REVIVE3D: REtrieval-based Volumetric Infusion via Visual Editing

Hankyeol Lee Wooyeol Baek Seongdo Kim Jongyoo Kim[†]
Yonsei University
{guts4, wooyeol.baek, sdokim07, jy.kim}@yonsei.ac.kr

Figure 1. REVIVE3D. Generating high-fidelity 3D models from flat-colored images

Abstract

*We introduce REVIVE3D, a **retrieve-and-edit framework** for single image-to-3D generation. Our method is designed for **flat-colored images** such as cartoons and drawings with minimal shading. Instead of 2D preprocessing, REVIVE3D first retrieves a shape-aligned 3D prior and then edits it directly in a 3D latent space. The edit is guided by the visual difference between the input image and an aligned render of the retrieved prior. This direct 3D operation injects the missing volumetric cues and preserves the global structure of the shape. The method is **plug-and-play and requires no retraining**. It produces results in approximately 2 minutes using only 0.6 GB of memory on a single GPU. On the Art3D test set, REVIVE3D achieves state-of-the-art image-to-3D alignment and consistently reconstructs complete geometry with fine details, while also performing well on standard images. These results demonstrate that direct latent editing of a retrieved 3D prior is an effective and practical route to high-fidelity 3D from flat-colored images.*

1. Introduction

The creation of 3D assets from 2D images is crucial for various applications, including virtual reality, video games and animations. While manual modeling achieves high quality, it is a time-consuming and labor-intensive process that creates a significant bottleneck in content production. To address this challenge, numerous high-performance models for automatic single image-to-3D generation have been developed [11, 12, 17, 18, 27, 28, 33]. However, a common limitation of existing image-to-3D models is their inability to effectively handle flat-colored images such as cartoons, line drawings, and flat-shaded art. This creates a significant domain gap, as these images present several distinct challenges for photorealistically-trained models, including: (1) strong view-dependent contour lines often misinterpreted as texture, (2) flat-color regions that offer no shading cues for inferring volume, and (3) exaggerated or physically implausible geometries. As shown in Fig. 2, these factors cause state-of-the-art methods like Wonder3D [18] and Hunyuan3D-2.1 [12] to struggle with these inputs. Both methods fail to generate sufficient volume and com-

[2]Corresponding author.

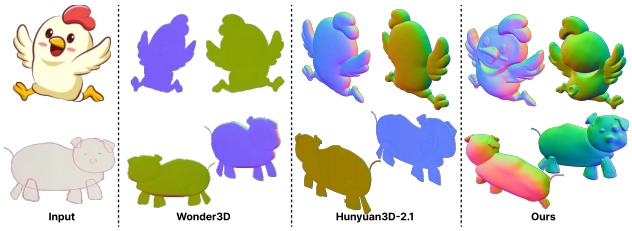

Figure 2. **Normal map comparison for 3D reconstruction from flat-colored image.** While state-of-the-art methods like Wonder3D [18] and Hunyuan3D-2.1 [12] yield results lacking volumetric completeness and fine-grained details, our method successfully reconstructs a complete and detailed 3D shape.

pletely miss fine-grained details such as the eyes, nose, and mouth, resulting in incomplete geometry.

Prior works have attempted to address this challenge by performing an explicit 2D-to-2D transformation to make the ambiguous input more compatible with standard generators. For example, Art3D [6] uses a VLM-generated caption to synthesize a new 2D proxy image based on depth and Canny edge conditions. DrawingSpinUp [35] takes a similar approach by first removing artistic contour lines that can confuse pre-trained models. Although these methods aim to inject 3D information, their reliance on this 2D preprocessing step is often insufficient, as the transformed images can still lack the consistency needed to generate a complete 3D shape.

Our approach bypasses such 2D transformations entirely. Instead of reconstructing a shape from ambiguous 2D cues, our framework is designed to edit a complete 3D reference until it matches the input's visual details. To achieve this, our REVIVE3D framework directly addresses the aforementioned challenges. First, by retrieving a similar 3D reference, we provide a robust geometric foundation that compensates for flat colors and exaggerated structures. Second, by editing this reference directly in its 3D latent space using visual difference, we resolve geometric ambiguities arising from flat colors and misleading contour lines. This process also allows us to incorporate fine details without the information loss inherent in 2D-based methods. Our resulting framework is training-free and highly efficient (0.6 GB memory, 2 mins/result), and as our evaluations show, it not only excels on ambiguous inputs but also generalizes to produce more detailed results on standard images.

2. Related Work

2.1. Stylized Image-to-3D Generation

Generating 3D assets from stylized single images like drawings or cartoons is a unique and significant challenge. These inputs often lack the photorealistic cues that standard image-to-3D models rely on, leading to common failures such as collapsed geometry. Prior works have attempted to bridge this domain gap through two main strategies.

The first strategy involves preprocessing the 2D input to make it more compatible with standard generators. These methods explicitly transform the stylized image, for instance by adding synthetic depth and edge cues (Art3D [6]), or by removing artistic contour lines that can cause ambiguity (DrawingSpinUp [35], PAniC-3D [4]). The second strategy, particularly for characters, involves fitting a parametric 3D model or canonicalizing the input. Methods like RaBit [19] and MagiCartoon [25] learn a controllable character model, while CharacterGen [23] aligns the input to a neutral A-pose before reconstruction.

While these approaches can improve results, they are often limited. Preprocessing in 2D can lead to information loss, and fitting to a generic template may not capture the unique style of the input. In contrast, our retrieve-and-edit method provides a high-quality 3D prior for each input, avoiding these limitations by editing directly in 3D.

2.2. Retrieval-Based 3D Generation

Retrieval-augmented generation, which leverages an external 3D model as a structural prior, is a promising direction for improving 3D consistency. However, state-of-the-art methods still process this 3D information through 2D-based processes. For instance, Retrieval-Augmented Score Distillation [24] uses the retrieved asset to guide the gradients of a 2D diffusion model during optimization, while Phidias [26] converts the 3D reference into 2D conditional maps to guide a multi-view image generator. In both approaches, the 3D reference acts as an indirect guide for a 2D synthesis task, which results in an incomplete transfer of its geometric information. In contrast, our pipeline operates more directly by using the retrieved reference as the primary 3D prior and explicitly deforming it in its latent space. This direct 3D operation bypasses the limitations of 2D-based guidance, preserving the geometric consistency of the prior while accurately incorporating the input's details.

