3D Mesh Editing using Masked LRMs

Will Gao[1,2] Dilin Wang[2] Yuchen Fan[2] Aljaz Bozic[2] Tuur Stuyck[2]
Zhengqin Li[2] Zhao Dong[2] Rakesh Ranjan[2] Nikolaos Sarafianos[2]
[1]University of Chicago, [2]Meta Reality Labs

MaskedLRM Website

Abstract

We present a novel approach to shape editing, building on recent progress in 3D reconstruction from multi-view images. We formulate shape editing as a conditional reconstruction problem, where the model must reconstruct the input shape with the exception of a specified 3D region, in which the geometry should be generated from the conditional signal. To this end, we train a conditional Large Reconstruction Model (LRM) for masked reconstruction, using multi-view consistent masks rendered from a randomly generated 3D occlusion, and using one clean viewpoint as the conditional signal. During inference, we manually define a 3D region to edit and provide an edited image from a canonical viewpoint to fill that region. We demonstrate that, in just a single forward pass, our method not only preserves the input geometry in the unmasked region through reconstruction capabilities on par with SoTA, but is also expressive enough to perform a variety of mesh edits from a single image guidance that past works struggle with, while being $2-10\times$ *faster than the top-performing prior work.*

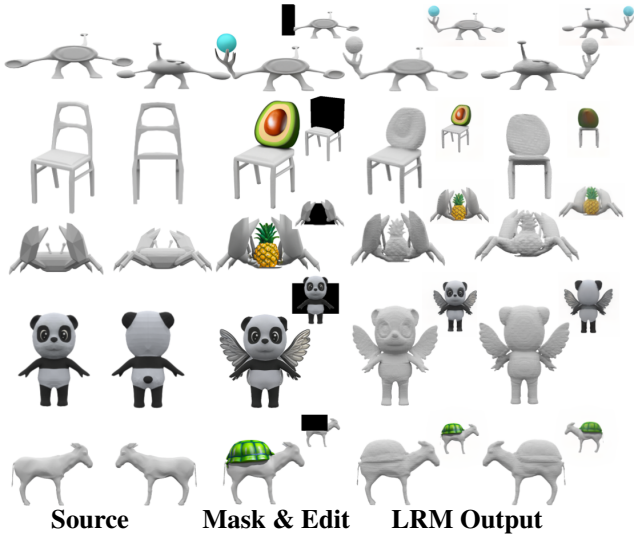

Figure 1. **Mesh Editing using MaskedLRMs**: The inputs comprise front/back views of the source mesh (1^{st} column) and a frontal image used as the conditional view. The 2^{nd} column shows the masked area rendered from the front (inset) and a 2D edit. The last column shows our generated mesh from the front/back views.

1. Introduction

Automated 3D content generation has been at the forefront of computer vision and graphics research, due to applications in various visual mediums like games, animation, simulation, and more recently, virtual and augmented reality. As research on neural methods for content generation has progressed, there has been significant progress in modifying and applying well-studied 2D methods into the 3D domain.

Recent developments in 3D content generation initially followed a similar path to 2D content generation. Operating in 3D voxel space instead of pixel space, models like VAEs [28, 55, 56, 70] and GANs [14, 98] were built, and trained on small-scale datasets [9]. These works often demonstrated limited editing capabilities through simple latent operations on their learned representations. Efforts have been made to extend generative diffusion to 3D [45, 53, 82, 99]. There has also been work done in generative autoregressive models for 3D, which tokenize 3D data in a unique way [60, 62, 77, 79]. Furthermore, neural representation techniques such as NeRFs [58] and Gaussian splatting [40] have introduced an entirely new paradigm for 3D content generation.

Despite significant progress in 3D content generation from scratch, research in editing the shape of *existing* 3D models is underdeveloped. Image editing methods benefit from a nearly endless source of data from scraping the internet, while 3D assets typically require a higher level of expertise and specialized tools to create and thus are scarce in comparison. The difference in scale is staggering, with the largest image datasets containing billions of samples [72] while the largest 3D datasets contain only millions [20]. A common approach to tackling the issue of 3D data scarcity is to exploit existing text-image foundation models. Recent efforts in 3D editing involve using these huge models to

provide guidance to an optimization process by giving them differentiably rendered images of the manipulated geometry as input [6, 27, 57, 59]. While these approaches demonstrated some success, they face several major challenges. Firstly, the gradients obtained using foundation models as guidance are often extremely noisy, leading to unstable and unpredictable optimizations [84]. Furthermore, since these methods often use text as input in lieu of visual input, they are hard to control. Finally, these techniques typically directly optimize properties of an explicit 3D mesh, which severely constrains the type of possible edits. For example, it is impossible to add a hole to a shape, since such a modification is not topology-preserving.

Recent works follow a different path and utilize a two-stage approach, placing the brunt of the "creative effort" onto 2D models, using them to generate and edit content. Then, a pipeline that lifts 2D images into 3D content produces the final output [44, 90]. Thus, by giving the model edited image inputs, a 3D edit is obtained. However, these methods rely on diffusion models that produce multi-view images [50, 52, 54, 75] which then are passed to a 3D reconstruction model [34, 89]. While editing a single image is no longer a challenging task, this multi-view generation procedure often suffers from ambiguous 3D structure in the occluded regions and does not accurately reconstruct what a shape looks like from every viewpoint. Efforts have been made to adapt multi-view diffusion models specifically for text-driven editing instead of the single-view-to-multi-view task [5, 24]. As we qualitatively demonstrate, editing multi-view images in a realistic manner remains a challenging task.

Our proposed approach falls into the second direction: lifting 2D images to 3D. Instead of using a 3D model to simply reconstruct geometry, our model is inherently trained to "inpaint" regions of multi-view images. The inpainting task is performed directly in 3D, instead of in multi-view image space. Specifically, the inputs to our method are a set of masked renders and a single clean conditional image that is provided to infer the missing information from. Our approach solves the issues present in both approaches to shape editing. In contrast to optimization methods, our model is efficient as it constructs shapes in a single, fast forward pass. Furthermore, the output of our model is highly predictable, as it is trained to reconstruct and inpaint geometry to a high degree of accuracy. This predictability gives a high degree of control to our method via the conditioning image. Our approach addresses the multi-view consistency and ambiguity problems of reconstruction methods by relying on a single conditional image while propagating the conditional signal to the rest of the multi-view inputs.

