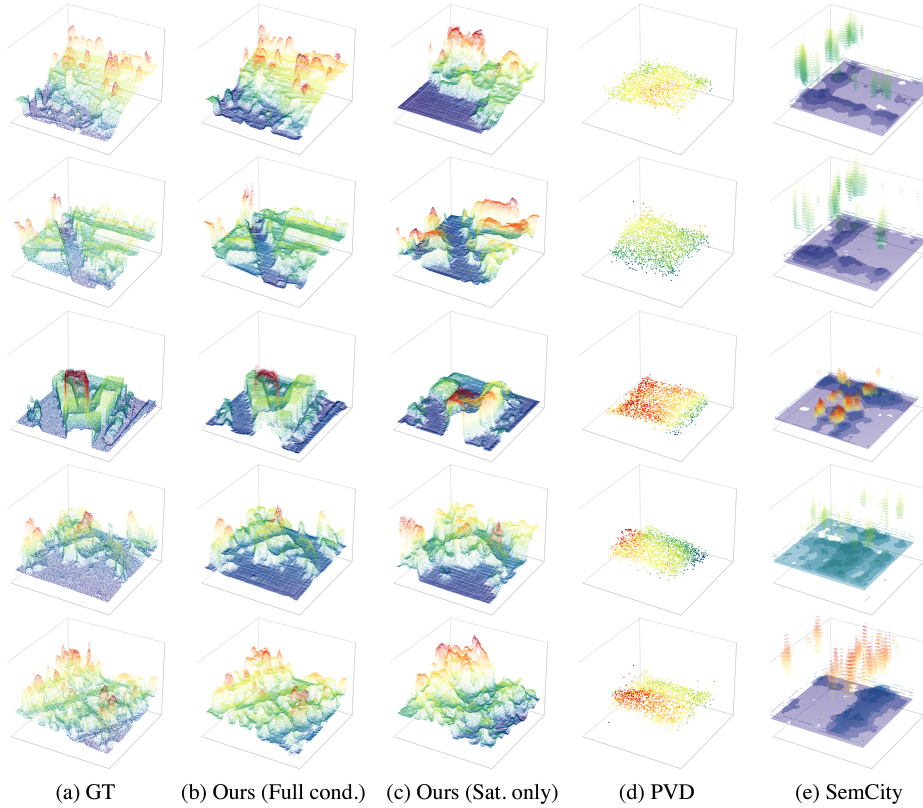
#### 4.4 Implementation Details

The three stages are trained sequentially on the City of Melbourne 3D Point Cloud [7], each freezing all upstream models: the VAE for 150 epochs, the flow model for 200 epochs, and the colorization model for 50 epochs. AdamW with cosine learning rate decay is used throughout. The color model trains on VAE-reconstructed (rather than ground-truth) geometry with latent noise augmentation to improve robustness to imperfect flow-generated latents at inference. Training the full pipeline takes approximately 55 days on a single NVIDIA RTX 5090 GPU (29/14/12 days for the GA-VAE, flow, and colorization stages, respectively). At inference, GridFlow generates one  $150\text{m}\times 150\text{m}$  tile in  $\sim 4.9\text{s}$  and a  $1\text{km}^2$  area ( $\sim 49$  tiles) in  $\sim 4$  minutes.

## 5 Experiments

**Setup.** We train GridFlow on the Melbourne split of City3D-MultiGen and evaluate on a held-out test set of 13,363 tiles. Following the evaluation protocol of Achlioptas *et al.* [1], we report Chamfer Distance (CD), F-Score ( $\tau=0.05$ ), Jensen–Shannon Divergence (JSD) on a  $28^3$  voxel grid, Coverage (COV-CD), and 1-Nearest-Neighbor Accuracy (1-NNA-CD). All metrics are computed on 500 randomly sampled test tiles. Given the lack of city-scale point cloud generation methods, we adapt four baselines to our setting for comparison. These include: PointFlow [47], DPM [28], PVD [53], and SemCity [20]. For visual rendering evaluation, we additionally compare with CityDreamer [44] and Skyfall-GS [19], using PSNR, SSIM, LPIPS, FID, and histogram correlation against Google Earth reference views.

