

# WAVELET LATENT DIFFUSION (WALA): BILLION-PARAMETER 3D GENERATIVE MODEL WITH COMPACT WAVELET ENCODINGS

Aditya Sanghi      Aliasghar Khani\*      Pradyumna Reddy\*      Arianna Rampini  
Derek Cheung      Kamal Rahimi Malekshan      Kanika Madan      Hooman Shayani

<https://autodeskailab.github.io/WaLaProject>

## ABSTRACT

Large-scale 3D generative models require substantial computational resources yet often fall short in capturing fine details and complex geometries at high resolutions. We attribute this limitation to the inefficiency of current representations, which lack the compactness required to model the generative models effectively. To address this, we introduce a novel approach called **Wavelet Latent Diffusion**, or **WaLa**, that encodes 3D shapes into a wavelet-based, compact latent encodings. Specifically, we compress a  $256^3$  signed distance field into a  $12^3 \times 4$  latent grid, achieving an impressive 2,427 $\times$  compression ratio with minimal loss of detail. This high level of compression allows our method to efficiently train large-scale generative networks without increasing the inference time. Our models, both conditional and unconditional, contain approximately one billion parameters and successfully generate high-quality 3D shapes at  $256^3$  resolution. Moreover, WaLa offers rapid inference, producing shapes within two to four seconds depending on the condition, despite the model's scale. We demonstrate state-of-the-art performance across multiple datasets, with significant improvements in generation quality, diversity, and computational efficiency. We open-source our code and, to the best of our knowledge, release the largest pretrained 3D generative models across different modalities: <https://github.com/AutodeskAILab/WaLa>.

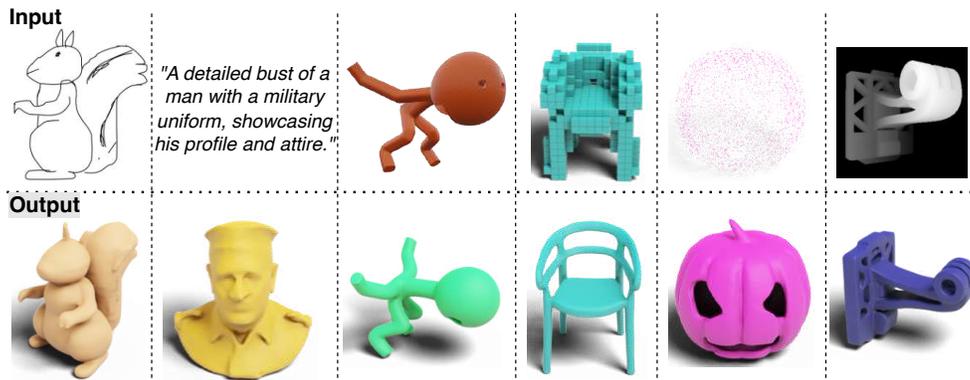


Figure 1: We propose a new 3D generative model, called WaLa, that can generate shapes from conditions such as sketches, text, single-view images, low-resolution voxels, point clouds & depth-maps.

\* Equal contribution. For further inquiries, please email [aditya.sanghi@autodesk.com](mailto:aditya.sanghi@autodesk.com)

# 1 INTRODUCTION

Training generative models on large-scale 3D data presents significant challenges. The cubic nature of 3D data drastically increases the number of input variables the model must handle, far exceeding the complexity found in image and natural language tasks. This complexity is further compounded by storage and streaming issues. Training such large models often requires cloud services, which makes the process expensive for high-resolution 3D datasets as these datasets take up considerable space and are slow to stream during training. Additionally, unlike other data types, 3D shapes can be represented in various ways, such as voxels, point clouds, meshes, and implicit functions. Each representation presents different trade-offs between quality and compactness. Determining which representation best balances high fidelity with compactness for efficient training and generation remains an open challenge. Finally, 3D representations often exhibit complex hierarchical structures with details at multiple scales, making it challenging for a generative model to capture both global structure and fine-grained details simultaneously.

To address these challenges, current state-of-the-art methods for large generative models typically employ three main strategies. The first strategy involves using low-resolution representations, such as sparse point clouds (Nichol et al., 2022c; Jun & Nichol, 2023b), low-polygon meshes (Chen et al., 2024b), or coarse grids (Cheng et al., 2023; Sanghi et al., 2023b). While these approaches reduce computational complexity, they are limited in their ability to model the full distribution of 3D shapes, struggle to capture intricate details, and often lead to lossy representations. The second approach represents 3D shapes through a collection of 2D images (Yan et al., 2024a) or incorporates images (Hong et al., 2023; Li et al., 2023a; Liu et al., 2024; Xu et al., 2023b; Siddiqui et al., 2024; Bensadoun et al., 2024) into the training loss. However, this method suffers from long training times due to the need for rendering and can fail to capture internal details of 3D shapes, as it primarily focuses on external appearances. The third strategy introduces more compactness into the input representations (Hui et al., 2024; Zhou et al., 2024; Ren et al., 2024; Yariv et al., 2024; Xiong et al., 2024; Zhang et al., 2024) to reduce the number of variables the generative model must handle. While these representations can be sparse (Ren et al., 2024; Yariv et al., 2024; Xiong et al., 2024), they are often irregular or discrete in nature making it challenging to be modeled via neural networks and can still be relatively large compared to image or natural language data (Hui et al., 2024; Zhou et al., 2024), thus making it difficult to scale the model parameters efficiently.

