Appendix

https://itingtsai.github.io/syn_arch_2025/

A. Implementation Details

A.1. Generating Coarse Geometry

In this paper, we presented our method for generating building massing using random Constructive Solid Geometry (CSG) objects. Here, we provide additional details on the process of aggregating cuboids and the constraints applied based on the initial cuboid and the specific building type.

A.1.1 Determining the Building Type

We define three distinct building types, enabling users to specify the desired type for generation. If no building type is specified, the system will randomly select one. When a specific building type is chosen, the corresponding keyword (e.g., "low-rise building") is appended to the user-input prompt to guide the diffusion model.

A.1.2 Creating the Initial Cuboid

To define the initial cuboid, we start by randomly selecting its base dimensions, adjustable based on the desired cuboid size. Next, we determine the height range based on the selected building type, using a height-to-width ratio, where the width corresponds to the base dimension of the building.

For low-rise buildings, the height is set to 1 to 2 times the width of the base. For mid-rise buildings, it ranges from 3 to 4 times the width, and for high-rise buildings, it falls between 5 and 6 times the width. Finally, a height is uniformly sampled from this range, completing the initial cuboid generation.

A.1.3 Attaching Additional Cuboids

The dimensions of the attached cuboids are defined relative to the base cuboid they are connected to, ensuring that the overall 3D massing model stays within the height limits of the designated building type.

The building massing guidelines set specific dimensional constraints to ensure balanced proportions in low-rise, mid-rise, and high-rise structures. For low-rise buildings that expand horizontally, new cuboids must have heights between 2 units and the smaller of 3 units or half the base cuboid's height, with widths and lengths ranging from 2 units up to twice the base dimensions. High-rise buildings focus on vertical growth, requiring new cuboids to have widths and lengths between 2 units and the smaller of 3 units or half the base dimensions, and heights from 2 units up to the full height of the base cuboid. Mid-rise buildings adopt dimensions that fall between these two extremes, maintaining a balanced massing model across all building types.

After generating an additional cuboid, it is attached to the existing structure. If the model already includes multiple cuboids, a random one is selected as the attachment point. Users can control the number of cuboids to be created and added, allowing them to customize the building's massing model according to their requirements.

To ensure contact between the new cuboid and the target, we randomly select a side of the target cuboid and compute a position vector to align the new cuboid

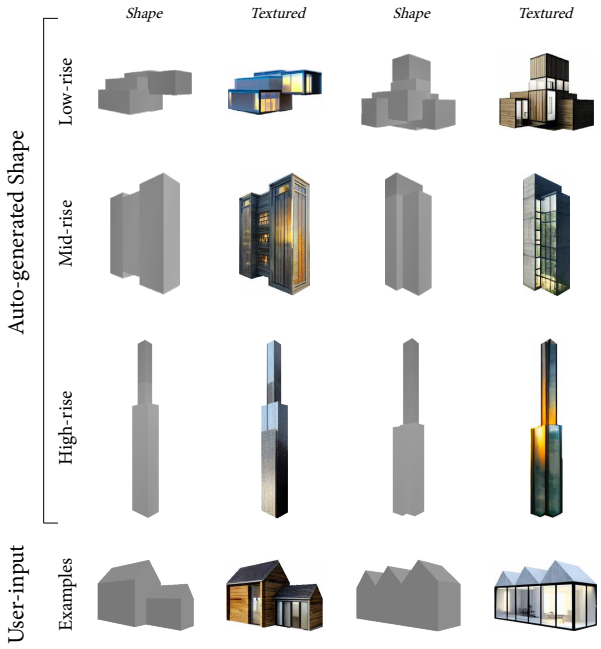


Figure 1: **Shape and Texture Generation Results.** The first and third columns show different CSG results, while the second and fourth display their textured versions. The bottom row presents user-input shapes and their generated outputs.

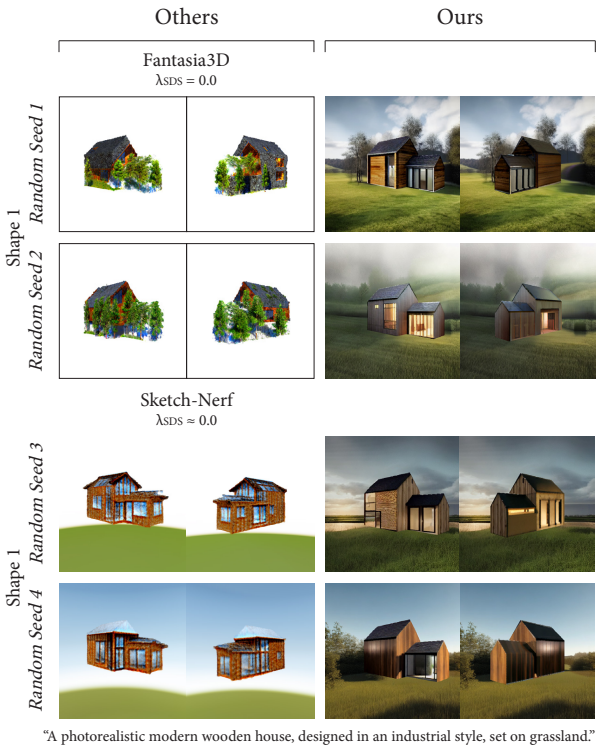


Figure 2: **Texture Synthesis Comparison.** Our method (right) generates diverse design variations from the same base geometry (Shape 1) as depicted in the bottom left of Fig. 1, using identical text prompts while preserving stylistic consistency. In contrast, existing approaches [2, 5] (left) yield repetitive, low-quality results, frequently blending backgrounds into facades or rendering portions of the geometry invisible by merging with elements like the sky and trees. $\lambda_{SDS} \approx 0$ for these techniques implies that the initial geometry is preserved.

precisely. For example, when placing a cuboid on top of another, the bottom of the new cuboid aligns with the top of the target. To introduce variation, we add randomness by shifting the new cuboid along the plane of attachment, allowing for varied configurations while maintaining proper alignment. The results, along with their textures, are illustrated in Fig. 1. Users also have the option to import custom meshes, as demonstrated in the bottom row of Fig. 1.

A.2. Synthesizing Textures

In this section, we describe the detailed methodologies that enhance the texture quality and consistency of our 3D massing models. These techniques, integrated into our workflow, are essential for achieving high-quality results and demonstrate the comprehensive approach we employ in our 3D synthesis process.

A.2.1 Auto-adjusting FOV

To handle the varying dimensions of our 3D massing models, we employ an automatic adjustment mechanism for the camera's Field of View (FOV). This mechanism ensures the complete visibility of the model within the scene by dynamically adjusting the FOV parameters. We continuously monitor the image boundaries and expand the FOV until all the pixels at the edges align with the background color, thereby guaranteeing the entire model is fully captured within the rendered image.

A.2.2 Coloring the Mesh

We extract the dominant color from the initial ControlNet [11] output and apply it to the 3D model to ensure consistent texture tones during inpainting. Refer to Sec. B.2 for additional information.

A.2.3 Inpainting with Different Seeds

Our approach provides extensive design flexibility, allowing for variations even with the same shape and prompt, as demonstrated in Fig. 2. Diverse textures can be applied to the same geometric model to create stylistic variation (Fig. 9). Moreover, beginning with an initial render, our methods can generate multiple facade designs for the same building by employing different random seeds during the facade-by-facade inpainting process (Fig. 10). This capability allows users to easily explore a wide range of design possibilities.