A key challenge is designing a training procedure that allows the model to learn how to use the conditional information in a multi-view consistent manner. To accomplish this, we introduce a new 3D masking strategy. We mask each input view in a consistent manner by rendering an explicit occluding mesh. Then, by supervising both the occluded and unoccluded regions with multi-view reconstruction targets, our model learns to not only fill in the occluded region, but also to accurately reconstruct the rest of the shape. Unlike previous works such as NeRFiller [88] which used fixed masks at a per-scene basis, training with randomly generated masks allows our model to generalize to arbitrary shapes and test-time masks. We demonstrate that this training method allows our model to be used downstream for editing tasks while maintaining strong quantitative performance on reconstruction baselines. By manually defining an editing region analogous to the train-time occlusions, and using a single edited canonical view, users can use our model to generate a shape that is faithful both to the original shape, and the edited content. In summary, our contributions are as follows:

- We design a novel conditional LRM trained with a new 3D-consistent multi-view masking strategy that enables our LRM to generalize to *arbitrary* masks during inference.
- Despite not being our primary intention, our architecture matches SoTA reconstruction metrics, while concurrently learns to use the conditional input to fill 3D occlusions.
- We show that our LRM can be used for 3D shape editing while being $2-10\times$ faster than optimization- and LRM-based edit methods. It synthesizes edits that optimization cannot (*e.g.* genus changes) and does not suffer from the multi-view consistency and occlusion ambiguity issues that approaches trained without masking suffer from.

2. Related Work

Large Reconstruction Models: LRM [34] and its recently introduced variants [8, 32, 44, 81, 85, 89–91, 95] showcase the solid capabilities of the transformer architecture for sparse reconstruction. Trained on large-scale 3D [19, 20] and multi-view image datasets [93], these models reconstruct geometry and texture details from sparse inputs or a single image in a feed-forward manner. Most LRMs focus on reconstructing radiance fields, which cannot be consumed by standard graphics pipelines for editing. MeshLRM [89] and InstantMesh [90] extract mesh and texture maps, but it remains a challenging to perform shape editing in an intuitive and fast manner. Furthermore, while these models achieve quite high reconstruction quality when given at least four complete views as input [44, 95], the problem is much more ambiguous when given only a single (possibly edited) image [34]. In this work we investigate how to utilize the LRM representation power for 3D shape editing, given a handful incomplete views as input for the shape reconstruction. This makes the reconstructed geometry of the non-edited content match significantly better to the original geometry, while ensuring view-consistency of the edited parts.

Shape Editing: Editing 3D shapes has been an active area of research for at least four decades. Early works focused

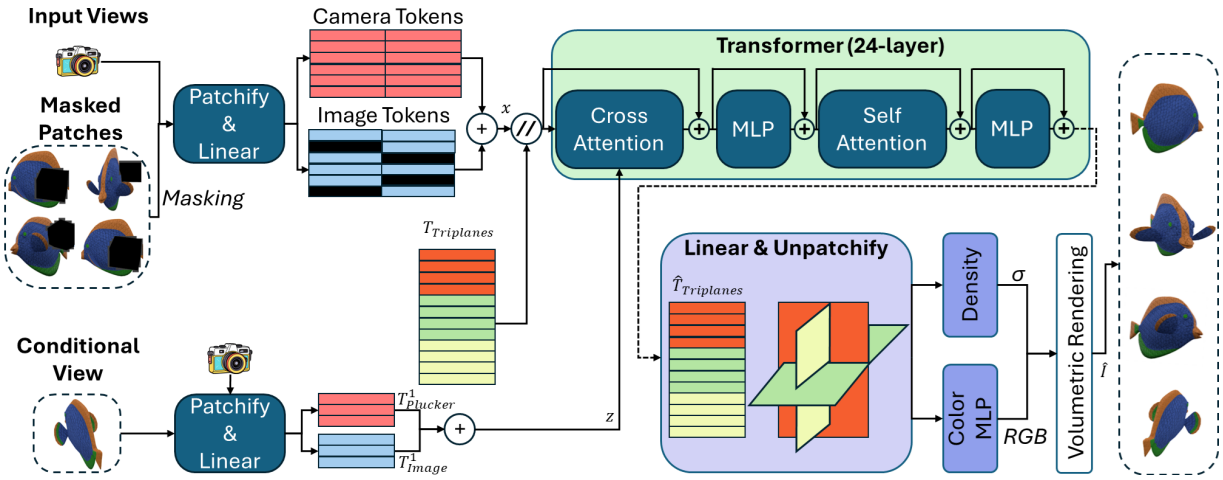

Figure 2. **Training Pipeline**. The images and camera poses are patchified and projected into tokens. A random 3D mask is generated and tokens corresponding to occluded patches are replaced by a learnable mask token. Camera and image tokens are summed and concatenated with learnable triplane tokens to form the transformer input. A clean conditional image is tokenized, forming the cross-attention input. The output triplane tokens are upsampled and decoded into colors and SDF values, which are transformed into densities for volumetric rendering.

on deformation [17, 73], cut and paste [7, 68], Laplacian surface editing [26, 48, 76] or fusion [39]. Recent works have tackled this task from different viewpoints depending on the geometry representation, the losses, and the downstream application. Regarding representation, research has been conducted on implicit surface editing [15, 33], voxels [74], mesh-based deformations [27, 41, 71], NeRF [3, 11, 30, 36, 38, 88, 94] and Gaussian splatting [13, 42, 67, 87]. Another line of work focused on generative local editing using diffusion models. MagicClay [6] sculpted 3D meshes using a 2D hybrid representation where part of the geometry is frozen and the remaining is optimized using SDS loss [46, 64]. 3D shape editing has been explored in the context of sketches, [49, 61, 78], faces [2, 10, 25, 29, 65] or in an interactive manner [21]. Recent approaches build upon progress in LRMs [34, 89], performing multi-view editing using diffusion models and then using LRMs to reconstruct an edited shape [5, 24, 66]. In contrast, our work introduces a novel architecture trained on multi-view consistent masked data that bypasses the need for inconsistent diffusion editing and enables 3D mesh editing within seconds.

Masked Vision Models: The original Denoising Autoencoder [83] used random noise as a "masking" method on images during training with the objective of learning better semantic representations. More recently, methods using transformers convert images into sequences and predict unknown pixels [4, 12] which culminated in the development of the Vision Transformer (ViT) [22] as the backbone of modern masked image models. Models like the Masked Autoencoder [31] use ViTs to process randomly masked tokens, where every token represents a distinct, non-overlapping image patch Research in diffusion models which also uses random noise as a "masking" procedure, has exploded in popularity, producing increasingly impressive generated images. By taking random Gaussian noise and constraining it to a specific region, diffusion models can be used for image inpainting [18, 43]. Masked autoencoders have been built for 3D data types such as point clouds [37, 63, 97], meshes [47], and NeRFs [35], with each work developing a different way to "patchify" their respective 3D representations. Point clouds have the most natural representation for next token prediction [62, 77], while efforts have also been made into tokenizing triangle meshes for generation as sequences of vertices and faces [60, 79]. Our paper presents a new approach to combine masking with LRMs for editing.

