SAT-SKYLINES: 3D Building Generation from Satellite Imagery and Coarse Geometric Priors

Zhangyu Jin[1] Andrew Feng[1]

[1] University of Southern California, Institute for Creative Technologies

{zjin,feng}@ict.usc.edu

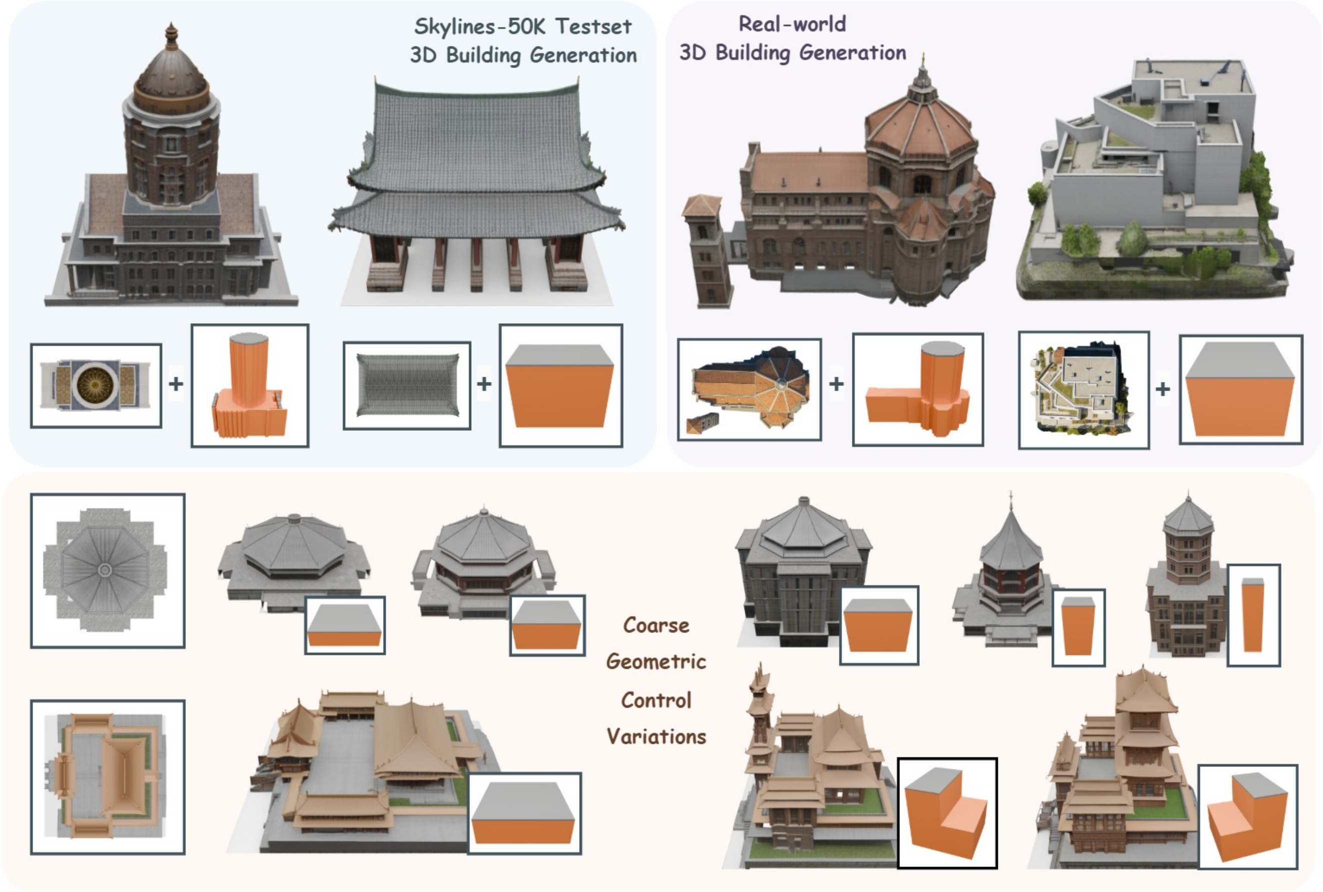

Figure 1. 3D building assets generated by our method using satellite imagery and coarse geometric priors. Our method not only demonstrates versatile generation, but also produces realistic performance in real-world scenarios.

Abstract

*We present **SatSkylines**, a 3D building generation approach that takes satellite imagery and coarse geometric priors. Without proper geometric guidance, existing image-based 3D generation methods struggle to recover accurate building structures from the top-down views of satellite images alone. On the other hand, 3D detailization methods tend to rely heavily on highly detailed voxel inputs and fail to produce satisfying results from simple priors such as cuboids. To address these issues, our key idea is to model the transformation from interpolated noisy coarse priors to detailed geometries, enabling flexible geometric control without additional computational cost. We have further de-*

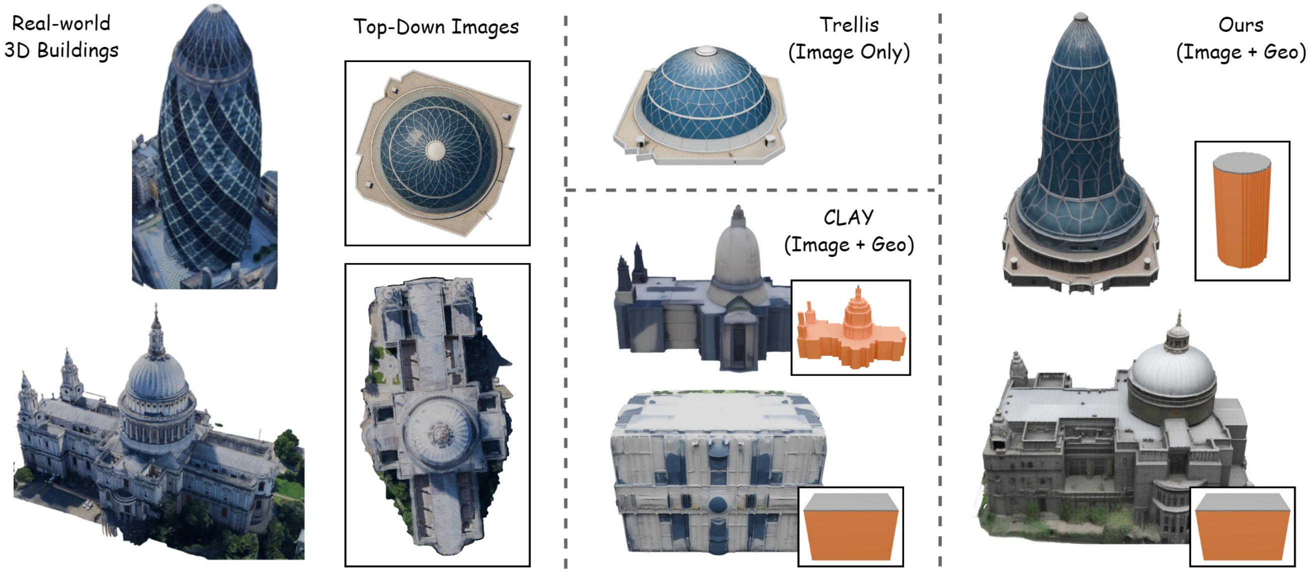

Figure 2. **Necessity of our method**. Trellis fails to recover building heights (upper-middle). CLAY requires highly detailed voxels to work well (lower-middle). Our method takes top-down images and coarse geometric priors to generate realistic 3D buildings (right).

