DetailGen3D: Generative 3D Geometry Enhancement via Data-Dependent Flow

Ken Deng[1,2,3] Yuan-Chen Guo[2] Jingxiang Sun[1] Zi-Xin Zou[2] Yangguang Li[2] Xin Cai[4]
Yan-Pei Cao[2] Yebin Liu[1] Ding Liang[1,2]

[1]Tsinghua University [2]VAST [3]Sun Yat-sen University [4]The Chinese University of Hong Kong

Figure 1. Our method demonstrates effective geometry refinement across various tasks and representations. In the images, the coarse geometry is displayed in gray, while the refined geometry produced by our approach is shown in red. On the right, zoomed-in details are provided to better observe the refinement effects.

Abstract

Modern 3D generation methods can rapidly create shapes from sparse or single views, but their outputs often lack geometric detail due to computational constraints. We present DetailGen3D, a generative approach specifically designed to enhance these generated 3D shapes. Our key insight is to model the coarse-to-fine transformation directly through data-dependent flows in latent space, avoiding the computational overhead of large-scale 3D generative models. We introduce a token matching strategy that ensures accurate spatial correspondence during refinement, enabling local detail synthesis while preserving global structure. By carefully designing our training data to match the characteristics of synthesized coarse shapes, our method can effectively enhance shapes produced by various 3D generation and reconstruction approaches, from single-view to sparse multi-view inputs. Extensive experiments demonstrate that DetailGen3D achieves high-fidelity geometric detail synthesis while maintaining efficiency in training. Our project page is https://detailgen3d.github.io/DetailGen3D/

1. Introduction

Obtaining high-quality 3D geometry has been a long-standing research focus in the fields of computer vision and graphics. High-quality 3D models are not only valuable in the film industry, video games, and virtual reality but also play a crucial role in the rapidly advancing field of embodied intelligence, contributing significantly to simulation environments. Early approaches for high-quality multi-view stereo reconstruction rely on dense multi-view inputs [1, 8, 17, 19, 27, 85]. Although recent methods [5, 26, 67, 69] based on Neural Radiance Field

(NeRF) [45] and 3D Gaussian Splatting (3DGS) [31] have improved the end-to-end performance, reconstructing high-quality geometry remains a challenging problem.

Recent methods for 3D generation from sparse or single views have evolved into two main paradigms: 1) optimization using 2D generative models and 2) direct training with 3D data. The first approach leverages pre-trained 2D vision models - notably, DreamFusion [50] introduced Score Distillation Sampling (SDS) to align rendered views with text-conditioned distributions. Related methods [59, 70] have extended this paradigm, though they consistently face geometric inconsistency issues, such as the "Janus problem" where details conflict across viewpoints. The second paradigm, training with 3D data, has seen multiple technical advances. Multi-view diffusion approaches [33, 37, 38, 41, 43, 44, 51, 56, 57, 66] achieve view consistency but struggle with geometric coherence in fine details. Feed-forward Large Reconstruction Models [25, 32, 40, 68, 72, 82, 94, 99] offer impressive speed but face resolution constraints. While recent 3D diffusion models [53, 75, 91, 92, 95, 96] demonstrate quality improvements through direct 3D training, computational demands continue to limit their achievable resolution and detail.

While recent advances in 3D generation have shown promising results, the generated shapes often lack fine geometric details. Simply scaling up these generative models demands prohibitive computational resources. Traditional geometry refinement approaches that rely on high-resolution dense multi-view images are ill-suited for enhancing such generated shapes, as these additional inputs are typically unavailable. Furthermore, existing optimization-based methods that utilize normal or shading information [46, 73, 74, 79, 88] require precise texture alignment and struggle to preserve global shape coherence. To address these challenges, we propose *DetailGen3D*, a generative approach to 3D geometry refinement that learns to synthesize plausible fine-scale details directly from high-quality 3D shapes. By capturing the underlying geometric patterns through data-driven priors, our method can enhance coarse shapes while maintaining structural consistency, even under noisy or incomplete information.

To tackle geometry refinement tasks with a generative model, a straightforward approach is to train a coarse geometry conditioned diffusion model, as seen in 2D image restoration methods [3, 87]. However, this method can be resource-intensive and may not be optimal for 3D tasks. Instead, inspired by Fischer et al. [14], who achieved effective image super-resolution without relying on large generative models, we propose a more training-efficient strategy. Rather than learning a complex mapping from noise to fine details, we focus on directly modeling the transformation between coarse and fine geometry using data-dependent rectified flow [42]. This approach utilizes optimal transport, which provides a direct and structured mapping between coarse and fine geometry. By exploiting this coupling, we eliminate the need for the random coupling of noise and fine details, thus significantly reducing the training cost. This method is better suited for scaling up, offering a more efficient path to geometry refinement.

Specifically, we introduce a training method called token matching. Establishing the refinement process locally is essential; otherwise, global refinement leads to inefficient training and hinders detail capturing. Token matching matches the coarse geometry latent code with the fine geometry latent code in latent space one-to-one, improving training efficiency. It prevents the network from learning unnecessary operations (e.g., swapping positions between two latent codes) and focuses solely on local refinement operations. This approach enables the capture of geometry details even with a small network.

Effective geometry refinement requires carefully designed training data. While high-quality 3D shapes are available, obtaining matching coarse-fine pairs poses a significant challenge. Simply applying traditional mesh degradation algorithms (e.g., simplification or smoothing operations) leads to simple objects remaining unchanged or extreme degradation on complex objects, resulting in low-quality coarse-fine pairs. To address this, we leverage an LRM-based model that reconstructs 3D geometry from sparse-view renderings of high-quality fine models. This approach enables consistent degradation across objects of varying complexity, enhances the utilization of existing 3D objects, and increases both the quality and quantity of coarse-fine pairs.

In summary, our contributions are as follows:

1. We develop a novel generative geometry refinement algorithm, demonstrating its highly effectiveness for different geometric representations.
2. We propose a data-dependent rectified flow to incorporate coarse geometry information, enabling local distribution transformation from coarse to refined geometry.
3. We introduce a token matching training method that significantly enhances training efficiency and spatial correspondence accuracy.

2. Related Works

3D generation using 2D generative model. With the significant advancements in text-to-image generation models [2, 54, 55], methods for text to 3D generation based on SDS loss optimization [9, 35, 36, 50, 59, 62, 70, 80, 86] have emerged, allowing the acquisition of 3D models without relying on multi-view inputs. However, these image diffusion models lack 3D priors, leading to the generation of 3D models that often suffer from the "Janus problem", with poor alignment between geometry and texture, as well as substantial time overhead.

3D generation using 3D data training. Finetuning image diffusion models using Objaverse [11], Objaverse-XL [12] to generate reasonably consistent multi-views with 3D consistency, which could then be input into multi-view reconstruction models to obtain 3D geometry [33, 37, 38, 41, 43, 44, 51, 56, 57, 66]. Some methods trying to improve multi-view consistency [63, 71, 81, 83], but geometry quality is even worse than previous methods since only 2D supervision is used. Using triplane as 3D representation, large-scale reconstruction model [25, 32, 60, 65, 68, 72, 82, 84, 99] utilizing transformer architectures and triplane representations with fixed-view multi-view inputs, enabling reconstruction in just a few seconds. However, the extensive training time and GPU memory requirements of large reconstruction models severely limit their resolution. Generative models using triplane as 3D representation [21, 24, 58, 76, 93] meet the same problem as well. Using point cloud as 3D representation improves the efficiency [30, 34, 47, 91, 95–97] also brings high training costs. Some recent methods use voxel [39, 53, 78] or gaussian [63, 92, 94] as 3D representation are also efficient.

3D shape detailization. SketchPatch [16] uses a PatchGAN [29] discriminator to mimic the local style of a reference image, stylizing plain solid-lined sketches. DECORGAN [10], which employs PatchGAN to generate detailed voxel shapes from input coarse voxels, with the geometric style of the generated shape derived from a detailed reference voxel model. ShaDDR [6] enhances the generated geometry of DECORGAN by utilizing a 2-level hierarchical GAN and introducing texture generation. Meanwhile, DECOLLAGE [7] further improves shape detailization through structure-preserving losses and adaptive weighting of style and global discriminators. However, these methods primarily focus on enhancing the voxel resolution of low-quality inputs, which is different from our method: pushing the boundaries of detail enhancement beyond current state-of-the-art generative models, capturing finer details missed by coarser methods.

3. Methods

Our method F is designed to refine coarse geometry G_{Coarse} into fine geometry G_{Fine}, guided by an image prompt I. Here, G_{Coarse} may originate from reconstruction or generation processes, while G_{Fine} represents an enhanced version with improved surface quality and additional geometric details, such as noise removal and refinement:

$$G_{Fine} = \mathcal{F}(G_{Coarse}, I) \tag{1}$$

Considering this process has strong uncertainty, we model this process using generative model. In particular, we select rectified flow [42] because of its efficiency based on optimal transport and its ability to model the mapping relationship between two data distributions, i.e., between the distribution of coarse geometry and fine geometry. Our inference pipeline is shown in Fig 2 (1).

3.1. Data Dependent Rectified Flow

We use data-dependent rectified flow to model the distribution mapping of coarse and fine geometry due to its efficiency. Compared to modeling the transformation between Gaussian noise and fine geometry, directly modeling the transformation between coarse and fine geometry can, to some extent, reconstruct the coarse shape without extensive denoising training, as demonstrated in Fig 7. In contrast, modeling the distribution mapping between noise and fine geometry, with coarse geometry as a condition via cross-attention, requires longer training time to achieve noise-to-coarse mapping.

Model Architecture. To improve efficiency, we build our refinement process in latent space, and our network consists of two parts: 3D Variational Autoencoder (3D-VAE) and Diffusion Transformer [49] (DiT).

