Restore3D: Breathing Life into Broken Objects with Shape and Texture Restoration

Anonymous Author(s)
Affiliation
Address
email

Figure 1: Completion Results. Our **Restore3D** is among the first to simultaneously restore the shape and texture of relatively complex and diverse objects, producing highly plausible and realistic results.

Abstract

Restoring incomplete or damaged 3D objects is crucial for cultural heritage preservation, occluded object reconstruction, and artistic design. Existing methods primarily focus on geometric completion, often neglecting texture restoration and struggling with relatively complex and diverse objects. We introduce Restore3D, a novel framework that simultaneously restores both the shape and texture of broken objects using multi-view images. To address limited training data, we develop an automated data generation pipeline that synthesizes paired incomplete-complete samples from large-scale 3D datasets. Central to Restore3D is a multi-view model, enhanced by a carefully designed Mask Self-Perceiver module with a Depth-Aware Mask Rectifier. The rectified masks, learned through the self-perceiver, facilitate an image integration and enhancement phase that preserves shape and texture patterns of incomplete objects and mitigates the low-resolution limitations of the base model, yielding high-resolution, semantically coherent, and view-consistent multi-view images. A coarse-to-fine reconstruction strategy is then employed to recover detailed textured 3D meshes from refined multi-view images. Comprehensive experiments show that Restore3D produces visually and geometrically faithful 3D textured meshes, outperforming existing methods and paving the way for more robust 3D object restoration. Project page: `https://nip-ss.github.io/NIPS-anonymous/`.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

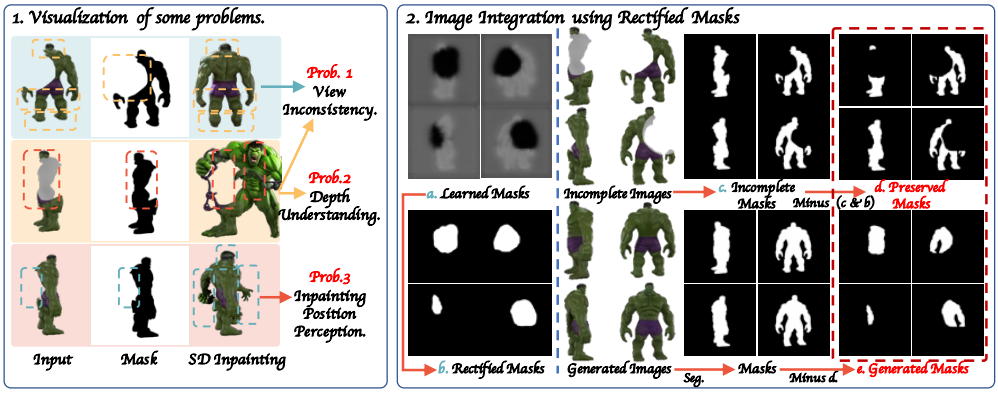

Figure 2: **The importance of masks.** In single-view inpainting, user-provided masks define the regions requiring inpainting. However, in a multi-view context, manually creating consistent masks across all views is impractical. Directly inverting object masks to serve as inpainting masks inevitably causes issues (see Prob. 1 & 3). Moreover, manually adjusting masks based on depth information (see Prob. 2) is labor-intensive and time-consuming. As shown in the right figure (a), our mask self-perceiver can automatically indicate the regions that need to be completed. By leveraging both preserved and generated masks (d & e), our approach retains the incomplete object's patterns, ensuring accurate and consistent multi-view inpainting. These masks are also used for the image enhancement stage to yield high-resolution restored images (see Fig. 4).

1 Introduction

Recent advances in 3D generation and reconstruction techniques [12, 45, 30, 29, 69, 31, 56] have demonstrated impressive capabilities, paving the way for innovative applications across diverse fields. Despite these strides, a significant gap remains in the comprehensive restoration of both shape and texture for broken or incomplete 3D objects. This challenge is particularly relevant for some applications such as cultural heritage preservation, occluded objects reconstruction, and artistic creation, where high-fidelity restoration/completion is crucial.

In this study, we aim to develop a robust framework that can simultaneously restore the shape and texture of incomplete 3D objects while handling complex and diverse data types. Key challenges in achieving this goal include: *i) Data Collection*. Existing 3D datasets [6, 16, 48] focus primarily on shape completion, often neglecting the equally critical aspect of texture restoration. Furthermore, these datasets typically contain simple objects. Creating a diverse, high-quality dataset remains labor-intensive and time-consuming. *ii) Complexity of Object Completion*. Addressing the intricacies of restoring complex and general objects requires a robust framework, as simpler methods often fall short. *iii) Consistency Preservation of Broken Parts*. Incomplete objects may exhibit varying degrees of degradation in shape and texture. Therefore, preserving the integrity of original components, including consistent color, style, and structural coherence, is crucial for realistic restoration.

To address these challenges, we propose several complementary solutions: **i) Synthetic Data Generation**. To overcome the limitations of existing datasets, we propose to synthesize paired broken and complete data. **ii) Leveraging Foundation Models**. Recent advancements in foundation models [23, 52, 50, 43, 28, 71] have demonstrated exceptional generalizability, due to their extensive architectures, large-scale datasets, and adaptability through fine-tuning. We incorporate foundation models to provide prior knowledge, enabling our framework to effectively handle complex and diverse cases. **iii) Task-Specific Structures**. While foundation models offer valuable priors, task-specific components are necessary to tailor their application. Motivated by studies [80, 73, 40], we guide these models toward optimal probability distributions with specialized modules, achieving more accurate and contextually appropriate restorations.

