LAM3D: Large Image-Point-Cloud Alignment Model for 3D Reconstruction from Single Image

Ruikai Cui[1] * **Xibin Song**[2] † **Weixuan Sun**[2] **Senbo Wang**[2] **Weizhe Liu**[2]
Shenzhou Chen[2] **Taizhang Shang**[2] **Yang Li**[2] **Nick Barnes**[1] **Hongdong Li**[1] **Pan Ji**[2]
[1]Australian National University [2]Tencent XR Vision Labs

Abstract

Large Reconstruction Models have made significant strides in the realm of automated 3D content generation from single or multiple input images. Despite their success, these models often produce 3D meshes with geometric inaccuracies, stemming from the inherent challenges of deducing 3D shapes solely from image data. In this work, we introduce a novel framework, the Large Image and Point Cloud Alignment Model (LAM3D), which utilizes 3D point cloud data to enhance the fidelity of generated 3D meshes. Our methodology begins with the development of a point-cloud-based network that effectively generates precise and meaningful latent tri-planes, laying the groundwork for accurate 3D mesh reconstruction. Building upon this, our Image-Point-Cloud Feature Alignment technique processes a single input image, aligning to the latent tri-planes to imbue image features with robust 3D information. This process not only enriches the image features but also facilitates the production of high-fidelity 3D meshes without the need for multi-view input, significantly reducing geometric distortions. Our approach achieves state-of-the-art high-fidelity 3D mesh reconstruction from a single image in just 6 seconds, and experiments on various datasets demonstrate its effectiveness.

1 Introduction

High-quality 3D mesh creation has drawn increasing attentions for its great potential in numerous fields, such as video games, virtual reality, and films. Traditionally, 3D assets are usually created manually by expert artists or alternatively reconstructed from multi-view 2D images, which are time-consuming. Recent proposed Large Reconstruction Models (LRMs) [15, 62, 48, 52] from a single view or few-shot images bring significant advancements for high-fidelity 3D content creation.

Single view based LRMs [15, 13] take a single-view image as input and use transformers to directly regress tri-plane-based [3] neural radiance fields (NeRF) [31] for 3D content creation. However, as shown in Fig. 1 (b), due to ambiguity of 3D geometry from a single image, single-view-based LRMs commonly suffer from geometric distortions, especially for the unseen areas of the input view. Inspired by Hong *et al*. [15], LRMs with multi-view inputs [62, 48, 52, 20] are proposed, where they typically first feed the single-view image into multi-view image diffusion models [51, 43, 28] for few-shot multi-view images generation. Then multi-view tri-plane features are generated and converted into 3D representations (NeRF [31], SDFs [34], *etc.*) for 3D mesh creation. However, as shown in Fig.1 (c), the generated multi-view images lack guaranteed multi-view consistency, which can lead to geometric distortions.

To relieve these problems, we propose to introduce a 3D point cloud prior for single image based 3D mesh reconstruction with Large Image-Point-Cloud Alignment Model (LAM3D). LAM3D aims to

*The contribution of Ruikai Cui was made during an internship at Tencent XR Vision Labs.
†Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-0145

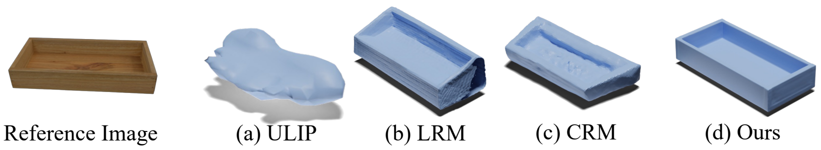

Figure 1: An example of single-image reconstruction from state-of-the-art methods: (a) ULIP [63], (b) LRM [13], (c) CRM [54], and (d) Ours (LAM3D).

transfer single image features into point cloud features, enriching the image features with accurate and effective 3D information for high-fidelity 3D mesh reconstruction. The motivation behind this choice is obvious: 3D point cloud contains effective structure prior of 3D geometries. Meanwhile, approaches [63, 64] of modality feature alignment have been proposed, aiming to learn unified representations of language, images, and point clouds for 3D understanding. Nevertheless, as shown in Fig. 1 (a), these techniques fall short when they use the aligned image features to replace point clouds features for 3D mesh creation. This is largely due to the exclusive usage of contrastive loss [63, 64] and the aligned feature size is commonly with (512×1)[63] and (1024×1)[64], which is inadequate to fully capture the rich diversity of 3D shapes.

In this work, we extend the image and point cloud alignment for 3D mesh creation, and present the Large Image-Point-Cloud Alignment Model (LAM3D) which utilize the prior of 3D point cloud for high-fidelity feature alignment. We start by designing an effective point cloud-based network with hierarchical tri-planes [3] for 3D mesh reconstruction. Following this, taking single image as input, image features are extracted using DINO [2], whilst LAM3D is employed to transpose these image features into tri-planes obtained via the point-cloud-based network. The tri-plane representation [3] allows for the tuning of image features with precise and effective 3D information, which is essential for high-fidelity 3D mesh reconstruction. Meanwhile, to better retain 3D structure information and reduce the interference of each plane (XY, XZ, YZ) in tri-plane features, we enlist the use of independent diffusion operations to transfer image features to each respective plane. Note that the transferred features are with size of 3×C×M×N ((3×2×32×32) in our experiments), which is larger than previous representation size of (512×1)[63] and (1024×1)[64], thus contains more information for better 3D mesh creation. As shown in Fig. 1 (d), more reasonable geometric mesh can be reconstructed by our approach.

The main contributions of our work are as follows:

- We propose an effective Large Image-Point-Clouds Alignment Model (LAM3D) for 3D mesh reconstruction from a single image. By utilizing point cloud priors, LAM3D delivers precise image tri-plane feature transformation, thereby enhancing the quality and accuracy of 3D mesh creation.
- To preserve 3D structural information more effectively, we employ independent diffusion processes that transmit image features to each respective tri-plane (XY, XZ, and YZ) for accurate 3D mesh reconstruction.
- Our method significantly relieves geometric distortions caused by ambiguities of 3D shape inferred from the image modality, and experiments on various of dataset demonstrate the effectiveness of our approach.

2 Related Works

Single Image to 3D Reconstruction: Recently, LRM [15] was the first to demonstrate that a regression model can be trained to robustly predict tri-plane-based NeRF from a single-view image. Inspired by LRM, numerous large models[25, 26, 36, 54, 69, 53, 50] for 3D reconstruction have since been proposed. Among these, Instant3D [20] trains a text-to-few-shot multi-view image diffusion model, as well as a multi-view LRM, to enable fast and diverse text-to-3D generation. Subsequent works have extended LRM by incorporating pose prediction from multi-view images [52], combining it with diffusion [62], and specializing on human data [56]. Tang et al.[48] introduced the Large Multi-View Gaussian Model to generate high-resolution 3D models from text prompts or single-view images. Wang et al.[54] presented a single-image-to-3D generative model that generates six orthographic images, utilizing Flexicubes [42] as a 3D representation. Concurrently, Wu et al. [58]

proposed Direct3D, a method that differs from ours primarily in its approach to feature extraction from point clouds. Direct3D utilizes learnable tokens for feature extraction, whereas our method introduces a projection module to explicitly convert point cloud features into a tri-plane structure.

Generative Modeling of 3D shapes: In recent years, a multitude of studies have been dedicated to the generation of 3D shapes. Existing 3D generative models have been developed using a variety of frameworks, including generative adversarial networks (GANs)[57, 1, 67], variational autoencoders (VAEs)[47, 11, 17, 32, 12], normalizing flows [65, 32], autoregressive models [45, 32], energy-based models [61, 8], and more recently, denoising diffusion probabilistic models (DDPMs) [30, 68, 27, 49, 24, 4, 44, 60]. Luo *et al*. [30] pioneers the application of DDPMs for modeling the distribution of raw point clouds. Fellowing Luo *et al*, several works [49, 24, 4] explore generative modeling of 3D shapes in latent space to reduce computational complexity and enhance generation quality. For instance, LION [49] proposed an effective latent-space diffusion model to generate novel point clouds, while 3DQD [24], SDFusion [4] utilize DDPM to model Signed Distance Function (SDF) in the latent space of an autoencoder for 3D shape generation. Concurrently, other research efforts have investigated DDPMs for 3D shape generation with other representations, such as mesh [27], occupancy grid [44], and neural radiance fields [33].