3. Method

Our large reconstruction model, shown in Figure 2, reconstructs a 3D shape from input images. Specifically, the model maps a sparse set of renders from various viewpoints into a latent triplane representation. We sample this representation to obtain latent features at different 3D points, which are then decoded into distance and RGB values for volumetric rendering. At training, we predict output renders from arbitrary camera poses. During inference, we use marching cubes to produce the reconstructed geometry. Unlike existing LRMs, our model uses a conditional branch to accept an additional view of the target shape. The inputs are then corrupted by a random masking procedure, forcing the model to learn to "inpaint" the input views using the conditional branch signal.

3.1. Masked LRM: Architecture

Image and Pose Tokens. The raw input to our model is a set of images with known camera parameters. During both training and inference, the input shapes are oriented in an arbitrary manner. Since we cannot guarantee a canonical viewpoint in the data, we remove the dependence on absolute poses by computing all camera parameters relative to the

first randomly selected image which we use as the conditional input. These camera parameters are represented by Plücker rays, forming a 6-channel grid with the same spatial dimensions as RGB pixels. We apply the standard ViT tokenization [22] to the image and the Plücker rays independently, dividing both grids into non-overlapping patches, and linearly projecting them into a high-dimensional embedding.

Masking. After the input images are tokenized, we randomly select a subset of tokens to mask out. For general masked image modeling, [31] demonstrated that dropping out random patches from the encoded image enabled a desirable balance between reconstruction and learned representation quality. However, since our goal is to train a model that fills in the missing geometry from the content of a single clean view, it is not suitable to occlude random patches since they lack correspondence for each input view. Instead, we require a structured, *3D-consistent* form of occlusion. Specifically, we generate a 3D rectangular mesh with uniformly random side lengths. We then render the depth map of this mesh from the same cameras as the input images, obtaining a set of multi-view consistent occlusions. Patches containing pixels that would be occluded by this random mesh are masked out. Instead of dropping the masked patches entirely as in [31], we propose to replace them with a learnable token. This does not suffer the same train-test gap, as occluded images are passed to the model during inference as well. This allows the model to maintain the 3D spatial context of the occlusion. Hence, our masking strategy is specifically designed with downstream editing of an occluded shape in mind.

Model Formulation. Using the above input tokenization and masking procedures, we can write a complete description of our model. Let $\mathcal{S}$ be a shape rendered from n camera poses described by the Plücker ray coordinates $\{\mathbf{C_i}\}_{i=1}^n$ producing RGB renders $\{\mathbf{I_i}\}_{i=1}^n$. The input token sequence to our model for any image are given by:

$$T^i_{\text{Image}} = \textbf{PatchEmbed}(\mathbf{I}_i), T^i_{\text{Plücker}} = \textbf{PatchEmbed}(\mathbf{C}_i), \tag{1}$$

where **PatchEmbed** is the operation of splitting images into non-overlapping patches and applying a linear layer to the channels. We reserve T^1_{Image} and $T^1_{\text{Plücker}}$ for the clean conditional signal. Now, we sample a random rectangular mesh $\mathcal{O}$ and render it from the same camera poses as $\mathcal{S}$. Comparing the depth maps of $\mathcal{S}$ and $\mathcal{O}$, we produce modified tokens $\tilde{T}^i_{\text{Image}}$ where the token for any patch that contains an occluded pixel is replaced by a learnable mask token. Then, the input tokens to the model are constructed as:

$$\mathbf{x} = \Big\|_{i=2}^{n} (\tilde{T}^i_{\text{Image}} + T^i_{\text{Plücker}}), \quad \mathbf{z} = T^1_{\text{Image}} + T^1_{\text{Plücker}}, \tag{2}$$

where z is the condition passed to the model and $\|$ the iterated concatenation operation. We choose to add the Plücker ray tokens after masking such that the model can differentiate between different occluded patches. Note that adding a Plücker ray token for each patch means the model does not need a positional embedding to differentiate patches. We use three sequences of learnable tokens $\mathbf{T}_{\text{Triplanes}} = \mathbf{T}_{xy}||\mathbf{T}_{yz}||\mathbf{T}_{xz}$ to produce the predicted triplanes. These tokens are passed to the transformer body of the model, which comprises of iterated standard cross-attention and self-attention operations equipped with MLPs and residual connections:

$$\hat{\mathbf{x}}, \hat{\mathbf{T}}_{\text{Triplanes}} = \textbf{Self-Att}\left(\textbf{Cross-Att}\left(\mathbf{x}||\mathbf{T}_{\text{Triplanes}}\right), \mathbf{z}\right), \tag{3}$$

with $\mathbf{x}$ and $\mathbf{T}_{\text{Triplanes}}$ coming from the previous transformer block (or the input). Finally, we upsample each triplane token to a patch using a single layer MLP, evaluate the learned triplanes at densely sampled points, and decode the latents using MLPs. We obtain predicted images and $\hat{\mathbf{I}}$ pixel-ray opacity maps $\hat{\mathbf{M}}$ through volumetric rendering, and normal maps $\hat{\mathbf{N}}$ by estimating normalized SDF gradients:

$$\begin{aligned}
\textbf{Triplanes} &= \textbf{MLP}_{\text{Upsample}}(\hat{\mathbf{T}}_{\text{Triplanes}}) \\
\textbf{SDF}(x,y,z) &= \textbf{MLP}_{\text{Distance}}(\textbf{Triplanes}(x,y,z)) \\
\textbf{RGB}(x,y,z) &= \textbf{MLP}_{\text{Color}}(\textbf{Triplanes}(x,y,z)) \\
\sigma &= \textbf{Density}(\textbf{SDF}) \\
\hat{\mathbf{N}} &= \textbf{NormGrad}(\textbf{SDF}) \\
\hat{\mathbf{I}}, \hat{\mathbf{M}} &= \textbf{VolRender}(\sigma, \textbf{RGB})
\end{aligned}$$

where we convert the SDF values to densities σ for rendering following [92]. The learned image tokens $\hat{\mathbf{x}}$ are not used for any remaining task and are thus discarded.

3.2. Supervision

Our LRM is trained with L2 reconstruction and LPIPS perceptual losses. Given a ground truth image $\mathbf{I}$, normal map $\mathbf{N}$, and binary silhouette mask $\mathbf{M}$, we use the following losses:

$$\mathcal{L}_{\text{Recon}} = w_I\|\hat{\mathbf{I}} - \mathbf{I}\|_2^2 + w_N\|\hat{\mathbf{N}} - \mathbf{N}\|_2^2 + w_M\|\hat{\mathbf{M}} - \mathbf{M}\|_2^2$$
$$\mathcal{L}_{\text{Percep}} = w_P\mathcal{L}_{\text{LPIPS}}(\hat{\mathbf{I}}, \mathbf{I})$$

where w_I, w_N, w_M, w_P are tunable weights, and $\mathcal{L}_{\text{LPIPS}}$ is the image perceptual similarity loss proposed in [96]. For the results in this paper, we choose simply $w_I = w_M = w_P = 1$ and $w_N = 0$ or 1 depending on the stage of training.