Concretely, we first produce an automatic pipeline to construct paired data, which uses the Boolean modifier in Blender. It offers diverse and large-scale data that are difficult to acquire manually. Second, we propose an innovative framework named **Restore3D**, comprising two key components, *i.e.*, **multi-view image inpainting and reconstruction**. There are several foundational models [52, 31, 69] in these two components that we can leverage prior knowledge to further handle more diverse incomplete objects effectively. However, simply applying foundational models to multi-view images introduces several **challenges**, as shown in Fig. 2, including: *1) View Inconsistency*: Generated results often

differ across views, leading to visual incoherence. *2) Depth Understanding*: Existing models often lack robust depth perception, resulting in failures to recognize occlusions and spatial relationships. *3) Inpainting Position Perception*: Accurately identifying regions requiring inpainting can be difficult, especially for large masks.

To address these issues, we propose a **multi-view** base model combined with a specially designed **mask self-perceiver** module incorporating a **depth-aware mask rectifier**. This module autonomously perceives and reconstructs missing components, preserving the integrity of original broken regions and ensuring consistent results across multiple views. Additionally, by leveraging the preserved and generated masks predicted by the self-perceiver, we can develop an image integration and enhancement pipeline (see Fig. 2 & 4), yielding high-quality and consistent results. To convert high-quality multi-view images into 3D objects, we employ large reconstruction models (LRMs)[23, 56, 69, 29, 63], which offer efficient single- and multi-view object reconstruction capabilities. To overcome the limitation of coarse outputs from these models, we adopt a coarse-to-fine refinement approach. Leveraging recent advances in surface normal prediction models[3, 72], we inject normal priors to progressively enhance geometric quality, and refine texture based on updated geometry by using enhanced images. This ensures that our refined shapes and textures maintain high fidelity, even for complex scenarios.

We conduct extensive experiments on Objaverse [17], GSO [18], and OmniObject3D [67] to validate the quality of inpainting and reconstruction. The results demonstrate that our inpainting method significantly outperforms previous approaches [36, 80, 50], *e.g.*, $\uparrow$ 13 in PSNR compared to Nerfiller [62]. By carefully designing a mask self-perceiver, our method can alleviate view inconsistency, understand depth concepts, and capture inpainting regions, achieving consistent structure and texture styles without requiring user-provided masks to indicate inpainting regions. For reconstruction, our approach enhances both geometric and texture quality as shown in Fig. 1, indicating that our proposed framework is capable of producing complete shapes and textures with relatively high fidelity compared to baseline methods [22, 69]. Overall, our contributions are summarized as follows,

- To the best of our knowledge, we are among the first to explore the completion of relatively complex shapes and textures. To support this task, we introduce an automated data synthesis pipeline that generates paired incomplete and complete shapes and textures, providing a rich source of training data named RestoreIt-3D.
- We propose Restore3D, a novel framework to tackle shape and texture completion through a combination of multi-view image inpainting and reconstruction. In multi-view image inpainting, we design a mask self-perceiver with a depth-aware mask rectifier for autonomous perception and reconstruction of missing components, ensuring preservation of original features. Moreover, we introduce an image integration and enhancement pipeline to restore fine details. We refine coarse meshes by using normal priors and enhanced images.
- Comprehensive experiments validate the effectiveness of Restore3D, demonstrating its ability to produce complete and high-quality textured meshes.

2 Related Work

2D Inpainting and Generation models 2D inpainting methods are designed to complete missing content in an image using a given image and mask. LaMa [54] utilizes fast Fourier convolutions, a large receptive field, and extensive training masks to effectively fill large missing areas, producing plausible inpainting results. Recent advancements in image generation [50, 80] have demonstrated superior performance and can be adapted for inpainting tasks with high-quality outcomes. RePaint [36] modifies the diffusion generation process, allowing it to be used for inpainting. NeRFiller [62] uses grid priors to make the 2D diffusion model produce more consistent multi-view inpainting results. However, these methods require a user-defined mask to specify the regions that need inpainting.

3D Generation and Completion Recent 3D generation models [61, 30, 9] showcase promising results. DreamFusion [45] and SJC [59] are first proposed to generate 3D assets from text using the strong 2D text-to-image generation model [50]. As 2D diffusion models easily lead to 3D inconsistency, some works [31, 82, 57, 55, 58, 70] focus on consistent multi-view image diffusion models. MVDream [52] uses 3D self-attention and camera embedding to achieve multi-view text-to-image generation. Considering the time-consuming nature of SDS-based methods, there are some works [20, 34, 29, 33, 56, 65, 35] that use multi-view diffusion models and reconstruction models.

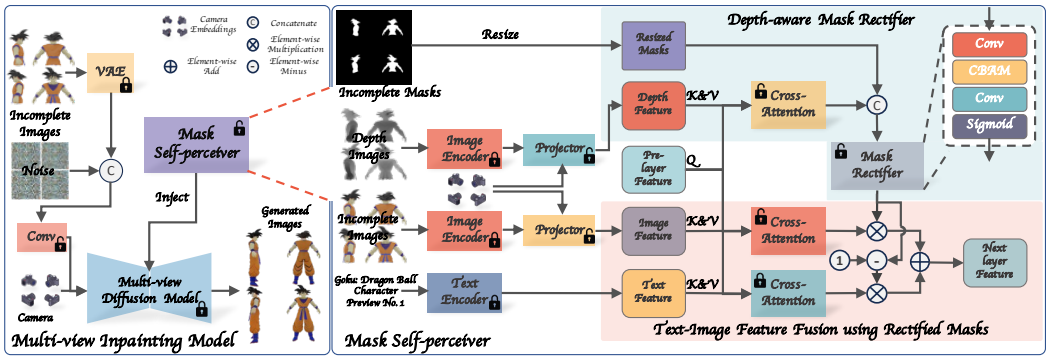

Figure 3: **An overview of multi-view image inpainting.** We carefully design a mask self-perceiver based on a multi-view diffusion model that composes the image and text features with a spatial mask predicted by a depth-aware mask rectifier, therefore the model can automatically perceive the missing part and further generate it meanwhile preserving the original parts.