*veloped **Skylines-50K**, a large-scale dataset of over 50,000 unique and stylized 3D building assets in order to support the generations of detailed building models. Extensive evaluations indicate the effectiveness of our model and strong generalization ability. See the project page for more results: https://jinzhangyu.github.io/projects/SatSkylines/*

1. Introduction

Condtional 3D Building Generation is a growing research direction in 3D World Generation [12], attracting attention from both academic and industry. Instead of relying on oblique aerial-view or street-view images, which are commonly used as the conditions for generating 3D models, we choose to leverage satellite imagery for its globally available large-scale data. This enables scalable 3D building modeling in regions where ground-level or oblique data is unavailable despite its lower resolution and fixed top-down view.

Although numerous models [5–7, 10, 11, 15, 20, 25, 29, 32, 33, 35, 54, 55, 62, 65, 67] have been proposed, achieving high-quality 3D building generation from top-down satellite images and coarse geometric priors remains challenging due to the following difficulties:

Inefficient Usage of Satellite Images and Coarse Geometries. Many existing image-based 3D generation methods [20, 25, 29, 32, 33, 35, 54, 55, 62, 67] use the Sphere Hammersley Sequence [53] to render image prompts from 3D assets during training, but they typically ignore pure top-down views. Top-down or satellite-view images are more challenging than traditional front or side views, because they provide no explicit height data and omit vertical surface details. In challenging cases, the lack of vertical cues in satellite imagery can result in ambiguous or inaccurate predictions, including failure to generate building-like structures. Consequently, such methods, like Trellis [55], often fail to reconstruct accurate building geometries using satellite imagery (see Fig.2). Other methods support both image and voxel controls, but require highly detailed geometric voxels to achieve satisfactory results [11, 65]. As shown in Fig. 2, CLAY [65] requires voxels to be detailed enough to do further refinement. However, if provided with only a simple cuboid, it fails to add necessary geometric complexity. Moreover, approaches [11, 65] that support both image prompts and geometric priors suffer from prohibitively slow inference, limiting their potential applicability in large-scale building model generations. To address these limitations, we design our method to efficiently leverage both top-down satellite imagery and flexible coarse geometry priors while maintaining fast inference speed.

Lack of High-Quality 3D Building Datasets. Scale, diversity, and quality have been crucial factors in recent AI. However, existing 3D building datasets [8, 9, 14, 18, 30, 45, 46] often fall short in these aspects: (*a*) **Scale**. Although some datasets [16, 22, 59, 60] include collections of 3D building assets, they are in the form of colorless meshes, point clouds, or images. Datasets with textured meshes remain relatively small in scale. (*b*) **Diversity**. Certain datasets [23, 36] focus on buildings from only a few cities (e.g., NYC, LA), lacking the architectural variety all over the world. This narrow coverage lowers their ability

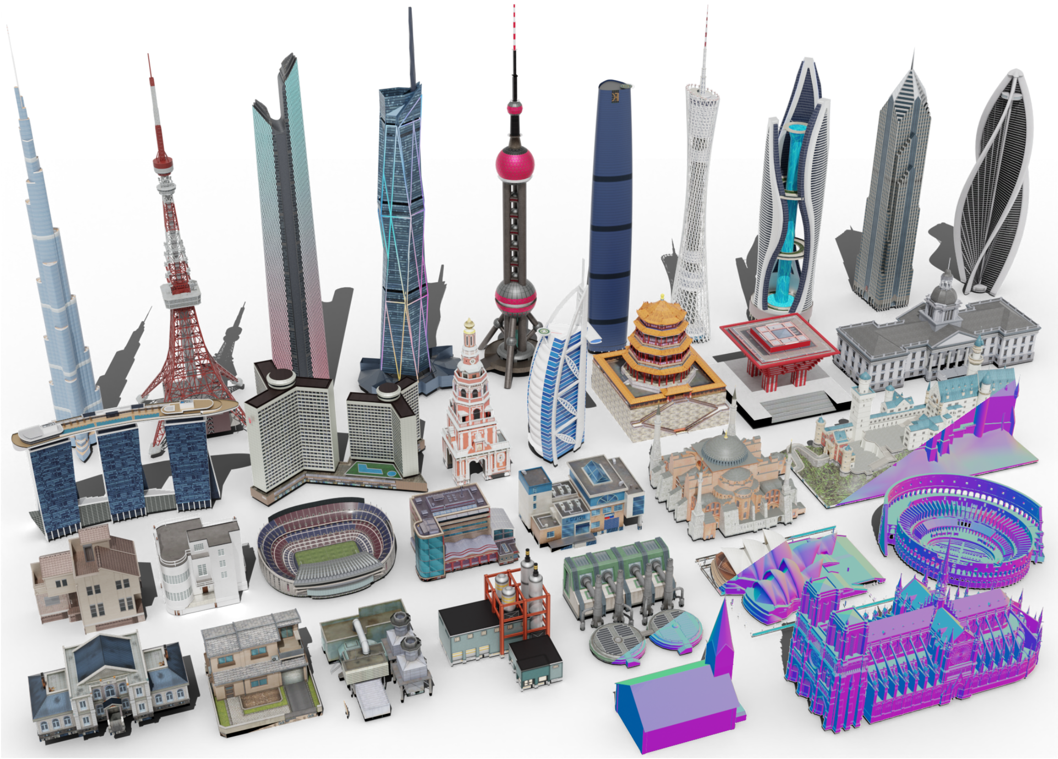

Figure 3. **Skylines-50K** is a large-scale, diverse, high quality 3D building dataset. These assets are sourced from the Steam Workshop of the famous city-building and simulation game '*Cities: Skylines*'. Examples of rendered buildings are shown here to demonstrate style diversity.

to in-the-wild settings. In our work, we address these limitations by constructing a large-scale, globally diverse, and high-quality dataset of textured 3D building assets, enabling models to learn from both geometric detail and realistic appearance.