A.2.4 Inpainting Check and Re-inpaint

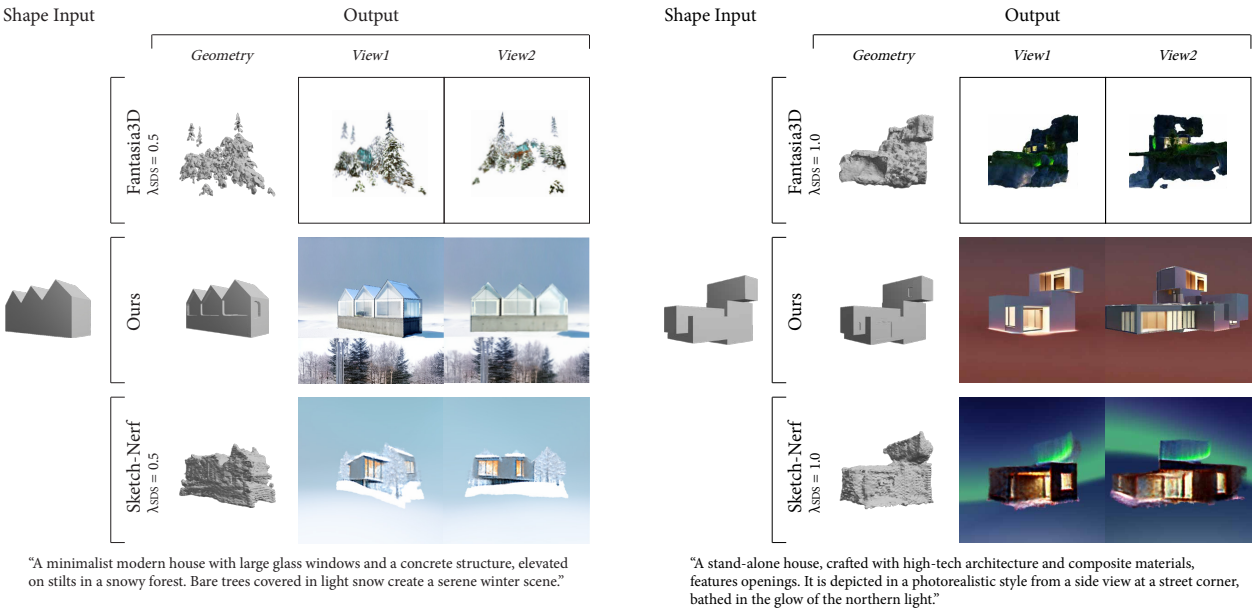
We ensure accurate 2D-to-3D texture mapping by checking contour alignment and white pixel count; if misalignment or excess white pixels are found, inpainting is repeated. Details are provided in Sec. B.5.

A.2.5 Inpainting the Background

The background for our model is derived from a static image generated by the initial ControlNet output. We isolate the building area and use "dilation" as a keyword to inpaint a complete background. This inpainted background is then applied to all rotated views of our 3D model, ensuring visual coherence across all perspectives.

A.3. User-defined Hyperparameters

In this project, we utilized several key user-defined hyperparameters to let users customize and optimize the architectural design generation process. Below is a detailed description of these parameters and their significance:



(a) Auto-generated Shape

(b) User-input Shape

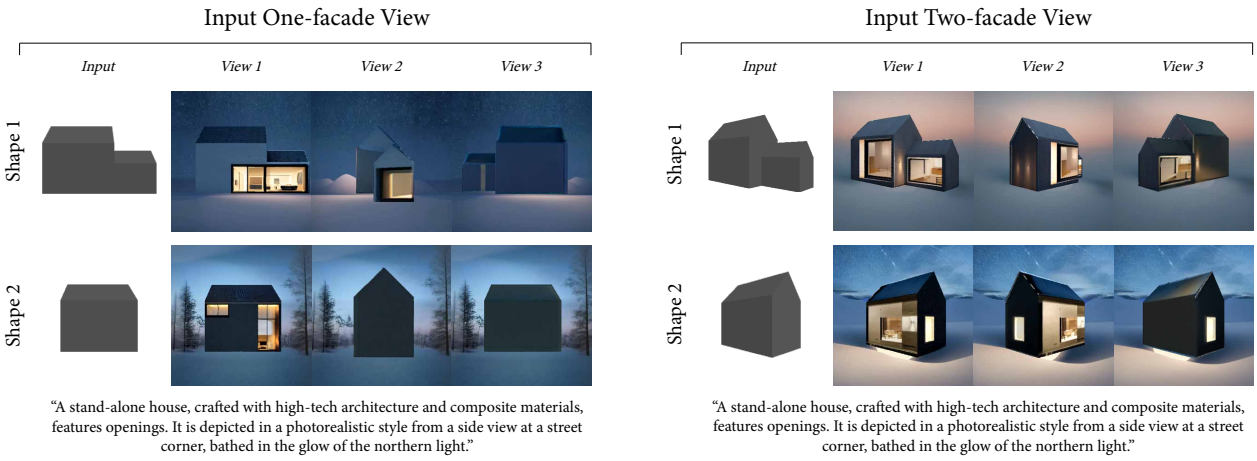
Figure 3: **Shape-guided Synthesis Comparison.** We apply open-vocabulary recognition to automatically detect specific elements using keywords such as "windows" in examples (a) and (b). This approach allows us to precisely edit details, such as pushing windows inward where needed, compared to the global edits seen in previous work (top and bottom rows in (a) and (b)) [2, 5].

- **User-defined Prompt:** Users can input a text prompt to define the architectural design they wish to generate, enabling a customizable and user-centric process.
- **Building Type Specification:** Users can input numerical values to indicate their preference for an auto-generated building and specify the type of building (low-rise, mid-rise, or high-rise). This structured approach allows for the generation of various building typologies, ensuring that the resulting designs align with the user's intended use cases.
- **Open-vocabulary Object Detection Keywords & Threshold:** Keywords allow users to specify elements they wish to adjust on the building facade. While "windows" is the default keyword, it can be changed to other architectural elements such as "doors" or "balcony," or a combination of these keywords, as needed. This flexibility is essential for adapting specific design features and enhancing the realism and functionality of the generated buildings (Fig. 3).

  The optimal threshold varies depending on the texture characteristics of each building, requiring adjustments to achieve the best results in geometry modification.
- **Image Dimensions:** Users can define the width and height of the rendered image, allowing control over the resolution and aspect ratio of the output. We default to 512×512, as this is the most commonly accepted dimension for diffusion models.
- **Starting Angle:** Users can define the initial viewing angle of the model as the primary render view of the design. For example, setting the starting angle to 45 degrees establishes this view as the reference point for subsequent 3D rotation and inpainting. This primary render view is ensured to be clear and well-defined, as the 3D texture mapping for this view is directly derived from the initial 2D image generated by the diffusion model.
- **ControlNet:** We employed the official ControlNet [11] with SD1.5 model to control stable diffusion [7] using Midas depth estimation [1], which processes the full 512×512 depth map.

B. Ablation Study

We conduct an ablation study to validate the key design choices within our 3D synthesis framework, systematically comparing the results of the complete model against those without specific components.