Another line for 3D generation is that directly train 3D generative models using 3D representations like point cloud [42, 77, 37, 81], meshes [32, 21], neural fields [27, 1, 41, 24, 78, 19, 8]. In addition to 3D generation, recent 3D shape completion works [26, 79, 66, 68, 16, 15, 39, 44, 12, 13] usually use different types of 3D representations and networks to model global and local structures, *e.g.*, point cloud, sdf, GAN, VAE, and diffusion models. However, they all learn models on small-scale datasets, therefore the modeling capacity is limited compared with some 3D generation models trained on large-scale datasets (*e.g.*, Objavese [17]). Moreover, these works do not consider the texture.

Texture Generation. Several texture generation works [49, 5, 7] use an iteratively texturing strategy based on the pre-trained depth-to-image diffusion models, yielding high-quality texture. However, these methods tend to error lighting inherited from training data. Paint3D [76] proposes a shape-aware UV Inpainting and a shape-aware UVHD diffusion model to alleviate this situation. There is another line to learn texture. Texturify [53] employs texture maps on the surface of meshes and uses StyleGAN [25] to predict texture. Mesh2Tex [4] incorporates an implicit texture field for texture prediction. These methods are lacking in global information modeling. PointUV [75] first trains a diffusion model specifically for mesh texture generation, and the proposed coarse-to-fine framework allows it to enjoy the efficiency of 2D representation while enhancing 3D consistency. Other approaches like AUV-net [10], LTG [74], and TUVF [11] learn to generate UV-Maps for 3D shapes. However, they typically focus on the texture generation starting from a complete shape.

3 Method

3.1 Data Preparation

Motivation. We browse the datasets of related tasks and find that the existing datasets [6, 17, 67, 18, 14] are not sufficient to handle the shape and texture completion of broken objects, which suggests the need to construct specific broken and complete paired data. However, collecting large-scale paired data in the real world is *time-consuming and labor-intensive*. Thus we propose to *synthesize* broken and complete paired data.

Data Collection. We select the recent dataset, G-objaverse [46] that has *more diverse and general objects*, and sample about 83K 3D objects from this dataset.

Synthesis Pipeiline. Specifically, we propose an automatic data processing technique using Boolean operations (*i.e.*, Difference and Intersect) of Blender. Additionally, we equip the dataset with text captions using Cap3D [38]. Subsequently, we normalize and merge the prepared 3D data. The use of Boolean operations requires the introduction of another object. Therefore, we use an ico sphere or cube with random size and rotation angle and then randomly place them inside the 3D bounding box of the prepared 3D data to ensure that the objects can be realistically segmented. After that, it is essential to render this processed data in the format of RGB images to facilitate model learning. We execute the rendering at a resolution of 256×256. The camera settings include a randomly chosen elevation between -10° and 30°. Additionally, the azimuth values are uniformly rendered from 0° to

360° with a randomly sampled start view, producing a total of 32 images per object. The Fov of the camera is randomly from 35° to 45° and the distance is always 2.

3.2 Multi-view Image Inpainting

Motivation. Traditional single-view image inpainting methods [54, 50, 80] rely on the user-provided masks that indicate the areas to be inpainted. While this approach works well in the context of single-view images, it presents significant challenges when extended to multi-view contexts as shown in Fig. 2. *1. View inconsistency.* In a multi-view scenario, the user is required to manually provide a mask for each of the views (*e.g.*, four views in our case). This also introduces the risk of errors, as the mask needs to be accurately aligned across different perspectives to maintain 3D consistency. *2. Uncertainty Regarding Inpainting Areas.* These models cannot autonomously perceive the regions that require inpainting when a large mask is applied. Additionally, they do not incorporate depth perception, limiting their understanding of occlusion and spatial relationships. To address these challenges, we propose an innovative approach that enables the model to *ensure view consistency* and *self-perceive the mask*. Concretely, we design the following two parts.

Mask Self-perceiver. We propose a mask self-perceiver module based on a multi-view image generation model as shown in Fig. 3. It has two projectors that consist of transformer-based blocks and camera modulation layers, which project the depth and image features (f_d, f_r) extracted from CLIP [47] to the diffusion feature space. The camera modulation helps the model to discriminate the feature under different cameras. Then these projected features (p_d, p_r) will be fed to the respective cross-attention blocks as key and value ($\mathbf{K_d}, \mathbf{K_r}, \mathbf{V_d}, \mathbf{V_r}$). The process can be formulated as follows,

$$p_* = \mathbf{Proj}(f_*, c) = \mathbf{Trans}(\mathbf{Mod}(f_*, c)) \tag{1}$$

$$s_* = \mathbf{Softmax}(\frac{\mathbf{Q}\mathbf{K}_*^{\mathbf{T}}}{\sqrt{d}})\mathbf{V}_* \tag{2}$$

where f_* can be depth or image features, p_* is the projected features of them. Similarly, $s_*, \mathbf{K}_*$ and $\mathbf{V}_*$ are the results of p_* via cross-attention and linear layers. $\mathbf{Q}$ originates from the pre-layer features in the diffusion model.