In this paper, we introduce **SatSkylines**, a 3D building generative approach that takes top-down satellite imagery together with coarse geometric priors as control. Following Trellis [55], the top-down satellite image is injected into the sparse structure flow transformer as the condition. Rather than feeding in pure noise (as in Trellis [55]) or using just the coarse geometry prior (like DetailGen3D [10]), we apply a cosine interpolation between those two. This process is further refined with a latent normalization on the coarse geometry priors to ensure both items follow the same gaussian distribution before interpolation. This also allows controlling the intensity of the geometry priors as the constraints by simply changing the interpolation parameters to balance between higher fidelity and higher creativity. By exploiting these techniques, our method can support both top-down satellite images and coarse geometry priors for generating high quality 3D building models.

We also propose **Skylines-50K**, a large-scale 3D building dataset containing over 50,000 unique and stylized assets (Fig. 3). The dataset is sourced from the Steam Workshop of the popular city-building and simulation game '*Cities: Skylines*'. Over the past 10 years, dedicated players all over the world have created and shared 3D building assets from their own countries, resulting in a rich collection that spans diverse architectural styles. The dataset includes landmarks, buildings of varying heights and scales, and distinctive structures, forming a valuable resource for developing high-quality, generalizable 3D building generation models.

Based on the proposed model, we have also developed an end-to-end pipeline that generate 3D building assets from real world satellite imagery. Given a GPS coordinate, our pipeline automatically gathers and enhances satellite imagery while generating coarse 3D geometry as priors. The prepared data is then fed into the generative model, providing a streamline process for real-world 3D building generation. We also evaluated our method on the Skylines-50K dataset and a diverse set of randomly selected real-world buildings. Our experiment results showed that SatSkylines is able to generate reliable outputs that align well with both the satellite imagery and the coarse geometry constraints.

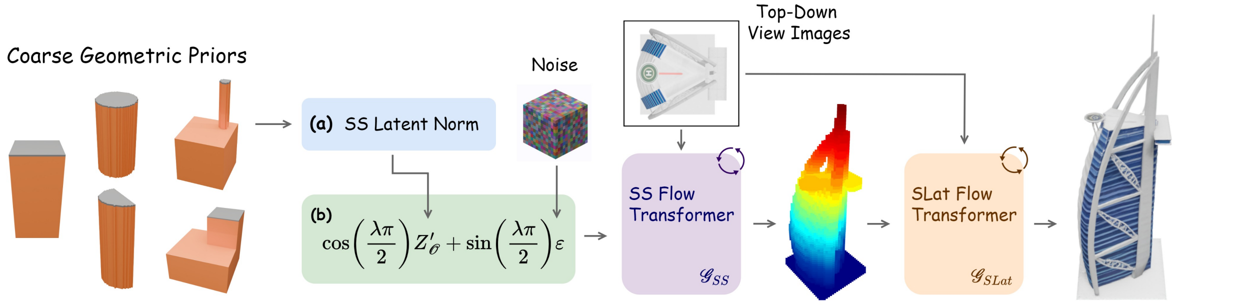

Figure 4. **SatSkylines Architecture**. The coarse geometric prior $\mathcal{O}$ is encoded by the SS VAE to obtain $Z_{\mathcal{O}}$. (a) A channel-wise latent normalization is applied to produce $Z'_{\mathcal{O}}$. (b) The cosine geometric interpolation is then performed between $Z'_{\mathcal{O}}$ and gaussian noise ϵ, with λ controlling geometric guidance strength. Finally, the SS and SLat flow transformers generate detailed geometry and appearance.

Our main contributions are summarized as follows.

(*a*) A large-scale and high quality 3D building dataset.

(*b*) An effective 3D building generation architecture using both satellite imagery and coarse geometry priors.

(*c*) An end-to-end pipeline that produces detailed 3D building assets from real world satellite imagery.

2. Related Works

3D Building Datasets. Existing 3D building datasets fall short in the following aspects: (*a*) Non-textured 3D format. Some provides only colorless meshes, including Building3D [50] and so on [24, 31, 52]. Others are available only as point clouds, including SensatUrban [21] and others [4, 51, 63]. A large number of them are only composed of images, such as CityTopia [57] and some else [27, 28, 34, 38, 47, 49, 56, 58, 61, 66]. Additionally, BuildingNet [44] has colorized 3D buildings but with non-photorealistic colors. (*b*) Small at scale. Existing datasets with good textured 3D buildings have limited number of assets, such as UrbanBIS [60], SS3DM [22], SUMParts [16], and TwinTex [59]. (*c*) Lack of diversity. Current large-scale textured 3D building datasets usually focus on only a few cities (e.g., NYC, LA), lacking the architectural variety all over the world, such as UrbanScene3D [36] and Sat2City [23]. (*d*) Quality constraints. Many recent large-scale, diverse datasets are not fully handcrafted, resulting in bumpy surfaces and low-resolution textures, such as Google Earth [37] and other digital twin cities datasets [8, 9, 14, 18, 30, 45, 46]. In contrast, our Skylines-50K dataset offers over 50,000 handcrafted and textured 3D building assets from diverse locations worldwide.

3D Generation Control Types. High-resolution image-based 3D generation models have advanced rapidly, such as Trellis [55] and others [20, 25, 29, 32, 33, 35, 54, 62, 67]. But they are neither trained on top-down satellite imagery nor designed to support geometric control. Conversely, some geometry-controlled 3D generation models lack the ability to take image prompts. For instance, Cube [1] supports text prompts with 3D bounding box control, and Sat2City [23] uses only height maps. 3D detailization methods refine coarse geometric priors into detailed geometry but also face limitations. Mars [15] and DECOLLAGE [6] only accept coarse geometry without image inputs, while DetailGen3D [10] supports both image prompts and coarse voxels but outputs geometry without texture or color. ART-DECO [7] can produce textured 3D assets but requires additional text prompts. ShaDDR [5] and Coin3D [11] accept both image and coarse geometry inputs and output textured assets, but they demand unrealistically detailed input to achieve satisfactory results. For example, when provided only with a simple cuboid, their outputs are of low quality. Rodin, a commercial product built on CLAY [65], supports diverse inputs including images, text, bounding boxes, voxels, and point clouds, and can generate high-quality textured assets. However, it requires around 5–10 minutes per forward pass, making it impractical for large-scale automated use. In comparison, our SatSkylines model efficiently processes top-down satellite imagery together with flexible coarse geometric priors to produce textured, detailed 3D building assets in around 15 seconds.