**Geometry Comparison.** Table 1 and Fig. 4 present the quantitative and qualitative geometry results on the Melbourne test set. GridFlow achieves the best performance across all five metrics, reducing Chamfer Distance by 29% relative to the next-best method (PVD) and reaching an F-Score of 0.884. The



**Fig. 4:** Qualitative geometry comparison on Melbourne test tiles. GridFlow produces detailed building structures and sharp boundaries compared with other methods.

distributional metrics further highlight the gap: our JSD (0.010) is nearly  $4\times$  lower than PVD (0.037), and our COV-CD of 51.8% indicates that our generated tiles cover more than half of the test distribution, compared to 24.4% for PVD. The 1-NNA-CD of 46.5%, close to the ideal 50%, confirms that GridFlow samples are nearly indistinguishable from real tiles in the metric space. PointFlow and DPM fail to capture the structural complexity of city tiles, producing amorphous geometry. SemCity generates recognizable urban layouts via triplane diffusion but introduces coarse voxelization artifacts and lacks fine-grained building detail.

**Visual Comparison.** To evaluate the full pipeline including colorization, we render top-down views of generated point clouds and compare against satellite imagery from the same locations (Table 2, Fig. 5). GridFlow achieves an SSIM of 0.70 and FID of 58.52. That GridFlow slightly exceeds Google Earth on SSIM and FID is expected: our method is conditioned on the evaluation satellite image, whereas Google Earth renders come from independent multi-view capture under different lighting. We therefore include Google Earth as a reference point, not

**Table 1:** Quantitative geometry comparison on the Melbourne test set. Best results are shown in **bold**.

| Method      | CD ↓           | F-Score ↑     | JSD ↓           | COV-CD ↑     | 1-NNA-CD (→50%) |
|-------------|----------------|---------------|-----------------|--------------|-----------------|
| <b>Ours</b> | <b>0.00272</b> | <b>0.8840</b> | <b>0.009752</b> | <b>51.8%</b> | <b>46.5%</b>    |
| PVD         | 0.00382        | 0.7939        | 0.036975        | 24.4%        | 69.8%           |
| SemCity     | 0.00451        | 0.7800        | 0.144380        | 4.1%         | 96.8%           |
| DPM         | 0.01936        | 0.5692        | 0.279454        | 5.2%         | 98.7%           |
| PointFlow   | 0.12071        | 0.2513        | 0.786199        | 0.2%         | 100.0%          |

**Table 2:** Rendered image quality compared against the conditioning satellite image. Results reflect conditioning fidelity; the satellite-conditioned method has a structural advantage over methods using independent inputs.

| Method       | PSNR ↑       | SSIM ↑      | LPIPS ↓     | FID ↓        | Hist-Corr ↑ |
|--------------|--------------|-------------|-------------|--------------|-------------|
| Ours         | 22.82        | <b>0.70</b> | 0.30        | <b>58.52</b> | 0.88        |
| Google Earth | <b>26.11</b> | 0.67        | <b>0.28</b> | 62.83        | <b>0.90</b> |
| CityDreamer  | 13.21        | 0.26        | 0.62        | 155.27       | 0.38        |
| Skyfall-GS   | 16.50        | 0.38        | 0.52        | 115.68       | 0.58        |

a baseline to surpass. CityDreamer produces plausible but location-agnostic imagery (PSNR 13.21, SSIM 0.26), and Skyfall-GS yields blurred outputs (PSNR 16.50, LPIPS 0.52) due to the limited satellite baseline.

**Large-Tile Visualization.** Fig. 6 shows multiple independently generated tiles assembled from the Melbourne test set. The latent-space edge consistency mechanism (Sec. 4.2) enables seamless tiling: buildings, roads, and vegetation continue naturally across tile boundaries without visible seams, demonstrating practical viability for city-scale deployment.

**Cross-Dataset Generalization.** To assess generalization beyond the training city, we evaluate the GridFlow trained on Melbourne data (as is without retraining) on the HoliCity [54] London test split. As shown in Table 3 and Fig. 7, the model achieves a CD of 0.00394 and F-Score of 0.783, degrading only moderately from the Melbourne results despite the substantial domain shift: London features denser, taller buildings with more complex structural layouts compared to Melbourne’s largely suburban environment. The COV-CD of 41.8% and 1-NNA-CD of 63.7% show that, while not fully capturing London’s structural diversity, the model produces plausible outputs that reflect the conditioning inputs, confirming cross-city generalization of the structured latent representation.