Depth-aware Mask Rectifier. Since depth effectively captures the incomplete shape while disregarding texture information, the rectifier can focus solely on identifying the regions that require generation and preservation. Moreover, the depth can help the model understand the spatial relation and occlusion. Specifically, This module leverages depth features obtained after the cross-attention layer, along with incomplete masks, and inputs them into a mask rectifier. The rectifier then outputs a mask indicating where needs to be generated *i.e.*, leveraging the text features and where needs to be preserved *i.e.*, using the image features. The process can be formulated as follows,

$$\mathcal{M}_r = \mathbf{Sigmoid}(\mathbf{Conv}(\mathbf{CBAM}(\mathbf{Conv}[s_d, \mathcal{M}_o]))) \tag{3}$$

$$f_n = (\mathbf{1} - \mathcal{M}_r)s_t + \mathcal{M}_r s_r \tag{4}$$

where $\mathbf{Conv}$ is a convolution layer, and $\mathbf{CBAM}$ is Convolutional Block Attention Module [64]

Training objectives Given training samples, including incomplete images $\mathcal{I}$, depth images $\mathcal{D}$, incomplete masks $\mathcal{M}$, text prompts $\mathcal{P}$ and camera embedding $\mathcal{C}$, the multi-view inpainting loss can be formulated as follows,

$$\mathcal{L} = \min_{\theta} \mathbb{E}_{z,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t} \|\epsilon - \epsilon_\theta(z_t; t, \mathcal{I}, \mathcal{D}, \mathcal{M}, \mathcal{P}, \mathcal{C})\|_2^2. \tag{5}$$

3.3 Image Integration and Enhancement

Motivation. The input resolution of multi-view model is 256 x 256, which is subsequently encoded to 32 x 32 using a Variational Autoencoder. As a result, *local details are compressed, leading to a loss of clarity in both the original and generated regions of the image.* This compression often causes the inpainted part to be unclear, and the reconstructed image may lose fine details that are essential for achieving high-quality results. Moreover, *high-quality images* will *help* the next *reconstruction* stage to give accurate and detailed textured meshes. To address these challenges, we propose a pipeline that enables the model to *restore local details and preserve the original patterns*.

Enhancement Models. We explore two types of enhancement models. *Real-ESRGAN* [60] is effective at preserving the patterns of low-resolution images with minimal misalignment, making it ideal for recovering the overall structure. *ControlNet-Tile* [80] offers advanced capabilities for enhancing image details, but will modify the original pattern when a high denoising step is used.

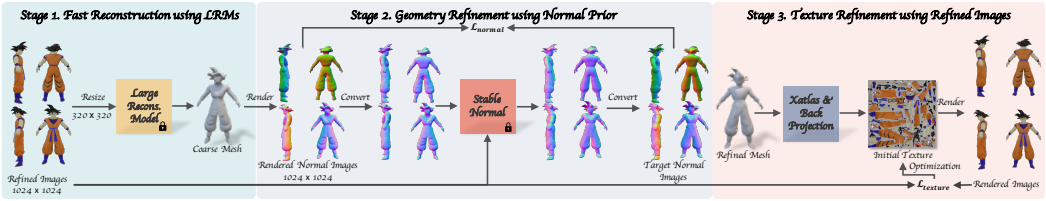

Figure 5: **Geometry and Texture Refinement.** We separately refine the geometry and texture of the coarse results inferred by LRMs [69].

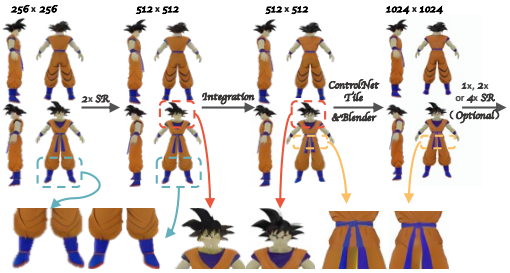

Figure 4: **Image Integration and Enhancement Pipeline using Rectified Masks.**

Based on these properties, we design the following enhancement pipeline. *1. Input resolution alignment using Real-ESRGAN.* Before integrating with the original images, we need to align the resolution. Using Real-ESRGAN effectively preserves the overall structure and does not introduce content that is not related to the original style. *2. Integration of generated and original parts using rectified masks.* As depicted in Fig. 4, this procedure infers the preserved and generated masks used to compose the images, which preserves the original parts as soon as possible. However, this procedure inevitably leads to some artifacts, *e.g.*, inconsistent color transitions. To address these artifacts, we leverage the mentioned property of ControlNet-Tile to enhance the images. *3. Image harmonizing using ControlNet-Tile with a blending strategy.* Directly using ControlNet-Tile will alter the original pattern and destroy the integration step. Inspired by previous works [2, 36], we incorporate a mask blending technique within the diffusion process. This technique helps maintain the original patterns, eliminates any gaps caused by integration in image space, and enhances the image quality.

3.4 Multi-view Image Reconstruction

Fast Reconstruction using Large Reconstruction Models (LRMs). Recent advancements in LRMs [23, 56, 69], which leverage sophisticated architectures, large-scale datasets, and extensive model parameters, have demonstrated impressive capabilities in 3D object reconstruction from single or sparse-view images. These models are particularly well-suited for tasks requiring fast mesh reconstruction. However, while LRMs can produce initial reconstructions efficiently, the results are often *coarse and lack the fine details* necessary for high-quality 3D representations. To address this limitation, we adopt a coarse-to-fine schema and refine the shapes and textures of the outputs generated by LRMs, separately, as shown in Fig. 5.

Geometry Refinement using Normal Prior. A key component in optimizing shape structure is to obtain high-quality surface normals. Recent surface normal estimation methods [3, 72] have demonstrated the ability to predict relatively accurate normals for in-the-wild monocular images or videos. Therefore, we can employ an *off-the-shelf* normal estimation model to provide normal priors and then use it to optimize the shape structure of 3D objects. Since these models are primarily trained on monocular images or videos, the predicted normals are typically in camera space. Thus we need to convert these normals into world space using camera extrinsic parameters. Specifically, we select StableNorm, a model that accepts coarse rendered normals and RGB images as inputs to predict refined normal outputs. The consistency of the rendered normals contributes to the stability and accuracy of the predicted normals, allowing for more precise geometry refinement.