3D Generation Control Mechanisms. Images are usually encoded using DINOv2 [41] or CLIP [43] to extract visual features, which are then injected as key–value pairs into the cross-attention layers of a DiT [42], as illustrated in Trellis [55] and related works [35, 54]. In some cases, image features are concatenated directly with the noise latent and fed into the DiT, as in [25, 33, 67]. We adopt the Trellis design [55] for its strong trade-off between generation quality and computational efficiency. For 3D geometry control, LION [48] proposes a training-free diffuse–denoise approach for point clouds control, but the absence of training also limits its generalization ability. CLAY [65] and others [11, 23] incorporate additional ControlNet [64] or Adapters [40] for voxel control, which increases model complexity and slows inference. Cube [1] encodes 3D bounding box coordinates as text via the CLIP text encoder [43], but this design is difficult to extend to more complex voxel control. DetailGen3D [10] feeds coarse geometry directly into the DiT, directly modeling the transformation between coarse and fine geometry. However, this is unsuitable for our setting, where the model must handle cases ranging from a simple cuboid to diverse building shapes guided by various image prompts. Such a "one-to-many" mapping increases the difficulty of Rectified Flow [39] learning. Instead, our SatSkylines model, built on Trellis [55], interpolates between pure noise and coarse geometric priors as the DiT input, preserving high inference speed and enabling simultaneous image and geometry control.

3. Method

An overview of our **SatSkylines** model is shown in Fig. 4. Inspired by Trellis [55], we first uses a Sparse Structure (SS) Rectified Flow [39] transformer to produce geometry latents Z_{ss}, followed by a Structured Latent (SLat) transformer to generate appearance latents Z_{slat}. Top-down view images are introduced through cross-attention layers. However, Trellis relies only on noise inputs and does not incorporate necessary geometric controls.

3.1. Coarse Geometric Control

Existing research, such as [10], demonstrated potential in directly modeling transformation between coarse geometric priors $Z_{\mathcal{O}}$ and detailed ones Z_{ss}. However, these methods [5–7, 10, 11, 15, 65] require highly detailed $Z_{\mathcal{O}}$ to achieve satisfactory results Z_{ss}, making them unsuitable for our use case. Our goal is to develop a model capable of generating detailed geometry from simple starting priors, such as a basic cuboid. This challenge is a 'one-to-many' mapping problem, where a single, simple coarse prior could lead to many different detailed outcomes, which is more difficult for the Rectified Flow to learn. To address this, we take an interpolation between $Z_{\mathcal{O}}$ and gaussian noise ϵ as the input for our SS Flow Transformer. This approach not only provides the model with geometric prior information but also effectively addresses the "one-to-many" problem by varying the starting point with different level of noises. Unlike other techniques [25, 29, 67] that increase computational costs by concatenating conditions with noise latent, this method does not affect training or inference speeds.

As shown in Fig. 4, any coarse geometric priors are first converted into a N^3 grid voxels $\mathcal{O} \in \{0,1\}^{N\times N\times N}$, where 1 indicates an occupied voxel and 0 means empty. We then apply the SS VAE encoder to get the SS latent representation $Z_{\mathcal{O}} \in \mathbb{R}^{D\times D\times D\times C'}$, where D is down-sampled spatial resolution and C' is the corresponding feature dimension.

SS Latent Normalization. After obtaining the $Z_{\mathcal{O}}$, we perform channel-wise normalization.

$$Z'_{\mathcal{O}} = \frac{Z_{\mathcal{O}} - Mean(Z_{\mathcal{O}})}{Std(Z_{\mathcal{O}})}$$

In our experiments, the distribution of $Z_{\mathcal{O}}$ does not follow a standard gaussian distribution but roughly $\mathcal{N}(0, 0.2)$. Without normalization, it makes the model less responsive to the geometric priors, leading to poor geometric control.

Cosine Geometric Interpolation. We define the interpolated geometric latent as follows.

$$Z''_{\mathcal{O}} = \cos(\frac{\lambda\pi}{2})Z'_{\mathcal{O}} + \sin(\frac{\lambda\pi}{2})\epsilon$$

where ϵ is the pure gaussian noise. Since $Z'_{\mathcal{O}}$ and ϵ follow gaussian distributions, their linear combination remains gaussian, owing to the identity $\cos^2(\frac{\lambda\pi}{2}) + \sin^2(\frac{\lambda\pi}{2}) = 1$. Here $\lambda \in [0, 1]$ controls the degree of coarse geometric guidance: when $\lambda = 0$, the interpolation fully reflects the geometry priors ($Z''_{\mathcal{O}} = Z'_{\mathcal{O}}$); when $\lambda = 1$, it ignores the geometry priors and use only noise. At inference, λ can be manually set to adjust the amount of geometric influence.

3.2. Dataset Curation

Existing 3D building datasets [4, 8, 9, 14, 16, 18, 21–23, 27, 28, 30, 34, 36–38, 45–47, 49, 51, 56–61, 63, 66] are often limited in scale, diversity, or quality. To address these gaps, we introduce a new dataset, **Skylines-50K**, as illustrated in Fig. 3. We collect raw asset files from the Steam Workshop of '*Cities: Skylines*' and convert them into meshes with PBR materials. All of the assets are then exported into the GLB format.

For each 3D asset, we render eight image conditions around the top view, ensuring that one of them is a pure top view. Since the Skylines-50K dataset does not provide OpenStreetMap data, we generate coarse geometric priors ourselves. Each asset is assigned three levels of priors (LOD 0–2): LOD 0 corresponds to a bounding box or cuboid, LOD 1 uses one unique cross-section along the height, and LOD 2 employs two distinct cross-sections. This design is intended to mimic OpenStreetMap building data, which typically contains 2D footprints and height attributes. During training, these four geometric priors together with a pure Gaussian noise are uniformly sampled within each batch.