3D Variational Autoencoder. The design of our 3D-VAE is primarily inspired by 3DShape2VecSet [91] and CLAY [95] and shares the same architecture with CLAY's VAE. To transform 3D geometry into latent space, we first sample point cloud X from 3D geometry's surface and adopt a two-stage downsampling (random downsample firstly then apply farthest point sampling [18]) for X to get query points X_0. Lastly, the cross attention is applied to X and X_0 with learnable positional embedding to obtain latent code z.:

$$z = \mathcal{E}(X) = \text{CrossAttn}(\text{PosEmb}(X_0), \text{PosEmb}(X)) \tag{2}$$

The 3D-VAE decoder, composed of several self attention layers and a cross attention layer, transform latent code z to signed distance field (SDF) with preset query points q.:

$$SDF = \mathcal{D}(q, z) = \text{CrossAttn}(\text{PosEmb}(q), \text{SelfAttn}(z)) \tag{3}$$

Diffusion Transformer. Our DiT network, comprised of 24 DiT blocks with a width of 768 and totaling 368M parameters, is designed with efficiency in mind. Each DiT block consists of a multi-head cross-attention layer (MCA), a multi-head self-attention layer (MSA), and a feedforward network (FFN), interleaved with layer normalization (LN). To accommodate the relatively small width, we inject the time step t using the adaptive layer normalization (adaLN) [28], modulating MSA, MCA, and FFN via factors g, γ, and s, obtained by a MLP conditioned on t.:

$$z = z + g_{\text{msa}} \cdot \text{SelfAttn}(\text{mod}(\text{LN}_{\text{msa}}(z), s_{\text{msa}}, \gamma_{\text{msa}})) \tag{4}$$

Turning to conditioning on the image prompt y, we adopt cross-attention to ensure spatial alignment between latent code and image feature, which is extracted by DINO V2 [48]. The conditioning is defined as:

$$z = z + g_{\text{mca}} \cdot \text{CrossAttn}(\text{mod}(\text{LN}_{\text{mca}}(z), s_{\text{mca}}, \gamma_{\text{mca}}), y) \tag{5}$$

At last, tokens will pass through a feedforward network:

$$z = z + g_{\text{ffn}} \cdot \text{FFN}(\text{mod}(\text{LN}_{\text{ffn}}(z), s_{\text{ffn}}, \gamma_{\text{ffn}})) \tag{6}$$

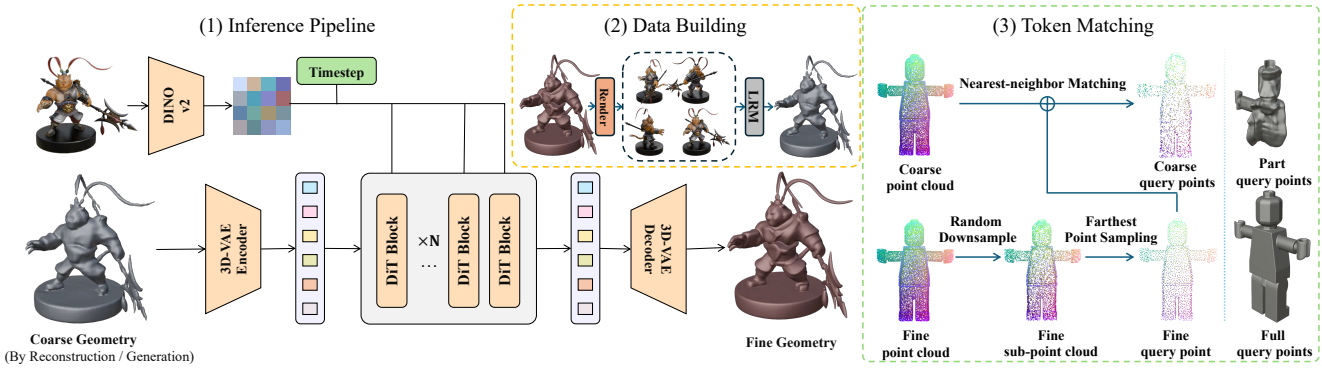

Figure 2. (1) Inference pipeline. We use 3D-VAE to extract tokens of the coarse geometry generated or reconstructed, then input the coarse token and DINO feature [48] of the image prompt to DiT [49]. After the refinement process, we decode the predicted token using a 3D-VAE decoder to obtain refined geometry. The inference process takes only a few seconds. (2) For training data, we use reconstruction results reconstructed by LRM using multi-views rendered from fine geometry as coarse geometry. (3) We demonstrate the token matching process on the left. On the right, for the top one, we only use part query points, which are located in quadrant one, and for the bottom one, we use full query points, which demonstrate that tokens represent the space around the corresponding query points.

Rectified Flow and Loss Functions. Unlike previous generative models that model the mapping between a Gaussian distribution and the ground truth data distribution, we model the mapping between the coarse geometry distribution and the fine geometry distribution. Let z_1 represent the fine geometry's latent code and z_0 corresponds to the coarse geometry's latent code. The geometry latent code at time t, z_t, is given by:

$$z_t = (1-t)\cdot z_0 + t\cdot z_1 \tag{7}$$

The conditional probability $P(z_1|z_t)$ is defined as:

$$P(z_1|z_t) = \mathcal{N}(z|(1-t)\cdot z_0 + t\cdot z_1, \sigma_{\min}^2\cdot I) \tag{8}$$

The loss function $L(\theta)$ is:

$$L(\theta) = \mathbb{E}_{t,z_0,z_1,y}\left[\|v_\theta(t,z_t,y) - (z_1 - z_0)\|^2\right] \tag{9}$$

where v_θ is parameterized by the DiT with learnable parameters θ and y is the image prompt. The L2 loss constrains the prediction of DiT with ground truth geometry in the form of v-prediction in latent space.

Noise Augmentation. The degraded geometric distribution is relatively independent so direct modeling of distribution transformation between coarse and fine geometry cannot well learn comprehensive refinement rules from the dataset, leading to noisy results. Inspired by [15], we apply noise augmentation to the latent code of coarse geometry, which helps improve the quality. The noise ϵ is added to z_0 before obtaining z_t instead of after, as applying noise after obtaining z_t would affect the efficiency of optimal transport, leading to an optimization path that is not straight. See supplementary material for more details.

3.2. Token Matching

We propose token matching, a training method to match the latent codes of coarse and fine geometries for efficient training, as shown in Fig 2 (3). Training without token matching leads the network to learn not only refinement rules but also unnecessary operations (e.g., swapping latent codes), significantly reducing training efficiency. However, measuring the similarity between latent codes is non-trivial. Since the latent code represents the space area around the query points X_0 (Fig 2 (4)), we are inspired to match latent codes in latent space similarly to matching query points in 3D space.

To balance effectiveness and efficiency, we propose a novel token matching method. First, we obtain the query points of fine geometry following 3D-VAE's two-stage downsampling (applying random downsampling then farthest point sampling on fine geometry's point cloud). Next, we obtain coarse geometry's query points by applying a nearest-neighbour algorithm to identify the closest fine geometry's query points X_0 in the coarse point cloud rather than downsampling coarse geometry's point cloud. This method is computationally efficient, maintaining model generalization and offering robust performance across both simple and complex geometries while ensuring local refinement. We further discuss token matching in the supplementary material.

3.3. Data Curation

To achieve effective local transformation between coarse and fine geometries, well-aligned coarse–fine pairs are crucial. Traditional geometric degradation techniques—such as Taubin smoothing [64]- can effectively reduce high-frequency noise in complex geometries but tend to be too subtle for simpler ones. Moreover, applying the same degradation across all objects risks distorting complex shapes while failing to degrade simpler ones sufficiently. To address these issues, we adopt the reconstruction outputs from an LRM as shown in Fig 2 (2), which naturally introduce a balanced level of geometric degradation that adapts to an object's inherent complexity. This choice not only preserves spatial correspondence but also better replicates the artifacts typical of neural reconstructions. Further details of data cu-

ration strategy are provided in the supplementary material.

Considering that fine objects are essential for training, we select Objaverse as our dataset and filter out the low-quality objects by evaluating each object's quality based on CLIP [52] feature of four ortho rendering views. After filtering, our training set consists of about 110,000 objects, most of them are rich in details. We randomly partition the dataset into a training set and an evaluation set, with the latter containing 350 objects. The coarse geometry is reconstructed using a reimplemented version of Instant3D [32], leveraging four orthographic views of objects at a resolution of 512.

4. Experiment

Our comprehensive evaluation of the model's 3D geometric refinement capabilities is divided into three parts: 1). *FeedForward-based Reconstruction*: We tested the refinement ability of our method on the reconstruction results of various LRM models to demonstrate its effectiveness. 2). *Generation:* We assessed our method's performance on the generation outputs from different LRM models, showcasing its versatility across diverse 3D representations. 3). *More 3D representations:* We applied our method to reconstruction results reconstructed by NeuS-like [67] optimization-based and generation results generated by Rodin Gen-1[1] and Neural4D[2], achieving impressive results that highlight its strong generalization. We also performed ablation studies on our design choices. See the supplementary for more implementation and experiment details.

For the metric, we select FID [22] as our metric and obtain each object's FID result following the setting of SDF-styleGAN [98], which can assess visual quality, capture statistical differences, detect mode collapse, and serve as a standardized benchmark despite not measuring fine-grained consistency perfectly. Considering FID may not fully capture the perceptible difference in fine details, we also mixed results from different datasets, methods, and tasks for a user study.

4.1. Feed-forward Reconstruction

Considering the coarse-fine pairs used in the training process, where all coarse models are reconstructed by Instant3D [32], this experiment aims to evaluate the generalizability by refining the reconstruction results of different kinds of LRM, including Instant3D as well.

For the evaluation set, we randomly sampled 199 models from the GSO [13] dataset and selected 350 unseen models from [11] as the evaluation set. For the LRM, We used Instant3D [32], CRM [71], InstantMesh [81] (testing both NeRF and Mesh, the checkpoints used are all the official large versions provided). We render multi-view images according to different LRM's requirements and use their reconstruction results as coarse model.

Visualization results can be found in Fig 3 and FID results can be found in Tab 1. Our method can highlight the blurry details in the coarse model and add details only exist in the image prompt. In addition, for the aliasing existing on the input geometric surface, our method can eliminate them and show a smoother geometric surface. The FID measurement results reflect the effectiveness and robustness of our method for different models and different objects.