Texture Refinement using Enhanced High-quality Images. Since the current shape differs from the coarse shape, the original texture no longer aligns with the updated geometry. Thus we propose to learn the textures that better match the optimized shape. Concretely, we can use Xatlas to obtain UV coordinates, enabling us to back-project the colors from the inpainted images onto the UV textures. After that, we treat the UV textures as parameters and use the enhanced high-quality images to optimize the texture maps.

Training Objectives. We apply a normal loss $\mathcal{L}_{normal}$ based on the rendered normals $\mathcal{I}_n$ and the target normals $\hat{\mathcal{I}}_n$. Additionally, we apply a mask loss $\mathcal{L}_{mask}$ to ensure that the optimization regions

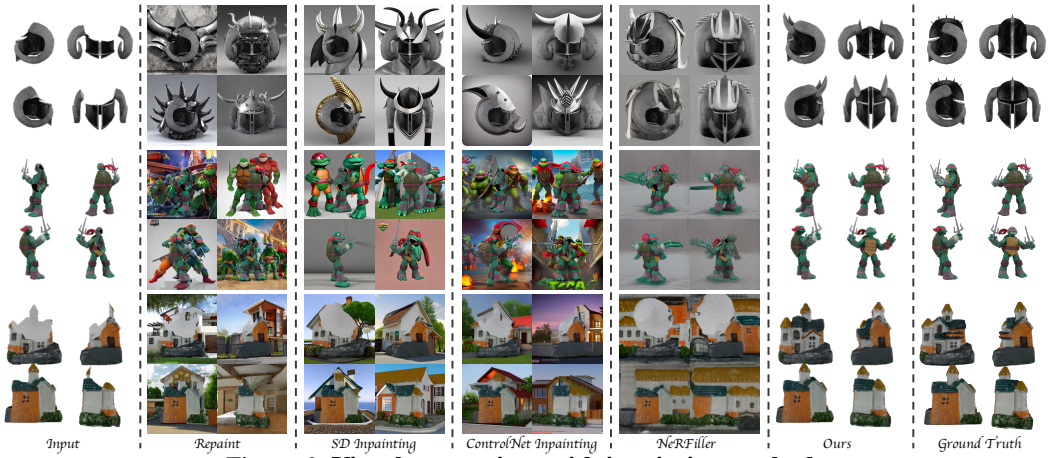

Figure 6: **Visual comparison with inpainting methods.**

Table 1: **Comparison with the previous inpainting and reconstruction methods.** $\star$ means inpainting, while $\triangle$ means using Depth-Anything [71] to obtain the depth images. Note that we do not apply image integration and enhancement pipelines. IM means InstantMesh [69].

(a) **Inpainting.**

Method	PSNR ↑	LPIPS ↓	FID ↓	SSIM ↑
Repaint [36]	10.55	0.31	69.57	0.76
SD $\star$ [50]	12.58	0.22	61.15	0.83
ControlNet $\star$ [80]	10.66	0.30	69.91	0.76
NeRFiller [62]	12.03	0.25	65.20	0.82
Ours $\triangle$	25.29	0.07	32.05	0.95
Ours	**25.50**	**0.06**	**31.82**	**0.95**

(b) **Reconstruction.**

Method	PSNR ↑	LPIPS ↓	CD ↓	F-Score ↑
Open-LRM [22]	16.90	0.15	0.011	0.179
IM [69]	20.60	0.11	0.006	0.321
Ours	**23.35**	**0.09**	**0.005**	**0.389**

are correctly aligned. The loss function is defined as follows,

$$\mathcal{L}_{shape} = \mathcal{L}_{normal} + \mathcal{L}_{mask} = \|\mathcal{I}_n - \hat{\mathcal{I}}_n\|_2^2 + \|\mathcal{M} - \hat{\mathcal{M}}\|_2^2. \tag{6}$$

To optimize the texture, we use a RGB loss $\mathcal{L}_{rgb}$ on the rendered images $\mathcal{I}_{rgb}$ and enhanced images $\hat{\mathcal{I}_{rgb}}$. The mask loss $\mathcal{L}_{mask}$ is also applied. Moreover, the SSIM $\mathcal{L}_{ssim}$ loss is introduced to improve the texture quality. The loss functions are defined as follows,

$$\mathcal{L}_{tex} = \mathcal{L}_{rgb} + \mathcal{L}_{mask} + \lambda\mathcal{L}_{ssim} = \|\mathcal{I}_{rgb} - \hat{\mathcal{I}_{rgb}}\|_2^2 + \|\mathcal{M} - \hat{\mathcal{M}}\|_2^2 + \lambda\mathbf{SSIM}(\mathcal{I}, \hat{\mathcal{I}}), \tag{7}$$

where λ is a weight parameter.

4 Experiments

Dataset & Metrics. For model training, we sample approximately 83K data from the G-objaverse [46] dataset and process them using our proposed pipeline. For model testing, we sample approximately 350 data from the GSO [18], Omniobject [67], and Objaverse [17] datasets. **Inpainting.** To assess image quality, We choose Peak Signal-to-Noise Ratio (PSNR), Frechet Inception Distance (FID), Learned Perceptual Image Patch Similarity (LPIPS), and Structural Similarity Index Measure (SSIM). **Reconstruction.** In addition to the metrics mentioned above, we evaluate geometry quality using Chamfer Distance (CD) and F-scores.