2.3. 3D Shape Deformation

Once a 3D reference is retrieved, the core challenge becomes how to directly deform this prior to match the input image, avoiding the limitations of indirect 2D synthesis. Prior work has explored this deformation in two main domains: direct mesh deformation and latent space editing.

A common strategy is direct mesh deformation, which edits vertices using either CLIP-similarity objectives [5, 20, 21] or Jacobian-based guidance from 2D diffusion priors [1, 8, 14]. However, these techniques are unsuitable for flat-colored images. Similar to the generation methods discussed previously, their reliance on indirect 2D signals like text prompts and novel-view synthesis is often insufficient and leads to distorted geometry.

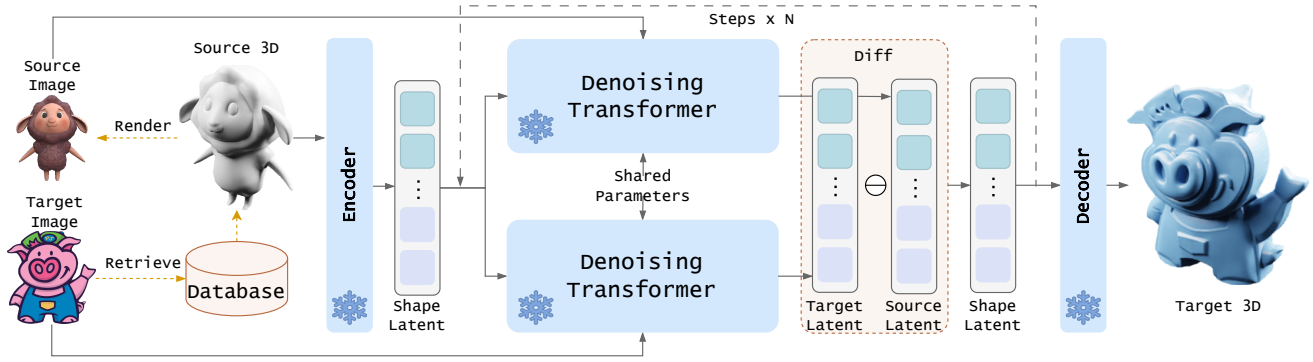

Figure 3. **Retrieve-and-edit pipeline.** We first retrieve a Source 3D model that is similar to the input Target Image. The model is then edited using the visual difference between the Target Image and a rendered Source Image, allowing it to reflect the target's fine details while preserving its global structure.

To better preserve 3D structure, an alternative is to perform edits in a latent space. Prior works like SHAP-EDITOR and Michelangelo [2, 32] have shown this is effective, typically by learning to align latents with text or image prompts. However, the key limitation is that while these methods align the latent space to inject 3D information, the alignment target is too generic. It lacks the specific geometric information required to precisely reconstruct the unique structure of the input image. In contrast, our training-free approach edits directly on the latent space of a retrieved reference, using the visual difference between the input and reference to preserve instance-specific geometry.

3. Method

To address the flat-colored images, our approach for 3D reconstruction is a retrieve-and-edit framework. We first retrieve a 3D shape from a database to serve as a structural prior, using the input image as a query (Sec. 3.2). We then edit this reference shape in its latent space, guided by the visual difference between a render of the 3D shape and the target image (Sec. 3.3). The method thereby preserves the reference's 3D structure while accurately capturing the details and style of the input. Fig. 3 provides an overview of our pipeline.

3.1. Background

3D Latent Representation. To efficiently generate complex 3D shapes, many recent models [3, 30–33] first compress high-dimensional geometric data, such as a point cloud X, into a compact latent representation $\mathbf{z}_0 = E_{\text{pc}}(X)$.

To achieve a high-fidelity latent that captures fine geometric details, our method employs the ShapeVAE from Hunyuan3D-2.0 [33], which uses a dual-sampling strategy. This combines a uniformly sampled point cloud for overall structure with an importance-sampled one that concentrates on complex regions like edges and corners. This richer representation provides a robust foundation for our editing process.

Difference-based Latent Editing. A powerful method for latent space editing is to guide a generative process using the difference between conditional predictions [9, 16]. Given a denoising model $\Phi(\mathbf{z}_t, t \mid c)$, the update direction Δ_t is computed as the difference between its outputs conditioned on a target (c_{tar}) and a source (c_{src}):

$$\Delta_t = \Phi(\mathbf{z}_t, t \mid c_{\text{tar}}) - \Phi(\mathbf{z}_t, t \mid c_{\text{src}}) \tag{1}$$

Iteratively updating a latent along this direction transforms a shape toward the target state by capturing their difference while preserving shared global information. This principle has proven highly effective in the 2D domain for tasks like text-guided image editing. However, its application to direct 3D shape modification remains largely unexplored. Therefore, a key contribution of our work is to adapt this powerful 2D editing approach to the 3D latent space, using visual differences between images to guide the modification process.

3.2. Image-Based 3D Reference Retrieval

Our approach to handling flat-colored images is to provide them with instance-specific geometric cues by retrieving a structurally similar 3D reference (hereafter, the source 3D). We retrieve this source 3D from a database of 40K objects from Objaverse [7], selecting the asset with the highest cross-modal similarity to ensure it is a relevant prior. Following the methodology of Phidias [26], we measure this by computing the cosine similarity between the input image's feature embedding (from OpenCLIP [13]) and the pre-computed 3D embeddings of all assets in the database (from Uni3D [34]). The selected asset, M_{ref} , thus serves as the initial geometric prior for the editing stage.

Figure 4. **Visualization of the 3D reference retrieval process.** Given a Target Image (left), we retrieve a similar 3D shape (Source 3D, right) and render it from an aligned viewpoint to generate the Source Image (middle), which serves as the starting point for our editing stage.

The editing stage is guided by the visual difference between the input image and a 2D rendering of the source 3D (M_{ref}). For this difference to be meaningful, the source 3D must be rendered from a viewpoint that is precisely aligned with the input. We therefore produce the source image, I_{src}, by rendering the source 3D from a manually aligned viewpoint (see Fig. 4). This carefully prepared source image then provides the initial visual condition for our shape modification process.

3.3. Latent Space Editing via Visual Difference

With the source 3D (M_{ref}) and its rendered image (I_{src}) prepared, we can now proceed to the core of our method: editing the shape in its latent space to match the target image(I_{tar}). To do this, we first obtain a comprehensive initial latent representation using the Hunyuan3D-2.0 [33] Shape-VAE encoder, E_{pc}. This encoder applies a dual-sampling strategy to the source 3D, processing both a uniformly sampled and an importance sampled point cloud. The resulting latent, $\mathbf{z}_0^{\text{src}}$, thus faithfully captures the complete geometry of the source 3D. This is because while uniform samples provide the overall structure, the importance samples offer more complete information for complex regions like edges and corners, enabling the latent to better represent intricate details. This detail-rich latent serves as the starting point for our editing state, initialized as $\mathbf{Z}_0 = \mathbf{z}_0^{\text{src}}$.