FID ↓	Reconstruction		Generation	
	Coarse	Fine	Coarse	Fine
Instant3D [32]	20.33	**19.07**	x	x
TripoSR [65]	x	x	51.80	**33.33**
CRM [71]	40.13	**25.29**	48.91	**33.74**
InstantMesh (NeRF) [81]	58.35	**29.51**	50.12	**25.75**
InstantMesh (Mesh) [81]	35.32	**24.45**	33.85	**25.79**

Table 1. FID comparison of applying our method to different LRM results on reconstruction/generation task using Objaverse.

4.2. Generation

For the generation refinement task, we randomly sampled 199 models from the GSO [13] dataset and selected 350 unseen models from Objaverse [11] as the evaluation set, same as mentioned before. Additionally, to explore the refinement ability, we used 110 images provided by GPTEval3D [77] as input, employing TripoSR [65], CRM [71] and InstantMesh [81] (testing both NeRF and Mesh, the checkpoints used are all the official large versions provided.) as generation models for the coarse geometry.

As the visualization results illustrated in Fig 4 and FID results shown in Tab 1, our method performs particularly well on complex objects, while coarse models' surface have large geometric noise and lack details. Furthermore, our method shows outstanding performance in FID scores.

For generation results, some of them are not well aligned with the input image. However, our method can still refine them in a proper way, where aligning the original shape and adding finer local detail. It shows that although our method learns refinement ability with aligned image-coarse shape pairs, our method knows how to use image prompt to add local details.

For experiments on GPTEval3D generation results, refinement results shows the robustness of our method to such challenging data, as illustrated in Fig 5. Each image from GPTEval3D [77] is used to obtain coarse models through image-to-3D models (i.e., TripoSR, CRM, InstantMesh) and serves as an image prompt in the refinement process. Due to the high level of difficulty, most image-to-3D generation models struggled to produce reasonable 3D models based on these images. We selected the plausible 3D models from them and applied our refinement method.

[1] https://hyperhuman.deemos.com/rodin
[2] https://www.neural4d.com/

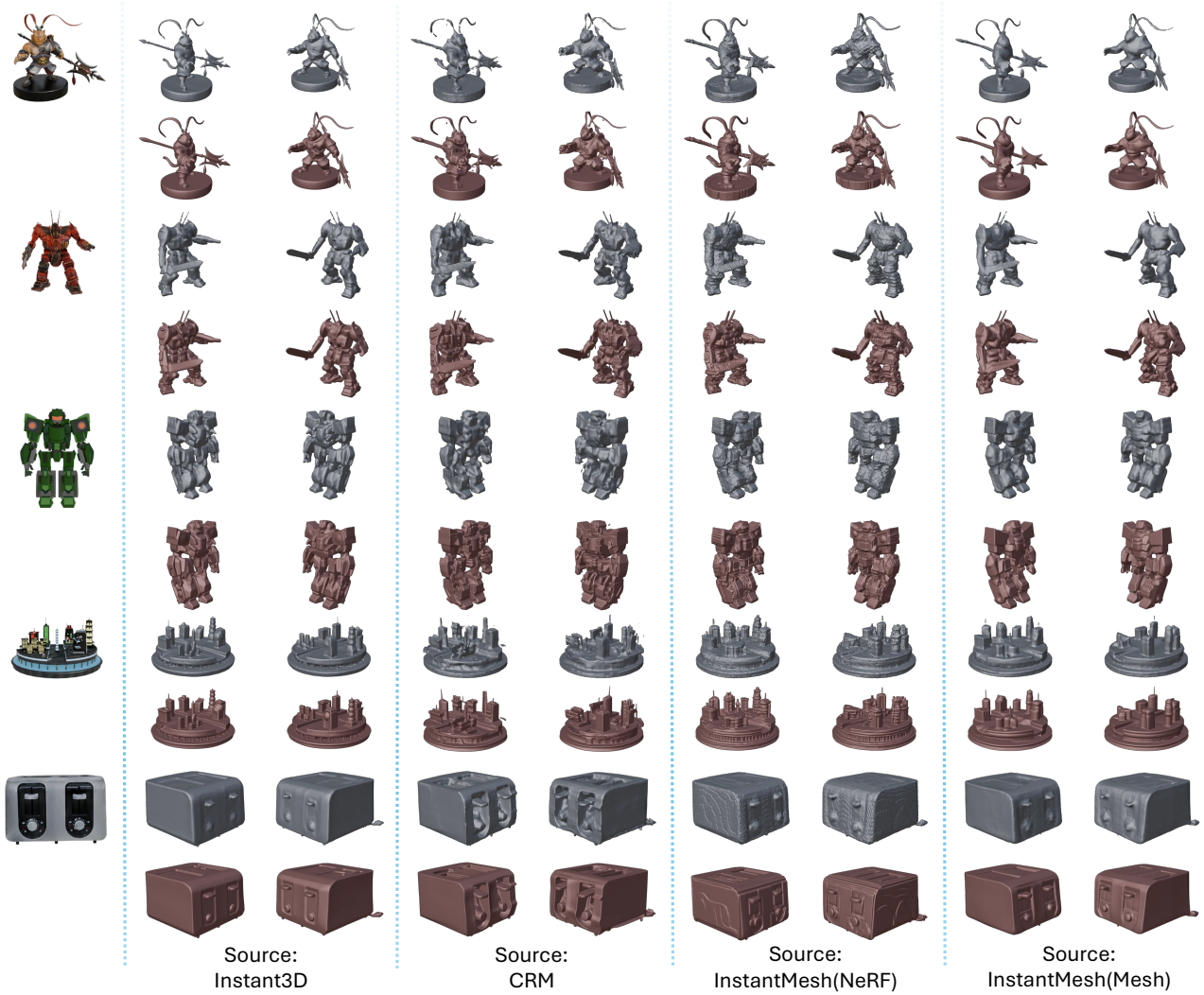

Figure 3. We apply our method on input meshes reconstructed by different approaches (Instant3D [32], CRM [71], InstantMesh [81]). ■ represent coarse, ■ represent fine refinement results from our method. The top three objects are from Objaverse, while the bottom object is from GSO. More results can be found in the supplementary.

4.3. More 3D representations

To further evaluate the generalizability on other 3D representations, we tested our refinement ability on reconstruction results of NeuS [67] and generation results of two commercial products, i.e., Rodin Gen-1, Neural4D.

Considering the long optimization time of NeuS and limited quota of Rodin Gen-1 and Neural4D, randomly sampled a subset of evaluation of Objaverse and GSO used in previous experiments for evaluation.

The refinement results are all illustrated in Fig 6 and demonstrate that our method successfully refines and corrects irregular details. The results of NeuS are reconstructed with 40 views evenly distributed around each object. However, due to the lack of prior guidance, they are prone to artifacts in scenarios with complex occlusions, and the smoothness of the surface remains imperfect. After refining by our method, the shredded parts are became continuously and details are more obvious. Considering commercial products are trained on huge 3D data with large amount of GPUs, we selected images from GPTEval3D as input. The generation results have smooth surface but still lacked of details. After refinement, more details are carved onto geometry, and shapes are more plausible.

4.4. User Study

The FID metric, while showing incremental improvement, may not fully capture the perceptible difference in fine details achieved by our method, especially on complex geometries. We invited 32 researchers, primarily experts in 3D vision, and compared the geometry quality of 10 models (reconstructed and generated by Instant3D [32], InstantMesh [81], CRM [71], TripoSR [65]) before and after refinement. In the questionnaire, we presented four views (azimuths: 45°, 135°, 225°, 315°; elevation: 15°)