3.3. Real-world 3D Building Generation Pipeline

Image-based 3D generation methods [20, 25, 29, 32, 33, 35, 54, 55, 62, 67] require users to provide an image prompt with foreground masks. In practice, Rembg [17] is usually applied to separate the foreground building from the background automatically. However, this approach tends to perform poorly on satellite images, since most background-removal models are not trained for this specific domain and

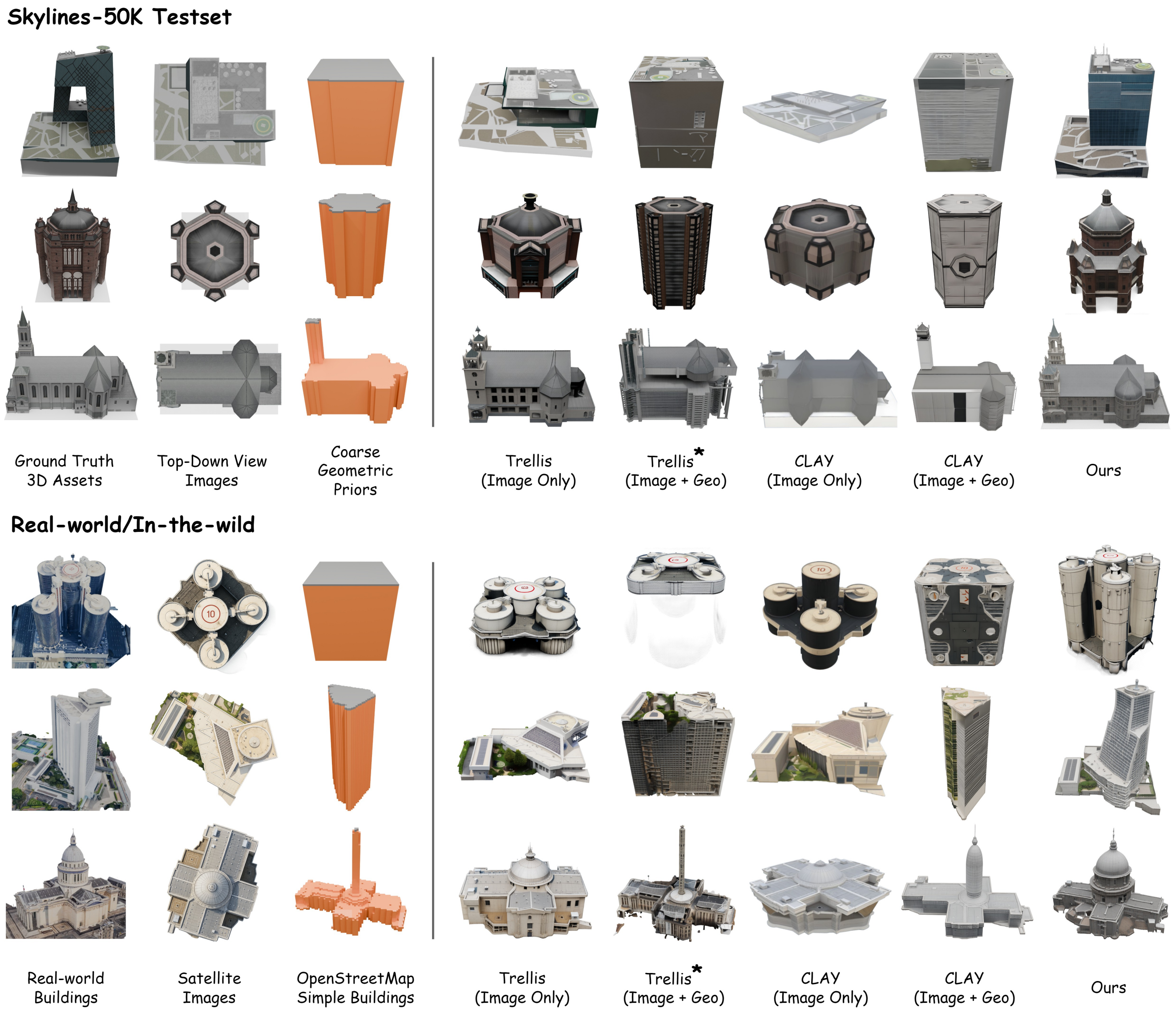

Figure 5. Visual comparisons of generated 3D building assets between our method and previous approaches. The first three rows are from the Skylines-50K test set, and the last three rows are real-world examples. Here ⋆ indicates that Trellis does not natively support coarse geometric control, but our cosine geometric interpolation can be applied to incorporate geometric priors into its input.

fail to create clean masks. In addition, geometry-controlled 3D generation models [5–7, 10, 11, 15, 65] require users to give a geometric prior. Typically, this involves creating voxel representations using software such as Mesh Editor [26], Blender [2], or Unreal Engine [13]. This presents a steep learning curve for both novice users and experienced artists to prepare the inputs for large scale building model generations.

To address these issues and to demonstrate the proposed method, we have developed an end-to-end pipeline that requires only a geo-spatial bounding box of (min-lat, max-lat, min-lon, max-lon) to simplify the data preparation and building model generations. Specifically, from the bounding box information, it automatically retrieves 2D building footprints and height attributes from OpenStreetMap to generate coarse geometric priors, avoiding manual voxel creation. Satellite imagery from sources like Google Maps or Mapbox can also be combined with OpenStreetMap outlines to directly extract building masks without additional segmentation tools. Finally, since satellite imagery is often blurry or low-resolution, the pipeline also leverages '*gpt-image-1*' to enhance the image quality with super-resolution.

Methods	# Assets	Geometry			Appearance		
		IoU ↑	CD ↓	F Score ↑	PSNR ↑	LPIPS ↓	CLIP ↑
Trellis [55]	500	0.4415	0.0632	0.6295	11.9705	0.4529	0.6461
CLAY [65]	20*	0.6859	0.0574	0.6516	13.3745	0.4136	0.6549
SatSkylines	500	**0.9381**	**0.0168**	**0.8684**	**19.2386**	**0.1678**	**0.7860**
	20*	0.9278	0.0080	0.9091	18.6003	0.1786	0.7958

Table 1. **Quantitative Comparisons in Skylines-50K Dataset**. Trellis uses image prompts only, while CLAY and our SatSkylines additionally use LOD 1 coarse geometric priors. Here * means, due to CLAY's slow inference speed, we evaluate it only on a smaller sub-test set of 20 assets.

4. Experiments

Implementation details. Our model is built upon Trellis with the 1.1B image-large configuration. We initialize from Trellis pretrained weights and finetune on Skylines-50K using a learning rate of $1e^{-4}$, batch size of 32, and $40K$ total steps on 4 A100 GPUs. Channel-wise mean and standard deviation of Z_O are fixed constants for both training and inference. The interpolation factor λ follows a logit-normal distribution with parameters $\mu = 1, \sigma = 1$ during training, while being fixed to 0.5 at inference. At inference, classifier-free guidance (CFG) scales are set to 7.5 for SS and 3.0 for SLat, with 50 sampling steps for both modules.

4.1. 3D Building Generation

In this section, we evaluate our 3D building generation quality. We first present various 3D generation results of our method, and then compare with other baseline methods.