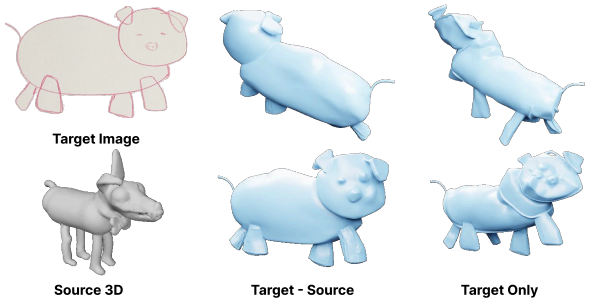

Figure 5. **Justification for Difference-Based Editing.** A comparison between a naive Target Only approach and our difference-based method (Target - Source). The target-only guidance yields a distorted shape that fails to capture the target's features, while our method produces a high-fidelity result.

Our editing process uses the visual difference between the source and target images to preserve the source's global structure while incorporating the target's fine-grained details. Computing this difference requires first extracting rich visual features from each image that capture both high-level semantics and fine-grained details. For this purpose, we employ the powerful pre-trained DINOv2-giant [22] model as our vision encoder ($\mathcal{E}_{\text{vision}}$). The resulting feature vectors serve as our conditioning vectors: $c_{\text{src}} = \mathcal{E}_{\text{vision}}(I_{\text{src}})$ and $c_{\text{tar}} = \mathcal{E}_{\text{vision}}(I_{\text{tar}})$. These conditions, containing the essential visual information, then guide the denoising process of our generative model, Φ, which we implement using the Hunyuan3D-2.0 [33] DiT backbone.

We iteratively refine the latent state $\mathbf{Z}_t$ by progressively injecting the visual information from the target image at each step. A naive approach would be to guide this process using only the target condition (c_{tar}). However, as shown in Fig.5 (Target Only), this method is often insufficient. The model's strong geometric priors can lead to a distorted shape that incorrectly blends features from both the source and the target, failing to faithfully capture the intended geometry. To enable more accurate edits, we instead guide the process using the difference between the source- and target-conditioned predictions. As seen in Fig.5 (Target - Source), this approach effectively isolates the desired changes from the shared information, resulting in a high-fidelity edit that successfully modifies the shape's specific features while preserving overall structural coherence. To implement this, we adapt the general update direction from Eq. (1) with a regularization mechanism, formulating the final update as:

$$\Delta_t = \Phi(\mathbf{Z}_t + \widetilde{\mathbf{Z}}_t^{\text{src}} - \mathbf{z}_0^{\text{src}}, t \mid c_{\text{tar}}) - \Phi(\widetilde{\mathbf{Z}}_t^{\text{src}}, t \mid c_{\text{src}}) \quad (2)$$

where $\widetilde{\mathbf{Z}}_t^{\text{src}}$ is the source latent noised to timestep t. The input to the first term, $\mathbf{Z}_t + \widetilde{\mathbf{Z}}_t^{\text{src}} - \mathbf{z}_0^{\text{src}}$, anchors the current

editing state to the source's noise trajectory, acting as a geometric anchor. This anchoring mechanism is particularly effective because our initial latent $\mathbf{z}_0^{\text{src}}$, derived from dual sampling, provides a comprehensive geometric foundation. The uniform samples ensure the latent robustly encodes the global structure, which this term helps to preserve. Concurrently, the update direction Δ_t is guided by the visual difference, which effectively performs a type of implicit inversion. Instead of a separate inversion step, this allows us to directly guide the sampling process to modify latent features that correspond to the fine-grained details initially captured by the importance samples. By applying this directional guidance Δ_t at each step of a pre-defined schedule, we obtain a final latent that balances the reference's 3D structure with the target's distinctive appearance and details.

4. Experiments

In this section, we conduct a comprehensive evaluation of our proposed method for 3D generation from flat-colored images. We perform both qualitative and quantitative comparisons against a diverse set of baselines. These include models specifically designed for character generation [23, 35], a retrieval-augmented method [26], a latent-aligned model [32], and state-of-the-art generalist models [12, 33].

4.1. Experimental Setup

Dataset Publicly available datasets focusing exclusively on flat, stylized images are scarce. Therefore, for our evaluation, we utilize the test set released by Art3D [6], which is specifically curated for this task and consists of over 100 images. As a consistent preprocessing step for all methods, we remove the background from each image using the Segment Anything Model [15] to ensure a clean and uniform input.

Evaluation metrics To assess the quality and semantic alignment of the final 3D mesh with the input 2D image, we compute cross-modal similarity scores using both ULIP [29] and Uni3D [34]. These models are specifically designed to learn a unified feature space by aligning 3D point cloud representations with image and text features. We therefore leverage their powerful encoders to extract a feature embedding from the input image and another from the generated mesh. A high cosine similarity between these embeddings indicates a strong semantic alignment, as a higher score means the generated 3D mesh is more semantically consistent with the input image. In addition, we follow [10] and use the Average LPIPS (A-LPIPS) metric to measure the multi-view consistency of the generated shapes. This is calculated by rendering 100 views of each mesh by rotating the azimuth at a fixed vertical angle of 20° and then averaging the LPIPS between adjacent frames, reported using both VGG and AlexNet backbones. Finally, to specifically analyze the editing process for retrieval-based methods, we also measure the feature distance between the initial retrieved 3D reference and the final generated mesh, allowing for a direct comparison of the modification's impact.

Implementation details. Our method is entirely training-free. For the core components of our framework, we utilize the memory-efficient Mini versions of the ShapeVAE and DiT from the Hunyuan3D-2.0 [33] baseline. This lightweight configuration, with a memory footprint of only 0.6 GB, enables our model to generate a high-quality 3D result from a single image in approximately 2 minutes on a single NVIDIA RTX 6000 Ada GPU.

For all baseline models, we use their officially released code and pre-trained weights following the recommended settings. For Phidias, which also employs a retrieval mechanism, we retrieve the same number of 3D candidates as our method. To ensure a fair comparison, we then select and report the result from the candidate that yields the highest Uni3D score with the input image.