Feature Alignment / Multimodal 3D: Multimodal approaches are mainly about image-text and image-text-point-clouds modalities. Several works [21, 23, 46] propose to learn interactions between image regions and caption words using transformer-based architectures, which show great predictive capability despite being costly to train. Meanwhile, CLIP [38] uses image and text encoders to output a single image/text representation for each image-text pair, and then aligns the representations to a unified latent space. Built upon CLIP [38], ULIP [63] and its successor, ULIP2 [64], aim to learn a unified representation that integrates images, text, and 3D point clouds. These methods establish a 3D representation space aligned with the CLIP latent space by leveraging synthesized 3D-text-image triplets. The most closely related work to ours is Michelangelo [66], which aligns point cloud features with frozen CLIP image/text embeddings through an autoencoder. Unlike Michelangelo, which employs contrastive loss to learn an aligned latent space, our approach uses diffusion to align 3D and 2D modalities. This results in a more continuous latent space, beneficial for 3D reconstruction.

3 Method

In this section, we present the detailed design of our approach. The full training process consists of two stages: *Point Cloud Compression* (Sec. 3.1) and *Image-Point-Cloud Alignment* (Sec. 3.2).

3.1 Stage 1: Point Cloud Compression

As shown in Fig. 2, we encode point clouds into a tri-plane representation and then decode SDF values from the reconstructed tri-plane to extract 3D meshes via marching cubes [29]. The point cloud compression consists of *Initial Point Feature Extraction* and *Tri-plane Compression*. Specifically, we propose a point-cloud-based 3D reconstruction network to convert point cloud into a latent tri-plane, which can retain rich 3D information. In the tri-plane compression module, the latent tri-plane is up-sampled to reconstruct tri-planes followed by an MLP to compute SDF values. As shown in Fig. 2, hierarchical tri-plane are used, including latent tri-planes and reconstructed tri-planes, where the latent tri-plane is designed for better image-point-cloud feature alignment and the reconstructed tri-plane is for better mesh reconstruction.

3.1.1 Initial Point Feature Extraction

To compress a point cloud into latent tri-planes, we employ a transformer-based approach to learn long-range interactions between points. We first use Furthest Point Sampling (FPS) to downsample the input point cloud to n center points. Then, we gather the K-nearest neighbors of the n center points together as a small point cloud and use a shallow PointNet [35] to embed each small point cloud as an embedding. By doing so, we reduce the sequence length of the point cloud, and then we can use a transformer to process these embeddings.

After the transformer step, we need to collect the embeddings to form the three planes of the tri-plane structure. To this end, we use a projection operation. In specific, each embedding feature is associated with a knn center. For each plane (XY, XZ, YZ), we diminish one axis of the center points and

https://doi.org/10.52202/079017-0145

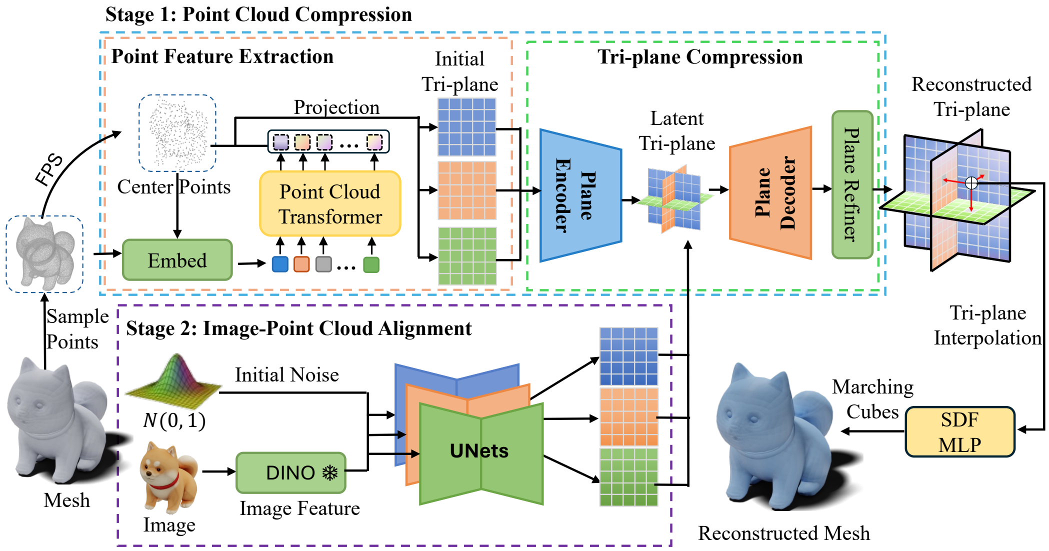

Figure 2: Overview of our method. Our method contains two training stage. **Stage 1**: we train an encoder-decoder structure to take point clouds as input and compress it to a latent tri-plane representation; **Stage 2**: we employ diffusion to align image modality to latent tri-planes obtained in stage 1. The diffusion step takes an initial noise and an image feature from a freezed DINO feature encoder and progressively align the image feature to the latent tri-plane. **Inference**: To reconstruct a 3D mesh from a single-view image, we use the alignment step, following the decoder (Plane Decoder, Plane Refiner) from the compression step, to predict a tri-plane. Then, we can use algorithms like marching cubes to extract 3D meshes from the reconstructed tri-plane.

voxelize the center points into a plane of size $\mathbb{R}^{C\times R\times R}$ where C is the channel dimension that is the same with the embedding dimension, and R is the plane resolution. We use average operation to aggregate embeddings associated with center points that land into the same voxel. As shown in Fig. 2, the point cloud is converted into a initial tri-planes that faithfully preserves 3D information.

3.1.2 Tri-plane Compression

The initial tri-planes is a sparse tensor as many of its grids are empty if there is no center points land there. To obtain a more compact and expressive latent tri-plane representation, we adopt an plane autoencoder to further compress initial tri-planes to latent tri-planes.

Plane Encoder We find that directly aligning image features with initial tri-planes leads to two drawbacks: (1) Aligning in high-resolution requires high computational complexity; (2) Empty grids and overwhelmed details in high-resolution tri-planes substantially increase training difficulty, leading to noisy results. To further compress the initial tri-planes into a more compact latent, we opt for a transformer-based plane encoder. It consists of a convolution layer to down-sample the initial tri-planes followed by successive transformer blocks to achieve inter-plane correlations. In each transformer block, we follow Cui *et al.* [6] to concatenate tri-planes into one sequence and adopt linear attention [37] to process the sequence to model inter-plane correlations. In our settings, the extracted initial tri-plane is down-sampled by 4 times to obtain a point cloud latent tri-plane $t \in \mathbb{R}^{(3\times 2\times 32\times 32)}$.

Plane Decoder We introduce a hierarchical scheme to decode the latent tri-plane into a high-resolution reconstructed tri-plane, as depicted in Fig. 2, and for predicting the Signed Distance Function (SDF). Initially, the latent tri-plane is processed through a standard decoder that mirrors the encoder. This decoder comprises successive transformer blocks coupled with a convolutional layer, which augments the latent tri-plane by a factor of four.

Plane Refiner After Plane Decoder, we adopt an intra-plane refinement module to further enhance the reconstruction quality. In latent tri-plane generation, the original point clouds are condensed to form a compact tri-plane latent, the compression inaccuracies are unavoidable. Therefore, we propose an asymmetric decoding of the tri-plane, which is accomplished by individually refining each plane with a plane refiner module subsequent to the conventional decoder. Specifically, the plane refiners are three small-scale UNet [41] models to refine each plane separately. We find that refining each plane individually extensively improves mesh quality while retains inter-plane correlations as we demonstrated in Appendices.Tab. 5. Finally, we can query SDF values of any spatial position using the SDF-MLP by interpolating features on the reconstructed tri-plane and extract 3D meshes using Marching Cubes [29].