3.3. Training Stages

Our model is trained in stages following [89], for training efficiency. However, the purpose of our stages differ. Since fully rendering 512×512 output images is computationally expensive, for every stage, we sample a random 128×128 crop from each output image to use as supervision. We maintain the full images for the input throughout every stage.

Stage 1: We downsample the output images to a 256×256 resolution, allowing the random crops to supervise 25% of the image. We use 128 samples per ray for the volumetric rendering. In this initial stage, we observe that the geometric supervision from the normal maps is not yet necessary, so we drop this portion of the reconstruction loss by setting $w_N = 0$ enabling a more efficient backwards pass.

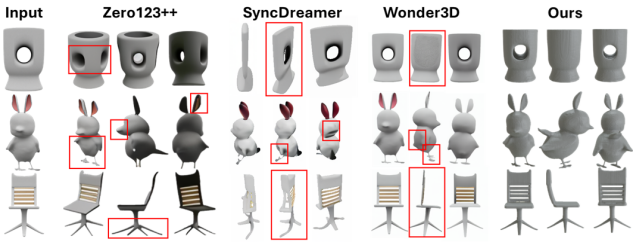

Figure 3. **Multi-view Diffusion vs MaskedLRM**: Multi-view diffusion models must infer occluded geometry from the input which leads to artifacts in the multi-view images (incorrect/distorted views). MaskedLRM which does not use multi-view diffusion, receives both the edited image and multi-view information, allowing it to bypass this problem and reconstruct the correct geometry.

Stage 2: We downsample the output images to 384×384, meaning that the random crops now only supervise 11% of the image and increase the samples per ray to 512. By increasing the rendering fidelity and decreasing the proportion of the image supervision, we train the model to focus more sharply on geometric details. We observed that without any geometric supervision, the LRM may produce degenerate solutions by generating textures that hide geometric artifacts. Thus, we introduce the normal loss by setting $w_N = 1$.

3.4. Mesh Shape Editing

Since our LRM is trained with 3D-consistent, multi-view occlusions, using the conditional branch to complete the partial observations, it is straightforward to use it for shape editing. Given a shape $\mathcal{S}$, we manually define an occlusion $\mathcal{O}$ that occludes the region of interest for editing. Then, we edit a representative image within the pixels that are occluded from its camera viewpoint. This may be done a variety of ways – for our results, we use a text-to-image masked diffusion method [18, 43]. The image edit is used as a conditional signal, while the rest of the occluded images are fed to the main transformer body of the LRM. The LRM is trained to inpaint the occluded region using the content from the conditional branch, and as such it propagates the 2D edit into 3D space. This approach to shape editing is much faster than optimization-based methods (see Table 2), requiring only a single forward pass to lift the desired edit into 3D. Our model also produces more realistic shapes since it is trained on a large collection of scanned objects instead of relying on diffusion model guidance and complex regularizations. It is also more expressive than optimization-based methods as it can generate arbitrary geometry based on the input condition. For example, it can change the geometric genus of a shape (adding a handle or a hole as in Figure 4), which deformation-based optimization methods cannot do as genus changes are not differentiable. Generative methods using LRMs such as InstantMesh [90] rely on methods such as Zero-123++ [75] to generate multi-view images, introducing view-consistency artifacts. In Figure 3 we show examples of such artifacts generated by recent models. Zero-123++ hallucinates additional holes in the vase, and generates a distorted and incorrect bird anatomy. SyncDreamer [52] generates unrealistically distorted views, such as a completely flattened vase, poor bird anatomy, and a warped chair. Wonder3D [54] is better, but it cannot capture the correct bird anatomy and chair structure. In contrast, our model requires only a single view as conditioning and uses the prior from the dataset to construct the shape in a consistent manner. Some recent concurrent work tackles editing directly in the multi-view image space. While this also handles ambiguity, we show in Figure 6 that our method produces more realistic edits.

4. Experiments

Training Data. We train our Masked LRM on a the Objaverse dataset [19] containing shapes collected from a wide variety of sources. Each shape is normalized to fit within a sphere of fixed radius. Our training data consists of 40 512×512 images of each shape, rendered from randomly sampled cameras. We also render the corresponding depth and normal maps for these camera poses. Every iteration, the model inputs and reconstruction targets are chosen randomly from these pre-fixed sets of images.

Evaluation. We evaluate the reconstruction quality of our model on the GSO [23] and ABO [16] datasets and compare the state-of-the-art MeshLRM [89]. Since MeshLRM cannot be easily repurposed for our editing task, we also compare the reconstruction quality with InstantMesh. We use PSNR, SSIM, and LPIPS on the output renders from novel poses as metrics. To remain consistent with the training setting, we randomly generate a rectangular mask to occlude the input views and provide a different clean view as conditioning for our method. Finally, we qualitatively demonstrate the main contribution of our model, the ability to propagate 2D edits from a single viewpoint into 3D. We compare our results to prior works for text and image based 3D generation.

4.1. Quantitative Comparisons

Table 1 shows novel-view synthesis metrics of our method when compared to InstantMesh and MeshLRM. Since our main goal is to edit existing shapes and not to completely generate shapes from scratch, we choose to train our model by randomly selecting 6-8 input views along with one conditional view, giving our LRM a denser input than MeshLRM. We show metrics for both 6 and 8 input views and we compute those on another set of 10 camera poses, different from the input poses. Our method is competitive with the state-of-the-art model on reconstruction, achieving a 2.56 PSNR improvement on the ABO dataset, and a comparable PSNR on GSO. We observe the same phenomenon in perceptual quality measured by LPIPS, where our method significantly outperforms on ABO shapes, and is comparable on GSO shapes. As expected, using 6 views under-performs using 8 views, but only by a slight margin. Furthermore, our method

Table 1. **Quantitative Evaluation**: We evaluate our model using test-set shapes and compare it to the state-of-the-art LRM and InstantMesh on the ABO and GSO shape datasets, reconstructing the meshes from 6 and 8 posed images. Despite direct reconstruction of new shapes not being our main goal, and our masking introducing extra difficulty in the task, we still achieve better than SoTA metrics on ABO, and comparable to SoTA metrics on the GSO dataset.

Method	ABO Dataset			GSO dataset		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
InstantMesh [90] (NeRF)	-	-	-	23.14	0.898	0.119
InstantMesh [90] (Mesh)	-	-	-	22.79	0.897	0.120
MeshLRM [89]	26.09	0.898	0.102	**27.93**	0.925	**0.081**
Ours (6 views)	28.37	0.946	0.081	27.24	0.931	0.088
Ours (8 views)	**28.65**	**0.947**	**0.078**	27.58	**0.933**	0.085

significantly outperforms InstantMesh. This is to be expected, as InstantMesh infers everything from a single view, while both MeshLRM and MaskedLRM access multi-view information. Our model achieves performance on par with SoTA on reconstructing a diverse set of output poses, indicating that it has learned to effectively "in-paint" the randomly occluded regions in the input views using context from the available unoccluded signal. Since our end goal is mesh shape editing, it is *not critical that we surpass the reconstruction quality* of prior works, as we only need to ensure a high quality baseline for the output geometry. We further demonstrate qualitatively in Sec. 4.3 that our model indeed learns to inpaint using the conditional signal, instead of only the context from multi-view images, thereby accomplishing feed-forward shape editing through a single view.