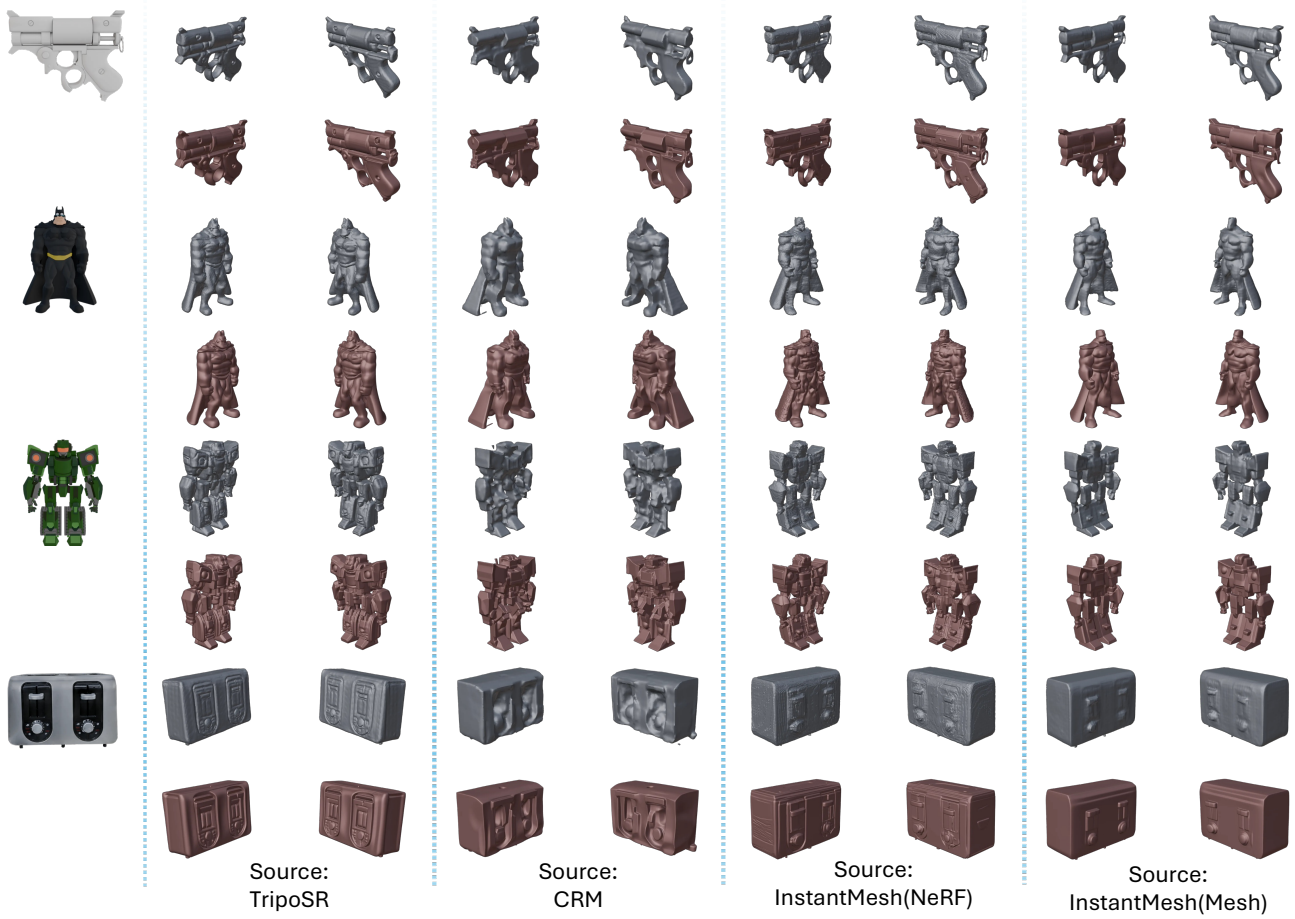

Figure 4. We apply our method on input meshes generated by different approaches (TripoSR [65], CRM [71], InstantMesh [81]). ■ represent coarse, ■ represent fine refinement results from our method. The top three objects are from Objaverse, while the bottom object is from GSO. More results can be found in the supplementary.

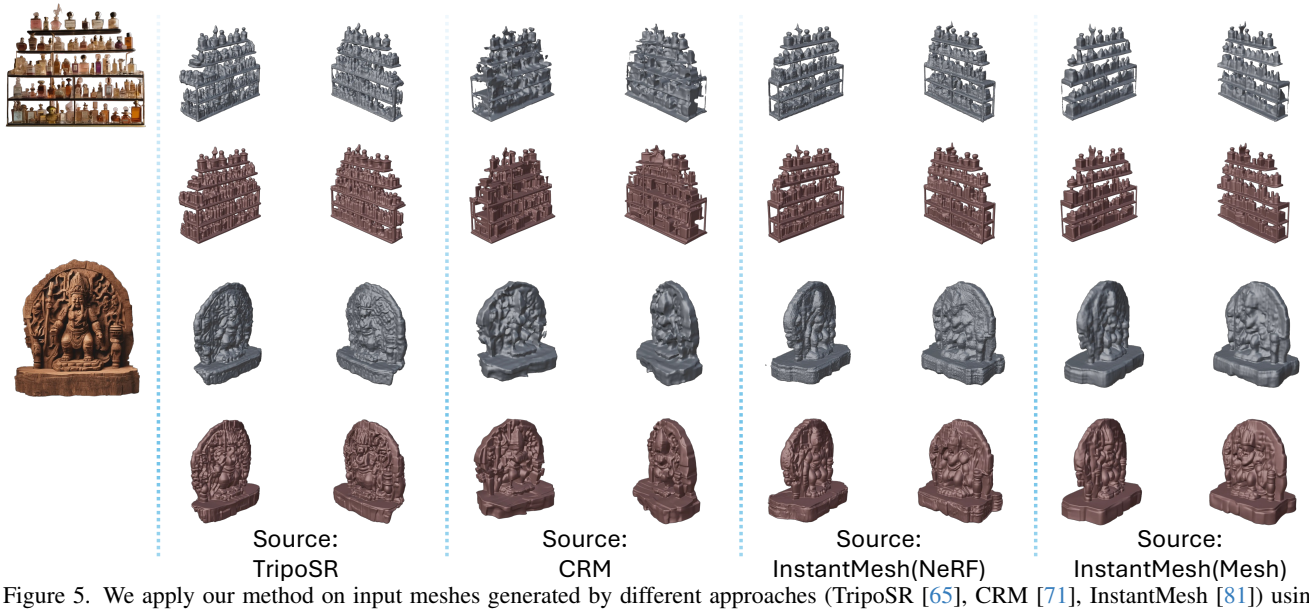

Figure 5. We apply our method on input meshes generated by different approaches (TripoSR [65], CRM [71], InstantMesh [81]) using GPTEval3D as input. ■ represent coarse, ■ represent fine refinement results from our method. More results can be found in the supplementary.

Figure 6. We apply our method on input meshes reconstructed by NeuS and generated by Rodin Gen-1 and Neural4D. ■ represent coarse, ■ represent fine refinement results from our method. More results can be found in the supplementary.

Method	Coarse	w/o Token matching	w/o Image condition	w/o Noise augmentation	Cross attention	Sorting	w/o training	Our full method
FID ↓	20.33	19.79	21.31	17.98	23.92	20.58	24.22	19.32

Table 2. FID scores evaluated on the Objaverse evaluation set (350 objects) for different model ablations.

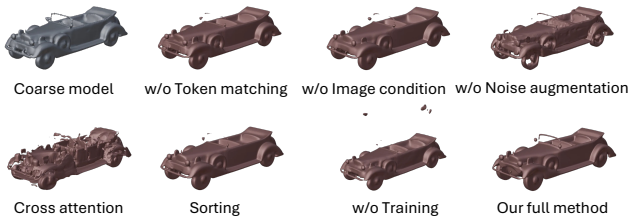

Figure 7. Visualization of inference results with different settings.

of both coarse and refined geometries, rendered with consistent code to ensure varied order and same color to avoid bias, and asked users: " *Please look at the following image and two 3D models and tell us which model you think is more detailed and closer to the input image.*". The statistics show that 94% of the votes chose that the geometry obtained by refining using our method is better, which strongly proved our method's refinement ability.

4.5. Ablation Study

In our ablation study, we tested the effects of removing token matching, using cross-attention to inject coarse geometry information, removing image prompts, removing noise augmentation, and using a point cloud sorting algorithm to query point cloud, inference without training for reference. For all ablation experiments, we trained for approximately 100,000 steps on a dataset of 40,000 objects (subsample from the training set), with the results tested on an evaluation set of 350 objects from Objaverse (same as in previous experiments). We used FID as the evaluation metric on the rendered images following the SDF-StyleGAN [98] settings. The FID results are shown in Tab 2, and the visualizations are presented in Fig 7.

Experimental results indicate that removing token matching slows convergence, as the model must simultaneously learn both refinement transformations and unnecessary operation (e.g., position swapping); it is hard to generate unseen details in the coarse geometry without the image condition; cross-attention modeling lacks the constraint of one-to-one token correspondence between coarse and fine models, making the noise-to-fine process inefficient and yielding worse geometric quality after refinement; applying point cloud sorting to the query point cloud yields no clear benefit, and inference without training is almost as effective in predicting coarse geometry.

Removing noise augmentation results in the lowest FID, as shown in Tab 2. We speculate that this is primarily due to two reasons: first, the Inception model [61], which computes FID, is not sensitive to the noise perception of the ensemble model, and second, the image conditioning ensures the results are more consistent with the ground truth, even without noise augmentation. While these factors contribute to low FID, the visual degradation remains clearly noticeable, as demonstrated in Fig 7 and further illustrated in the supplementary material.

5. Discussion and Conclusion

Limitation. Our method can robustly refine geometry with image prompt. However, it still struggles with ultra-precision details like extremly complex and small geometry shown in image. This issue might be solved by training a better 3D-VAE, collecting more diverse coarse and fine geometry pairs for training.

Conclusion. In this paper, we present a generative 3D geometry refinement method using an image as a prompt. We introduce a training technique called Token Matching for localized geometry refinement. This approach proves highly effective in both reconstruction and generation tasks, delivering refined results across diverse datasets, particularly for complex geometries.

DetailGen3D: Generative 3D Geometry Enhancement via Data-Dependent Flow

Supplementary Material

In this supplement, we first provide the implementation detail in Sec. 6. We also provide further experiment details in Sec. 7 and further discussion in Sec. 8. Finally, we provide additional visual results in Sec. 9. We encourage the readers to view our accompanying videos in the supplement, showcase the rotation of objects rendered with normals as presented in the paper.

6. Implementation Details

6.1. Model architecture

For the 3D-VAE encoder, after sampling N points from geometry's surface, we first randomly downsample it to $4\times M$, where M is the number of latent codes of each object and $M = 2048$, $N = 20480$. Next, obtaining query points X_0 by using farthest point sampling to further subsample it with $\frac{1}{4}$ ratio.

The 3D-VAE decoder, comprised 24 multi-head self attention layer and a multi-head cross attention layer. For the preset query points used in cross attention layer, it evenly distributed in 3D space, using for querying the corresponding spatial SDF value, which can adopt marching cube algorithm to convert it into mesh format.

For the Refinement DiT is comprised 24 DiT blocks with a width of 768, 12 attention heads, and a latent length of 2048, totaling 368M parameters. For the image prompt, we use the image feature extracted by DINO V2 [48]. The feedforward network in each DiT block consists of a two layer multi layer perception with GELU activation and the middle dimension is four times the input dimension.

6.2. Noise Augmentation

During training, we apply noise to z_0 according to the DDPM [23] linear schedule at 400 timesteps. During the inference stage, noise augmentation is optionally, which can reduce the impact of floating objects in the coarse model on the final refined results. For our experiment, we all add noise at 100 time steps.

6.3. Data Curation

To construct satisfying coarse-fine pairs, we choose to obtain coarse model by using Instant3D to reconstruct through four ortho views at a resolution of 512. It is worth mention that some of the objects are textureless. For the training dataset, we select Instant3D [32] to obtain coarse model, and the objects are all from Objaverse [11]. It is worth noting that the Instant3D we use is reimplemented by us because its code has not been released yet.

In order to align the coarse geometry with the fine geometry in space, we first normalize and rescale each object to fit within a bounding box of side length 1, then translate the object so that the bounding box is centered at the origin.