3.1.3 Compression Loss Function

The training objective for point cloud compression comprises a KL penalty and a geometry loss. To align image features onto a 3D-aware latent space, we attempt to preserve 3D prior to the greatest extent, even in the most compact compressed tri-plane latent. However, we discovered that the compressed tri-plane latent fails to maintain the 3D prior, leading to artifacts in the mesh outcomes if we merely employ an arbitrary encoder to compress the initial tri-plane without explicit constraints. Hence, we propose to concurrently supervise the latent tri-plane t and reconstructed tri-plane P using the ground truth SDF and two distinct SDF MLPs (Φ_t and Φ_P). To this end, we combine the geometric SDF loss and latent regularization to train the compression module in an end-to-end manner. Formally, to allow the predicted tri-plane recover SDF values, we use the geometry loss:

$$\mathcal{L}_{geo} = \mathcal{L}_{sdf} + \mathcal{L}_{normal}. \tag{1}$$

These two terms are defined as:

$$\mathcal{L}_{sdf} = \lambda_1 \sum_{p \in \Omega_0} ||\Phi_P(p)|| + \lambda_2 \sum_{p \in \Omega} ||\Phi_P(p) - d_p||, \tag{2}$$

$$\mathcal{L}_{normal} = \lambda_3 \sum_{p \in \Omega_0} ||\nabla_p \Phi_P(p) - n_p||, \tag{3}$$

where Ω_0 and Ω are sets of query points sampled from object surface and off-surface points sampled from $[-1, 1]^3$. The grouth truth SDF values and surface normal are denoted as d_p and n_p, respectively. The gradient $\nabla_p \Phi_P(p) = \left(\frac{\partial \Phi_P(p)}{\partial X}, \frac{\partial \Phi_P(p)}{\partial Y}, \frac{\partial \Phi_P(p)}{\partial Z} \right)$ represents the direction of the steepest change in SDF. It can be computed using finite difference, *e.g.*, the partial derivative for the X-axis component reads $\frac{\partial \Phi(p)}{\partial X} = \frac{\Phi_P(p+(\delta,0,0)) - \Phi_P(p-(\delta,0,0))}{2\delta}$ where δ is the step size. Similarly, the latent sdf loss $\mathcal{L}_{lsdf}$ is defined as:

$$\mathcal{L}_{lsdf} = \lambda_4 \sum_{p \in \Omega_0} ||\Phi_t(p)|| + \lambda_5 \sum_{p \in \Omega} ||\Phi_t(p) - d_p||. \tag{4}$$

We use the corresponding MLP to predict a SDF value from either latent tri-plane t or the reconstructed tri-plane P. Specifically, given a query position p, we project it to each of the planes and retrieve the feature vectors F_{xy}, F_{xz}, F_{yz} via bilinear interpolation. Then, the signed distance can be predicted as: $\Phi(p) = \text{MLP}(F_{xy} \oplus F_{xz} \oplus F_{yz})$, where $\oplus$ denotes element-wise summation.

Overall, the training objective is defined as fellow:

$$\mathcal{L}_{comp} = \mathcal{L}_{geo} + \mathcal{L}_{lsdf} + \mathcal{L}_{KL}, \tag{5}$$

where $\mathcal{L}_{KL}$ is a KL penalty which leans towards a standard normal distribution on the learned latent tri-plane, akin to a Variational Autoencoder [19].

3.2 Stage 2: Image-Point-Cloud Alignment

We align image and point cloud features within the same feature space. Since the point clouds have been compressed into latent tri-planes with 3D-aware point cloud priors in stage 1, we can project the image features onto this 3D-aware latent. We experiment with two approaches to accomplish this goal. First, following reconstruction methods such as LRM [15], we can utilize a transformer-based structure to align the image features with the point cloud tri-planes, employing a distance metric like L1 or L2 loss. Alternatively, we can employ a probabilistic diffusion approach, wherein we initiate from random noise and denoise it, guided by the input image targeting the point cloud domain.

Our experiments reveal that the transformer-based model yields unstable results. This instability arises because 3D reconstruction from a single image is an ill-posed problem, as the image lacks information

https://doi.org/10.52202/079017-0145

about the occluded object parts. The 3D structural information in an image is not as rich as that in point clouds. Consequently, if we simply adopt a deterministic approach, the alignment result tends to be the average of all possible outcomes for regions that are not sufficiently observed [15]. Thus, we choose a diffusion model as a probabilistic alignment approach. Intuitively, the diffusion model can leverage its prior to envision unobserved regions and generate clear predictions for these areas, as demonstrated in previous 2D prior works [55, 22]. We provide more analysis in Appendices. D.2.

3.2.1 Alignment Model Structure

As shown in Fig. 2, first, we use a pre-trained DINO [2] image encoder to convert images to a $S \times D$ dim feature z_{img} where S is the squence length and D is the feature dimension. Subsequently, the image features serve as conditioning information in the diffusion process, mapping the image feature from the image modality to the point cloud modality.

Unlike previous methods [5, 6, 12] that depend on a single diffusion model to map an arbitrary 1D ordering vector, we employ three parallel UNets to map the image feature to each of the three latent planes, separately. Since each tri-plane contains a 2D-image-like feature, as demonstrated in Appendices, Fig. 6. Prior methods [12, 6] use a single UNet to denoise concatenated planes. However, the UNet is designed without 3D-aware prior, meaning that there will be adjacent pixels without a direct relation due to the convolution operation being a local operation, leading to artifacts.

3.2.2 Alignment Loss Function

We utilize a diffusion model to align the image and the point cloud features. Diffusion Models [14, 40] are designed to learn a data distribution $q(z_0)$ by progressively denoising a normally distributed variable. The learning process is equivalent to performing the reverse operation of a fixed Markov Chain with a length of T, which transforms latent z_0 into purely Gaussian noise $z_T \sim \mathcal{N}(0, I)$ over T time steps. The forward step in this process is defined as:

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1-\beta_t} z_{t-1}, \beta_t I), \tag{6}$$

where noisy variable z_t is derived by scaling the previous noise sample z_{t-1} with $\sqrt{1-\beta_t}$ and adding Gaussian noise following a variance schedule $\beta_1, \beta_2, \ldots, \beta_T$.

We train three parallel diffusion models on each of the three planes by learning to reverse the above diffusion process. To achieve this, we adopt the approach proposed by Aditya *et al.* [39], wherein we use three neural networks that takes z_t to directly predict z_0 corresponding to XY, XZ, and YZ planes. Specifically, given a uniformly sampled time step t from the set $\{1, ..., T\}$, we generate z_t by sampling noise from the input latent tri-plane z_0. A time-conditioned denoising autoencoder consisted of three parallel UNets [40], denoted by Ψ, is employed to reconstruct z_0 from z_t given the alignment source, *i.e.*, the image latent feature z_{img}. The objective of the alignment is given by:

$$\mathcal{L}_{align} = ||\Psi(z_t, z_{img}, \gamma(t)) - z_0||^2, \tag{7}$$

where $\gamma(\cdot)$ represents a positional embedding and $||\cdot||^2$ denotes the mean squared error (MSE) loss.

4 Experiments

Dataset: Following previous works [54, 15], our model is trained on a subset of Objaverse [9] with 140k filtered 3D assets, and evaluated on unseen Objaverse objects and Google Scanned Objects [10]. The Objaverse dataset contains over 800k 3D assets. It was filtered to remove objects with thin faces and repeated/similar buildings, resulting in a final training set of 140k objects. More details can be found in Appendices. B.

Baselines: To assess the effectiveness of our approach, we compare our method with state-of-the-art approaches, including One-2-3-45 [25], LRM [15], SyncDreamer [26], Wonder3D [28], Magic123 [36], TGS [69], LRM [48] and CRM [54].