4.2. Qualitative Evaluations

Using a bird mesh generated from a text-to-3D model, and several editing targets, we compare our method against other 3D editing methods. Full results are shown in Figure 5. We define a masked region to edit on the head of the bird (omitted from the figure for brevity). Conditional signals provided to our method generated by masked diffusion are shown in the 1^{st} row, and our results are in the last row.

Optimization methods. In the 2^{nd} and 3^{rd} rows of Figure 5, we show the results of two text-based mesh optimization methods. Instead of using the edited images themselves, we use the text prompts that we passed to the diffusion model as guidance. The first optimization-based method we compare to (2^{nd} row) is TextDeformer [27] which uses a Jacobian representation [1] for deformations to constrain the edits instead of explicit localization.TextDeformer struggles with the highly localized nature of our task and globally distorts the mesh, failing to produce an output of acceptable quality in all examples. We also compare with MagicClay [6], which optimizes both an SDF and a triangle-mesh representation. It also optionally uses a manual "seed" edit so that the optimization task is easier. However, since this requires an additional layer of modeling expertise (*i.e.* a manual user

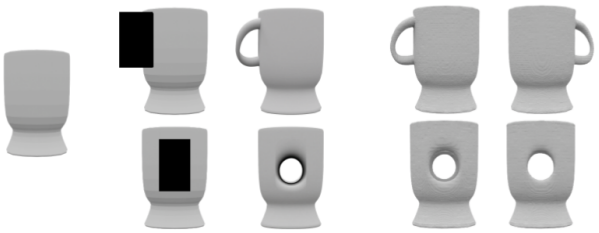

Figure 4. **Genus changes**: Our method unlocks genus-changing edits like adding a handle or a hole to the original vase. We show the output of our model from 2 opposing views in the 3^{rd} column.

intervention) our method does not require, we opt out of this step for our comparison. Unlike TextDeformer, MagicClay selects a subset of vertices to deform to combat noisy SDS [64] gradients. Since we have a 3D mask, we simply choose the vertices that lie within that region. Although this selection serves to localize the editing process, we observe that the deformations are still noisy. While sometimes MagicClay edits are semantically correct (flower and rabbit ears), in other cases such as the fedora (3^{rd} column) and top hat (6^{th} column), the optimization process collapses completely. In both cases, noisy gradients from text-guidance result in results in optimizations that are both unpredictable and uncontrollable. In contrast, the output of our LRM is highly predictable from the selected conditional view, which may be re-generated until desirable or even manually edited.

LRM-based Methods. We compare our method against recent top-performing methods that combine multi-view diffusion and reconstruction models. InstantMesh [90] is one such pipeline that may be used for shape editing. It relies on using Zero-123++ [75] to generate multi-view images from a single view and then passing these images to an LRM. To edit, we simply pass an edited image of the original shape As shown in the 4^{th} row of Figure 5, this results in a poorly reconstructed shape that is particularly thin when compared to the ground truth and the output quality suffers due to the inability of Zero-123++ [75] to generate faithful multi-view images, as discussed in Section 3.4. Methods that rely directly on a separate diffusion module to generate the LRM inputs will run the risk of generating artifacts from inaccurate multiview generation. In comparison, our method does not suffer any such reconstruction artifacts since it uses trivially consistent ground truth renders as the main LRM inputs.

In Figure 6 we also compare to two concurrent works namely PrEditor3D [24] and Instant3Dit [5] that tackle localized editing using multi-view diffusion via text prompting. PrEditor3D [24] performs multi-view editing by first inverting a multi-view diffusion model to obtain Gaussian noise images, and then using the forward process to edit the desired region with a separate text prompt. While PrEditor3D generates semantically correct edits based on the prompts, it produces undesirable artifacts in several examples. In particular, many of the edits lack detail, such as the bunny

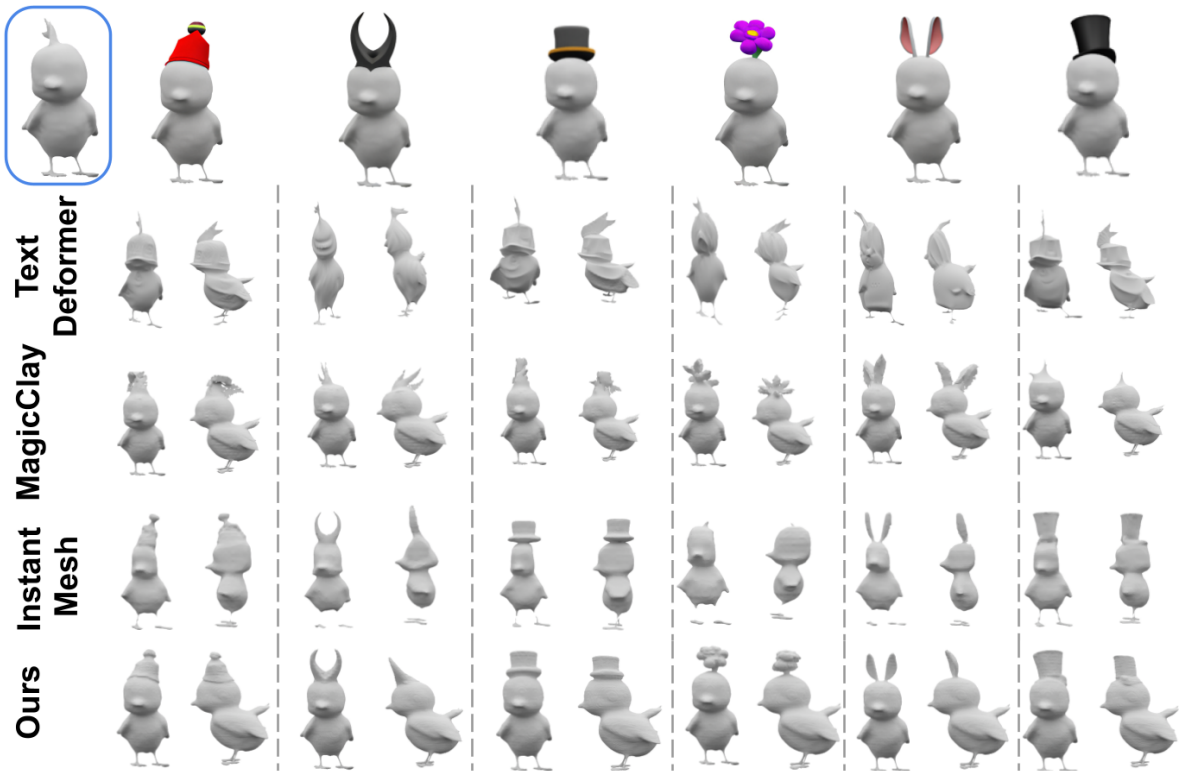

Figure 5. **Mesh Editing Comparisons**: Given a mesh (top left) and various image edits as guidance we demonstrate that our approach is the only one that generates multi-view consistent shape edits that follow the guidance. Colors omitted to clearly visualize the edited geometry.