4.2. Qualitative Evaluation

Fig. 6 presents a qualitative comparison of our method against several state-of-the-art baselines on a diverse set of flat images from the Art3D test set, including animals, characters and line drawings.

State-of-the-art generalist models like Hunyuan3D-2.0[33] struggle significantly with these flat-colored images, often producing collapsed meshes that fail to infer a plausible 3D structure. CharacterGen[23], which is specialized for generating humanoids in an A-pose, exhibits a strong prior bias; it fails to adapt to the varied poses and non-human subjects in the test set, often defaulting to a generic A-posed human shape regardless of the input. We note that while it performs poorly on these examples, its performance on its target domain is strong, as shown in Fig. 10.

DrawingSpinUp[35], which infers volume by first removing the input's outlines, produces results with high variance. While this approach can occasionally yield impressive volumetric shapes, it often leads to severe artifacts and unnatural deformations in the final mesh. Michelangelo[32], which aligns a 3D latent space, consistently produces voluminous shapes. However, it often fails to preserve the identity and specific structure of the input image, resulting in generic shapes that do not match the source. The retrieval-based baseline, Phidias[26], leverages a 3D reference to achieve good volumetric results. However, its 2D-based application of 3D information struggles

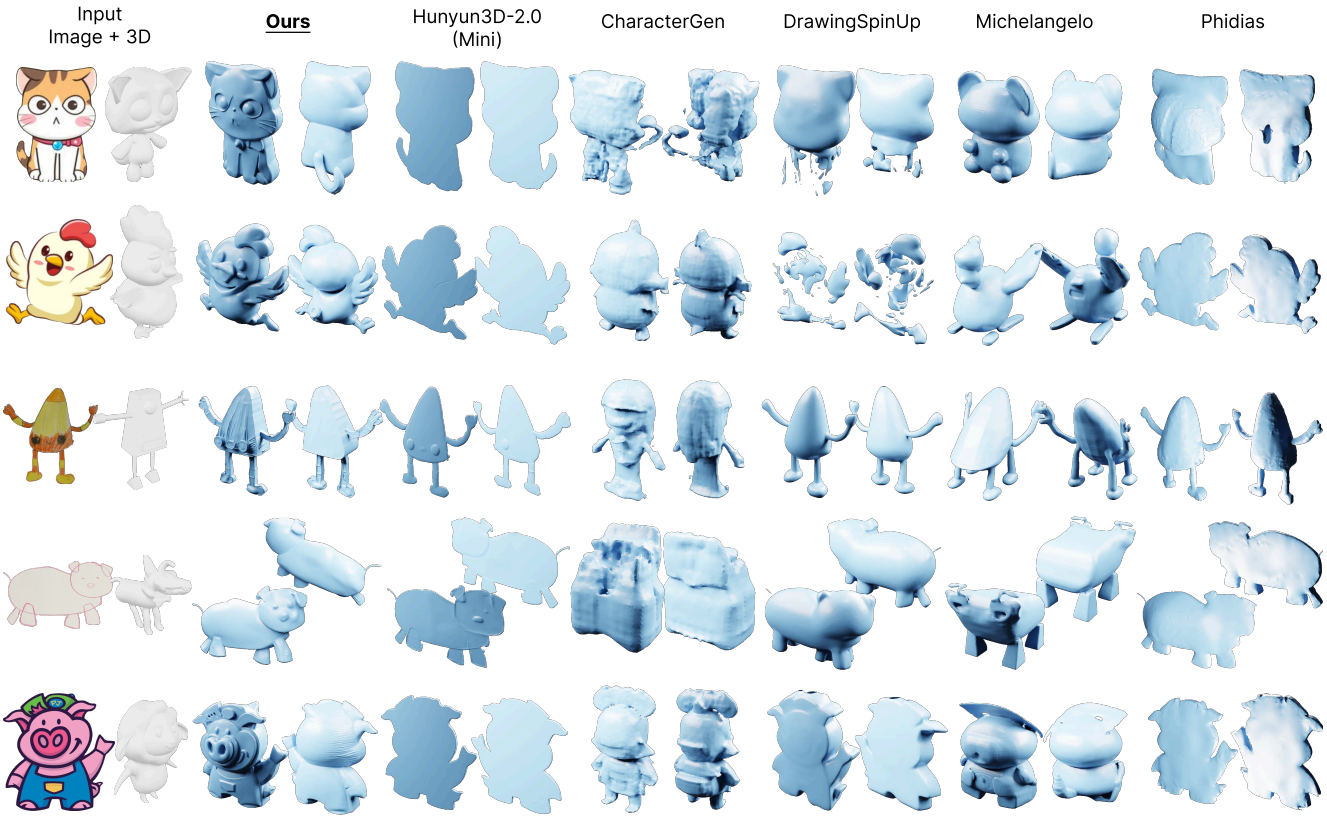

Figure 6. **Qualitative comparison of our method with several baselines**, including Hunyuan3D[33], CharacterGen[23], DrawingSpinUp[35], Michelangelo[32], and Phidias[26]. For each example, the leftmost column shows the 2D input image and its retrieved 3D reference. The subsequent columns display the output meshes from our method and the baselines, each rendered from two viewpoints.

to fully transfer the reference's structure, leading to geometrically inconsistent shapes.

In summary, all baseline methods exhibit critical failures, whether in inferring volume, preserving the input's identity, or avoiding artifacts. Most are unable to reconstruct fine details such as facial features. In contrast, our method consistently produces high-quality results across all categories. By retrieving a suitable 3D reference and performing editing directly in the 3D latent space, our approach generates meshes that are both structurally sound and rich in detail, faithfully capturing the essence of the input image from its overall volume to its distinctive features.

4.3. Quantitative Evaluation

We quantitatively evaluate our method by measuring both the alignment of the generated mesh to the input image and its internal 3D consistency.

Image-Mesh Alignment. Tab. 1 shows the similarity scores between the input image and the output mesh, evaluated using Uni3D-I and ULIP-I. Our method significantly outperforms all baselines, achieving the highest scores of 0.3537 in Uni3D-I and 0.1599 in ULIP-I. Baselines with specific failure modes score lower; for instance, DrawingSpinUp is penalized for its high result variance, while CharacterGen performs poorly on non-humanoid characters. Hunyuan3D series show progressive improvement

	Uni3D-I(↑)	ULIP-I(↑)
Hunyuan3D-2.1	0.3039	0.1044
Hunyuan3D-2.0 (Original)	0.2574	0.1020
Hunyuan3D-2.0 (Mini)	0.2384	0.0926
DrawingSpinUp	0.2217	0.0933
CharacterGen	0.2076	0.0945
Phidias	0.2075	0.1352
Michelangelo	0.2784	0.0989
REVIVE3D (Ours)	**0.3537**	**0.1599**

Table 1. **Quantitative comparison of image-to-mesh alignment.** This table evaluates how faithfully each model embeds the information from the input 2D image into the final 3D mesh.