**Ablation: Conditioning Modalities.** We ablate the contribution of each input modality and classifier-free guidance (CFG) in Table 4. All variants use the



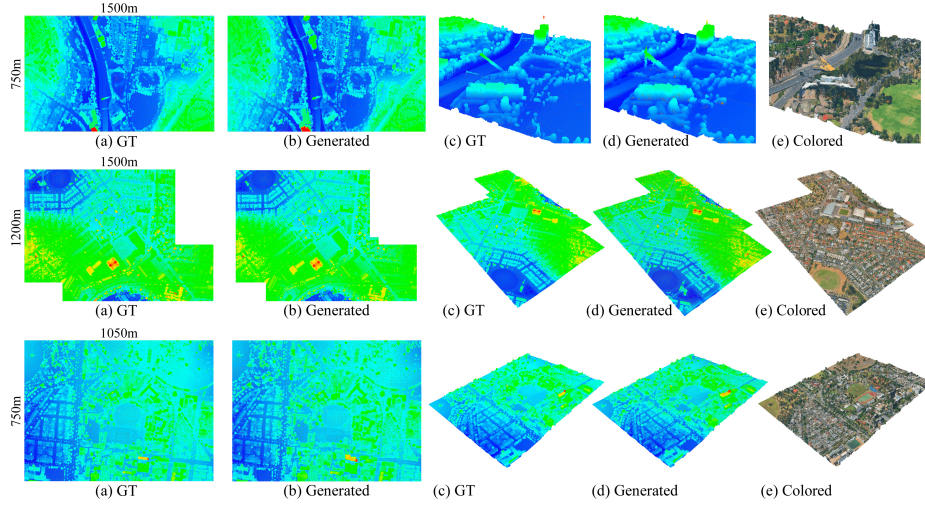
**Fig. 5:** Visual comparison of rendered outputs against Google Earth ground truth. GridFlow produces scene-specific geometry and appearance closest to the reference, while CityDreamer generates city-like but location-agnostic content, and Skyfall-GS produces blurred reconstructions.

same checkpoint; structured condition dropout ( $p=0.1$ ) during training makes inference-time removal methodologically consistent. Every modality contributes: removing any single condition degrades CD by 25–43%, with DSM being especially critical for distributional coverage (COV-CD drops to 23.0% without it). Disabling CFG degrades both fidelity and diversity, confirming its role in sharpening conditional generation. The VAE Recon upper bound shows that the latent space preserves substantial geometric detail, with the remaining gap attributable to the flow-based generation process.

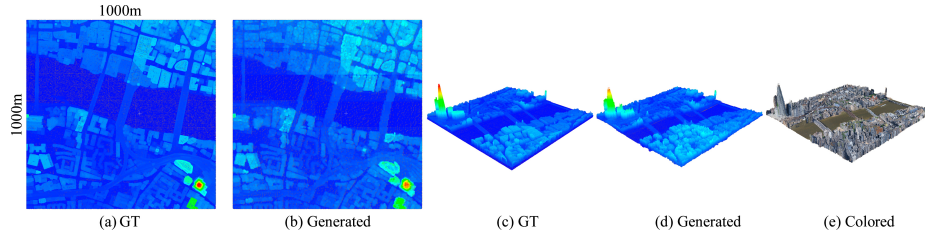
## 6 Conclusion

We presented GridFlow, a multi-stage framework for generating dense, colored city-scale point clouds from satellite imagery and auxiliary remote sensing modalities. The key insight is a Grid-Aligned VAE that compresses  $10^5$ -point city tiles into a topology-preserving latent grid, enabling spatially coherent multi-modal conditioning and an efficient latent-space edge consistency mechanism for seamless cross-tile generation. A conditional rectified flow model synthesizes geometry latents, and an orientation-aware diffusion colorizer handles the observability asymmetry between horizontal and vertical surfaces. We also introduced City3D-MultiGen, a benchmark of 163K tiles from Melbourne and London with





**Fig. 6:** Large-scale visualization of seamlessly tiled Melbourne point clouds generated by GridFlow. Adjacent tiles are independently generated with latent-space edge consistency, producing continuous geometry across tile boundaries.