4.1 Implementation Details

During training, we employ a two-stage training pipeline. In stage 1, a point cloud compressor is trained, comprising a transformer-based point feature extractor and a tri-plane autoencoder. Given

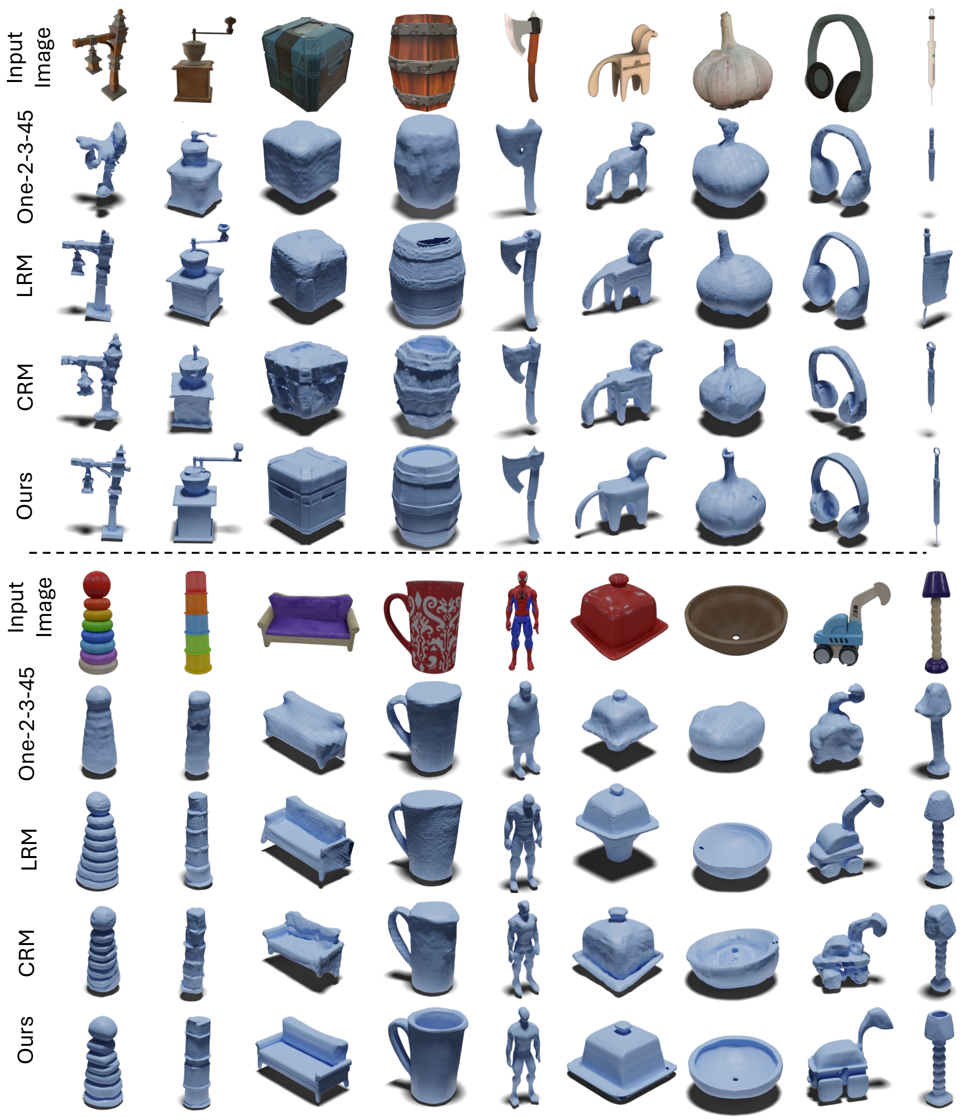

Figure 3: Rendered images of shapes reconstructed by various methods from single images. The upper samples are from Objaverse and the lowers are from Google Scanned Objects.

a point cloud, we sample 8192 points to feed them into the point feature extractor followed by a KNN-based sampler to obtain 512 point features. Subsequently, the point features are projected onto three orthogonal planes to generate a tri-plane with a 128×128 spatial resolution. Then we compress the tri-plane into a latent tri-plane of size ($3{\times}2{\times}32{\times}32$) which has a feature dimension of 2. All components including the point feature extractor, tri-plane autoencoder, and two SDF MLPs, are trained in an end-to-end manner. We utilize 32 NVIDIA V100 GPUs to train for 20 epochs with a batch size of 25 and Adam [18] optimizer, which takes approximate 3 days.

In stage 2, the compressed latent tri-planes are utilized as supervision signals to train our alignment network. For an input image of size ($512{\times}512$), we use a pre-trained DINO [2] to extract image feature of size 1025×768, where 768 is the feature dimension and 1025 is 32×32 image patches plus 1 global image feature. The image features serve as conditioning signal through cross-attention, guiding the diffusion process of the three UNets to generate three planes. During the alignment training, only the UNets are trained, while the DINO remains fixed. The training cost for the alignment is relatively low since we can pre-encode all the point clouds as well as all the source images. We employ 8 V100 GPUs to train the second stage model for 2 days with a batch size of 64.

Table 1: Quantitative comparisons for the geometry quality between our method and baselines.

Method	Chamfer Dist.↓	Vol. IoU↑	F-Sco.(%)↑
One-2-3-45 [25]	0.0172	0.4463	72.19
SyncDreamer [26]	0.0140	0.3900	75.74
Wonder3D [28]	0.0186	0.4398	76.75
Magic123 [36]	0.0188	0.3714	60.66
TGS [69]	0.0172	0.2982	65.17
OpenLRM [15, 13]	0.0168	0.3774	63.22
LGM [48]	0.0117	0.4685	68.69
CRM [54]	0.0094	0.6131	79.38
Ours	**0.0083**	**0.6235**	**85.40**

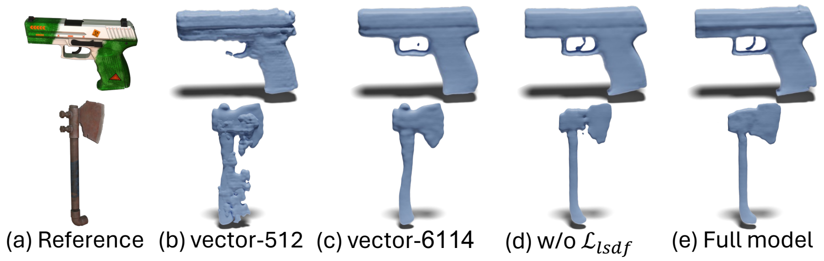
(a) Reference (b) vector-512 (c) vector-6114 (d) w/o $\mathcal{L}_{lsdf}$ (e) Full model

Figure 4: Qualitative comparisons of different latent representations.

The inference stage requires only a single-view image of an object. The process starts by encoding this image into a feature representation using the DINO image encoder. This image feature is then aligned to a latent tri-plane representation through three independent diffusion UNets. The latent tri-plane is subsequently upsampled to a high-resolution tri-plane using the plane decoder and refiner from the first training stage. We then employ the SDF MLP to decode the signed distance for any query position on the tri-plane, which allows us to use the Marching Cubes algorithm to reconstruct a mesh from the tri-plane.

4.2 Comparisons with state-of-the-arts

Qualitative Results: Fig. 3 presents a qualitative comparisons between our approach and other state-of-the-art approaches including One-2-3-45 [25], LRM [15] and CRM [54]. Since LRM is not open-sourced, we use OpenLRM [13], an open-sourced implementation of LRM for comparisons. For the other baselines, we use their official codes and model weights. As shown in Fig. 3. While impressive results can be obtained with these methods, geometric distortions often occur due to ambiguities caused by single image or multi-view inconsistency. Additionally, these approach commonly fail for unseen areas of the input image, such as the inside of a cup. In contrast, our method leverages point cloud priors to reconstruct 3D meshes with better geometry and more details than all other baselines. Moreover, our model does not rely on multi-view consistency, unlike One-2-3-45, LRM, and CRM, thanks to the point cloud prior. This results in significantly reduced geometric distortions, as seen in the box, sofa and bowl examples in Fig. 3.

Quantitative Results: following previous studies [54, 48, 15], we use the Google Scanned Objects (GSO) dataset [10] to evaluate the effectiveness of our method. In line with CRM [54] and One-2-3-45 [25], we use Chamfer Distance (CD) [59, 7], Volume IoU and F-Score with a threshold of 0.05 to evaluate the mesh geometry. The results are shown in Tab. 1. It can be seen that our method outperforms all of the baselines on all the evaluation metrics. This demonstrates the effectiveness of our method for 3D reconstruction. Notably, our model generates mesh within only 6 seconds on an NVIDIA V100 GPU.