Skylines-50K Dataset. Our evaluation is conducted on the Skylines-50K test set of 500 representative instances. For the geometry evaluation, we empoly Chamfer Distance (CD), IoU, and F-score of generated sparse structure voxels. For appearance quality, each instance is rendered into four views with FoV 40°, radius 2, pitch 30°, and yaw angles $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$. We then compute PSNR, LPIPS, and CLIP similarity against ground truth images. Following [7], in order to enable fair comparisons with Rodin (the commercial version of CLAY), we additionally define a 20-instance sub-test set. We conduct the comparison with CLAY on this sub-test set, as Rodin is both costly and requires approximately 5–10 minutes to generate a single building, making large-scale automatic evaluation impractical.

Real-world 3D Building Generation. We have also evaluated our model qualitatively in real-world, in-the-wild settings. Since no ground truth 3D assets are available, the qualitative results are provided with Google Earth screenshots as reference.

As shown in Fig. 5, our method produce better quality 3D building models compared to previous approaches, offering not only top-down satellite image control but also more precise alignment with flexible coarse geometric priors. Tab. 1 further demonstrates that our method shows competitive performance across all evaluation metrics. Trellis [55] generates reasonably good 3D buildings from satellite images but fails to capture realistic building geometries. Trellis [55] can also use our cosine geometric interpolation to accept geometric controls. However, the performance is not as reliable since it is not trained on such distributions with top-down view conditions, as shown in Fig. 5 and Tab. 2. CLAY [65] relies heavily on voxel priors and lacks further refinement under satellite prompts.

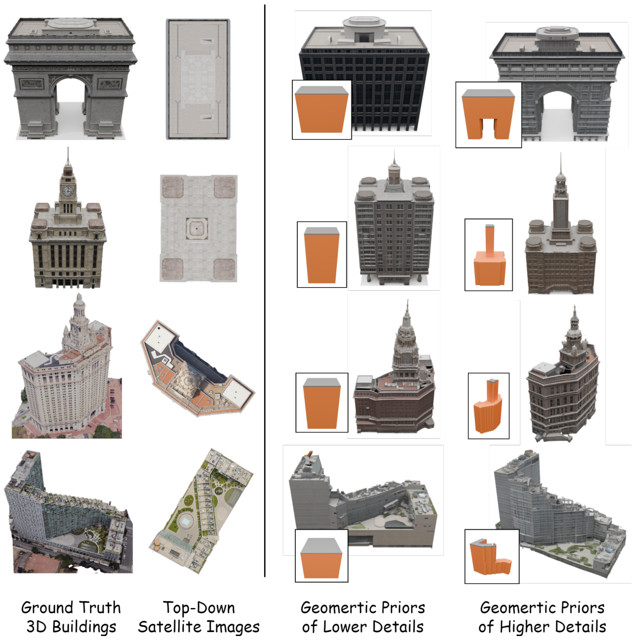
Ground Truth 3D Buildings
Top-Down Satellite Images
Geomertic Priors of Lower Details
Geomertic Priors of Higher Details

Figure 6. **Visualization of Coarse Geometric Prior Variations**. The first two rows are examples from the Skylines-50K dataset, while the last two rows are real-world cases.

Methods	Geormtric Priors Coarseness Level	Geometry		
		IoU ↑	CD ↓	F Score ↑
Trellis [55]	LOD 2	0.6367	0.0529	0.6495
SatSkylines	LOD 0	**0.9515**	0.0222	0.8315
	LOD 1	0.9381	0.0168	0.8684
	LOD 2	0.9441	**0.0141**	**0.8943**

Table 2. **Quantitative Analysis of Coarse Geometric Prior Variations**. Trellis does not natively support coarse geometric control, but our cosine interpolation enables it to use geometric priors. LOD 0 denotes a simple cuboid, while LOD 2 specifies two distinct cross-sections along the height.

4.2. Coarse Geometric Prior Variations

SatSkylines can generate reasonable 3D buildings from very coarse priors, such as a simple cuboid. When provided with more detailed priors, the generated geometry further improves, producing more accurate and faithful building shapes. As illustrated in Fig. 6, for example, the Arc de Triomphe can be approximated as a cuboid, leading the model to generate a generic office-like structure. However, if the

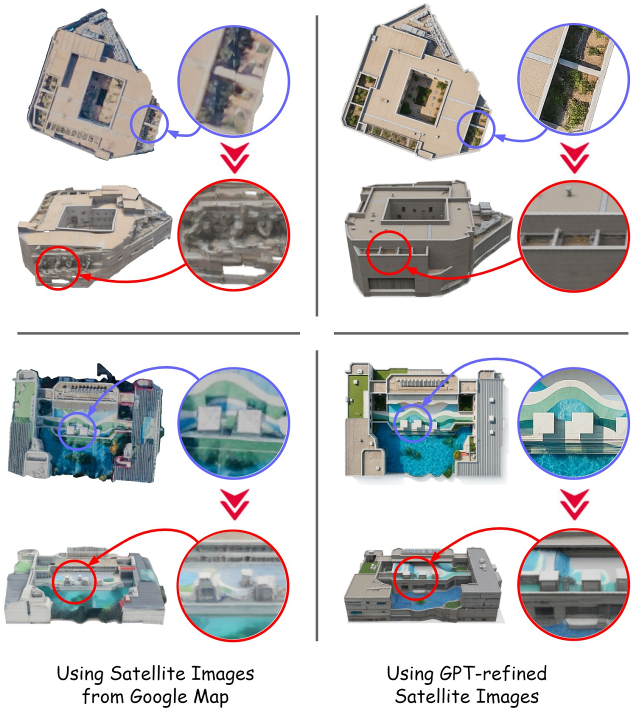

Figure 7. **Visualization of GPT Satellite Image Refinement**. Blue circles highlight zoomed-in details from 2D satellite images, while red circles mark the corresponding regions in SatSkylines generated 3D buildings. All examples are randomly sampled from real-world data.

Methods	Geometry		
	IoU ↑	CD ↓	F Score ↑
Trellis [55]	0.4415	0.0632	0.6295
w finetune on Skylines-50K	0.6599	0.0273	0.8047
SatSkylines	**0.9381**	**0.0168**	**0.8684**
wo ss latent norm	0.9300	0.0196	0.8598

Table 3. **Ablation Studies** include finetuning the image-only Trellis on our Skylines-50K dataset, and '*wo ss latent norm*' refers to applying cosine geometric interpolation between the unnormalized $Z_{\mathcal{O}}$ and noise ϵ.

geometric prior includes an empty space in the middle, the model is able to infer a reasonable arch structure. In the second row of Fig. 6, the target building consists of a lower cuboid base and an upper tower. From the simple cuboid, the model struggles to estimate the relative tower height. When the prior explicitly specifies two stacked cuboids, SatSkylines successfully reconstructs the correct proportion between the upper and lower components. Similar improvements are also observed in additional real-world examples.