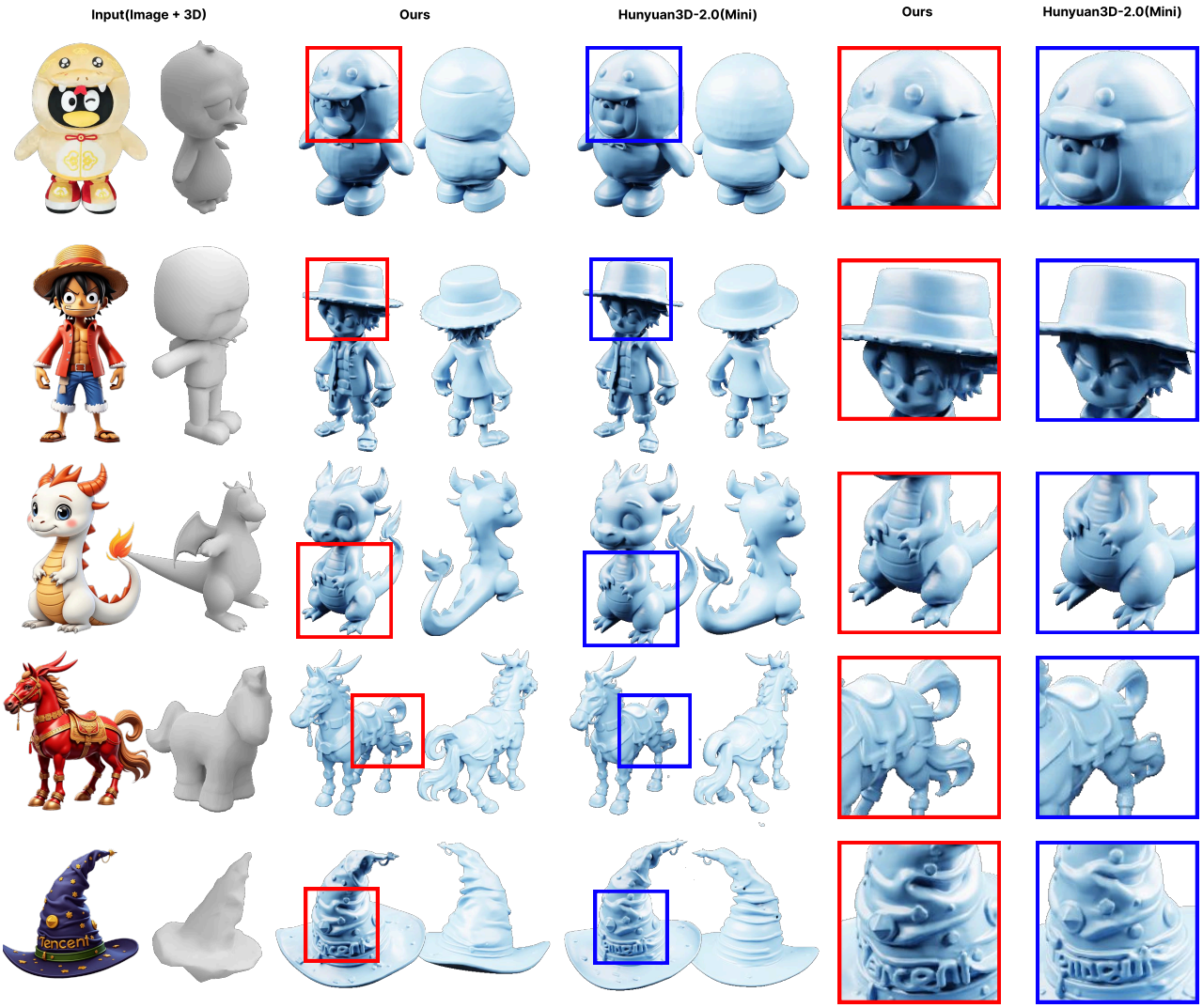

Figure 7. **Generalization to standard (non-flat) images.** A qualitative comparison of our method and the Hunyuan3D-2.0 (Mini) baseline on official example images from the Hunyuan3D project. Our method successfully generates high-quality 3D shapes and often captures finer surface details than the baseline, such as the lettering on the wizard hat and the correct number of toes on the dragon's foot.

across versions, yet our method still demonstrates a substantial improvement of approximately 0.12 in Uni3D-I and 0.07 in ULIP-I over our direct baseline, Hunyuan3D-2.0 (Mini).

	Uni3D(gen - ref)	ULIP(gen - ref)
Phidias	-0.0365	0.0320
REVIVE3D (Ours)	**0.0960**	**0.0412**

Table 2. **Modification distance from the initial 3D reference.** This table quantifies the change between the initial retrieved reference and the final generated mesh, measured by the difference in Uni3D and ULIP similarity scores.

A potential concern is that high scores could arise from simply finding a good 3D reference. To verify that our performance gain comes from effective editing, not just good retrieval, Tab. 2 measures the modification distance: the change in similarity scores after our editing process. While Phidias shows minimal improvement or even degradation, our method demonstrates a significant increase in both scores. This demonstrates our performance gain comes from our effective latent editing, not just the initial retrieval.

3D Consistency. Tab. 3 presents the A-LPIPS results for multi-view consistency, where our method reports higher scores than some baselines like DrawingSpinUp. We argue this is a characteristic of the A-LPIPS metric when applied

	A-LPIPS(↓)	
	VGG	Alex
Hunyuan3D-2.1	0.0408	0.0435
Hunyuan3D-2.0 (Original)	0.0405	0.0461
Hunyuan3D-2.0 (Mini)	0.0411	0.0486
DrawingSpinUp	**0.0315**	**0.0371**
CharacterGen	0.0462	0.0392
Phidias	0.0405	0.0428
Michelangelo	0.0473	0.0464
REVIVE3D (Ours)	0.0418	0.0410

Table 3. **Quantitative comparison of multi-view consistency.** This table reports the multi-view consistency of the generated meshes, measured using the Average LPIPS (A-LPIPS) metric with both VGG and AlexNet backbones.