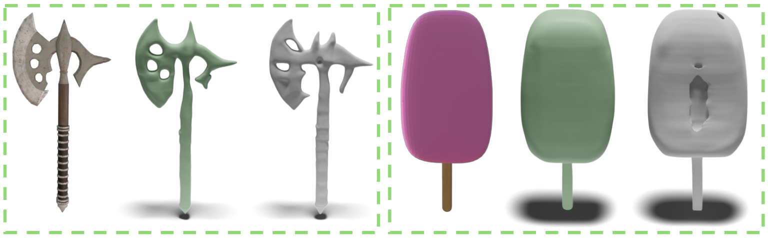

Figure 5: Green objects are generated from parallel UNets and gray samples are from single UNet.

4.3 Ablation Study

For our ablation study, we use a small subset of the Objaverse dataset containing 3000 game assets (weapons, gear, *etc.*) for training, and 216 objects for evaluation.

Effectiveness of latent tri-plane representation: In our method, we compress the point cloud as a latent tri-plane representation, which is then used for image-point-cloud feature alignment. In this study, we evaluate the effectiveness of this proposed latent tri-plane representation.

We compare three different variants in our ablation study: a) latent tri-plane w/ $\mathcal{L}_{lsdf}$: our introduced hierarchical decoding strategy, constraining latent tri-plane with a SDF loss; b) latent tri-plane w/o $\mathcal{L}_{lsdf}$, latent tri-plane without the SDF loss; c) *vector*, a vector with dim (6114×1) (same size with (3×2×32×32)) is used to represent the point cloud latent. We train the stage 1 compression models for point cloud-based reconstruction. As shown in Tab. 2, simply using a latent vector representation without spatial structure does not perform well with a performance of 4.81. However, converting the vector representation to tri-plane improves the performance to 1.95. Moreover, tri-plane with latent SDF loss (Ours in Tab. 2) can further refine the final results to 1.81. This emphasizes the importance of retaining 3D-aware structure in the point cloud latent representation and the necessarity of introducing the latent SDF loss $\mathcal{L}_{lsdf}$.

In Fig. 4, we provide illustrations of different sizes and representations. We observe that a vector size of 512 performs poorly, which is inadequate to represent the 3D shapes. A vector with dimensions of 6114×1 can recover the structure of a 3D mesh, but details such as the trigger are missing. In Fig. 4 (d), we show that tri-plane structure can improve the visual performance with better trigger. In Fig. 4 (e), further refined results with better details can be obtained, validating the effectiveness of the latent SDF loss.

Table 2: Comparisons of different representations.

Method	CD($\times 10^4$)↓
ULIP [63]	35.23
vector (6114×1)	4.81
latent tri-plane w/o $\mathcal{L}_{lsdf}$	1.95
latent tri-plane w/$\mathcal{L}_{lsdf}$ (Ours)	**1.81**

Table 3: Quantitative comparison between parallel diffusion UNet and single UNet.

Method	Single Diffusion	Parallel Diffusion
CD($\times 10^3$)↓	3.92	2.59

Image-point-cloud alignment model vs. ULIP: In order to validate the effectiveness of our proposed image-point-cloud alignment model, we compare our method with ULIP [63], which is designed for point cloud feature alignment with text and images.

ULIP-based image reconstruction: ULIP uses a pre-trained CLIP and point cloud encoder to encode text, images, and point clouds to a feature of size (512×1). Contrastive loss is utilized for aligning the features of point clouds, text, and images. To evaluate the performance of ULIP for 3D reconstruction, we design a decoder network that decodes the (512×1)-dimension point cloud feature vector to a 3D mesh. We then use the decoder network to directly replace the point cloud features with the aligned image feature obtained by ULIP for 3D mesh reconstruction. As shown in Tab. 2, results obtained by ULIP [63] are unstable, leading to worse performance in charmer distance, which proves that the scheme of feature alignment used in ULIP [63] works poor for 3D reconstruction. On contrary, our method can obtain better results.

https://doi.org/10.52202/079017-0145

Parallel Diffusion vs. Single Diffusion: To better retain 3D structure information and reduce the interference of each plane (XY, XZ, YZ) in latent tri-planes, we enlist the use of independent parallel diffusion operations to transfer image features to each respective plane. We valid the effectiveness of using three parallel diffusion operations to do the feature alignment instead of using a single UNet in Tab. 3. Parallel UNet variant has smaller CD error, and we show qualitative comparisons in Fig. 5. We can observe that, thanks to the parallel UNet design, our model can focuses on aligning image feature to a specific plane and thus presents better visual quality. For example, the surface of the ice cream bar is broken in the reconstruction of the single UNet variant, while ours can maintains the continuity of the surface. Notably, to compare equally, we reduce the parameters of parallel diffusion to guarantee the same parameter size of parallel diffusion and single diffusion.

We provide more ablation studies as well as more visualization results in the appendix.

5 Conclusions

In this work, we present the Large Image and Point Clouds Alignment Model (LAM3D), which harnesses the prior of 3D point clouds for high-fidelity 3D mesh creation. We design an effective point-cloud-based network to compress point cloud into compact and meaningful latent tri-plane. Subsequently, taking a single image as input, Image-Point-Clouds Feature Alignment is designed to transfer image features to latent tri-planes, enriching the image features with accurate and effective 3D information for high-fidelity 3D mesh reconstruction. We believe that LAM3D has potential to improve 3D content creation and assist the workflow of digital artists.

Limitations and Future Work: The main goal of the proposed LAM3D is geometry reconstruction but texture reconstruction is not included, so that our method cannot achieve textured mesh reconstruction. We will further extend LAM3D to geometric and texture reconstruction in the future.

Broader Impact: We hold the view that LAM3D has the potential to enhance 3D content creation and aid digital artists in their workflow. LAM3D was conceived with these applications in mind, and we hope it can evolve into a useful tool that reduce the labour cost in 3D asset production. While we do not foresee any immediate harmful uses for LAM3D, we believe it is crucial for users to exercise caution to minimize impacts, considering that the alignment framework can also be employed for harmful intents.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022.

[4] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2023.

[5] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2262–2272, 2023.

[6] Ruikai Cui, Weizhe Liu, Weixuan Sun, Senbo Wang, Taizhang Shang, Yang Li, Xibin Song, Han Yan, Zhennan Wu, Shenzhou Chen, et al. Neusdfusion: A spatial-aware generative model for 3d shape completion, reconstruction, and generation. *arXiv preprint arXiv:2403.18241*, 2024.

https://doi.org/10.52202/079017-0145

[7] Ruikai Cui, Shi Qiu, Saeed Anwar, Jiawei Liu, Chaoyue Xing, Jing Zhang, and Nick Barnes. P2c: Self-supervised point cloud completion from single partial clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14351–14360, 2023.

[8] Ruikai Cui, Shi Qiu, Saeed Anwar, Jing Zhang, and Nick Barnes. Energy-based residual latent transport for unsupervised point cloud completion. *arXiv preprint arXiv:2211.06820*, 2022.

[9] 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.

[10] 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.

[11] Lin Gao, Tong Wu, Yu-Jie Yuan, Ming-Xian Lin, Yu-Kun Lai, and Hao Zhang. Tm-net: Deep generative networks for textured meshes. *ACM Transactions on Graphics (TOG)*, 40(6):1–15, 2021.

[12] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023.

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

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, pages 6840–6851, 2020.

[15] 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.

[16] Jingwei Huang, Hao Su, and Leonidas Guibas. Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*, 2018.

[17] Jinwoo Kim, Jaehoon Yoo, Juho Lee, and Seunghoon Hong. Setvae: Learning hierarchical composition for generative modeling of set-structured data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15059–15068, 2021.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[19] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of International Conference on Learning Representations*, 2014.

[20] 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.

[21] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

[22] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2142–2152, 2023.