**Fig. 7:** Qualitative visualization on the HoliCity (London) dataset. The model, trained on Melbourne, generalizes to the structurally distinct London cityscape.

aligned point clouds, satellite images, semantic maps, and elevation data. Experiments demonstrate that GridFlow outperforms existing point cloud generation baselines across all geometry metrics and produces visually coherent colored outputs with seamless boundaries over arbitrarily large urban extents.

**Limitations and Future Work.** The sequential three-stage training may accumulate errors across stages; joint fine-tuning could mitigate this. Façade colorization relies on learned priors since satellite imagery provides no vertical supervision; incorporating street-level images could improve quality. Cross-city generalization degrades moderately (CD from 0.00272 to 0.00394 on London), and the fixed tile size and cell resolution may not optimally serve all urban morphologies. Future directions include temporal generation for urban change simulation, multi-resolution latent grids, and scaling the benchmark to additional cities.



**Table 3:** Evaluating GridFlow with other methods, trained on the Melbourne dataset, on the HoliCity (London) test split without retraining or finetuning.

| Method      | CD ↓           | F-Score ↑     | JSD ↓         | COV-CD ↑     | 1-NNA-CD (→50%) |
|-------------|----------------|---------------|---------------|--------------|-----------------|
| <b>Ours</b> | <b>0.00394</b> | <b>0.7826</b> | <b>0.0149</b> | <b>41.8%</b> | <b>63.7%</b>    |
| PVD         | 0.00542        | 0.7113        | 0.0526        | 23.5%        | 78.9%           |
| SemCity     | 0.00613        | 0.6942        | 0.1681        | 6.8%         | 94.5%           |
| DPM         | 0.02174        | 0.5221        | 0.3017        | 5.4%         | 97.8%           |
| PointFlow   | 0.11261        | 0.2394        | 0.8125        | 0.3%         | 100.0%          |

**Table 4:** Ablation on conditioning modalities. VAE Recon encodes and decodes ground-truth tiles without the flow model, serving as an upper bound. Best generative results in **bold**.

| Variant               | CD ↓           | F-Score↑      | JSD ↓         | COV-CD ↑     | 1-NNA-CD (→50%) |
|-----------------------|----------------|---------------|---------------|--------------|-----------------|
| VAE Recon (upper-bnd) | 0.00131        | 0.9861        | 0.0033        | 85.8%        | 47.2%           |
| <b>Full model</b>     | <b>0.00272</b> | <b>0.8840</b> | <b>0.0098</b> | 51.8%        | <b>46.5%</b>    |
| w/o Semantic          | 0.00350        | 0.8477        | 0.0340        | <b>59.2%</b> | 75.1%           |
| w/o DSM               | 0.00340        | 0.8490        | 0.0206        | 23.0%        | 82.2%           |
| Satellite only        | 0.00389        | 0.8244        | 0.0185        | 23.0%        | 83.9%           |
| w/o CFG               | 0.00362        | 0.8300        | 0.0108        | 41.0%        | 76.0%           |

## Acknowledgements

This research was supported by the Australian Research Council Discovery Project DP240101926.

## References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: ICML. pp. 40–49 (2018)
2. Bian, H., Kong, L., Xie, H., Pan, L., Qiao, Y., Liu, Z.: DynamicCity: Large-scale 4d occupancy generation from dynamic scenes. In: ICLR (2025)
3. Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., Hariharan, B.: Learning gradient fields for shape generation. In: ECCV. pp. 364–381 (2020)
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: ShapeNet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
5. Chen, H., Liu, Y., Li, M.: TRELLISWorld: Training-free world generation from object generators. arXiv preprint arXiv:2510.23880 (2025)
6. Chen, Z., Wang, G., Liu, Z.: SceneDreamer: Unbounded 3d scene generation from 2d image collections. IEEE TPAMI **46**(4), 2563–2576 (2024)
7. City of Melbourne: City of Melbourne 3d point cloud 2018. <https://data.melbourne.vic.gov.au/explore/dataset/city-of-melbourne-3d-pointcloud-2018/> (2018), open Data Portal, accessed 2026-06-30