6.4. Training

In our training setup, we start with a learning rate of 1e-10 and warm it up to 1e-4 over 5,000 steps. We use a total batch size of 256. We train our DetailGen3D model for 1,000 epochs, which takes approximately eight days on eight A800 GPUs. When training the data-dependent rectified flow, we randomly zero the DINO features with a probability of 10% to enable classifier-free guidance during inference, thereby improving the quality of conditional generation. For the DINO V2 [48] checkpoint, we use the ViT-L/14 distilled with the registered version, downloaded from the official DINO V2 GitHub repository[3]. For the image prompt, we select a forward view with same camera intrinsic and extrinsic for training because using more views or highly random camera poses leads to longer training time. We believe that introducing camera pose embedding (e.g., plucker embedding) will help.

7. Experiment Details

7.1. Other Methods

We didn't compare with other shape detailization methods as listed in related work because our tasks are different and they cannot handle our evaluation set, as illustrated in Fig.A 8. They aim at increasing the resolution of extremely low resolution voxels to relative higher resolution (e.g., 16^3 to 64^3). Our evaluation set consists of up to 450 objects, which are comes from GSO and Objaverse, however, their evaluation set only consists of tens objects and comes from the same category in ShapeNet [4] and have strict requirements on the orientation of objects (Refining a car from Objaverse using ShaDDR [6]'s checkpoint—which is trained on car objects from ShapeNet—produces worse results due to the difference in orientation compared to the training set.) and the input's orientation shown in 8 is manually adjusted to meet ShaDDR's requirements.

7.2. Evaluation Dataset

For the Objaverse [11] evaluation set used, the IDs for models trained with different LRM methods are varied and not publicly available, which may lead to unfair comparisons between methods. However, we emphasize that our refinement model has never seen these 3D models during training,

[3] https://github.com/facebookresearch/dinov2

Algorithm 1 Data-Dependent Rectified Flow

procedure $\mathcal{Z}$(RectFlow((X_0, X_1)))
Inputs: Draws from a coupling (X_0, X_1) of π_0 and π_1; velocity model $v_\theta : \mathbb{R}^d \to \mathbb{R}^d$ with parameter θ.
Training:
1. Obtain π_0 and π_1 using $\pi_0 = \mathcal{E}(G_0)$ and $\pi_1 = \mathcal{E}(G_1)$, where $\mathcal{E}$ denotes the VAE encoder, G_0 represents the coarse geometry, and G_1 represents the fine geometry.
2. Optimize $\hat{\theta} = \arg\min_\theta \mathbb{E}\left[\|X_1 - X_0 - v(tX_1 + (1-t)X_0, t)\|^2\right]$, with $t \sim \text{Uniform}([0,1])$.
Sampling:
1. Start with $Z_0 \sim \pi_0$, where $\pi_0 = \mathcal{E}(G_0)$. Here, $\mathcal{E}$ denotes the VAE encoder, G_0 represents the coarse geometry.
2. Generate (Z_0, Z_1) by solving $\mathrm{d}Z_t = v_{\hat{\theta}}(Z_t, t)\mathrm{d}t$, obtaining $\{Z_t : t \in [0,1]\}$.
Return: $\mathcal{Z} = \{Z_t : t \in [0,1]\}$.
end procedure

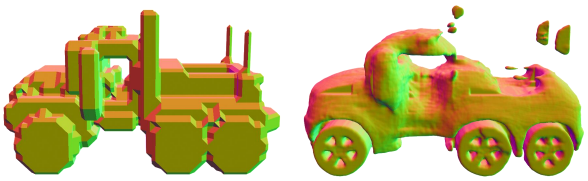

Figure 8. Coarse model (left) is degraded and rotated manually following ShaDDR training setting. The model refined by ShaDDR (right) is worse.

so its ability to refine the same object across different models fairly reflects the model's generalization capability.

7.3. SDF-stylegan Setting

For the feed-forward reconstruction and generation experiment, the FID [22] metric we evaluate following SDF-stylegan [98], rendering 20 views with preset random camera poses and same color (grey).

7.4. NeuS Reconstruction

For NeuS [67], to further improve speed, we used Instant-NSR [20] implementation. For the multi-view data, we rendered a uniformly distributed set of 40 views as input. The camera poses are elevation with -60°, -30°, 0°, 30°, 60° and azimuth with 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°.

8. Further Discussion

8.1. Token Matching

Although attempts were made to match latent code, the correspondence is difficult to model effectively. We explored the use of point cloud sorting algorithms, which work well for simple geometries but struggle with complex geometries due to their inability to preserve spatial relationships. Optimal Transport (OT) between coarse and fine query points can handle more complex shapes, but is computationally expensive and must be performed offline.

It is worth mentioning that using a learnable query in place of query points in the VAE does not solve the problem. The first reason is that the VAE quality with learnable queries is lower than that with query points. The second reason is that using learnable queries makes it impossible to ensure that the tokens with the same indices in the tensor, obtained after encoding the coarse geometry and fine geometry, have similar meanings.

Finally, we match the coarse geometry's latent code with fine geometry's latent code by applying nearest-neighbour algorithm to identify the closest fine geometry query points in the coarse geometry point cloud, using these as the coarse geometry's query points. While this design cannot theoretically guarantee one-to-one correspondence, it does so experimentally. This is because the 3D-VAE's two-stage downsampling process (i.e., downsampling point cloud X sampled from geometry surface to query points X_0)—random downsampling followed by farthest point sampling—results in query points X_0 of fine geometry being distributed far apart in space. As a result, the nearest-neighbour algorithm becomes a viable method for matching latent codes.

8.2. Data Curation

While our experimental results have shown our method has strong generalizability across different sources of coarse models, including both generation and reconstruction tasks, there is still room for improvement. Using only one type of LRM (e.g., Instant3D) may introduce bias, and we believe that applying geometry degradation and mixing multiple LRM reconstruction results will be of help.

8.3. Application

Our focus on geometry refinement stems from the fact that in many applications (e.g., design, simulation) replying on fine geometry, whereas color can be integrated later [53, 75]. For the texture, our method supports to simply reproject the original textures onto the refined mesh or

use separate texture-generation pipelines [89, 90]. This ensures high-quality geometry while preserving the flexibility to include color information as needed.

9. Additional Visual Results

We provide additional visual results in this section. Fig 9 shows GSO [13] generation results. Fig 10,11,12 shows Objaverse [11] generation results. Fig 13,14 shows GPTEval3D [77] generation results. Fig 17 shows GSO [77] generation results. Fig 18, 19, 20, 21, 22, shows Objaverse [11] generation results.

We also provide more ablation study visual results about noise augmentation, as presented in Fig 23, 24.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54 (10):105–112, 2011. 1

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 2

[3] Xin Cai, Zhiyuan You, Hailong Zhang, Wentao Liu, Jinwei Gu, and Tianfan Xue. Phocolens: Photorealistic and consistent reconstruction in lensless imaging. *arXiv preprint arXiv:2409.17996*, 2024. 2

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository, 2015. 1

[5] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction, 2024. 1

[6] Qimin Chen, Zhiqin Chen, Hang Zhou, and Hao Zhang. Shaddr: Interactive example-based geometry and texture generation via 3d shape detailization and differentiable rendering, 2023. 3, 1

[7] 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. 3

[8] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1538–1547, 2019. 1

[9] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation, 2023. 2

[10] Zhiqin Chen, Vladimir G. Kim, Matthew Fisher, Noam Aigerman, Hao Zhang, and Siddhartha Chaudhuri. Decorgan: 3d shape detailization by conditional refinement, 2021. 3

[11] 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, 2022. 3, 5, 1

[12] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects, 2023. 3

[13] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items, 2022. 5, 3

[14] Johannes S Fischer, Ming Gui, Pingchuan Ma, Nick Stracke, Stefan A Baumann, and Björn Ommer. Boosting latent diffusion with flow matching. *arXiv preprint arXiv:2312.07360*, 2023. 2

[15] Johannes S. Fischer, Ming Gui, Pingchuan Ma, Nick Stracke, Stefan A. Baumann, and Björn Ommer. Boosting latent diffusion with flow matching, 2024. 4

[16] Noa Fish, Lilach Perry, Amit Bermano, and Daniel Cohen-Or. Sketchpatch: sketch stylization via seamless patch-level synthesis. *ACM Trans. Graph.*, 39(6), 2020. 3

[17] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 1

[18] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997. 3

[19] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 1

[20] Yuan-Chen Guo. Instant neural surface reconstruction, 2022. https://github.com/bennyguo/instant-nsr-pl. 2

[21] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation, 2023. 3

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5, 2

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1

[24] Fangzhou Hong, Jiaxiang Tang, Ziang Cao, Min Shi, Tong Wu, Zhaoxi Chen, Shuai Yang, Tengfei Wang, Liang Pan, Dahua Lin, and Ziwei Liu. 3dtopia: Large text-to-3d generation model with hybrid diffusion priors, 2024. 3

[25] 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, 2024. 2, 3

[26] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 1

[27] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2821–2830, 2018. 1

[28] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017. 3

[29] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[30] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions, 2023. 3

[31] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2

[32] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model, 2023. 2, 3, 5, 6, 1

[33] Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d, 2023. 2, 3

[34] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, Dayou Chen, Jiajun He, Jiahao Li, Wenyue Li, Chen Zhang, Rongwei Quan, Jianxiang Lu, Jiabin Huang, Xiaoyan Yuan, Xiaoxiao Zheng, Yixuan Li, Jihong Zhang, Chao Zhang, Meng Chen, Jie Liu, Zheng Fang, Weiyan Wang, Jinbao Xue, Yangyu Tao, Jianchen Zhu, Kai Liu, Sihuan Lin, Yifu Sun, Yun Li, Dongdong Wang, Mingtao Chen, Zhichao Hu, Xiao Xiao, Yan Chen, Yuhong Liu, Wei Liu, Di Wang, Yong Yang, Jie Jiang, and Qinglin Lu. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding, 2024. 3

[35] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. *arXiv preprint arXiv:2311.11284*, 2023. 2