[23] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020.

https://doi.org/10.52202/079017-0145

[24] Yuhan Li, Yishun Dou, Xuanhong Chen, Bingbing Ni, Yilin Sun, Yutian Liu, and Fuzhen Wang. Generalized deep 3d shape prior via part-discretized diffusion process. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16784–16794, 2023.

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

[26] 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.

[27] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. In *The Eleventh International Conference on Learning Representations*, 2022.

[28] 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.

[29] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In Maureen C. Stone, editor, *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987*, pages 163–169. ACM, 1987.

[30] 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.

[31] 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.

[32] 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.

[33] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4328–4338, 2023.

[34] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[36] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023.

[37] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[42] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023.

[43] 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. *arXiv preprint arXiv:2310.15110*, 2023.

[44] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023.

[45] 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.

[46] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[47] Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. Variational autoencoders for deforming 3d mesh models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5841–5850, 2018.

[48] 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. *arXiv preprint arXiv:2402.05054*, 2024.

[49] 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.

[50] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, pages 439–457. Springer, 2025.

[51] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.

[52] 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.

[53] Yizhi Wang, Wallace P. Lira, Wenqi Wang, Ali Mahdavi-Amiri, and Hao Zhang. Slice3d: Multi-slice, occlusion-revealing, single view 3d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 9881–9891. IEEE, 2024.

[54] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*, 2024.

https://doi.org/10.52202/079017-0145

[55] Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16284–16294, 2023.

[56] Zhenzhen Weng, Jingyuan Liu, Hao Tan, Zhan Xu, Yang Zhou, Serena Yeung-Levy, and Jimei Yang. Single-view 3d human digitalization with large reconstruction models. *arXiv preprint arXiv:2401.12175*, 2024.

[57] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

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

[59] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. *arXiv preprint arXiv:2111.12702*, 2021.

[60] Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, et al. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. *arXiv preprint arXiv:2401.17053*, 2024.

[61] Jianwen Xie, Yifei Xu, Zilong Zheng, Song-Chun Zhu, and Ying Nian Wu. Generative pointnet: Deep energy-based learning on unordered point sets for 3d generation, reconstruction and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14976–14985, 2021.

[62] 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.

[63] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1189, June 2023.

[64] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. *arXiv preprint arXiv:2305.08275*, 2023.

[65] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019.

[66] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[67] Xinyang Zheng, Yang Liu, Pengshuai Wang, and Xin Tong. Sdf-stylegan: Implicit sdf-based stylegan for 3d shape generation. In *Computer Graphics Forum*, volume 41, pages 52–63. Wiley Online Library, 2022.

[68] 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.

[69] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. *arXiv preprint arXiv:2312.09147*, 2023.

https://doi.org/10.52202/079017-0145

A Diffusion Models Formulation

Diffusion Models are probabilistic models designed to learn a data distribution $z_0 \sim q(z_0)$ by gradually denoising a normally distributed variable. This process corresponds to learning the reverse operation of a fixed Markov Chain with a length of T. The inference process works by sampling a random noise z_T and gradually denoising it until it reaches a meaningful latent z_0. Denoising diffusion probabilistic model (DDPM) [14] defines a diffusion process that transforms latent z_0 to white Gaussian noise $z_T \sim \mathcal{N}(0, I)$ in T time steps. Each step in the forward direction is given by:

$$q(z_1, ..., z_T|z_0) = \prod_{t=1}^{T} q(z_t|z_{t-1}) \tag{8}$$

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1-\beta_t}z_{t-1}, \beta_t I) \tag{9}$$

The noisy latent z_t is obtained by scaling the previous noise sample z_{t-1} with $\sqrt{1-\beta_t}$ and adding Gaussian noise with variance β_t at timestep t. During training, DDPM reverses the diffusion process, which is modeled by a neural network Ψ that predicts the parameters $\mu_\Psi(z_t, t)$ and $\Sigma_\Psi(z_t, t)$ of a Gaussian distribution.

$$p_\Psi(z_{t-1}|z_t) = \mathcal{N}(z_{t-1}; \mu_\Psi(z_t, t), \Sigma_\Psi(z_t, t)) \tag{10}$$

With $\alpha_t := 1 - \beta_t$ and $\bar{\alpha} := \prod_{s=0}^{t} \alpha_s$, we can write the marginal distribution:

$$q(z_t|z_0) = \mathcal{N}(z_t; \bar{\alpha}_t z_0, (1-\bar{\alpha}_t)I) \tag{11}$$

$$z_t = \bar{\alpha}_t z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{12}$$

where $\epsilon \sim \mathcal{N}(0, I)$. Using Bayes' theorem, we can calculate $q(z_{t-1}|z_t, z_0)$ in terms of $\tilde{\beta}_t$ and $\tilde{\mu}_t(z_t, z_0)$ which are defined as follows:

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{13}$$

$$\tilde{\mu}_t(z_t, z_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}z_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}z_t \tag{14}$$

$$q(z_{t-1}|z_t, z_0) = \mathcal{N}(z_{t-1}; \tilde{\mu}_t(z_t, z_0), \tilde{\beta}_t I) \tag{15}$$

Instead of predicting the added noise as in the original DDPM, in this paper, we predict z_0 directly with a neural network Ψ following Aditya *et al.* [39]. The prediction could be used in Eq. 14 to produce $\mu_\Psi(z_t, t)$. Specifically, with a uniformly sampled time step t from $\{1, ..., T\}$, we sample noise to obtain z_t from input latent vector z_0. A time-conditioned denoising auto-encoder Ψ learns to reconstruct z_0 from z_t, guided by the image feature z_{img}. The objective of latent tri-plane diffusion reads

$$\mathcal{L}_{ldm} = \|\Psi(z_t, z_{img}, \gamma(t)) - z_0\|^2 \tag{16}$$

where $\gamma(\cdot)$ is a positional encoding function and $\|\cdot\|^2$ denotes the Mean Squared Error loss.

B Dataset Preparation

The Objaverse contains over 800k 3D assets. However, there are many low-quality 3D assets that are not suitable for training a 3D reconstruction, such as object with only thin faces. We filter those object by removing object with only thin faces, repeated and similar buildings and senses. Finally, we obtain 140k object for training.

For the first stage compression training, we sample data from the object surface. Given a 3D object mesh, we first pre-process it to ensure it forms a watertight shape using the method proposed by Huang *et al.* [16]. Then, we normalize the shape within the $[-1, 1]^3$ box. Subsequently, we randomly sample on-surface points from the shape surface and uniformly sample off-surface points with ground truth SDF values in the $[-1, 1]^3$ space. In addition to the signed distance supervision, we leverage normal directions as an extra guidance to achieve detailed surface modeling. Consequently, we also sample the normal vector for each on-surface point.

For the alignment stage training, we render each object from 48 fixed camera position equally distributed on a sphere covering the object. However, we do not use all 48 rendered images for the

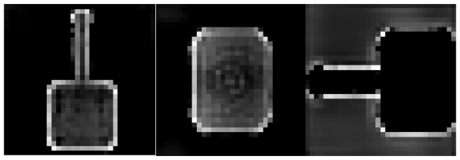

Figure 6: Visualization of latent tri-plane by converting the value to density.

second stage training. Instead, we count the area of observed region and only select the image with the largest observed area to align its feature to the point cloud modality at the second stage. This operation aims to reduce the ambiguity caused by symmetry of objects. By doing this operation, we also have the benefit that camera parameters are no longer needed to do the inference, the second stage model will determine this by itself in and end-to-end manner.

C Evaluation Metrics

We provide a detailed formulation of metrics that are not explicitly defined in the main manuscript to enhance understanding of our experimental outcomes

Chamfer Distance (CD): This metric measures the average distance between two point sets, usually the ground truth and the predicted 3D reconstruction. Given two point sets A and B, the Chamfer Distance is defined as:

$$CD(A,B) = \frac{1}{|A|}\sum_{a\in A}\min_{b\in B}||a-b||^2 + \frac{1}{|B|}\sum_{b\in B}\min_{a\in A}||a-b||^2, \tag{17}$$

where $|A|$and $|B|$ denote the number of points in sets A and B, respectively, and $||a-b||^2$ represents the squared Euclidean distance between points a and b. Note that CD is a symmetric and non-negative distance measure.