8. Du, Y., Zhao, Z., Su, S., Golluri, S., Zheng, H., Yao, R., Wang, C.: SuperPC: A single diffusion model for point cloud completion, upsampling, denoising, and colorization. In: CVPR. pp. 16953–16964 (2025)
9. Engstler, P., Shtedritski, A., Laina, I., Rupperecht, C., Vedaldi, A.: SynCity: Training-free generation of 3D worlds. In: ICCV (2025)
10. Gao, R., Xiang, T.Z., Lei, C., Park, J., Chen, Q.: Scene-level point cloud colorization with semantics-and-geometry-aware networks. In: ICRA. pp. 2818–2824 (2023)
11. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
12. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS. vol. 33, pp. 6840–6851 (2020)
13. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
14. Höllein, L., Cao, A., Owens, A., Johnson, J., Nießner, M.: Text2Room: Extracting textured 3D meshes from 2D text-to-image models. In: ICCV. pp. 7909–7920 (2023)
15. Hua, T., Jiang, L., Chen, Y.C., Zhao, W.: Sat2City: 3d city generation from a single satellite image with cascaded latent diffusion. In: ICCV (2025)
16. Jiang, L., Ren, K., Yu, M., Xu, L., Dong, J., Lu, T., Zhao, F., Lin, D., Dai, B.: Horizon-GS: Unified 3d gaussian splatting for large-scale aerial-to-ground scenes. In: CVPR (2025)
17. Kang, Y., Wang, X., Wu, Z., Shi, Y., Zhu, H.: Sat2RealCity: Geometry-aware and appearance-controllable 3d urban generation from satellite imagery. arXiv preprint arXiv:2511.11470 (2025)
18. Lee, H.H., Han, Q., Chang, A.X.: NuiScene: Exploring efficient generation of unbounded outdoor scenes. In: ICCV (2025)
19. Lee, J.Y., Liu, Y.R., Tsai, S.R., Chang, W.C., Wu, C.H., Chan, J., Zhao, Z., Lin, C.H., Liu, Y.L.: Skyfall-GS: Synthesizing immersive 3D urban scenes from satellite imagery. arXiv preprint arXiv:2510.15869 (2025)
20. Lee, J., Lee, S., Jo, C., Im, W., Seon, J., Yoon, S.E.: SemCity: Semantic scene generation with triplane diffusion. In: CVPR. pp. 28337–28347 (2024)
21. Liang, L., Akhtar, N., Vice, J., Kong, X., Mian, A.: Skip mamba diffusion for monocular 3d semantic scene completion. In: AAAI. vol. 39, pp. 5155–5163 (2025)
22. Liang, L., Miao, B., Wang, X., Akhtar, N., Vice, J., Mian, A.: CymbaDiff: Structured spatial diffusion for sketch-based 3d semantic urban scene generation. In: NeurIPS (2025)
23. Lin, C.H., Ma, H.Y., Torralba, A., Lucey, S.: InfiniCity: Infinite-scale city synthesis. In: ICCV. pp. 18405–18415 (2023)
24. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)
25. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: ICLR (2023)
26. Liu, Y., Li, X., Li, X., Qi, L., Li, C., Yang, M.H.: Pyramid diffusion for fine 3d large scene generation. In: ECCV. pp. 73–89 (2024)
27. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)
28. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: CVPR. pp. 2837–2845 (2021)
29. Meng, Q., Li, L., Nießner, M., Dai, A.: LT3SD: Latent trees for 3d scene diffusion. In: CVPR (2025)