[36] 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, 2023. 2

[37] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion, 2023. 2, 3

[38] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization, 2023. 2, 3

[39] Minghua Liu, Chong Zeng, Xinyue Wei, Ruoxi Shi, Linghao Chen, Chao Xu, Mengqi Zhang, Zhaoning Wang, Xiaoshuai Zhang, Isabella Liu, Hongzhi Wu, and Hao Su. Meshformer: High-quality mesh generation with 3d-guided reconstruction model, 2024. 3

[40] Minghua Liu, Chong Zeng, Xinyue Wei, Ruoxi Shi, Linghao Chen, Chao Xu, Mengqi Zhang, Zhaoning Wang, Xiaoshuai Zhang, Isabella Liu, et al. Meshformer: High-quality mesh generation with 3d-guided reconstruction model. *arXiv preprint arXiv:2408.10198*, 2024. 2

[41] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023. 2, 3

[42] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow

straight and fast: Learning to generate and transfer data with rectified flow, 2022. 2, 3

[43] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image, 2024. 2, 3

[44] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion, 2023. 2, 3

[45] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[46] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. *ACM transactions on graphics (TOG)*, 24(3):536–543, 2005. 2

[47] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts, 2022. 3

[48] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 3, 4, 1

[49] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 3, 4

[50] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 2

[51] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d, 2023. 2, 3

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 5

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

[54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

[55] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 2

[56] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model, 2023. 2, 3

[57] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation, 2024. 2, 3

[58] J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion, 2022. 3

[59] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. *arXiv preprint arXiv:2310.16818*, 2023. 2

[60] Jingxiang Sun, Cheng Peng, Ruizhi Shao, Yuan-Chen Guo, Xiaochen Zhao, Yangguang Li, Yanpei Cao, Bo Zhang, and Yebin Liu. Dreamcraft3d++: Efficient hierarchical 3d generation with multi-plane reconstruction model. *arXiv preprint arXiv:2410.12928*, 2024. 3

[61] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 8

[62] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 2

[63] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation, 2024. 3

[64] G. Taubin. Curve and surface smoothing without shrinkage. In *Proceedings of IEEE International Conference on Computer Vision*, pages 852–857, 1995. 4

[65] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image, 2024. 3, 5, 6, 7

[66] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation, 2023. 2, 3

[67] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction, 2023. 1, 5, 6, 2

[68] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction, 2023. 2, 3

[69] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction, 2023. 1

[70] 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, 2023. 2

[71] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model, 2024. 3, 5, 6, 7

[72] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality mesh, 2024. 2, 3

[73] Chenglei Wu, Kiran Varanasi, Yebin Liu, Hans-Peter Seidel, and Christian Theobalt. Shading-based dynamic shape refinement from multi-view video under general illumination. In *2011 International Conference on Computer Vision*, pages 1108–1115. IEEE, 2011. 2

[74] Chenglei Wu, Bennett Wilburn, Yasuyuki Matsushita, and Christian Theobalt. High-quality shape from multi-view stereo and shading under general illumination. In *CVPR 2011*, pages 969–976. IEEE, 2011. 2

[75] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *arXiv preprint arXiv:2405.14832*, 2024. 2

[76] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer, 2024. 3

[77] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v(ision) is a human-aligned evaluator for text-to-3d generation. In *CVPR*, 2024. 5, 3

[78] 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. *arXiv preprint arXiv:2412.01506*, 2024. 3

[79] Di Xu, Qi Duan, Jianmin Zheng, Juyong Zhang, Jianfei Cai, and Tat-Jen Cham. Shading-based surface detail recovery under general unknown illumination. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):423–436, 2017. 2

[80] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models, 2023. 2

[81] 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, 2024. 3, 5, 6, 7

[82] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model, 2023. 2, 3

[83] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation, 2024. 3

[84] Yunhan Yang, Yukun Huang, Xiaoyang Wu, Yuan-Chen Guo, Song-Hai Zhang, Hengshuang Zhao, Tong He, and Xihui Liu. Dreamcomposer: Controllable 3d object generation via multi-view conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8111–8120, 2024. 3

[85] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 1

[86] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *CVPR*, 2024. 2

[87] Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild, 2024. 2

[88] Lap-Fai Yu, Sai-Kit Yeung, Yu-Wing Tai, and Stephen Lin. Shading-based shape refinement of rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1422, 2013. 2

[89] Xin Yu, Ze Yuan, Yuan-Chen Guo, Ying-Tian Liu, Jianhui Liu, Yangguang Li, Yan-Pei Cao, Ding Liang, and Xiaojuan Qi. Texgen: a generative diffusion model for mesh textures. *ACM Transactions on Graphics*, 43(6):1–14, 2024. 3

[90] Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. Paint3d: Paint anything 3d with lighting-less texture diffusion models, 2023. 3

[91] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models, 2023. 2, 3

[92] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: A structured and explicit radiance representation for 3d generative modeling, 2024. 2, 3

[93] Bowen Zhang, Tianyu Yang, Yu Li, Lei Zhang, and Xi Zhao. Compress3d: a compressed latent space for 3d generation from a single image, 2024. 3

[94] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting, 2024. 2, 3

[95] 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, 2024. 2, 3

[96] 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, 2023. 2

[97] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, Huiwen Shi, Sicong Liu, Junta Wu, Yihang Lian, Fan Yang, Ruining Tang, Zebin He, Xinzhou Wang, Jian Liu, Xuhui Zuo, Zhuo Chen, Biwen Lei, Haohan Weng, Jing Xu, Yiling Zhu, Xinhai Liu, Lixin Xu,

Changrong Hu, Shaoxiong Yang, Song Zhang, Yang Liu, Tianyu Huang, Lifu Wang, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Chao Zhang, Yonghao Tan, Jie Xiao, Yangyu Tao, Jianchen Zhu, Jinbao Xue, Kai Liu, Chongqing Zhao, Xinming Wu, Zhichao Hu, Lei Qin, Jianbing Peng, Zhan Li, Minghui Chen, Xipeng Zhang, Lin Niu, Paige Wang, Yingkai Wang, Haozhao Kuang, Zhongyi Fan, Xu Zheng, Weihao Zhuang, YingPing He, Tian Liu, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, Jingwei Huang, and Chunchao Guo. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025. 3

[98] Xin-Yang Zheng, Yang Liu, Peng-Shuai Wang, and Xin Tong. Sdf-stylegan: Implicit sdf-based stylegan for 3d shape generation. In *Comput. Graph. Forum (SGP)*, 2022. 5, 8, 2

[99] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10324–10335, 2024. 2, 3

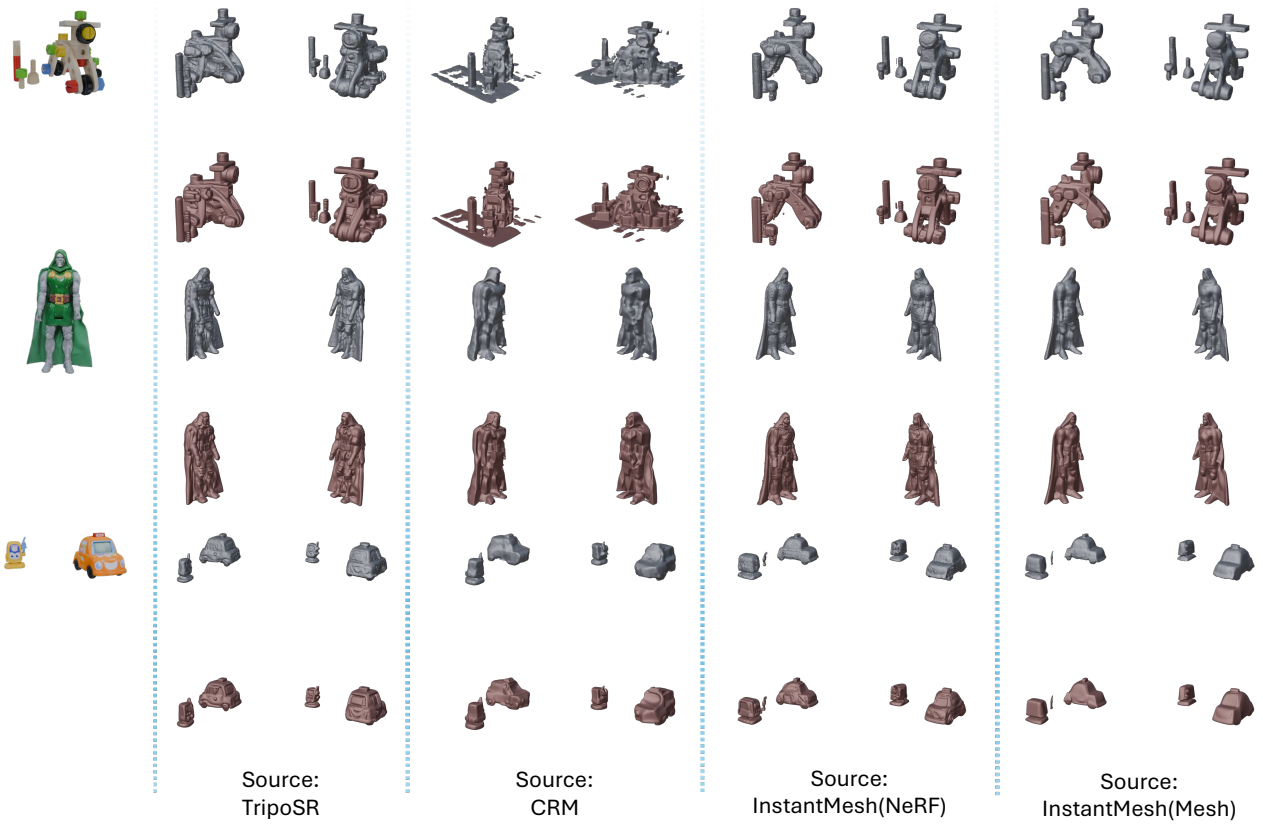

Figure 9. Generation results on GSO. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 10. Generation results on Objaverse. ■ represent coarse, ■ represent fine refinement results from our method.