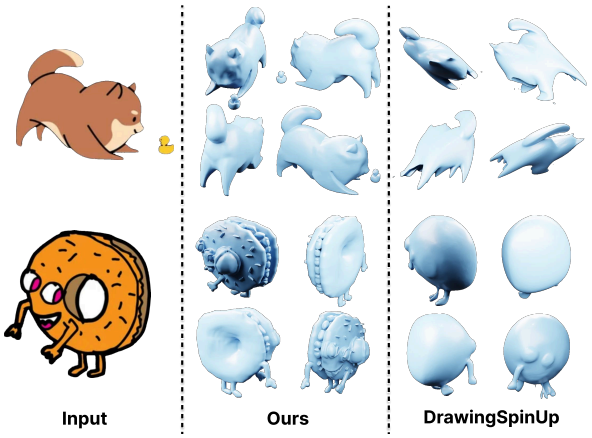

Figure 8. **Visual analysis of A-LPIPS scores.** A comparison of our detailed reconstructions (middle) against the smoother results from DrawingSpinUp (right). The higher fidelity to fine-grained geometric details in our results leads to higher A-LPIPS scores, despite strong visual consistency.

to highly detailed meshes, and Fig. 8 provides a visual explanation. For the top example in the figure, our detailed reconstruction scores 0.0580 / 0.0627 (VGG / Alex), while DrawingSpinUp's smoother mesh achieves lower scores of 0.0255 / 0.0298. This pattern holds for the bottom example, where our model reconstructs the complex donut character and scores 0.0580 / 0.0553, compared to the baseline's 0.0241 / 0.0214. These results demonstrate that our higher A-LPIPS scores are not due to a lack of consistency, but are a direct consequence of successfully reconstructing high-frequency geometric details, which cause larger variations between adjacent rendered views.

4.4. Generalization to Standard Images

While our method is specifically designed to address the challenges of flat images, we also evaluate its performance on standard, non-flat images to test its generalization capabilities. We conduct a qualitative comparison against our baseline, Hunyuan3D-2.0 (Mini), using several official example images from the Hunyuan3D project.

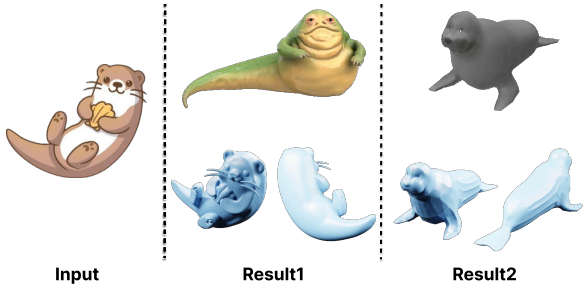

Figure 9. **Limitation of Semantic vs. Shape-based Retrieval.** This figure illustrates a failure case of our semantic-based retrieval. For the input image (left), an ideal retrieval would provide a reference with a similar pose and shape (middle). Instead, our current method retrieves a semantically related but geometrically dissimilar 3D model (right), which provides a poor initialization for our editing process.

As shown in Fig. 7, our method not only produces high-quality 3D shapes comparable to the baseline but often excels in reconstructing fine-grained details. For instance, when generating the wizard hat, our model successfully captures the intricate detail of the lettering on the texture, a feature that is lost or blurred in the baseline's output. This suggests that our direct latent editing mechanism is a robust and general approach that can enhance detail preservation for a wide range of input images, not just flat ones.

5. Conclusion

In this work, we have presented a novel retrieve-and-edit framework that demonstrates significant advancements in generating 3D shapes from flat-colored images. By operating directly on a 3D representation, our method bypasses the information loss inherent in 2D-centric approaches, successfully handling ambiguous inputs. The quality of our editing process fundamentally depends on the initial retrieved reference. Our current semantic retrieval can provide geometrically dissimilar priors (Fig. 9), which motivates the development of more advanced, shape-aware retrieval methods. Nevertheless, our framework introduces a robust solution that is plug-and-play, training-free, and remarkably efficient. It generates high-quality results in approximately 2 minutes using only 0.6 GB of memory. Our framework provides a practical and efficient pathway for generating high-fidelity 3D from flat-colored images, offering a valuable contribution and opening up new possibilities for content creation from the diverse world of 2D stylized art.

References

[1] Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *arXiv preprint arXiv:2205.02904*, 2022. 2

[2] Minghao Chen, Junyu Xie, Iro Laina, and Andrea Vedaldi. Shap-editor: Instruction-guided latent 3d editing in seconds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26456–26466, 2024. 3

[3] Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16251–16261, 2025. 3

[4] Shuhong Chen, Kevin Zhang, Yichun Shi, Heng Wang, Yiheng Zhu, Guoxian Song, Sizhe An, Janus Kristjansson, Xiao Yang, and Matthias Zwicker. Panic-3d: Stylized single-view 3d reconstruction from portraits of anime characters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21068–21077, 2023. 2

[5] Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. *Advances in Neural Information Processing Systems*, 35:30923–30936, 2022. 2

[6] Xiaoyan Cong, Jiayi Shen, Zekun Li, Rao Fu, Tao Lu, and Srinath Sridhar. Art3d: Training-free 3d generation from flat-colored illustration. *arXiv preprint arXiv:2504.10466*, 2025. 2, 5

[7] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023. 3

[8] William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023. 2

[9] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2328–2337, 2023. 3

[10] Susung Hong, Donghoon Ahn, and Seungryong Kim. Debiasing scores and prompts of 2d diffusion for view-consistent text-to-3d generation. *Advances in Neural Information Processing Systems*, 36:11970–11987, 2023. 5

[11] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 1

[12] 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. 1, 2, 5

[13] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 3

[14] Hyunwoo Kim, Itai Lang, Noam Aigerman, Thibault Groueix, Vladimir G Kim, and Rana Hanocka. Meshup: Multi-target mesh deformation via blended score distillation. *arXiv preprint arXiv:2408.14899*, 2024. 2

[15] 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. 5

[16] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. *arXiv preprint arXiv:2412.08629*, 2024. 3

[17] 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, 2024. 1

[18] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9970–9980, 2024. 1, 2

[19] Zhongjin Luo, Shengcai Cai, Jinguo Dong, Ruibo Ming, Liangdong Qiu, Xiaohang Zhan, and Xiaoguang Han. Rabit: Parametric modeling of 3d biped cartoon characters with a topological-consistent dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12825–12835, 2023. 2