(a) Input One-facade View

(b) Input Two-facade View

Figure 4: **Input Multi-facade View.** (a) In these examples, the inpainting model learns from the initial one-facade view but typically generates solid facades lacking details. This underscores the importance of a more informative initial view for realistic, interesting, and style-consistent results. (b) The input image shows two facades of the 3D coarse model from an angled perspective. This view, used as input for the ControlNet [11], provides a more comprehensive design by revealing additional lighting, textures, and window elements on the adjacent facade.

B.1. Two-facade View

We select an angle of the 3D coarse model that prominently displays two facades and render an image from this perspective. This rendered image is then used as input for the ControlNet model to generate the initial view. By capturing multiple facades, the ControlNet output provides a more comprehensive building design compared to single-view generation. For instance, as illustrated in Fig. 4 (b), the angled view reveals additional lighting, texture, and window elements on the adjacent facade that are absent in the initial single-facade view. While the inpainting model can learn from the initial one-facade view to produce similar textures, as shown in Fig. 4 (a), it typically generates images with solid facades lacking these details. Consequently, the richer informational content from the initial view enhances the inpainting model's ability to generate more realistic, design-wise interesting, and style-consistent results.

B.2. Mesh Color

We observed that the inpainting model adapts to the masked region's color distribution. By extracting the dominant color from the initial building view produced by ControlNet and applying it to the 3D coarse model at the start, we ensure consistent tones during the inpainting process, resulting in a harmonious final texture (Fig. 5 middle).

Our experiments reveal that the inpainting model learns not only from the visible unmasked areas but also from the regions underneath the mask. A significant portion of the generated facades exhibited noticeable color influences from the underlying mesh color render (Fig. 5 top). This issue was more pronounced with brighter colors, increasing the likelihood of color influence. When the coarse model's color diverges significantly from the building's dominant tone, the resulting facades appear unnatural and display inconsistent textures. This underscores the importance of ensuring color consistency between the coarse model and the target building texture to achieve realistic and cohesive outputs.

B.3. 2×2 Visual Prompt

With our unique 2×2 visual prompt as input, the inpainting model learns and applies textures to the coarse 3D model, ensuring clean, stylistically consistent, and 3D-aware results.

We observe that inpainting a building facade using a single image prompt produces consistent textures but lacks sufficient window and door openings compared to the other sides of the building (Fig. 5 bottom). The consistent texture and color are achieved due to the dominant mesh color applied beneath the masked region. However, the limited in-context information provided to the inpainting model results in fewer openings and a less visually interesting facade design.

To enable the inpainting model to perform in-context learning for better results, we propose using a 2×2 visual prompt as input. Our method integrates the initial output from ControlNet [11] and the front views of two

Figure 5: **Mesh Color, 2×2 Visual Prompt, and Single Image Prompt Comparison. Top Row:** Our experiments show that the inpainting model learns from both visible unmasked areas and regions beneath the mask. Significant color divergence between the coarse model and the building's dominant tone leads to unnatural and inconsistent textures. **Top & Middle:** We use a 2×2 visual prompt to enable the inpainting model to perform in-context learning. Our method combines the initial ControlNet [11] output with two adjacent facade views and an untextured view into a 2×2 layout. **Middle Row:** Our method ensures clean, style-consistent, and 3D-aware results by using a 2×2 visual prompt together with the assigned mesh color. **Bottom Row:** Inpainting a facade using a single image prompt often produces insufficient windows and doors because of the lack of contextual information.

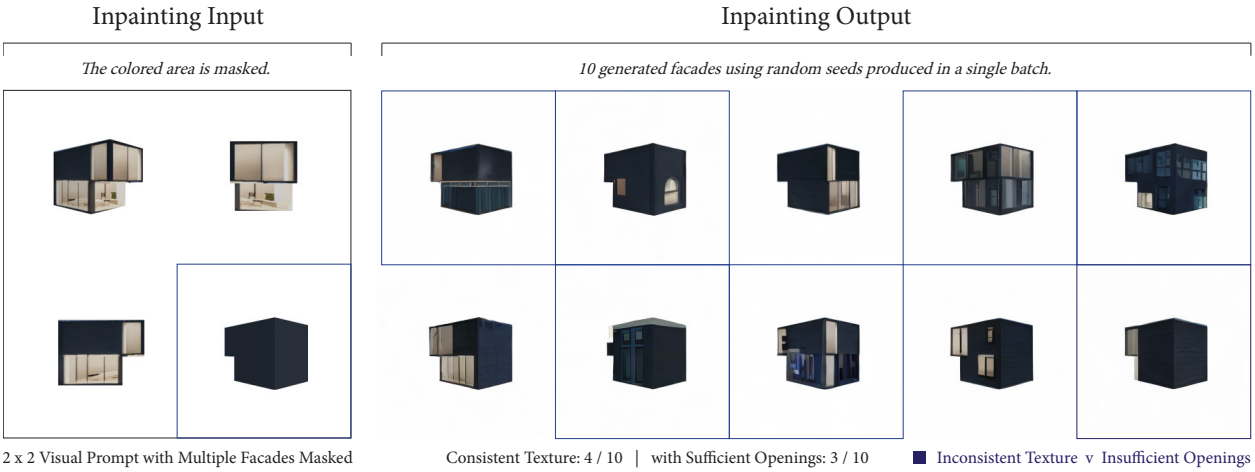


Figure 6: **Multi-facade Inpainting.** We conducted an experiment where multiple facades were inpainted simultaneously. In this setup, the model was rotated 180 degrees to inpaint the back view of the building. Most results exhibited a different texture style compared to the main building views and lacked sufficient window and door openings on the facades.

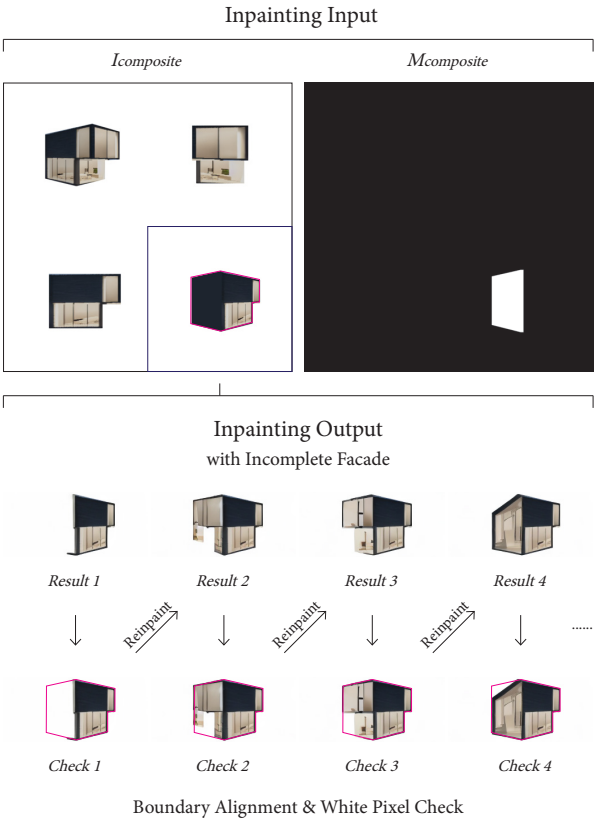


Figure 7: **Inpainting Check and Re-inpaint.** We compare boundaries of inpainted and unpainted models, repeating until alignment and inpainting are complete.

facades along with the untextured view into a 2×2 visual prompt layout. As demonstrated in Fig. 5, the middle row shows inpainted facades that achieve both consistent texture and sufficient openings, resulting in an improved design.