4.1 Inpainting Results.

Baselines. We compare our method with single-view image inpainting, *i.e.*, Repaint[36], Stable-Diffusion [50], Controlnet [80], and a multi-view inpainting method, *i.e.*, Nerfiller [62].

Qualitative Comparison. As shown in Fig. 6, the results demonstrate that our model produces plausible and coherent inpainting outcomes. Previous methods require user-provided masks to guide the model in generating missing parts. However, when given a relatively large mask, these methods struggle to capture the inherent structure of the objects, leading to less accurate and coherent inpainting.

Table 2: **Ablation studies for multi-view inpainting and reconstruction.** GR and TR mean geometry and texture refinements.

(a) **Inpainting.**

Method	PSNR ↑	LPIPS ↓	SSIM ↑
IF	22.65	0.14	0.90
IF + Conv	26.53	0.08	0.94
IF + Conv + DMR	**29.44**	**0.06**	**0.95**

(b) **Reconstruction.**

Method	PSNR ↑	LPIPS ↓	CD ↓	F-Score ↑
Baseline	20.60	0.11	0.006	0.321
GR	-	-	**0.005**	**0.389**
GR + TR	**23.35**	**0.09**	**0.005**	**0.389**

In contrast, our approach does not require predefined inpainting masks. Instead, it autonomously perceives and reconstructs the missing regions, capturing the underlying structure of the object without manual intervention. This capability allows our method to produce high-quality, structurally consistent inpainting results.

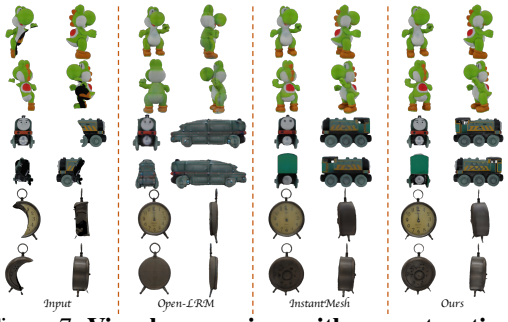

Figure 7: **Visual comparison with reconstruction methods.**

Quantitative Comparison. As illustrated in Table 1a, we observe the following: **1)** Our approach achieves the best performance in restoring shape and texture. **2)** When applying depth images predicted by Depth-Anything [71], our method yields results comparable to those obtained with ground truth depths. **3)** The compared methods produce noticeably inferior results in terms of inpainting quality.

4.2 Reconstruction Results.

Baselines. We compare our method against both single-view and multi-view LRMs, including LRM [22, 23] and InstantMesh [69]. For single-view baselines, we input the front-view image.

Quantitative & Qualitative Comparison. As shown in Table. 1b, our method achieves superior rendered image quality and geometry accuracy, with a substantial improvement over baseline methods. In Fig. 7, it is evident that our approach delivers clearer details and the most accurate geometry among the compared methods. **Training time.** Our approach is highly efficient, requiring 20 seconds per object for geometry and texture refinements.

4.3 Ablation Study

Multi-view Inpainting. We conduct ablation studies on the proposed multi-view inpainting module in the following components: **1) IF.** Only inputting incomplete images to the cross-attention layers. 2) **Conv.** Concatenating noise and incomplete images to a learnable convolutional layer. 3) **DMR.** Adding the designed Depth-aware Mask Rectifier. As shown in Table 2a, the results improve progressively with each added component, and using all designed components achieves the highest results. In the qualitative comparison shown in Fig. 8b, 1) IF Only: the model captures the general style of the object but lacks an understanding of spatial relationships and structure. 2) IF + Conv: This enables the model to capture spatial positioning and understand object structure. However, it is still prone to color inaccuracies, especially in areas like the head (blended with error black color). Additionally, the region that needs to be preserved is changed. 3) IF + Conv + DMR: This allows the model to improve its ability to handle occlusions and spatial relationships, producing the best inpainting quality, with coherent colors and well-preserved spatial structure.

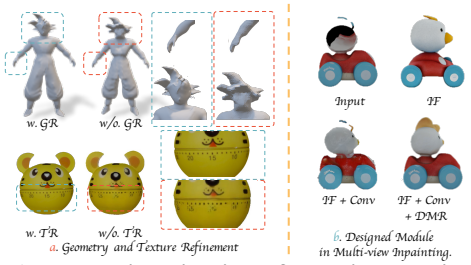

Figure 8: **Visualization of ablation studies.**

Reconstruction. We evaluate the impact of the following components: 1) Geometry Refinement (GR), and 2) Texture Refinement (TR). In Table 2b and Fig. 8a, incorporating GR leads to substantial improvements in geometry quality. TR improves the visual quality of the rendered images.

Figure 9: **Different lighting settings.**

Figure 10: **Visualization of occlusion cases.**

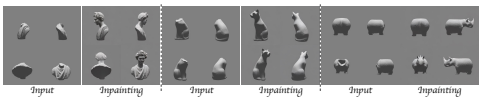

Figure 11: **Visual results on BBD [51].**

Table 3: **Different lighting settings.**

Method	PSNR ↑	LPIPS ↓	SSIM ↑
Top area light	25.18	0.06	0.95
Multiple area lights	25.50	0.06	0.95
Environment light	25.28	0.06	0.95

Table 4: **Occlusion results.**

Method	PSNR ↑	LPIPS ↓	SSIM ↑
1-view	27.16	0.06	0.95
4-view	25.62	0.07	0.95

Table 5: **BBD [51] results**

Method	PSNR ↑	LPIPS ↓	SSIM ↑
SD-inpainting	12.02	0.74	0.53
ControlNet	14.50	0.59	0.71
NeRFiller	17.66	0.52	0.79
Ours	**25.09**	**0.10**	**0.95**

Different lights. We render our test samples with different lights and test our inpainting model on these rendered images. In Table 3 and Fig. 9, the results show our model can achieve promising results under different lighting settings.