[20] Yiwei Ma, Xiaoqing Zhang, Xiaoshuai Sun, Jiayi Ji, Haowei Wang, Guannan Jiang, Weilin Zhuang, and Rongrong Ji. X-mesh: Towards fast and accurate text-driven 3d stylization via dynamic textual guidance. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2749–2760, 2023. 2

[21] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13492–13502, 2022. 2

[22] 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

[23] Hao-Yang Peng, Jia-Peng Zhang, Meng-Hao Guo, Yan-Pei Cao, and Shi-Min Hu. Charactergen: Efficient 3d character generation from single images with multi-view pose canonicalization. *ACM Transactions on Graphics (TOG)*, 43(4): 1–13, 2024. 2, 5, 6

[24] Junyoung Seo, Susung Hong, Wooseok Jang, Inès Hyeonsu Kim, Minseop Kwak, Doyup Lee, and Seungryong Kim. Retrieval-augmented score distillation for text-to-3d generation. *arXiv preprint arXiv:2402.02972*, 2024. 2

[25] Yu-Pei Song, Yuan-Tong Liu, Xiao Wu, Qi He, Zhaoquan Yuan, and Ao Luo. Magiccartoon: 3d pose and shape estimation for bipedal cartoon characters. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8219–8227, 2024. 2

[26] Zhenwei Wang, Tengfei Wang, Zexin He, Gerhard Hancke, Ziwei Liu, and Rynson WH Lau. Phidias: A generative model for creating 3d content from text, image, and 3d conditions with reference-augmented diffusion. *arXiv preprint arXiv:2409.11406*, 2024. 2, 3, 5, 6

[27] 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. 1

[28] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 1

[29] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1179–1189, 2023. 5

[30] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023. 3

[31] 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.

[32] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems*, 36:73969–73982, 2023. 3, 5, 6

[33] 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. 1, 3, 4, 5, 6

[34] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023. 3, 5

[35] Jie Zhou, Chufeng Xiao, Miu-Ling Lam, and Hongbo Fu. Drawingspinup: 3d animation from single character drawings. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–10, 2024. 2, 5, 6

A. Additional Results

This appendix provides supplementary qualitative results and explains the texture application process used for our main paper's visual comparisons.

A.1. Additional Qualitative Results

We present an extended gallery of generated meshes to complement the results in the main paper. Fig. 10 illustrates additional results, focusing on the challenging subcategory of humanoid characters from our flat-colored image test set.

A.2. Texture Application Details

The qualitative comparisons in the main paper (Fig. 6, Fig. 7) focus on the geometry of the generated meshes. To provide a comprehensive evaluation of appearance, this appendix presents the corresponding textured results. The texturing process varies across the compared methods. Baselines such as CharacterGen, DrawingSpinUp, and Phidias utilize their native, end-to-end texture generation pipelines. For the remaining methods (Ours, Hunyuan3D-DiT, and Michelangelo), which primarily focus on geometry, we painted the output meshes using the Hunyuan3D-2.0 Paint model to ensure a fair and consistent comparison of texturing quality. The final textured results for various flat and standard images are presented in Fig. 12, Fig. 13 and Fig. 14.

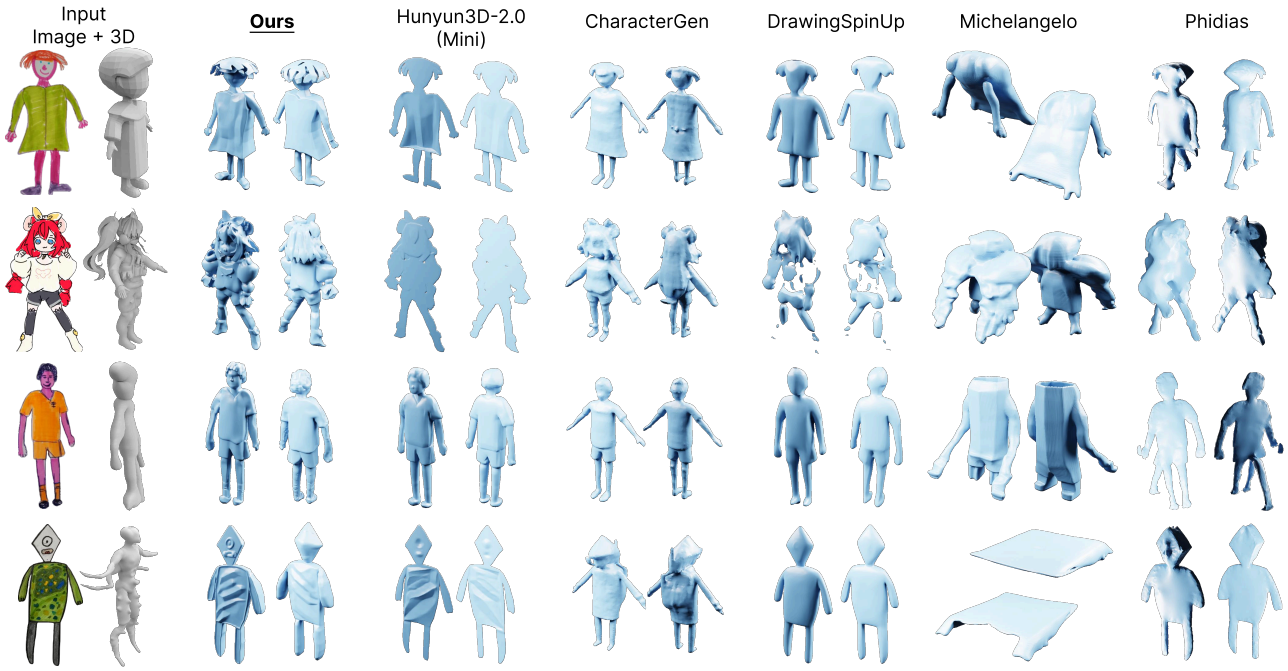

Figure 10. Geometry-only results for humanoid characters on flat-colored images.