To compute the CD for 3D reconstruction evaluation, we first sample 2048 points from the surfaces of the predicted mesh and the ground truth mesh. Then, we calculate the Chamfer distance between the point sets, as defined above.

F-Score: This metric is a measure of the trade-off between precision and recall, which are commonly used metrics for evaluating the quality of a 3D reconstruction. Given a threshold value τ, the F-Score is defined as:

$$F(\tau) = 2\frac{\text{precision}(\tau)\cdot\text{recall}(\tau)}{\text{precision}(\tau)+\text{recall}(\tau)}, \tag{18}$$

where precision(τ) and recall(τ) are the precision and recall values at the given threshold τ, respectively. In our experiment, we set the threshold to 0.05 *w.r.t.* the Chamfer distance.

Volume Intersection over Union (IoU): Volume Intersection over Union (IoU) is another metric used for evaluating the quality of 3D reconstructions. It measures the ratio of the volume of the intersection between the predicted mesh and the ground truth mesh to the volume of their union. The IoU is defined as:

$$IoU = \frac{\text{volume of intersection}}{\text{volume of union}} \tag{19}$$

To compute the IoU for 3D reconstruction evaluation, we first convert the predicted mesh and the ground truth mesh into Signed Distance Functions (SDFs) with a voxel resolution of $128\times128\times128$. All negative voxels in the SDFs represent the interior of the object. Then, we calculate the volume of the intersection and the volume of the union between the two SDFs. Finally, we compute the IoU using the formula above.

D More analysis

D.1 Effect of latent tri-plane resolution and feature dimension

We conduct experiments to study the effect of latent tri-plane resolution and feature dimension. We design four compared variants of our full model. There are two aspects that needs to be considered, *i.e.*

Table 4: Quantitative comparison for the geometry quality between different spatial resolution and feature dimension of latent tri-plane on point cloud to 3D mesh.

Spatial Reso.	Feature Dim.	#parameters	CD($\times 10^4$)↓
8×8	2	384	4.15
16×16	2	1536	2.12
32×32	1	3072	1.87
32×32	2	6144	**1.81**
32×32	4	12288	2.63

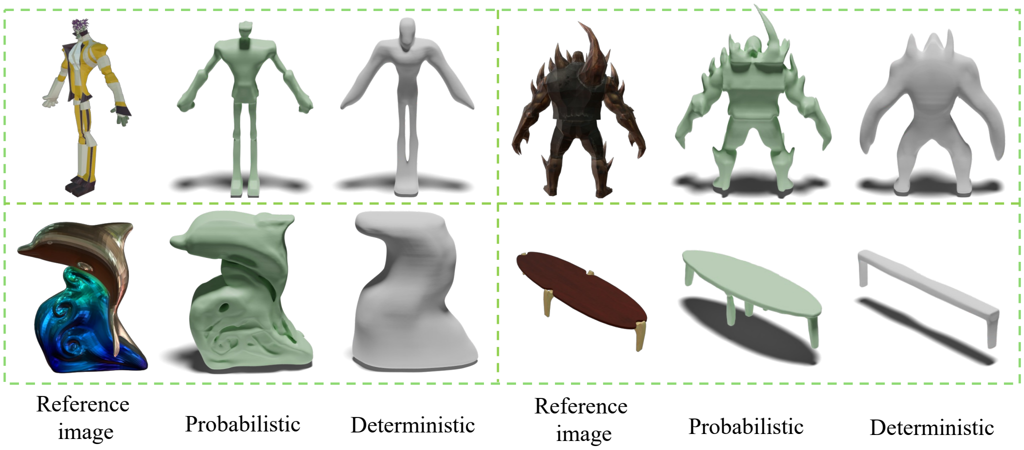

Figure 7: Alignment approach ablation study. We evaluate the single view reconstruction capability of deterministic vs. probabilistic approaches. Green objects are constructed from our probabilistic approach and gray samples are from a deterministic approach. We also present the reference image.

the spatial resolution and the feature dimension of each plane. We vary these parameters and evaluate the point cloud to mesh reconstruction capability in terms of CD. The result is presented in Tab. 4. From which we can observe that setting the spatial resolution and feature dimension as (32×32) and 2 are the best choice. We can also notice that, when we simply introduce a tri-plane structure, the reconstruction quality improved by 0.64 in terms of CD ($\times 10^4$) despite the number of parameters in the $2 \times 8 \times 8$ setting is significantly lower than it in the vector setting presented in Tab. 2. This improvement is brought by the tri-plane structure introduced 3D awareness to the latent structure, which benefits the decoding stage. Another founding is that improving the spatial resolution brings improvements to the reconstruction quality. However, the computation complexity will increase quadraticly as we increase the spatial resolution, therefore, increase the resolution beyond 32×32 becomes intractable in our experiment. Furthermore, we do ablation study on the feature dimension. We notice that increasing the feature dimension is not beneficial anymore, a potential reason is that increasing the feature dimension introduces too many parameters which leads to a latent space that is not compact enough to be continues. Therefore, it performs poorly when generalizing to novel objects. Also, decreasing the feature dimension to 1 leads to a worse performance as there is not adequate parameters to preserve the geometry information of the encoded object.

D.2 Effect of using probabilistic alignment rather than deterministic

In terms of the modality alignment strategy, a navie way to address this is using a projection network that directly project an image feature to the point cloud feature modality. However, since this is an ill-pose problem and a single-view image can corresponding to many possible valid 3D shapes due to occlusion. A deterministic approach tends to produce blurry predictions [15]. We demonstrate such an effect by designing a deterministic Image-Point-Cloud alignment network. To this end, we follow LRM [15] and use a 12-layer transformer to directly align image feature to point cloud feature modality and train on the 140k Objavese dataset. Fig. 7 shows the difference between our probabilistic alignment approach compared with a deterministic method. From which we can observe that the reconstruction with probabilistic process involved produces more clear and sharp details than a deterministic approach.

https://doi.org/10.52202/079017-0145

Table 5: Quantitative comparison for the geometry quality between our full model w/ or w/o the plane refiner on point cloud to 3D mesh.

Method	w/o Refiner	w/ Refiner
CD($\times 10^4$)$\downarrow$	2.37	**1.81**

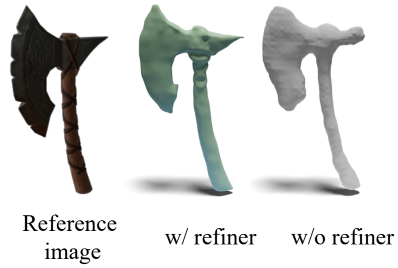

Figure 8: We evaluate the single view reconstruction capability of our model w/ and w/o the plane refiner. Green objects are constructed from our model with the plane refiner and gray samples are from a the model without the plane refiner. We also present the reference image.

D.3 Effect of plane refiner

To evaluate the effectiveness of our plane refiner module, we design a comparison about our full model with plane refiner and a variant without plane refiner. The result is shown in Tab. 5, and we also present a qualitative result in Fig. 8. We can observe that the plane refiner module improves the capability of our compression model, allowing it to recover more fine-grained details.

E More Visualization

We present more visualization for 3D reconstruction by aligning image and point cloud modality using our proposed method in Fig. 9 and Fig 10.