B.4. Facade-by-facade Inpainting

We introduce a novel facade-by-facade approach for generating realistic building textures using a 2×2 visual prompt. Example results are displayed in Fig. 5 (middle).

As shown in Fig. 6, inpainting two neighboring facades simultaneously may result in inconsistently styled facades. This occurs because the inpainting model learns from the adjacent unmasked areas. Our facade-by-facade approach addresses this issue effectively, allowing the inpainting model to learn from the neighboring pixels of the closed-adjacent facade, resulting in more cohesive and consistent outcomes.

To generate realistic building textures, we need to further constrain the generation process. This facade-by-facade method constrains the model to operate within the specific geometric context, ensuring that textures are applied accurately. This approach also mitigates the multi-face Janus problem.

B.5. Inpainting Check and Re-inpaint

To ensure accurate texture mapping from 2D to 3D, we first verify that the filled-in image completely covers the masked area. We compare a render of the object's

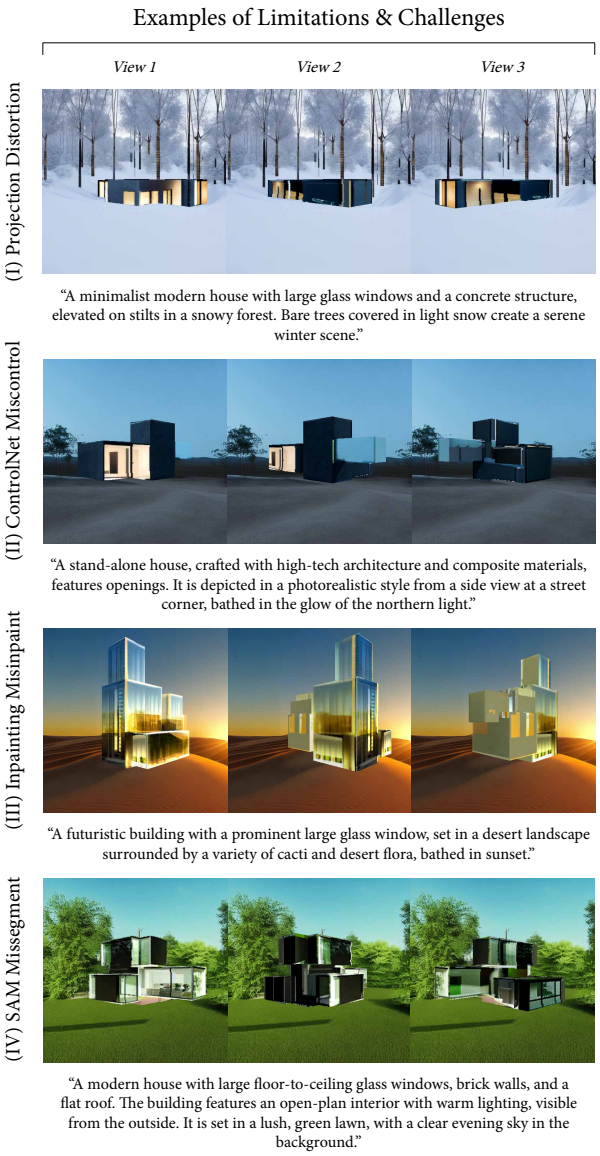


Figure 8: **Examples of Limitations and Challenges.** While our method performs well, challenges remain: (I) Vertex color mapping can distort textures in angled projections, especially in horizontally extensive buildings. (II) ControlNet [11] may miss 3D geometry details. (III) The inpainting model [7] can generate less realistic textures. (IV) Open-vocabulary object detection [6] and Segment Anything Model [3] need hyperparameter tuning to reduce misidentification.

boundary with a render of the original model's edges. Additionally, we count the white pixels within the boundary. If significant differences are detected or if there are too many white pixels, we repeat the process to ensure full coverage, as shown in Fig. 7.

C. Limitations and Challenges

Here we identified challenges that offer opportunities for future improvements.

First, while vertex color-based texture mapping is efficient, it may cause distortions in angled projections, especially in long, horizontally extended buildings (Fig. 8 (I)).

Second, the quality of our results is influenced by the ControlNet [11] and inpainting model, with occasional issues of the ControlNet model losing control and ignoring 3D geometry (Fig. 8 (II)). There are also instances where the inpainting model produces less convincing textures (Fig. 8 (III)).

Last, Open-vocabulary Object Detection [6] and the Segment Anything Model [3] require hyperparameter adjustments to improve detection accuracy and minimize misidentification of elements (Fig. 8 (IV)), which can be affected by variations in lighting and texture.

D. Additional Comparisons

Finally, we provide additional qualitative comparisons with text-guided synthesis models, including Prolificdreamer [10], MVDream [8], Dreamfusion-Hifa [12], Magic3D [4], and TextMesh [9], as shown in Figs. 11 to 19. These examples are selected to demonstrate the capabilities of our approach under various text prompts and to highlight the improvements achieved over existing techniques.

In conclusion, our method delivers higher-quality 3D synthesis for architectural design, resulting in more realistic and visually compelling outcomes.

References

[1] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023. 3

[2] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22246–22256, October 2023. 2, 3

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 7

[4] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 7, 13, 14, 15

[5] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022. 2, 3

[6] Matthias Minderer, Alexey A. Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. *ArXiv*, abs/2205.06230, 2022. 7