30. Mo, S., Xie, E., Chu, R., Yao, L., Hong, L., Nießner, M., Li, Z.: DiT-3D: Exploring plain diffusion transformers for 3D shape generation. In: NeurIPS (2023)
31. Morton, G.M.: A computer oriented geodetic data base and a new technique in file sequencing. Technical Report, IBM Ltd. (1966)
32. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-E: A system for generating 3D point clouds from complex prompts. In: ICML (2023)
33. Nunes, L., Marcuzzi, R., Mersch, B., Behley, J., Stachniss, C.: Scaling diffusion models to real-world 3d LiDAR scene completion. In: CVPR. pp. 14820–14829 (2024)
34. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV. pp. 4195–4205 (2023)
35. Ran, H., Guizilini, V., Wang, Y.: Towards realistic scene generation with LiDAR diffusion models. In: CVPR. pp. 14738–14748 (2024)
36. Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., Williams, F.: XCube: Large-scale 3d generative modeling using sparse voxel hierarchies. In: CVPR. pp. 4209–4219 (2024)
37. Schütz, M.: Potree: Rendering Large Point Clouds in Web Browsers. Master’s thesis, Vienna University of Technology (2016)
38. Shinohara, T., Xiu, H., Matsuoka, M.: Point2Color: 3d point cloud colorization using a conditional generative network and differentiable rendering for airborne LiDAR. In: CVPR Workshops. pp. 1062–1071 (2021)
39. Stücker, C., Richard, A., Wegner, J.D., Schindler, K.: ImpliciCity: City modeling from satellite images with deep implicit occupancy fields. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. vol. V-2-2022, pp. 193–201 (2022)
40. Tang, J., Nie, Y., Markhasin, L., Dai, A., Thies, J., Nießner, M.: DiffuScene: Denoising diffusion models for generative indoor scene synthesis. In: CVPR. pp. 4587–4597 (2024)
41. Wu, Z., Li, Y., Yan, H., Shang, T., Sun, W., Wang, S., Pan, R., Liu, W., Ouyang, S., Loy, C.C., et al.: BlockFusion: Expandable 3d scene generation using latent tri-plane extrapolation. vol. 43 (2024)
42. Wu, Z., Feng, M., Wang, Y., Xie, H., Dong, W., Miao, B., Mian, A.: External knowledge enhanced 3d scene generation from sketch. In: ECCV. pp. 286–304 (2024)
43. Wu, Z., Wang, Y., Feng, M., Xie, H., Mian, A.: Sketch and text guided diffusion model for colored point cloud generation. In: ICCV. pp. 8929–8939 (2023)
44. Xie, H., Chen, Z., Hong, F., Liu, Z.: Citydreamer: Compositional generative model of unbounded 3d cities. In: CVPR. pp. 9666–9675 (2024)
45. Xie, H., Chen, Z., Hong, F., Liu, Z.: CityDreamer4D: Compositional generative model of unbounded 4D cities. arXiv preprint arXiv:2501.08983 (2025)
46. Xie, H., Chen, Z., Hong, F., Liu, Z.: GaussianCity: Generative gaussian splatting for unbounded 3d city generation. In: CVPR (2025)
47. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: ICCV. pp. 4541–4550 (2019)
48. Yao, X., Liu, Z., et al.: MagicCity: Geometry-aware 3d city generation from satellite imagery with multi-view consistency. In: ICCV (2025)
49. Yoon, S., Kim, J., Park, J.: Extend3D: Town-scale 3D generation. arXiv preprint arXiv:2603.29387 (2026)
50. Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K.: LION: Latent point diffusion models for 3d shape generation. In: NeurIPS (2022)

51. Zhang, G., Zhu, A., Nalmpantis, C., Yan, Z.: CommonScenes: Generating commonsense 3D indoor scenes with scene graphs. In: NeurIPS (2023)
52. Zhang, J., Zhang, Q., Zhang, L., Kompella, R.R., Liu, G., Zhou, B.: Urban scene diffusion through semantic occupancy map. arXiv preprint arXiv:2403.11697 (2024)
53. Zhou, L., Du, Y., Wu, J.: 3d shape generation and completion through point-voxel diffusion. In: ICCV. pp. 5826–5835 (2021)
54. Zhou, Y., Huang, J., Dai, X., Luo, L., Chen, Z., Ma, Y.: HoliCity: A city-scale data platform for learning holistic 3d structures. In: CVPR. pp. 4610–4619 (2021)
55. Zhu, D., Hu, Y., Liu, Y., Lu, D., Kong, L., Ilic, S.: SPIRAL: Semantic-aware progressive LiDAR scene generation and understanding. arXiv preprint arXiv:2505.22643 (2025)