Tab. 2 reports results on Skylines-50K using geometric priors of different coarseness. We define three levels of details (LODs): LOD 0, a bounding box or simple cuboid; LOD 1, one unique cross-section repeated along the building height; LOD 2, two unique cross-sections. IoU is highest for LOD 0 because its definition directly matches cuboid overlap with ground truth. Beyond IoU, however, higher LOD priors consistently lead to more accurate reconstructions, showing that SatSkylines benefits from increased prior detail while remaining robust to coarse inputs.

4.3. GPT Satellite Image Refinement

As described in §3.3, in real-world settings, after obtaining the raw satellite image of a building, we refine it using '*gpt-image-1*'. This step is necessary since satellite imagery from Google Maps, Mapbox, or ArcGIS is often blurry and low-resolution. Since the quality of the image prompt directly influences the fidelity of 3D building generation, improving clarity is essential. As shown in Fig. 7 (left column), blurred satellite images lead to degraded 3D reconstructions. For example, gravel-like noise in the upper-left case and washed-out textures in the lower-left case.

Our pipeline employs '*gpt-image-1*' with carefully designed text prompts to edit the original satellite image into a sharper, more detailed, and blur-free version. As shown in Fig. 7 (right column), 3D buildings generated from refined images exhibit clearer geometry and more realistic textures compared to those produced from the original blurred inputs.

4.4. Ablation Studies

We conduct ablation studies to validate the design choices of our method. First, we finetune Trellis on the Skylines-50K dataset, which yields a noticeable improvement, reducing Chamfer Distance by 0.04. This demonstrates that our dataset facilitates the adaptation of a generalized 3D generation model to the satellite-view domain.

By adding our cosine geometric interpolation and SS latent normalization, we achieve an additional 0.011 reduction in Chamfer Distance, showing that the model successfully integrates coarse geometric priors into the generation process to produce more accurate building structures. Finally, removing SS latent normalization results in a 0.003 drop in Chamfer Distance, indicating that aligning $Z'_{\mathcal{O}}$ to have the same mean and standard deviation as pure Gaussian noise further improves generation quality.

5. Acknowledgement

The authors would like to thank our primary sponsors of this research: Mr. Clayton Burford of the Battlespace Content Creation (BCC) team at Simulation and Training Technology Center (STTC). This work is supported by University Affiliated Research Center (UARC) award W911NF-14-D-0005. Statements and opinions expressed and content included do not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] Kiran Bhat, Nishchaie Khanna, Karun Channa, Tinghui Zhou, Yiheng Zhu, Xiaoxia Sun, Charles Shang, Anirudh Sudarshan, Maurice Chu, Daiqing Li, et al. Cube: A roblox view of 3d intelligence. *arXiv preprint arXiv:2503.15475*, 2025. 4

[2] Blender Foundation. Blender — a 3d modeling and rendering package, 2024. Accessed: [Current Date]. 6

[3] Du Chen, Liyi Chen, Zhengqiang Zhang, and Lei Zhang. Generalized and efficient 2d gaussian splatting for arbitrary-scale super-resolution. *arXiv preprint arXiv:2501.06838*, 2025. 1

[4] Meida Chen, Qingyong Hu, Zifan Yu, Hugues Thomas, Andrew Feng, Yu Hou, Kyle McCullough, Fengbo Ren, and Lucio Soibelman. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. *arXiv preprint arXiv:2203.09065*, 2022. 4, 5

[5] Qimin Chen, Zhiqin Chen, Hang Zhou, and Hao Zhang. Shaddr: interactive example-based geometry and texture generation via 3d shape detailization and differentiable rendering. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 2, 4, 5, 6

[6] Qimin Chen, Zhiqin Chen, Vladimir G Kim, Noam Aigerman, Hao Zhang, and Siddhartha Chaudhuri. Decollage: 3d detailization by controllable, localized, and learned geometry enhancement. In *European Conference on Computer Vision*, pages 110–127. Springer, 2024. 4

[7] Qimin Chen, Yuezhi Yang, Yifang Wang, Vladimir G Kim, Siddhartha Chaudhuri, Hao Zhang, and Zhiqin Chen. Art-deco: Arbitrary text guidance for 3d detailizer construction. *arXiv preprint arXiv:2505.20431*, 2025. 2, 4, 5, 6, 7

[8] City of Espoo. Open city information model from the WFS interface service. https://kartat.espoo.fi/3d/services_en.html, 2025. 2, 4, 5

[9] City of Kuopio. 3D city model. https://kuopio.kunta3d.fi/Map.html?locale=en, 2025. 2, 4, 5

[10] Ken Deng, Yuan-Chen Guo, Jingxiang Sun, Zi-Xin Zou, Yangguang Li, Xin Cai, Yan-Pei Cao, Yebin Liu, and Ding Liang. Detailgen3d: Generative 3d geometry enhancement via data-dependent flow. *arXiv preprint arXiv:2411.16820*, 2024. 2, 3, 4, 5, 6

[11] Wenqi Dong, Bangbang Yang, Lin Ma, Xiao Liu, Liyuan Cui, Hujun Bao, Yuewen Ma, and Zhaopeng Cui. Coin3d: Controllable and interactive 3d assets generation with proxy-guided conditioning. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 2, 4, 5, 6

[12] Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. *arXiv preprint arXiv:2504.00983*, 2025. 2

[13] Epic Games. Unreal engine, 2024. Accessed: [Current Date]. 6

[14] Freie und Hansestadt Hamburg. Geoportal Hamburg. https://geoportal-hamburg.de/, 2025. 2, 4, 5

[15] Jingnan Gao, Weizhe Liu, Weixuan Sun, Senbo Wang, Xibin Song, Taizhang Shang, Shenzhou Chen, Hongdong Li, Xiaokang Yang, Yichao Yan, et al. Mars: Mesh autoregressive model for 3d shape detailization. *arXiv preprint arXiv:2502.11390*, 2025. 2, 4, 5, 6

[16] Weixiao Gao, Liangliang Nan, and Hugo Ledoux. Sum parts: Benchmarking part-level semantic segmentation of urban meshes. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24474–24484, 2025. 2, 4, 5

[17] Daniel Gatis. rembg: Remove image backgrounds. https://pypi.org/project/rembg/, 2022. Version 2.0.67, released 2025-07-05. 5

[18] Gemeente Rotterdam. 3d rotterdam. https://www.3drotterdam.nl/, 2025. 2, 4, 5

[19] Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mambairv2: Attentive state space restoration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28124–28133, 2025. 1

[20] Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. *arXiv preprint arXiv:2503.21732*, 2025. 2, 4, 5