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 3, 7

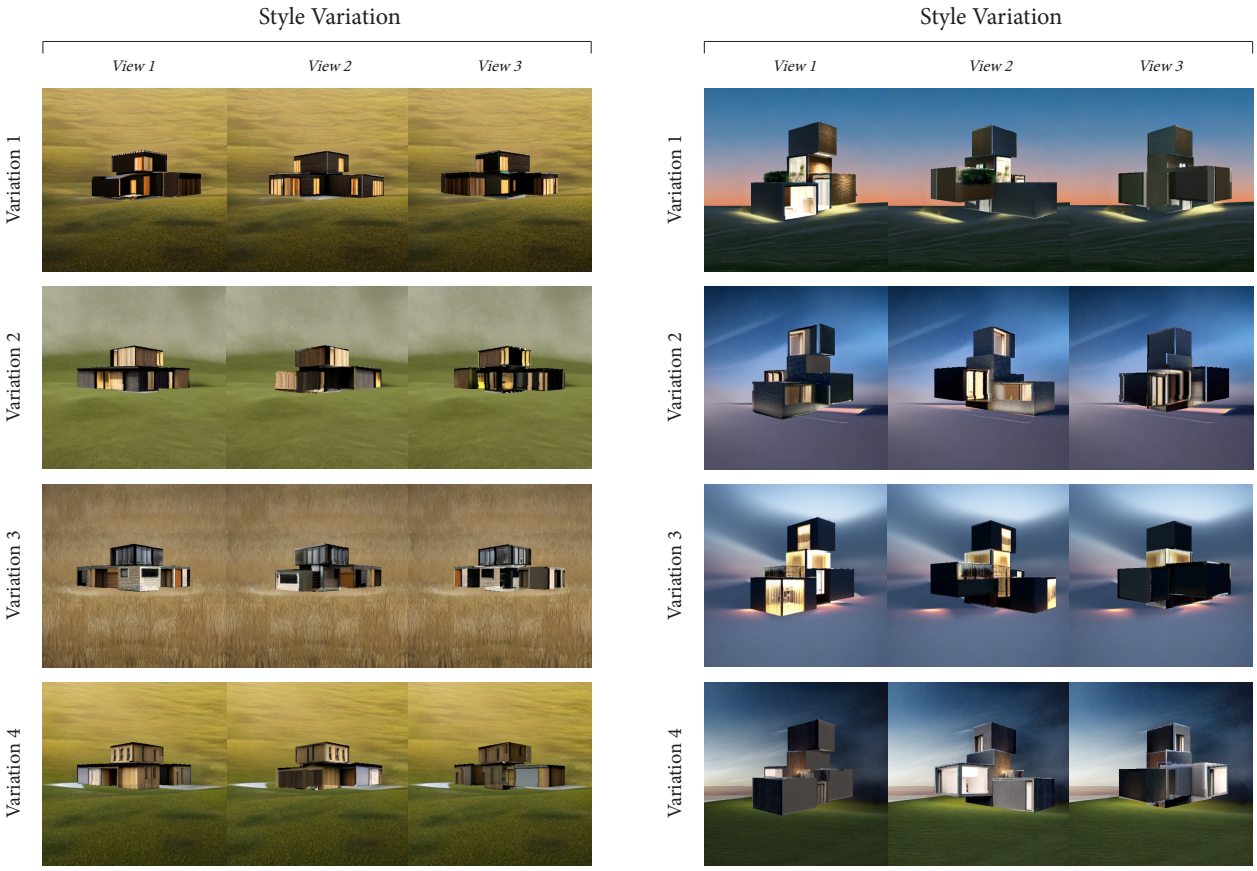
[8] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 7, 12, 14, 15

[9] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. In *International conference on 3D vision (3DV)*, 2024. 7, 11, 12, 13

[10] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 7, 11, 12, 13

[11] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2, 3, 4, 5, 7

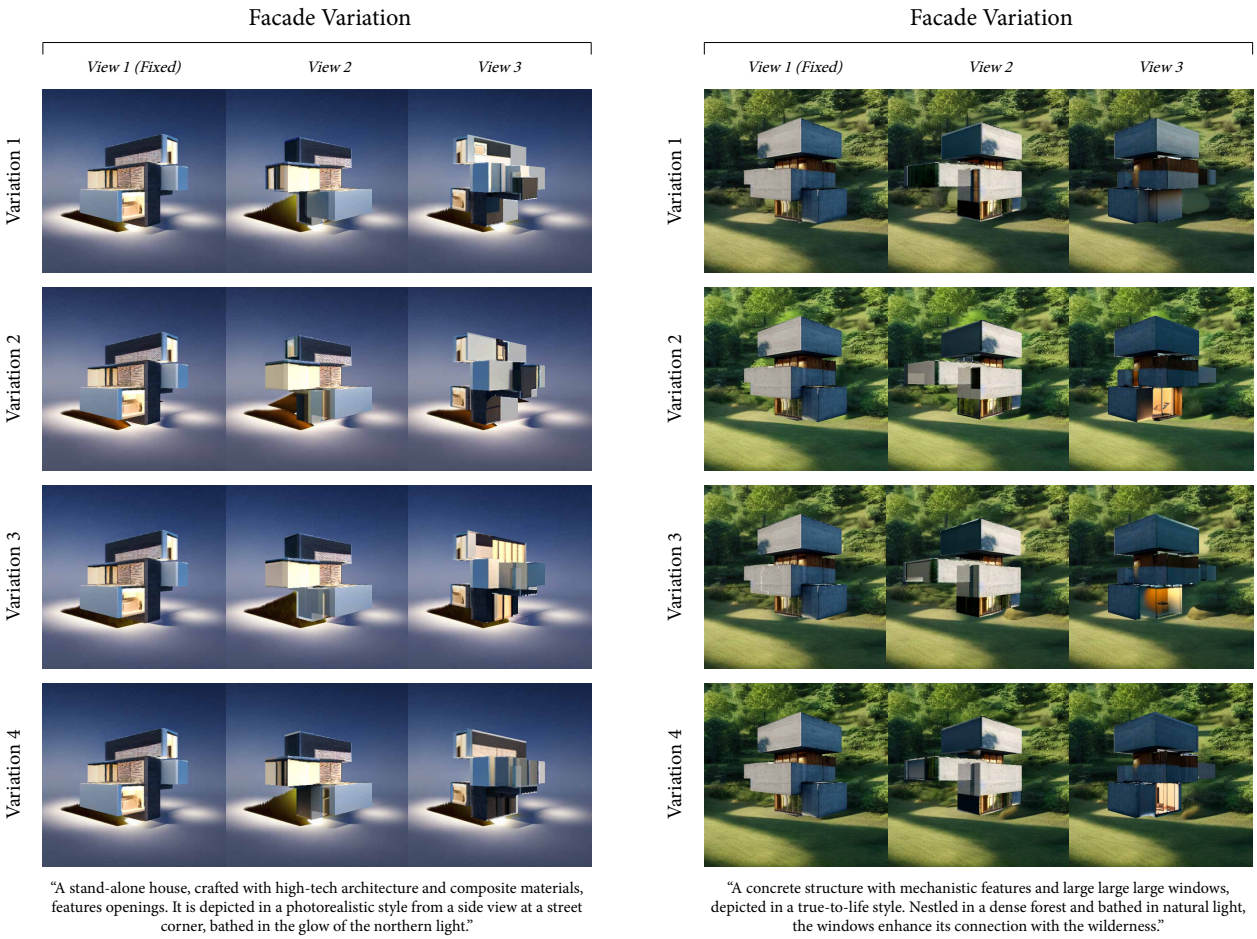
[12] Junzhe Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance, 2023. 7, 11, 14, 15



(a) Shape 1

(b) Shape 2

Figure 9: **Style Variation.** Our method can generate diverse facade textures and styles for buildings with the same shape. By leveraging the inpainting model and facade-by-facade approach, we produce a wide range of stylistic variations that maintain structural consistency.



(a) Shape 3

(b) Shape 4

Figure 10: **Facade Variation.** Here, we demonstrate that our method can generate diverse designs for each facade of the same building using a facade-by-facade approach. These designs maintain a consistent style while offering unique variations. This capability is particularly valuable in the early design stages, as it enables the exploration of multiple architectural options to support various design concepts and goals.