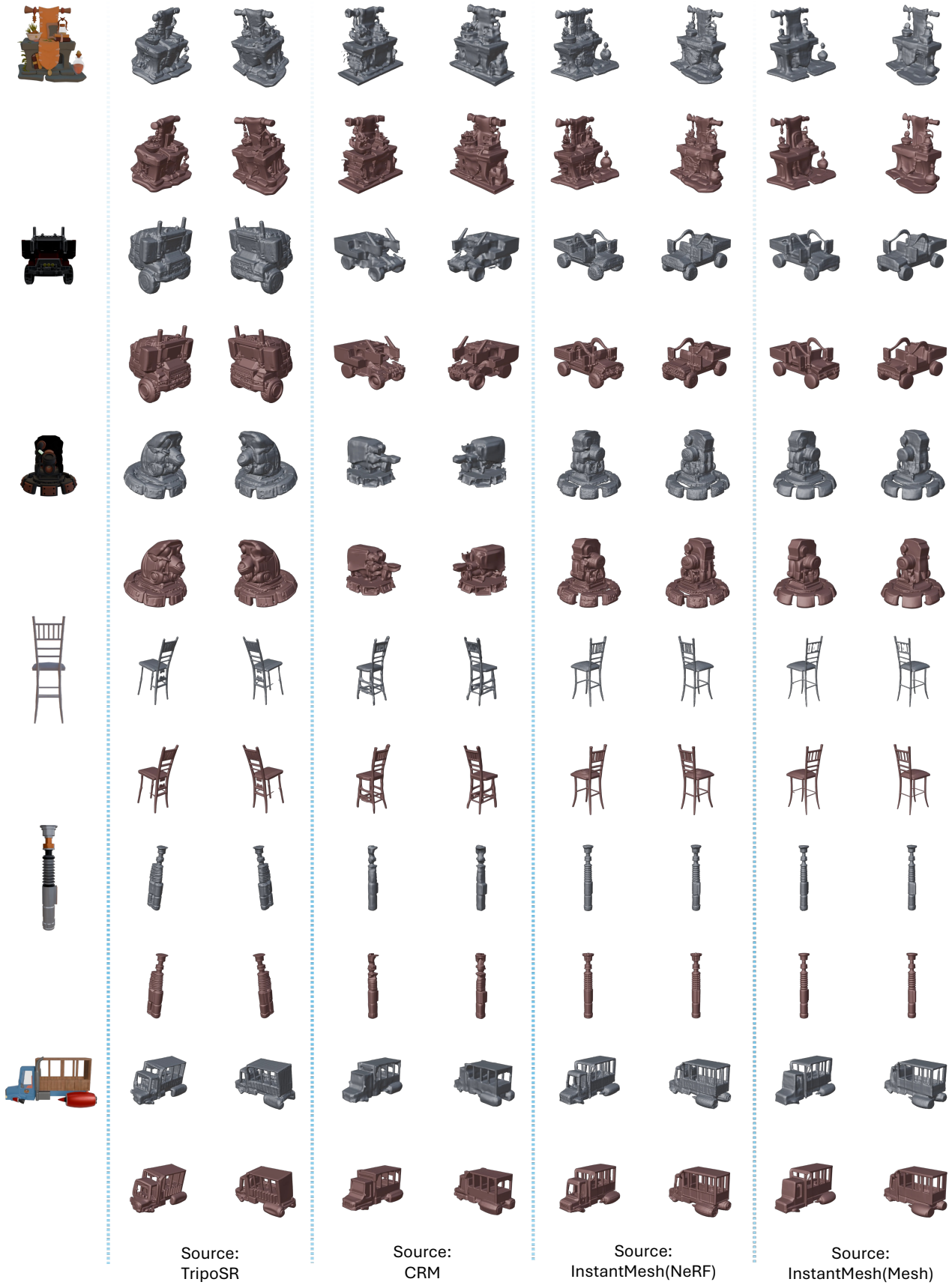

Figure 11. Generation results on Objaverse. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 12. Generation results on Objaverse. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 13. Generation results on GPTEval3D. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 14. Generation results on GPTEval3D. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 15. Optimization-based reconstruction results on GSO. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 16. Optimization-based reconstruction results on Objaverse. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 17. Feed-forward reconstruction results on GSO. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 18. Feed-forward reconstruction results on Objaverse. ■ represent coarse, ■ represent fine refinement results from our method.

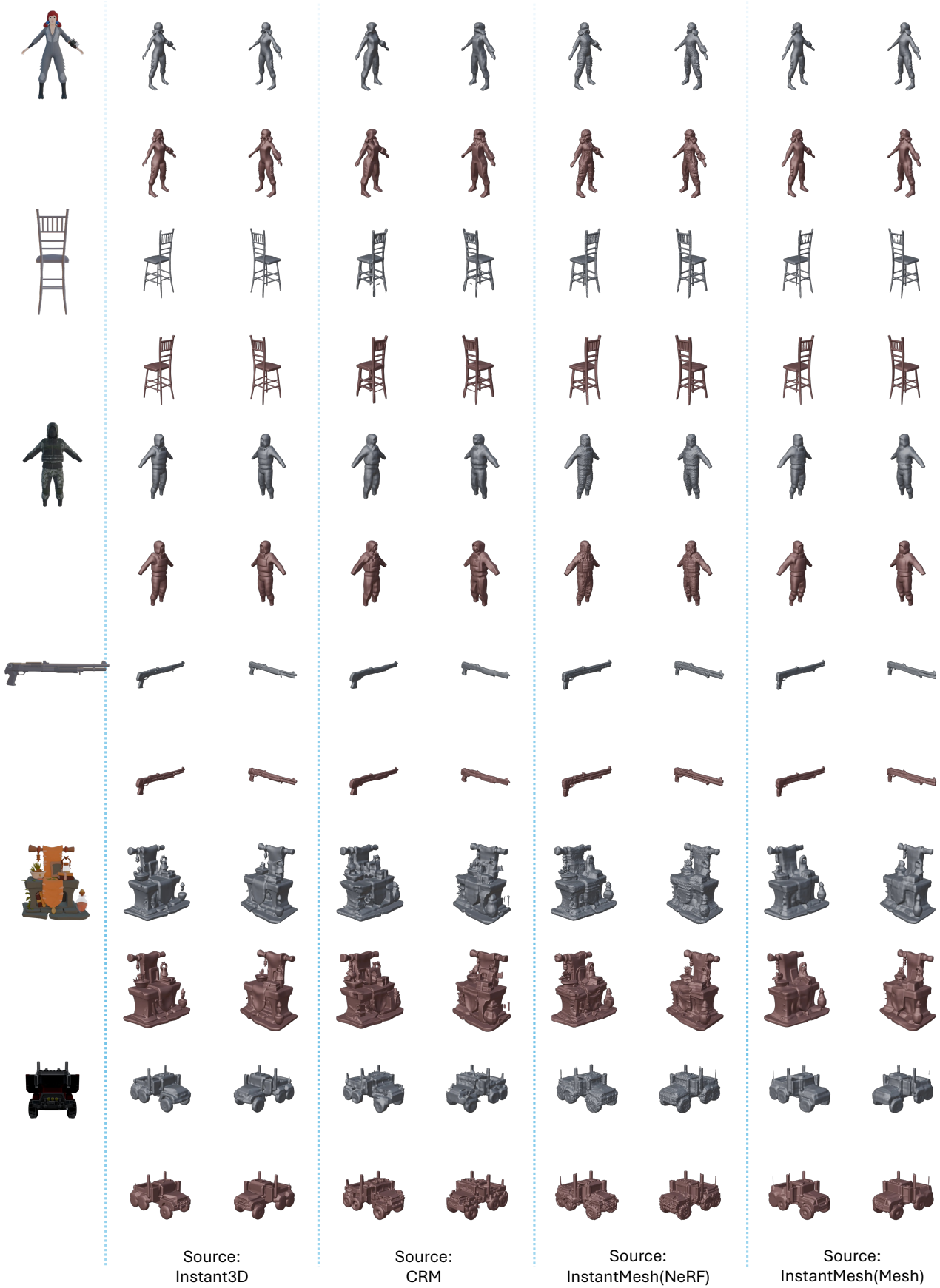

Figure 19. Feed-forward reconstruction results on Objaverse. represent coarse, represent fine refinement results from our method.