[21] Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. Sensaturban: Learning semantics from urban-scale photogrammetric point clouds. *International Journal of Computer Vision*, 130(2):316–343, 2022. 4, 5

[22] Yubin Hu, Kairui Wen, Heng Zhou, Xiaoyang Guo, and Yong-jin Liu. Ss3dm: Benchmarking street-view surface reconstruction with a synthetic 3d mesh dataset. *Advances in Neural Information Processing Systems*, 37:106649–106666, 2024. 2, 4

[23] Tongyan Hua, Lutao Jiang, Ying-Cong Chen, and Wufan Zhao. Sat2city: 3d city generation from a single satellite image with cascaded latent diffusion. *arXiv preprint arXiv:2507.04403*, 2025. 2, 4, 5

[24] Jin Huang, Jantien Stoter, Ravi Peters, and Liangliang Nan. City3d: Large-scale building reconstruction from airborne lidar point clouds. *Remote Sensing*, 14(9):2254, 2022. 4

[25] Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, et al. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025. 2, 4, 5

[26] Hyper3D. Mesh editor, 2024. Accessed: [Current Date]. 6

[27] Lihan Jiang, Kerui Ren, Mulin Yu, Linning Xu, Junting Dong, Tao Lu, Feng Zhao, Dahua Lin, and Bo Dai. Horizon-gs: Unified 3d gaussian splatting for large-scale aerial-to-ground scenes. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26789–26799, 2025. 4, 5

[28] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024. 4, 5

[29] Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui

Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025. 2, 4, 5

[30] Landeshauptstadt Dresden. 3-D-Modell der Landeshauptstadt Dresden. https://www.dresden.de/de/leben/stadtportrait/statistik/geoinformationen/3-d-modell.php?shortcut=3D, 2025. 2, 4, 5

[31] Han-Hung Lee, Qinghong Han, and Angel X Chang. Nuiscene: Exploring efficient generation of unbounded outdoor scenes. *arXiv preprint arXiv:2503.16375*, 2025. 4

[32] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024. 2, 4, 5

[33] Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025. 2, 4, 5

[34] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3205–3215, 2023. 4, 5

[35] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025. 2, 4, 5

[36] Liqiang Lin, Yilin Liu, Yue Hu, Xingguang Yan, Ke Xie, and Hui Huang. Capturing, reconstructing, and simulating: the urbanscene3d dataset. In *European Conference on Computer Vision*, pages 93–109. Springer, 2022. 2, 4, 5

[37] Richard J Lisle. Google earth: a new geological resource. *Geology today*, 22(1):29–32, 2006. 4

[38] Jin Liu, Jian Gao, Shunping Ji, Chang Zeng, Shaoyi Zhang, and JianYa Gong. Deep learning based multi-view stereo matching and 3d scene reconstruction from oblique aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 204:42–60, 2023. 4, 5

[39] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 5

[40] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4296–4304, 2024. 4

[41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4

[42] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 4

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 4, 5

[44] Pratheba Selvaraju, Mohamed Nabail, Marios Loizou, Maria Maslioukova, Melinos Averkiou, Andreas Andreou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. Buildingnet: Learning to label 3d buildings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10397–10407, 2021. 4

[45] Stadt Leipzig. Geoportal Leipzig. https://geoportal.leipzig.de/arcgis/apps/webappviewer3d/index.html?id=636b96152aac4769b6cf316312f3bf70, 2025. 2, 4, 5

[46] State of Victoria. Digital Twin Victoria. https://digitaltwin.vic.gov.au/public/, 2025. 2, 4

[47] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12922–12931, 2022. 4, 5

[48] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022. 4

[49] Guangyu Wang, Jinzhi Zhang, Fan Wang, Ruqi Huang, and Lu Fang. Xscale-nvs: Cross-scale novel view synthesis with hash featurized manifold. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21029–21039, 2024. 4, 5

[50] Ruisheng Wang, Shangfeng Huang, and Hongxin Yang. Building3d: A urban-scale dataset and benchmarks for learning roof structures from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20076–20086, 2023. 4

[51] Yao Wei, George Vosselman, and Michael Ying Yang. Buildiff: 3d building shape generation using single-image conditional point cloud diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2910–2919, 2023. 4, 5

[52] Congcong Wen, Wenyu Han, Lazarus Chok, Yan Liang Tan, Sheung Lung Chan, Hang Zhao, and Chen Feng. Realcity3d: A large-scale georeferenced 3d shape dataset of real-world cities. 4

[53] Tien-Tsin Wong, Wai-Shing Luk, and Pheng-Ann Heng. Sampling with hammersley and halton points. *Journal of Graphics Tools*, 2(2):9–24, 1997. 2

[54] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *arXiv preprint arXiv:2505.17412*, 2025. 2, 4, 5

[55] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 2, 3, 4, 5, 7, 8

[56] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. Citydreamer: Compositional generative model of unbounded 3d cities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9666–9675, 2024. 4, 5

[57] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. Citydreamer4d: Compositional generative model of unbounded 4d cities. *arXiv e-prints*, pages arXiv–2501, 2025. 4

[58] Butian Xiong, Nanjun Zheng, Junhua Liu, and Zhen Li. Gauu-scene v2: Assessing the reliability of image-based metrics with expansive lidar image dataset using 3dgs and nerf. *arXiv preprint arXiv:2404.04880*, 2024. 4

[59] Weidan Xiong, Hongqian Zhang, Botao Peng, Ziyu Hu, Yongli Wu, Jianwei Guo, and Hui Huang. Twintex: Geometry-aware texture generation for abstracted 3d architectural models. *ACM Transactions on Graphics (TOG)*, 42 (6):1–14, 2023. 2, 4

[60] Guoqing Yang, Fuyou Xue, Qi Zhang, Ke Xie, Chi-Wing Fu, and Hui Huang. Urbanbis: a large-scale benchmark for fine-grained urban building instance segmentation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2, 4

[61] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020. 4, 5

[62] Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236*, 3:2, 2025. 2, 4, 5

[63] Jun Yin, Wen Gao, Jizhizi Li, Pengjian Xu, Chenglin Wu, Borong Lin, and Shuai Lu. Archidiff: Interactive design of 3d architectural forms generated from a single image. *Computers in Industry*, 168:104275, 2025. 4, 5

[64] 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. 4

[65] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024. 2, 4, 5, 6, 7

[66] Saining Zhang, Baijun Ye, Xiaoxue Chen, Yuantao Chen, Zongzheng Zhang, Cheng Peng, Yongliang Shi, and Hao Zhao. Drone-assisted road gaussian splatting with cross-view uncertainty. *arXiv preprint arXiv:2408.15242*, 2024. 4, 5

[67] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 2, 4, 5