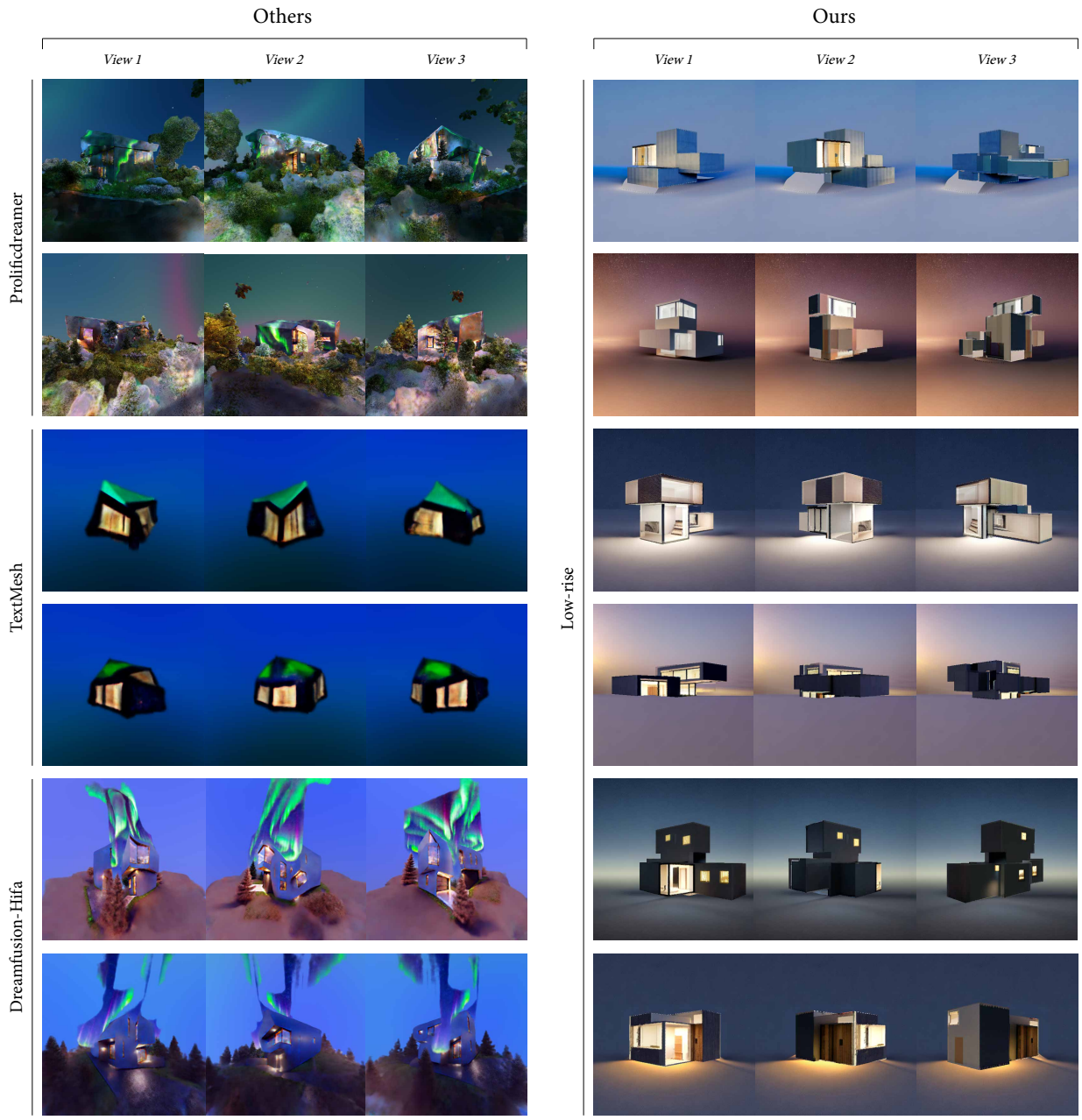


Figure 11: **Additional Comparisons between Ours and the Baselines: Prolificdreamer [10], TextMesh [9], and Dreamfusion-Hifa [12].**

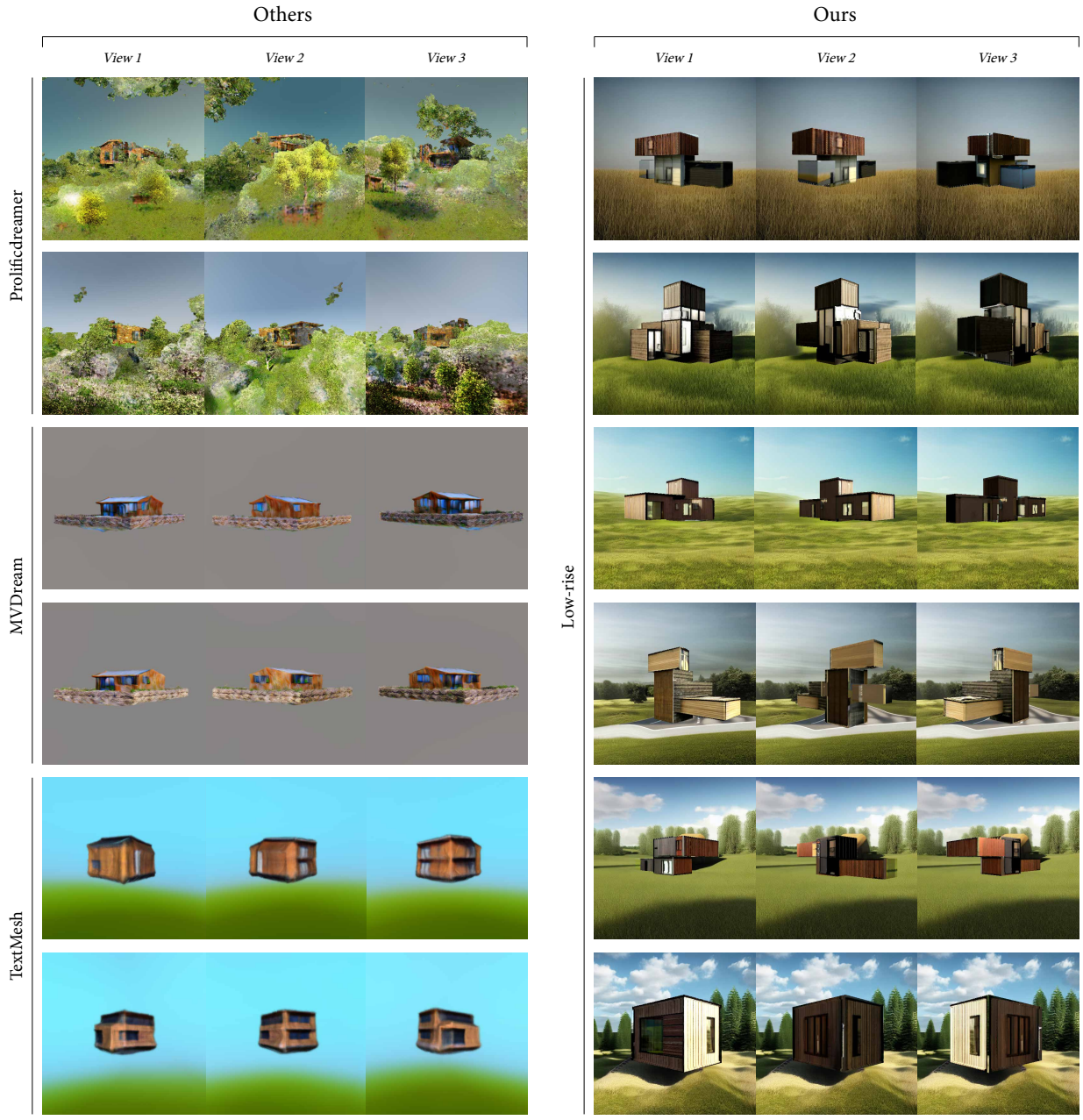


Figure 12: **Additional Comparisons between Ours and the Baselines: Prolificdreamer [10], MVDream [8], and TextMesh [9].**