ears in the top left and the wings in the bottom right. It also fails to produce a pineapple body in the bottom left. Instant3Dit [5] uses masking in multi-view diffusion training instead of LRM training. They use their text-prompted multi-view inpainting model to edit and then use an LRM generate an edited mesh. Similar to PrEditor3D, it produces semantically correct edits that lack realism due to artifacts. Instead of producing sharp bunny ears, Instant3Dit is only capable of adding vague pointed structures to the bird. In the second column, the flower it generates is plant-like but unrealistic. In the bottom row, we see that the pineapple and wings are again semantically correct but lacking in detail.

4.3. Mesh Editing Characteristics

In Figures 1 and 4, we show mesh editing examples demonstrating the capabilities of our method. The 1^{st} column shows the source mesh rendered from different viewpoints. The 2^{nd} column shows the edited conditional image with a render of the original masked region inset. Our LRM accepts the edited view along with a set of occluded ground-truth renders (omitted from the figure) and predicts an SDF. The last column shows the mesh extracted from the output while the insets depict volumetric renders of the predicted SDF.

Expressiveness. The edits throughout this paper show the expressiveness of our method. The meshes used in Figures 5 as well as in row 1 of Figure 1 are examples of non-standard shapes – a unique bird mesh generated from a text-to-multiview diffusion model [80] and a "Tele-alien" from the COSEG [86] dataset. Despite being novel shapes, our model is able to give the alien a staff and the bird a hat. The other four rows of 1 consist of edits that are "unnatural" – creating an avocado backrest, replacing the body of a crab with a pineapple, giving a panda wings, and giving a donkey a turtle shell. In every example, our method successfully translates the 2D edit into geometry in a realistic manner. The edits in Figure 4 show a critical benefit of our method. Since the final mesh is constructed entirely from the output of a neural network, there are no geometric restrictions on the type of edit we can do. The last two rows demonstrate the ability of our network to change the genus of a shape by adding a handle or a hole through the middle, which would be impossible for geometry optimization-based methods.

Identity Preservation. Although our model discards the initial shape in order to bypass editing limitations, we observe that the LRM still achieves highly faithful reconstructions of the initial geometry outside of the region of interest. This confirms our quantitative observations that our method has near-SoTA reconstruction quality. This also indicates that, due to multi-view masking, our method is able to constrain the edit inside selected 3D region without needing to perform expensive learning over explicit 3D signals.

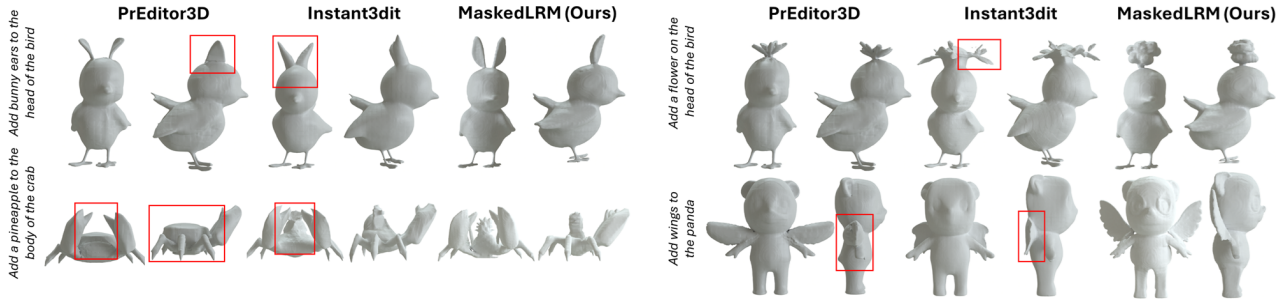

Figure 6. **Mesh Editing Comparisons with Concurrent work**: We compare our approach to concurrent work enabling localized 3D editing. While localized approaches better preserve the original structure of the shape, other methods are not able to produce edits as realistic as ours.

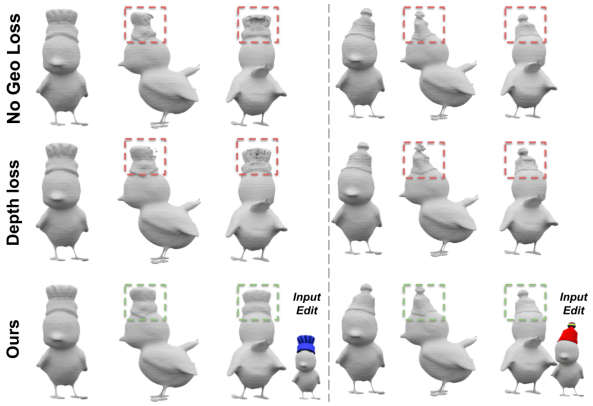

Figure 7. **Impact of Geometric Supervision**: Geometric losses are critical to produce high-quality surfaces. No geometric loss (top) causes severe hole and bump artifacts in the mesh. Depth loss (middle) is not as effective as normal loss (bottom) which allows our model to generate accurate and smooth reconstructions.

4.4. Ablations Studies & Discussion

Geometric Supervision. We investigate the effect of using geometric losses during training. Figure 7 compares three LRMs: a model trained by the pipeline described in Sec. 3, a model trained with no geometric supervision, and a model trained by replacing the normal map loss with a depth map loss. Using no geometric supervision results in poor surface quality, highlighted in the red boxes. Since the main training objective is multi-view image reconstruction, the model hallucinates correct geometry using colors, without producing an accurate surface. Supervising the predicted depth somewhat mitigates this issue, but the effect is weak and the surfaces are still incomplete. Normal map supervision gives high quality surfaces as shown in the green boxes.

Random Masking. We validate our choice in masking strategy by ablating our method using a uniformly random MAE-style mask across all views. This produces a clear train-test gap as during inference, we are always interested in editing contiguous 3D regions. This gap manifests in blurry and incorrect edits. We refer to Section B and Figure 2 of the supplementary material for details.

Runtime: In Table 2 we provide performance comparisons between our approach and several top-performing recent works. Our method is not only much faster than optimization-based approaches [6, 27] as it requires only one forward pass, but it also outperforms LRM approaches [5, 90] that make use of a multi-view generation model. PrEditor3D [24] requires the forward pass of several large pre-trained models [51, 69] resulting in a longer runtime.