5 Application

Our Restore3D can be directly used for some applications:

Object Restoration. We test our model on the validation set of Breaking Bad Dataset (BBD) [51], as shown in Fig. 11 and Table 5. This dataset is synthesized by a physically based method that simulates the natural destruction process of geometric objects.

Occluded Object Reconstruction. We arrange either a single object or four objects to create occluded scenarios with one view and four views, respectively, based on our 350 test samples. As shown in Table 4 and Fig. 10, the results indicate that the one-view occlusion scenario achieves higher performance, as the occluded regions can be inferred more easily from the visible areas. When applying four-view occlusion, our model still demonstrates strong performance. In addition, we present a real-world example in Fig. 10.

Figure 12: **Text-guided editing results.**

3D Object Editing. We can position a cube or sphere over the target region for editing and use a Boolean operation to segment the object. This enables us to render the object as an incomplete image. We then process them using our inpainting model with a text prompt for editing. Finally, we apply the reconstruction model. In Fig. 12, our approach successfully handles simple editing scenarios.

6 Conclusion

In this paper, we propose a novel framework named Restore3D, consisting of multi-view image inpainting and reconstruction, to simultaneously complete both the shape and texture of broken 3D objects. To facilitate this task, we develop an automated data processing pipeline that collects pair-wise data from a large-scale dataset [17]. In the multi-view image inpainting, we design a mask self-perceiver with a depth-aware mask rectifier. This component autonomously identifies and reconstructs missing regions while preserving the original patterns. To address the low resolution resulting from the base model [52], we implement an image integration and enhancement pipeline, allowing for seamless integration and detail enhancement by learned masks. For the reconstruction stage, we employ an LRM to quickly generate a coarse result, followed by separate geometry refinement using normal priors and texture refinement using enhanced images. Through this designed framework, our model produces coherent completions of broken objects as illustrated in Fig. 1.

References

[1] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *CVPR*, 2023.

[2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, June 2022.

[3] Gwangbin Bae and Andrew J. Davison. Rethinking inductive biases for surface normal estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[4] Alexey Bokhovkin, Shubham Tulsiani, and Angela Dai. Mesh2tex: Generating mesh textures from image queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8918–8928, 2023.

[5] Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4169–4181, 2023.

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[7] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023.

[8] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *ICCV*, 2023.

[9] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023.

[10] Zhiqin Chen, Kangxue Yin, and Sanja Fidler. Auv-net: Learning aligned uv maps for texture transfer and synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1465–1474, 2022.

[11] An-Chieh Cheng, Xueting Li, Sifei Liu, and Xiaolong Wang. Tuvf: Learning generalizable texture uv radiance fields. *arXiv preprint arXiv:2305.03040*, 2023.

[12] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *CVPR*, 2023.

[13] Ruihang Chu, Enze Xie, Shentong Mo, Zhenguo Li, Matthias Nießner, Chi-Wing Fu, and Jiaya Jia. Diffcomplete: Diffusion-based generative 3d shape completion, 2023.

[14] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. Abo: Dataset and benchmarks for real-world 3d object understanding. *CVPR*, 2022.

[15] Angela Dai and Matthias Nießner. Scan2mesh: From unstructured range scans to 3d meshes, 2019.

[16] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5868–5877, 2017.

[17] 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 *CVPR*, 2023.

[18] 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.

[19] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. *arXiv preprint arXiv:2303.17015*, 2023.

[20] Hugging Face. One-2-3-45. `https://huggingface.co/spaces/One-2-3-45/One-2-3-45`, 2023.

[21] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *NeurIPS*, 2022.

[22] Zexin He and Tengfei Wang. Openlrm: Open-source large reconstruction models. `https://github.com/3DTopia/OpenLRM`, 2023.

[23] 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.

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

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[26] Yoni Kasten, Ohad Rahamim, and Gal Chechik. Point-cloud completion with pretrained text-to-image diffusion models, 2023.

[27] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *CVPR*, 2023.

[28] 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.

[29] 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.

[30] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023.

[31] 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. In *ICCV*, 2023.

[32] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. In *ICLR*, 2023.

[33] 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.

[34] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*, pages 210–227. Springer, 2022.

[35] Yuanxun Lu, Jingyang Zhang, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, Long Quan, Xun Cao, and Yao Yao. Direct2.5: Diverse text-to-3d generation via multi-view 2.5d diffusion. *Computer Vision and Pattern Recognition (CVPR)*, 2024.

[36] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022.

[37] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021.

[38] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models, 2023.

[39] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022.

[40] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[41] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *CVPR*, 2023.

[42] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[43] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[44] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network, 2021.

[45] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[46] 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.

[47] 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.

[48] Yuchen Rao, Yinyu Nie, and Angela Dai. Patchcomplete: Learning multi-resolution patch priors for 3d shape completion on unseen categories, 2022.

[49] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023.

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[51] Silvia Sellán, Yun-Chun Chen, Ziyi Wu, Animesh Garg, and Alec Jacobson. Breaking bad: A dataset for geometric fracture and reassembly, 2022.

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

[53] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Texturify: Generating textures on 3d shape surfaces. In *European Conference on Computer Vision*, pages 72–88. Springer, 2022.

[54] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021.

[55] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. *arXiv preprint arXiv:2306.07881*, 2023.

[56] 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.

[57] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv preprint arXiv:2307.01097*, 2023.

[58] Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Joshua B Tenenbaum, Frédo Durand, William T Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *arXiv preprint arXiv:2306.11719*, 2023.