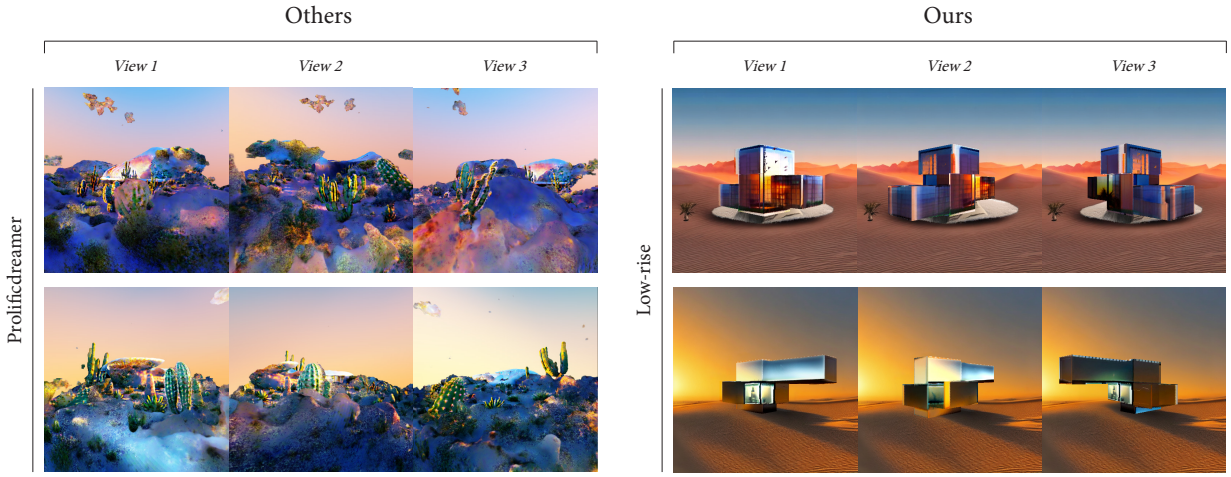


Figure 13: **Additional Comparisons between Ours and the Baselines: Prolificdreamer [10].**

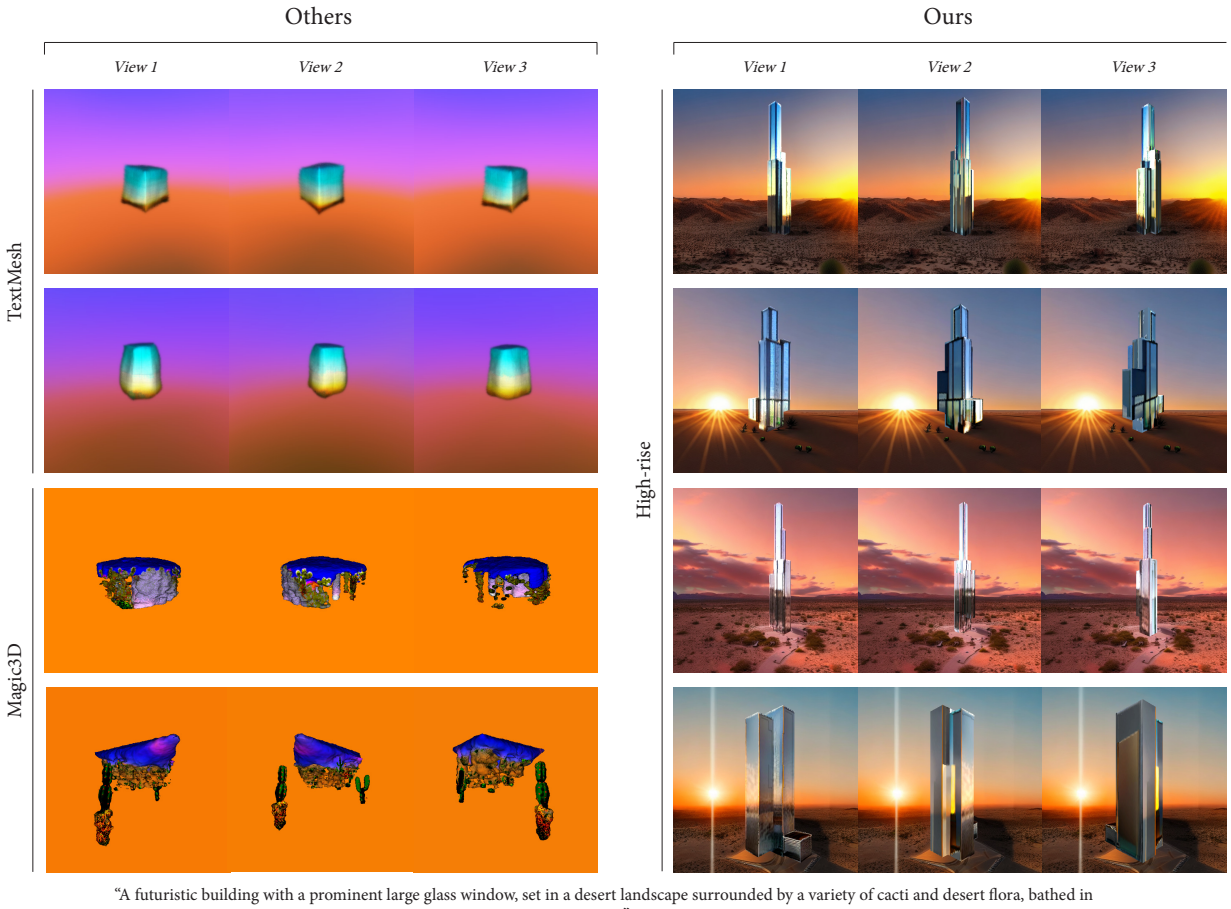
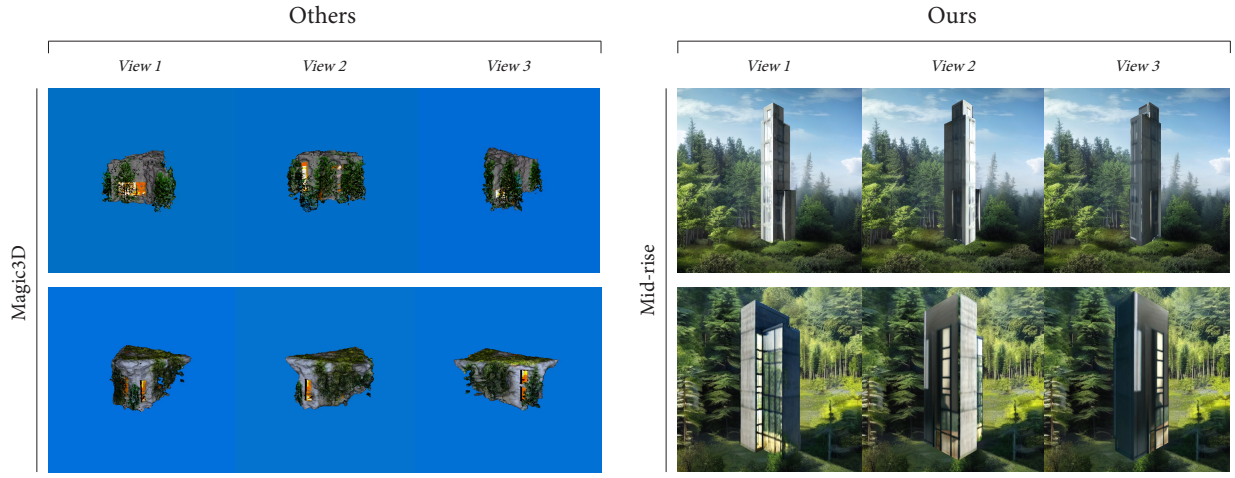