Table 2. **Runtime Comparison**: Our method is significantly faster than optimization methods as it is feed-forward and also faster than LRM-based approaches that must run multi-view diffusion.

	Optimization-based		LRM-based			
	TextDeformer	MagicClay	InstantMesh	Instant3dit	PrEditor3D	**Ours**
Runtime ↓	~20mins	~1hour	30sec	~6sec	80sec	< **3sec**

Limitations & Future Work: Our method is constrained by the expressiveness of editing in the canonical view. While text-to-image models can create a wide range of results, capturing a specific idea may require significant iteration. Our method is upper-bounded by the uniformity of the Marching Cubes triangulation, and the LRM reconstruction quality which makes performing edits that require extremely intricate details challenging. Blurry artifacts may arise when trying to reconstruct fine details (*e.g.* face) but we did not see such issues with shapes like chairs. MagicClay [6], manually freezes the un-edited geometry part but we designed a solution without such interventions. Future work could focus on improving localization by developing techniques to merge the existing triangulation with the edited output.

5. Conclusion

In this paper we introduced a new method to perform 3D shape editing. Our work builds upon the recent progress of LRMs by introducing a novel multi-view input masking strategy during training. Our LRM is trained to "inpaint" the masked region using a clean conditional viewpoint to reconstruct the missing information. During inference, a user may pass a single edited image as the conditional input, prompting our model to edit the existing shape in just one forward pass. We believe our method is an significant advancement in shape editing, allowing users to create accurate and controllable edits without 3D modeling expertise.

References

[1] Noam Aigerman, Kunal Gupta, Vladimir G. Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. In *ACM Transactions on Graphics (SIGGRAPH)*, 2022. 6

[2] Shivangi Aneja, Justus Thies, Angela Dai, and Matthias Nießner. Clipface: Text-guided editing of textured 3d morphable models. In *SIGGRAPH '23 Conference Proceedings*, 2023. 3

[3] Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023. 3

[4] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 3

[5] Amir Barda, Matheus Gadelha, Vladimir G. Kim, Noam Aigerman, Amit H. Bermano, and Thibault Groueix. Instant3dit: Multiview inpainting for fast editing of 3d objects, 2024. 2, 3, 6, 7, 8

[6] Amir Barda, Vladimir G. Kim, Noam Aigerman, Amit Bermano, and Thibault Groueix. Magicclay: Sculpting meshes with generative neural fields. In *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2024. 2, 3, 6, 8

[7] Henning Biermann, Ioana Martin, Fausto Bernardini, and Denis Zorin. Cut-and-paste editing of multiresolution surfaces. *ACM transactions on graphics (TOG)*, 21(3):312–321, 2002. 3

[8] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. *arXiv preprint*, 2024. 2

[9] Andrew Brock, Theodore Lim, James Millar Ritchie, and Nicholas J Weston. Generative and discriminative voxel modeling with convolutional neural networks. In *Neural Inofrmation Processing Conference: 3D Deep Learning*, 2016. 1

[10] Bindita Chaudhuri, Nikolaos Sarafianos, Linda Shapiro, and Tony Tung. Semi-supervised synthesis of high-resolution editable textures for 3d humans. In *CVPR*, 2021. 3

[11] Jun-Kun Chen, Jipeng Lyu, and Yu-Xiong Wang. NeuralEditor: Editing neural radiance fields via manipulating point clouds. In *CVPR*, 2023. 3

[12] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 3

[13] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting, 2023. 3

[14] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5939–5948, 2019. 1

[15] Chong Bao and Bangbang Yang, Zeng Junyi, Bao Hujun, Zhang Yinda, Cui Zhaopeng, and Zhang Guofeng. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[16] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21126–21136, 2022. 5

[17] Sabine Coquillart. Extended free-form deformation: A sculpturing tool for 3d geometric modeling. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 187–196, 1990. 3

[18] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 3, 5

[19] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2, 5

[20] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2

[21] Wenqi Dong, Bangbang Yang, Lin Ma, Xiao Liu, Liyuan Cui, Hujun Bao, Yuewen Ma, and Zhaopeng Cui. Coin3d: Controllable and interactive 3d assets generation with proxy-guided conditioning. In *SIGGRAPH*, 2024. 3

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3, 4

[23] 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. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 5

[24] Ziya Erkoç, Can Gümeli, Chaoyang Wang, Matthias Nießner, Angela Dai, Peter Wonka, Hsin-Ying Lee, and Peiye Zhuang. Preditor3d: Fast and precise 3d shape editing. In *CVPR*, 2025. 2, 3, 6, 8

[25] Anna Frühstück, Nikolaos Sarafianos, Yuanlu Xu, Peter Wonka, and Tony Tung. VIVE3D: Viewpoint-independent video editing using 3D-aware GANs. In *CVPR*, 2023. 3

[26] Ran Gal, Olga Sorkine, Niloy J Mitra, and Daniel Cohen-Or. iwires: An analyze-and-edit approach to shape manipulation. In *ACM SIGGRAPH 2009 papers*, pages 1–10, 2009. 3

[27] William Gao, Noam Aigerman, Groueix Thibault, Vladimir Kim, and Rana Hanocka. Textdeformer: Geometry manipula-

tion using text guidance. In *ACM Transactions on Graphics (SIGGRAPH)*, 2023. 2, 3, 6, 8

[28] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 484–499. Springer, 2016. 1

[29] Xiao Han, Yukang Cao, Kai Han, Xiatian Zhu, Jiankang Deng, Yi-Zhe Song, Tao Xiang, and Kwan-Yee K. Wong. Headsculpt: Crafting 3d head avatars with text. *arXiv preprint arXiv:2306.03038*, 2023. 3

[30] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *CVPR*, 2023. 3

[31] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 3, 4

[32] Zexin He and Tengfei Wang. Openlrm: Open-source large reconstruction models, 2023. 2

[33] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Spaghetti: Editing implicit shapes through part aware generation. *ACM Transactions on Graphics (TOG)*, 41(4):1–20, 2022. 3

[34] 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. 2, 3

[35] Muhammad Zubair Irshad, Sergey Zakharov, Vitor Guizilini, Adrien Gaidon, Zsolt Kira, and Rares Ambrus. Nerf-mae: Masked autoencoders for self-supervised 3d representation learning for neural radiance fields. In *European Conference on Computer Vision*, pages 434–453. Springer, 2025. 3

[36] Clément Jambon, Bernhard Kerbl, Georgios Kopanas, Stavros Diolatzis, Thomas Leimkühler, and George" Drettakis. Nerfshop: Interactive editing of neural radiance fields". *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 6(1), 2023. 3

[37] Jincen Jiang, Xuequan Lu, Lizhi Zhao, Richard Dazaley, and Meili Wang. Masked autoencoders in 3d point cloud representation learning. *IEEE Transactions on Multimedia*, 2023. 3