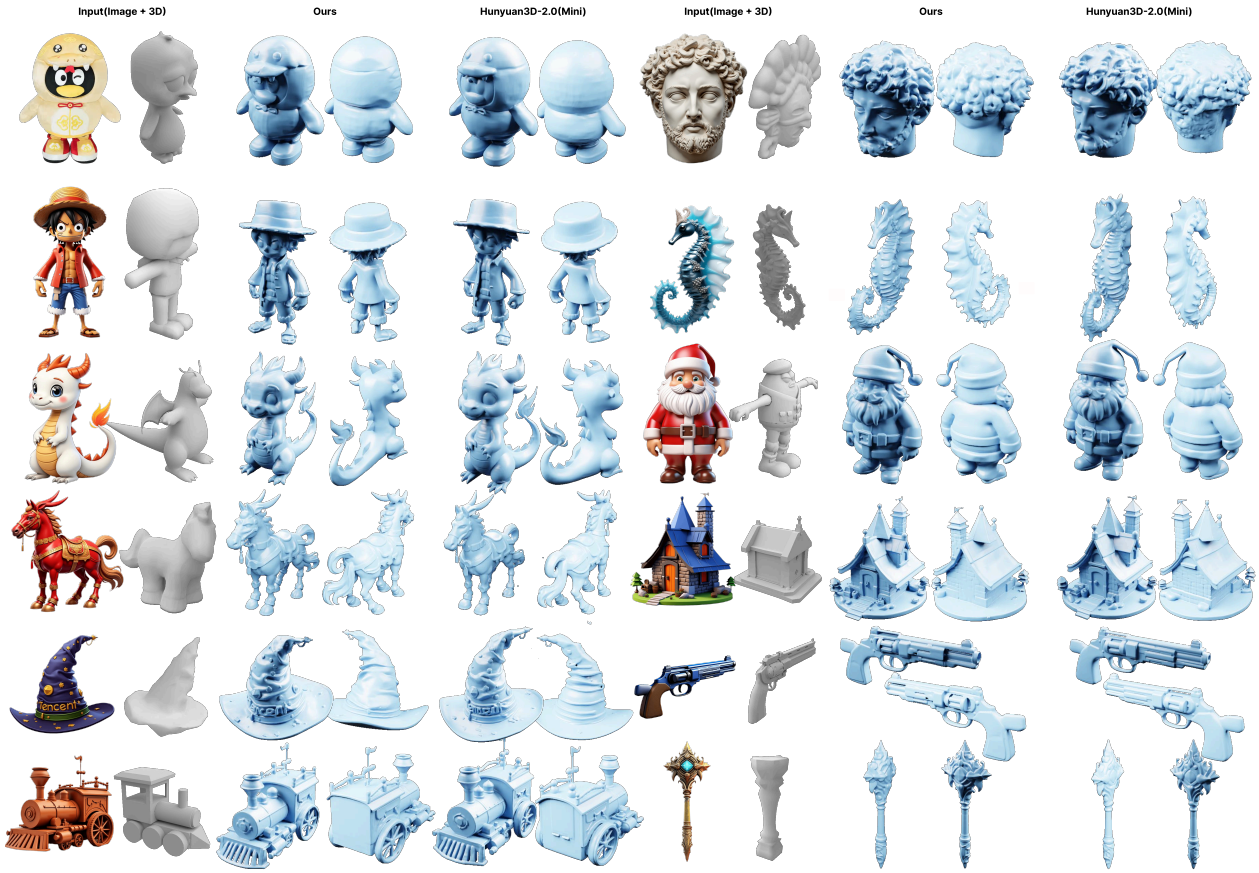

Figure 11. Additional geometry-only results on standard images.

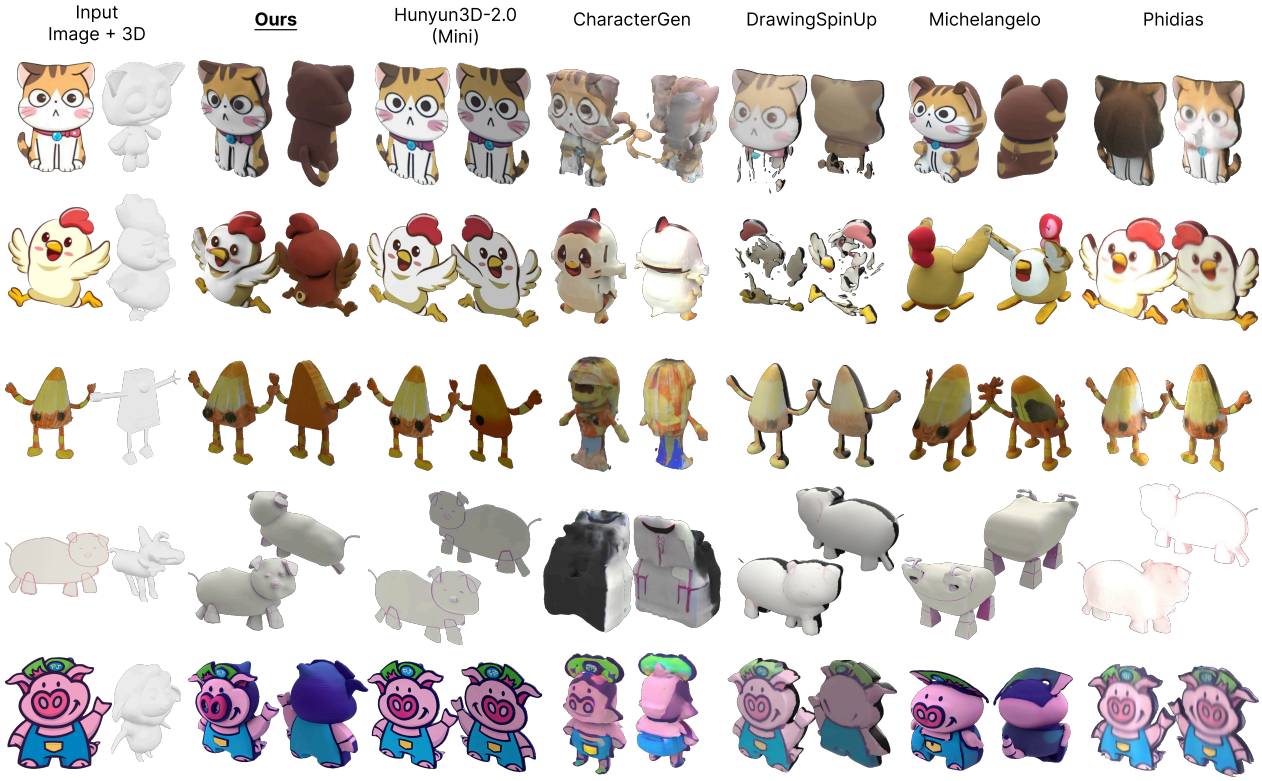

Figure 12. Textured qualitative results on flat-colored images.

Figure 13. Textured qualitative results for humanoid characters on flat-colored images.

Figure 14. Textured qualitative results on standard images.