Figure 14: **Additional Comparisons between Ours and the Baselines: TextMesh [9] & Magic3D [4].**

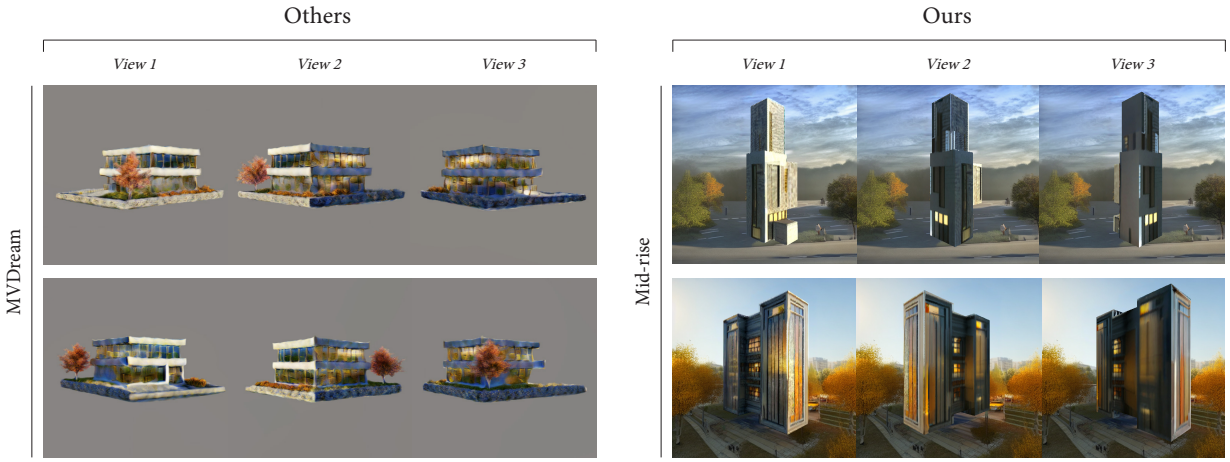


"A concrete structure with mechanistic features and large large large windows, depicted in a true-to-life style. Nestled in a dense forest and bathed in natural light, the windows enhance its connection with the wilderness."

Figure 15: **Additional Comparisons between Ours and the Baselines: Magic3D [4].**

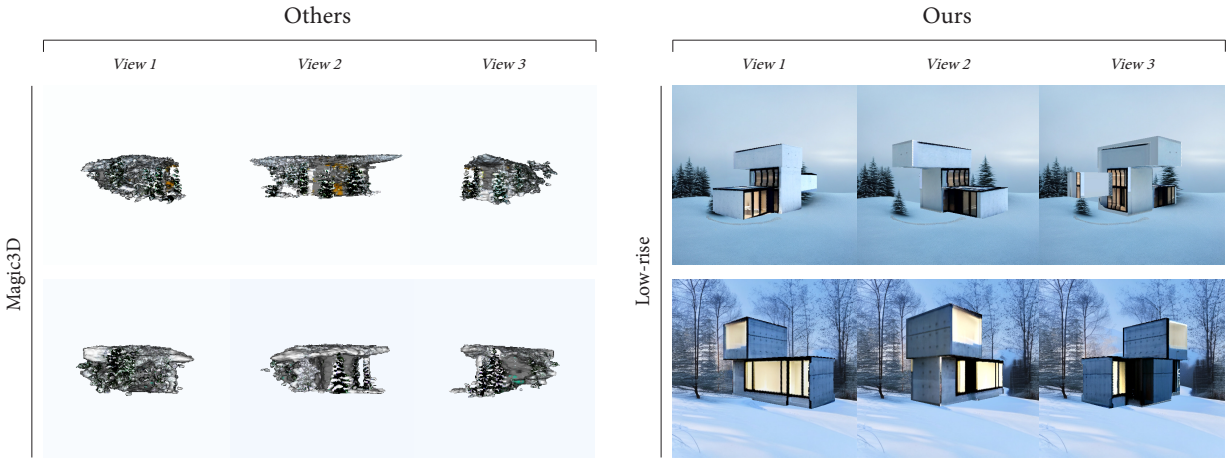
"A concrete structure with mechanistic features and large large large windows, depicted in a true-to-life style. Nestled in a dense forest and bathed in natural light, the windows enhance its connection with the wilderness."

Figure 16: **Additional Comparisons Between Ours and the Baselines: MVDream [8] & Dreamfusion-Hifa [12].**



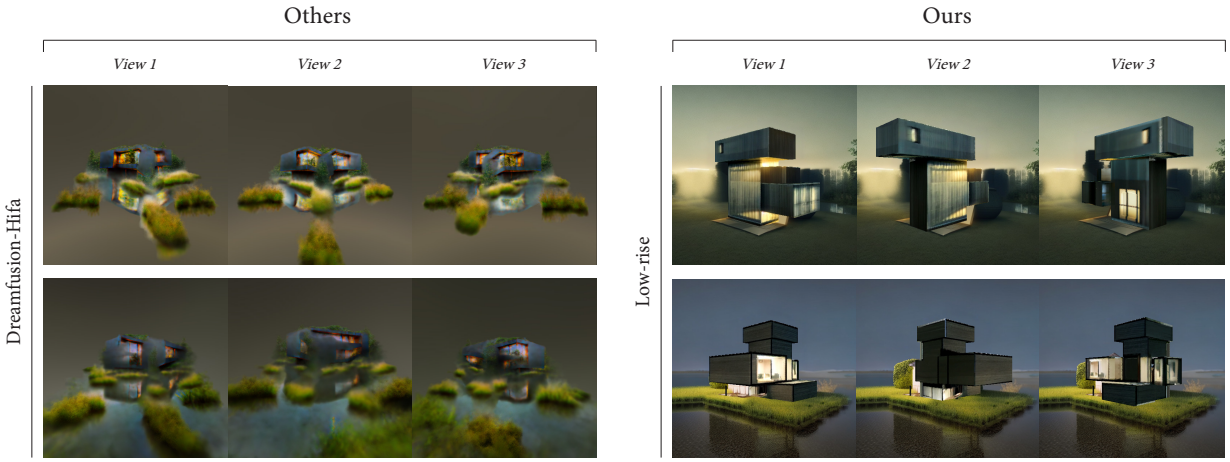
"A stand-alone building, crafted with organic architecture and ceramic detailing, features large windows. It is depicted in a photorealistic style from a two-point perspective, nestled in an autumn setting and bathed in direct sunlight."

Figure 17: **Additional Comparisons between Ours and the Baselines: MVDream [8].**



"A minimalist modern house with large glass windows and a concrete structure, elevated on stilts in a snowy forest. Bare trees covered in light snow create a serene winter scene."

Figure 18: **Additional Comparisons between Ours and the Baselines: Magic3D [4].**



"A house showcasing vintage aesthetics is captured in a lifelike, two-point perspective with dynamic glass and moody lighting, nestled within a wetland setting."

Figure 19: **Additional Comparisons between Ours and the Baselines: Dreamfusion-Hifa [12].**