[38] Hyunyoung Jung, Seonghyeon Nam, Nikolaos Sarafianos, Sungjoo Yoo, Alexander Sorkine-Hornung, and Rakesh Ranjan. Geometry transfer for stylizing radiance fields. In *CVPR*, 2024. 3

[39] Takashi Kanai, Hiromasa Suzuki, Jun Mitani, and Fumihiko Kimura. Interactive mesh fusion based on local 3d metamorphosis. In *Graphics interface*, pages 148–156, 1999. 3

[40] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[41] Hyun Woo Kim, Itai Lang, Thibault Groueix, Noam Aigerman, Vladimir G. Kim, and Rana Hanocka. Meshup: Multi-target mesh deformation via blended score distillation. In *arXiv preprint*, 2024. 3

[42] Dmytro Kotovenko, Olga Grebenkova, Nikolaos Sarafianos, Avinash Paliwal, Pingchuan Ma, Omid Poursaeed, Sreyas Mohan, Yuchen Fan, Yilei Li, Rakesh Ranjan, et al. Wast-3d: Wasserstein-2 distance for scene-to-scene stylization on 3d gaussians. In *ECCV*, 2024. 3

[43] Eran Levin and Ohad Fried. Differential diffusion: Giving each pixel its strength, 2023. 3, 5

[44] 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. *arXiv preprint arXiv:2311.06214*, 2023. 2

[45] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12642–12651, 2023. 1

[46] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly, 2023. 3

[47] Yaqian Liang, Shanshan Zhao, Baosheng Yu, Jing Zhang, and Fazhi He. Meshmae: Masked autoencoders for 3d mesh data analysis. In *European Conference on Computer Vision*, pages 37–54. Springer, 2022. 3

[48] Yaron Lipman, Olga Sorkine, Daniel Cohen-Or, David Levin, Christian Rossi, and Hans-Peter Seidel. Differential coordinates for interactive mesh editing. In *Proceedings Shape Modeling Applications, 2004.*, pages 181–190, 2004. 3

[49] Feng-Lin Liu, Hongbo Fu, Yu-Kun Lai, and Lin Gao. Sketchdream: Sketch-based text-to-3d generation and editing. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2024)*, 43(4), 2024. 3

[50] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 2

[51] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024. 8

[52] 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. *arXiv preprint arXiv:2309.03453*, 2023. 2, 5

[53] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. *arXiv preprint arXiv:2303.08133*, 2023. 1

[54] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023. 2, 5

[55] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. Vv-net: Voxel vae net with group convolutions for point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8500–8508, 2019. 1

[56] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 1

[57] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 2

[58] 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. 1

[59] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 conference papers*, pages 1–8, 2022. 2

[60] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020. 1, 3

[61] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. In *ACM SIGGRAPH 2005 Papers*, pages 1142–1147, 2005. 3

[62] 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. 1, 3

[63] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 3

[64] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 3, 6

[65] Rolandos Alexandros Potamias, Michail Tarasiou Stylianos Ploumpis, and Stefanos Zafeiriou. Shapefusion: A 3d diffusion model for localized shape editing. *arXiv preprint arXiv:2403.19773*, 2024. 3

[66] Zhangyang Qi, Yunhan Yang, Mengchen Zhang, Long Xing, Xiaoyang Wu, Tong Wu, Dahua Lin, Xihui Liu, Jiaqi Wang, and Hengshuang Zhao. Tailor3d: Customized 3d assets editing and generation with dual-side images, 2024. 3

[67] Aashish Rai, Dilin Wang, Mihir Jain, Nikolaos Sarafianos, Kefan Chen, Srinath Sridhar, and Aayush Prakash. Uvgs: Reimagining unstructured 3d gaussian splatting using uv mapping. In *CVPR*, 2025. 3

[68] Mervi Ranta, Masatomo Inui, Fumihiko Kimura, and Martti Mäntylä. Cut and paste based modeling with boundary features. In *Proceedings on the second ACM symposium on Solid modeling and applications*, pages 303–312, 1993. 3

[69] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. 8

[70] Abdrakhmanov Renat and Kerimbek Imangali. Learning latent representations for 3d voxel grid generation using variational autoencoders. In *2024 IEEE AITU: Digital Generation*, pages 169–173. IEEE, 2024. 1

[71] Nikolaos Sarafianos, Tuur Stuyck, Xiaoyu Xiang, Yilei Li, Jovan Popovic, and Rakesh Ranjan. Garment3dgen: 3d garment stylization and texture generation. In *3DV*, 2025. 3

[72] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 1

[73] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986. 3

[74] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *ICCV*, 2023. 3

[75] 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, 5, 6

[76] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, page 175–184. Association for Computing Machinery, 2004. 3

[77] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–70, 2020. 1, 3

[78] Kenshi Takayama, Daniele Panozzo, Alexander Sorkine-Hornung, and Olga Sorkine-Hornung. Sketch-based generation and editing of quad meshes. *ACM Trans. Graph.*, 32 (4):97–1, 2013. 3

[79] Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. Edgerunner: Autoregressive auto-encoder for artistic mesh generation. *arXiv preprint arXiv:2409.18114*, 2024. 1, 3

[80] Shitao Tang, Jiacheng Chen, Dilin Wang, Chengzhou Tang, Fuyang Zhang, Yuchen Fan, Vikas Chandra, Yasutaka Furukawa, and Rakesh Ranjan. Mvdiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. *arXiv preprint arXiv:2402.12712*, 2024. 7

[81] 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. *arXiv preprint arXiv:2403.02151*, 2024. 2

[82] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022. 1

[83] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 3

[84] Peihao Wang, Zhiwen Fan, Dejia Xu, Dilin Wang, Sreyas Mohan, Forrest Iandola, Rakesh Ranjan, Yilei Li, Qiang Liu, Zhangyang Wang, et al. Steindreamer: Variance reduction for text-to-3d score distillation via stein identity. *arXiv preprint arXiv:2401.00604*, 2023. 2

[85] 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. *arXiv preprint arXiv:2311.12024*, 2023. 2

[86] Yunhai Wang, Shmulik Asafi, Oliver Van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Active co-analysis of a set of shapes. *ACM Transactions on Graphics (TOG)*, 31 (6):1–10, 2012. 7

[87] Yuxuan Wang, Xuanyu Yi, Zike Wu, Na Zhao, Long Chen, and Hanwang Zhang. View-consistent 3d editing with gaussian splatting. In *ECCV*, 2024. 3

[88] 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. 2, 3

[89] 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. *arXiv preprint arXiv:2404.12385*, 2024. 2, 3, 4, 5, 6

[90] 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. 2, 5, 6, 8

[91] 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

[92] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 4

[93] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023. 2

[94] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 3

[95] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2025. 2

[96] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[97] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35:27061–27074, 2022. 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. 1

[99] 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. 1