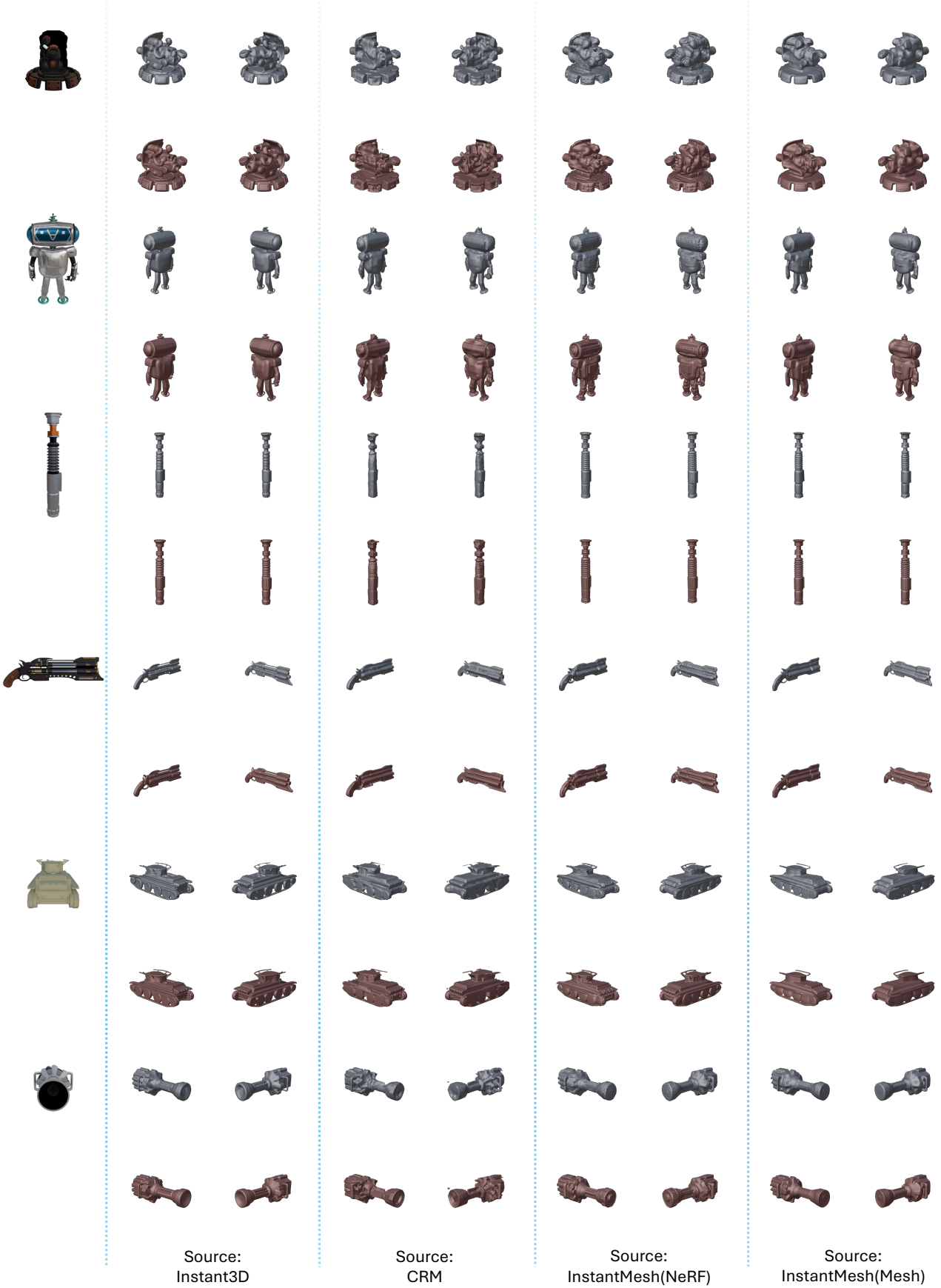

Figure 20. Feed-forward reconstruction results on Objaverse. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 21. Feed-forward reconstruction results on Objaverse. ■ represent coarse, ■ represent fine refinement results from our method.

Figure 22. Feed-forward reconstruction results on Objaverse. ■ represent coarse, ■ represent fine refinement results from our method.

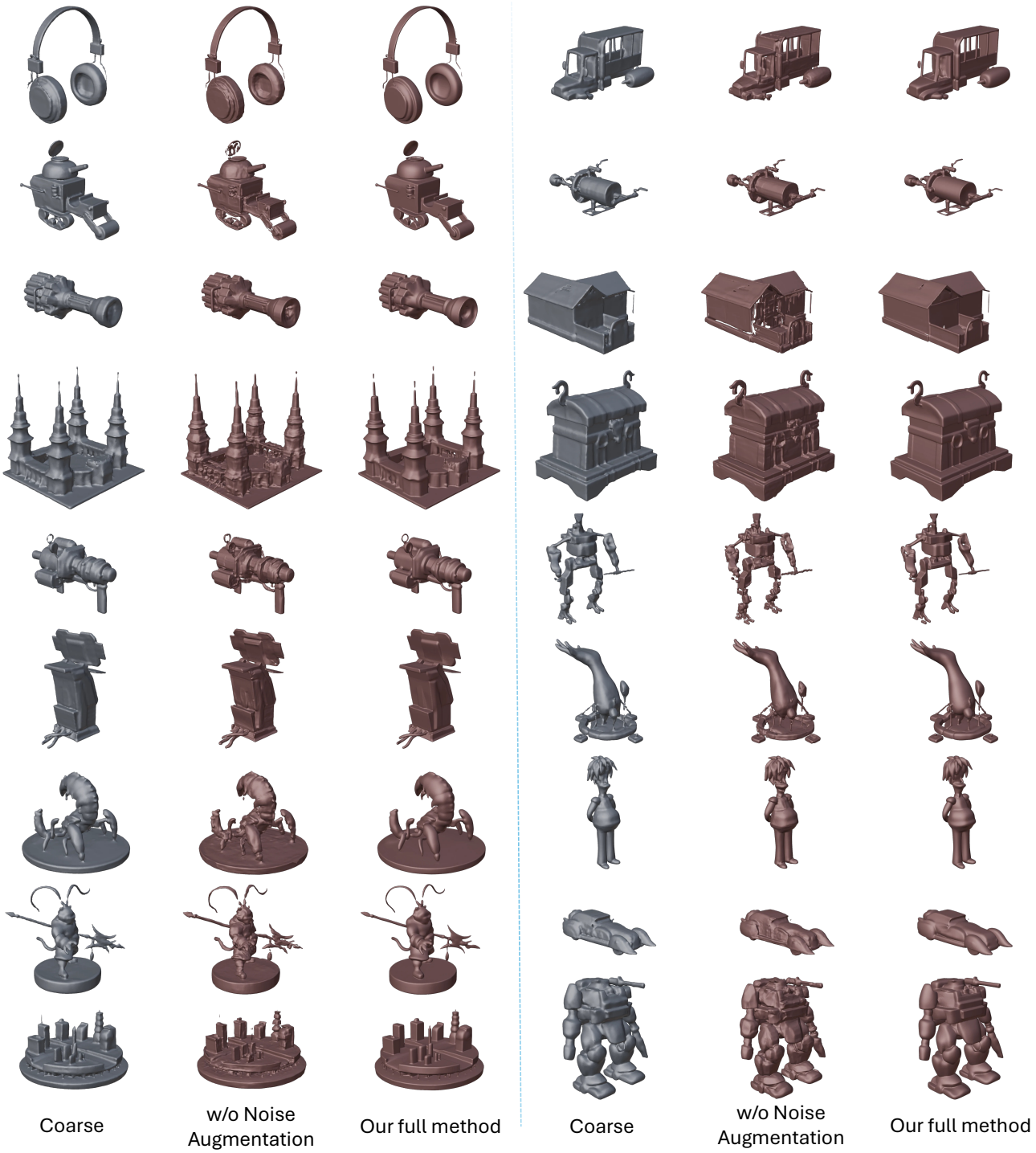

Figure 23. Ablation study visual results.

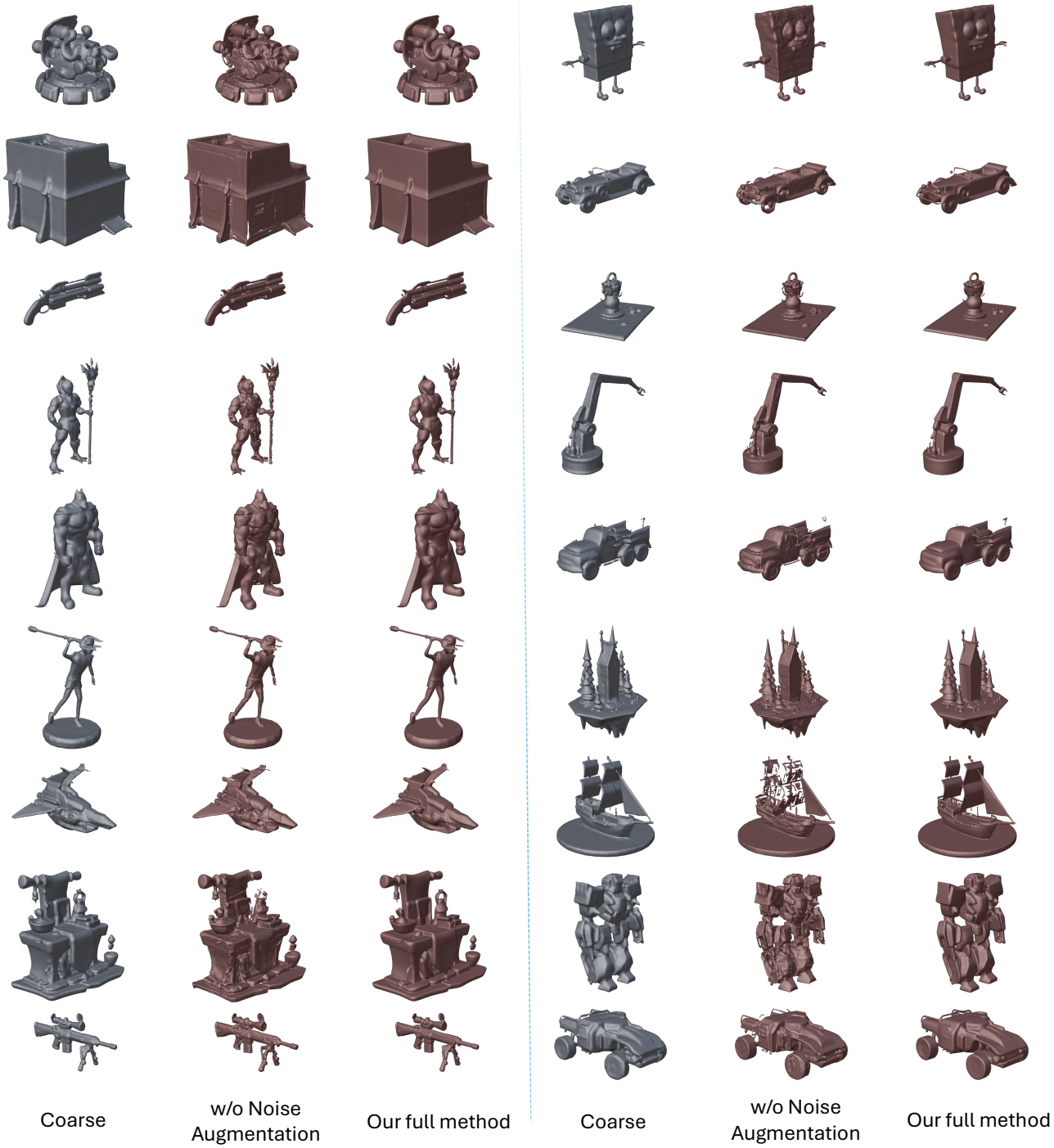

Figure 24. Ablation study visual results.