[59] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, 2023.

[60] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)*.

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

[62] Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. Nerfiller: Completing scenes via generative 3d inpainting. In *CVPR*, 2024.

[63] 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 meshes, 2025.

[64] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module, 2018.

[65] Kailu Wu, Fangfu Liu, Zhihan Cai, Runjie Yan, Hanyang Wang, Yating Hu, Yueqi Duan, and Kaisheng Ma. Unique3d: High-quality and efficient 3d mesh generation from a single image, 2024.

[66] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 281–296. Springer, 2020.

[67] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Liang Pan Jiawei Ren, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[68] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer, 2021.

[69] 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.

[70] 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.

[71] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

[72] Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *ACM Transactions on Graphics (TOG)*, 2024.

[73] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. 2023.

[74] Rui Yu, Yue Dong, Pieter Peers, and Xin Tong. Learning texture generators for 3d shape collections from internet photo sets. In *British Machine Vision Conference*, 2021.

[75] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengzhe Liu, and Xiaojuan Qi. Texture generation on 3d meshes with point-uv diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4206–4216, 2023.

[76] 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.

[77] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *NeurIPS*, 2022.

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

[79] Junzhe Zhang, Xinyi Chen, Zhongang Cai, Liang Pan, Haiyu Zhao, Shuai Yi, Chai Kiat Yeo, Bo Dai, and Chen Change Loy. Unsupervised 3d shape completion through gan inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1768–1777, 2021.

[80] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[81] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021.

[82] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *CVPR*, 2023.

A Technical Appendices and Supplementary Material

A.1 Preliminary

Multi-view Diffusion models. Extending 2D generation models to the multi-view domain has been explored in various works [31, 52]. These extensions often incorporate modifications like adding camera conditions and adjusting the attention mechanisms to enable effective multi-view synthesis. In this paper, we adopt MVDream as our base model. MVDream modifies the spatial attention mechanism in Stable Diffusion [50], allowing the attention to focus on corresponding features across different views.

A.2 Implementation Details

We train the multi-view inpainting model using four NVIDIA A100 GPUs. We use the Adam optimizer and incorporate classifier-free guidance. The training is conducted with a learning rate of 1e-4 and a batch size of 256. MVDream is utilized as the base model for multi-view inpainting, while InstantMesh is employed as the large reconstruction model. The input consists of 4-view images. For the sampling process, we employ DDIM with 50 steps and a guidance scale of 5.0.

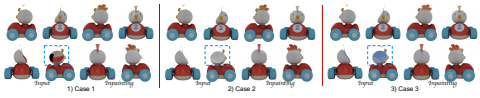

Figure 13: **Different color types.**

Table 6: **Ablation studies of views.**

Method	PSNR ↑	LPIPS ↓	FID ↓	SSIM ↑
4-view	25.50	0.06	31.82	0.95
6-view	25.00	0.07	24.70	0.95
8-view	25.17	0.07	20.49	0.95

A.3 More Results

More views. Although our model is trained on a 4-view setting, our model can be directly used to process inputs with more views. As shown in Table 6, the results show that their performance is comparable to the 4-view setting.

Different color types on the broken plane. As shown in Fig. 13, altering the broken plane (blue dotted box) with different colors does not affect our model's ability to complete the broken objects. This further validates that our model effectively distinguishes between regions that need to be preserved and those that require generation.

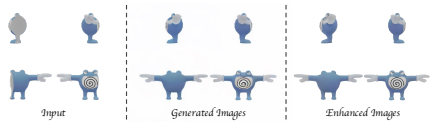

Figure 14: **Visualization of image integration and enhancement**.

Image Integration and Enhancement As shown in Fig. 14, we provide some results of this pipeline. The results show that the proposed pipeline restores the original pattern and improves the image quality.

Inpainting and reconstruction results on full GSO dataset (1030 Objects). As shown in Table 7 and Table 8, our model achieves the best performance on both inpainting and reconstruction results.

A.4 Limitations

Our approach builds upon a base model and thus inevitably inherits some of its limitations. For instance, the low resolution of the input restricts the ability to capture very fine details, such as the facial features of characters, even with the application of enhancement techniques. In addition, there is still a lot of room to enrich the quality of geometry and material details in the reconstruction.

Table 7: **Inpainting results on GSO**.

Method	PSNR ↑	LPIPS ↓	FID ↓	SSIM ↑
SD-inpainting	13.52	0.68	67.79	0.55
ControlNet	12.63	0.70	83.46	0.51
NeRFiller	17.07	0.60	75.24	0.72
Ours	**26.02**	**0.06**	**11.12**	**0.94**

Table 8: **Reconstruction results on GSO**.

Method	PSNR ↑	LPIPS ↓	CD ↓	F-Score ↑
Open-LRM	17.56	0.15	0.014	0.15
IM	22.15	0.11	**0.002**	0.36
Ours	**24.74**	**0.08**	**0.002**	**0.43**

A.5 Broader Impacts.

Object restoration will help cultural heritage preservation: restoring historical artifacts, sculptures, and architectural elements with accuracy. Negative impact: the ability to create highly accurate replicas can be misused for fraudulent purposes, such as creating counterfeit artifacts, artworks, or products, which can deceive consumers and harm original creators.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See § 1, we introduce our contribution and the scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See § A.4, we discuss our limitations.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See § 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer:[No]

   Justification: We will release the code and data after the publication of our paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See § 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not include it.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See § A.2. We provide it in the implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We acknowledge the NeurIPS Code of Ethics and obey them in our paper

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See § A.5. We provide the potential positive societal impacts and negative societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our model is based on publicly available datasets and models.

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the datasets and models.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not introduce new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not include this.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not include this.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.