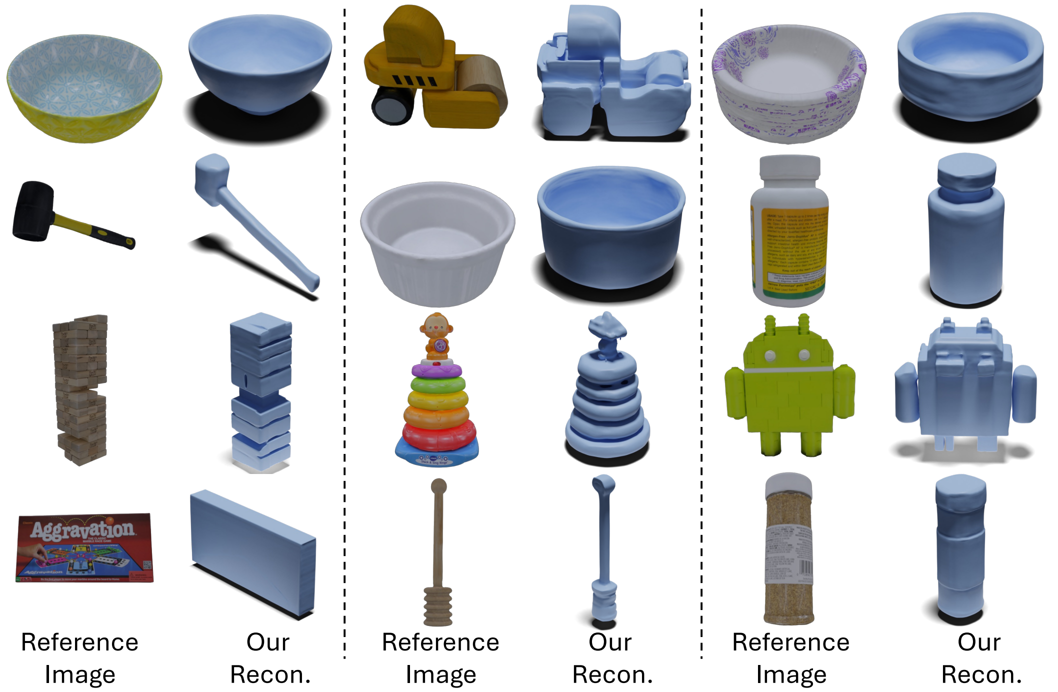

Figure 9: Rendered images of shapes reconstructed by our LAM3D from single images on the GSO dataset. For each tuple of samples, the left image is the reference image and the right image is the reconstructed geometry.

https://doi.org/10.52202/079017-0145

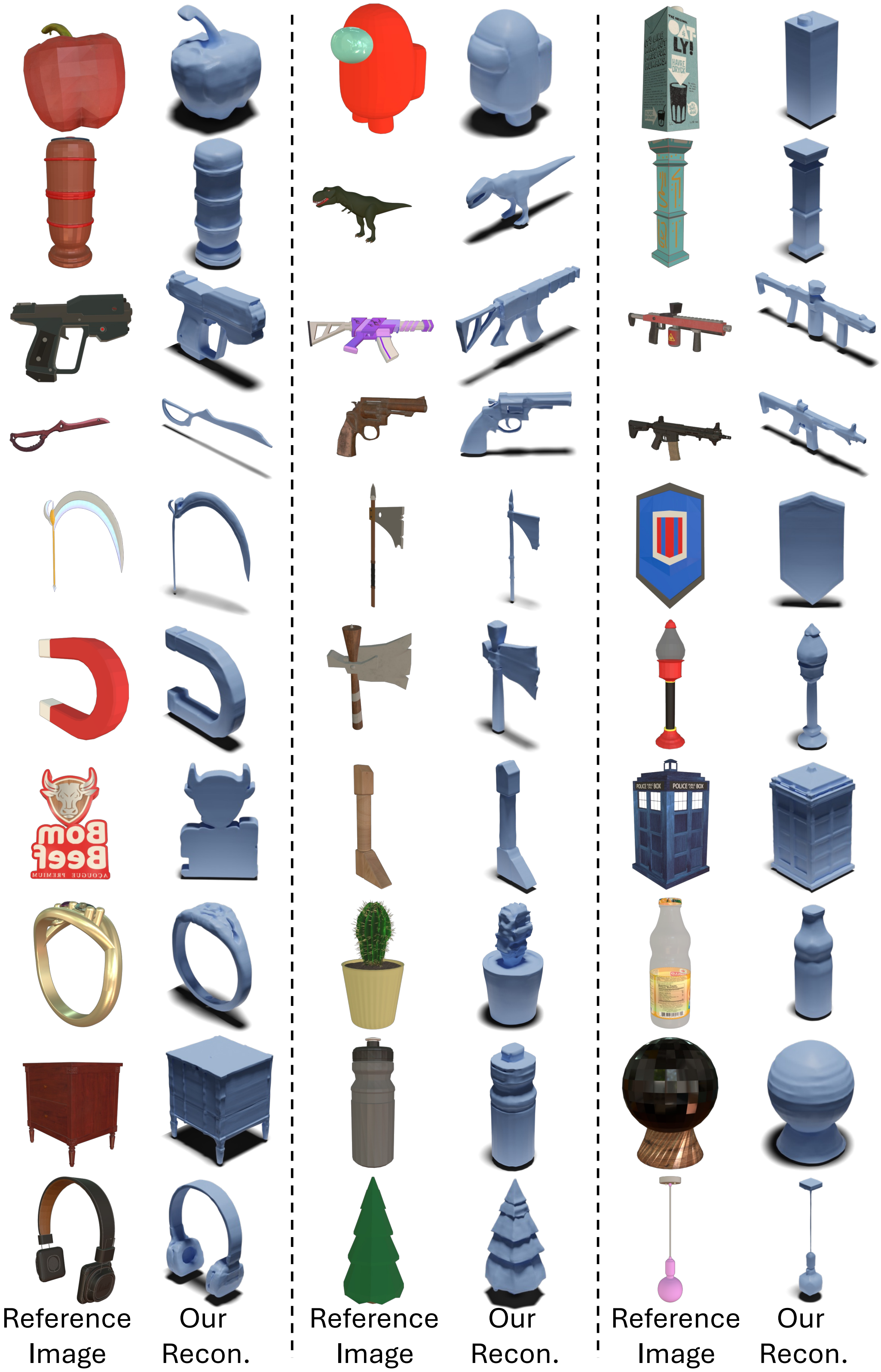

Figure 10: Rendered images of shapes reconstructed by our LAM3D from single images on the Objaverse dataset. For each tuple of samples, the left image is the reference image and the right image is the reconstructed geometry.

https://doi.org/10.52202/079017-0145

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We have clearly stated the claims made in the abstract and introduction, accurately reflecting the paper's contributions and scope, including the contributions made in the paper and important assumptions and limitations.

   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: We have discussed the limitations of our work in the paper, which include the absence of texture recovery and the occasional presence of local geometries in the reconstructed shape that are misaligned with the reference image.

   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**

https://doi.org/10.52202/079017-0145

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

Answer: [NA]

Justification: Our paper does not include theoretical results, and therefore, this question is not applicable to our work.

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: We provide a detailed description of the model and experimental settings in our paper, ensuring that readers have the necessary information to reproduce the main experimental results. Additionally, we plan to release the code to further enhance reproducibility and facilitate verification of our results.

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).

https://doi.org/10.52202/079017-0145

(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: While we currently do not provide open access to the data and code, we plan to release the code along with sufficient instructions to reproduce the main experimental results after the paper has been accepted.

   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: We have provided all the necessary details regarding the training and testing process, including data splits, hyperparameters, the rationale for their selection, and the type of optimizer used.

   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]

https://doi.org/10.52202/079017-0145

Justification: Our paper does not report error bars or other information about the statistical significance of the experiments. However, we follow common practices in our research area and have comprehensively evaluated our method to demonstrate its effectiveness, which ensures that our results are reliable and meaningful within the context of our work.

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: We provided sufficient information on the computer resources needed to reproduce the experiments in the implementation details section.

   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: Our research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   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: This paper discussed both potential positive societal impacts and negative societal impacts of the work performed

    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: [Yes]

    Justification: Our method presents a low risk for misuse or dual-use. Additionally, we have taken precautions to ensure responsible data handling by filtering our training data to avoid unsafe content.

    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?

https://doi.org/10.52202/079017-0145

Answer: [Yes]

Justification: We have ensured that all the assets used in our research, including code, data, and models, are properly credited. The licenses and terms of use for these assets have been explicitly mentioned and are adhered to in our work, demonstrating our respect for the rights and contributions of the original creators.

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 did not submit any new assets at the time of submission. However, we plan to release well-documented code after the paper's acceptance.

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: Our research does not involve crowdsourcing nor research with human subjects.

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.

https://doi.org/10.52202/079017-0145

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: Our research does not involve crowdsourcing nor research with human subjects.

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.

https://doi.org/10.52202/079017-0145