One prominent compact input representation is wavelet-based representation, which includes Neural Wavelet (Hui et al., 2022), UDiFF (Zhou et al., 2024), and wavelet-tree frameworks (Hui et al., 2024). These methods utilize wavelet transforms and their inverses to seamlessly convert between wavelet spaces and high-resolution truncated signed distance function (TSDF) representations. They offer several key advantages: data can be easily compressed by discarding selected coefficients with minimal loss of detail, and the interrelationships between coefficients facilitate efficient storage, streaming, and processing of large-scale 3D datasets compared to directly using TSDFs (Hui et al., 2024). However, despite these benefits, wavelet-based representations remain substantially large, especially when scaling up for large-scale generative models. For example, a  $256^3$  TSDF can be represented as a wavelet-tree of size  $46^3 \times 64$  (Hui et al., 2024), which is equivalent to a  $1440 \times 1440$  RGB image. Scaling within this space continues to pose significant challenges.

In this work, we build upon the wavelet representation described above and introduce the *Wavelet Latent Diffusion (WaLa)* framework. This framework further compresses the wavelet representation to obtain compact latent encodings without significant information loss, thereby efficiently enabling us to scale a diffusion-based generative model within this space. Starting with a truncated signed distance function (TSDF) of a shape, we first convert it into 3D wavelet tree representation as in Hui et al. (2024). Then, we train a convolution-based VQ-VAE model with adaptive sampling loss and *balanced fine-tuning* to compress a  $256^3$  TSDF into a  $12^3 \times 4$  grid, achieving a remarkable  $2,427 \times$  compression ratio while maintaining an impressive reconstruction without a significant loss of detail. For example, as shown in Table 1, an Intersection over Union (IOU) of 0.978 is achieved on the GSO dataset. Compared to other representations, this approach requires fewer input variables for the generative model while retaining high reconstruction accuracy. Consequently, the generative model does not need to model local details and can focus on capturing the global structure. Moreover, by significantly reducing the number of input variables that the generative model must handle due to this compression, we enable the training of large-scale 3D generative models with up to a billion parameters, producing highly detailed and diverse shapes. WaLa also supports controlled generation through multiple input modalities without adding significant inductive biases, making the framework



flexible and adaptable beyond single-view 3D reconstruction tasks. As a result, our model generates 3D shapes with complex geometry, plausible structures, intricate topologies, and smooth surfaces. This is demonstrated in Figures 1 and 2, where high-quality 3D meshes can be obtained by applying marching cubes to the SDF generated from different input modalities such as text, sketch, low-resolution voxel, point cloud, single-view, and multi-view images.

In summary, we make the following contributions:

- We introduce a *Wavelet Latent Diffusion (WaLa)* framework that tackles the dimensional and computational challenges of 3D generation with impressive compression while maximizing fidelity.
- Our large billion-parameter model generates high-quality 3D shapes within two to four seconds, significantly outperforming state-of-the-art benchmarks in 3D shape generation.
- Our model demonstrates exceptional versatility, accepting diverse input modalities such as single/multi-view images, voxels, point clouds, depth data, sketches, and textual descriptions (see Figure 1 and 2), making it applicable to a wide range of 3D modeling tasks.
- To foster reproducibility and stimulate further research in this domain, we release what we believe is, to the best of our knowledge, the largest 3D generative model to date that works across various input modalities, comprising approximately one billion parameters. The model is available at <https://github.com/AutodeskAILab/WaLa>.

Table 1: 3D representations compared on GSO dataset (Downs et al., 2022): Intersection over Union (IoU) for accuracy & number of input variables for generative models to evaluate complexity.

Representation	IoU	Number of Input Variables
Ground-truth SDF ( $256^3$ )	1.0	16, 777, 216 (~ 64MB)
Point Cloud (Nichol et al., 2022a)	0.8642	12, 288 (~ 0.05MB)
Latent Vectors (Jun & Nichol, 2023a)	0.8576	1, 048, 576 (~ 4MB)
Coarse Component (Hui et al., 2022)	0.9531	97, 336 (~ 0.4MB)
Wavelet tree (Hui et al., 2024)	0.9956	1, 129, 528 (~ 4.3MB)
<b>WaLa</b>	0.9780	6, 912 (~ 0.03MB)

## 2 RELATED WORK

**Neural Shape Representations.** Several representations have been explored for Deep learning for 3D data. Initially, volumetric methods using 3D convolutional networks were employed (Wu et al., 2015; Maturana & Scherer, 2015), but they were limited by resolution and efficiency. The field then advanced to multi-view CNNs that apply 2D processing to rendered views (Su et al., 2015; Qi et al., 2016), and further explored sparse point cloud representations with networks like PointNet and its successors (Qi et al., 2017a;b; Wang et al., 2019). Additionally, neural implicit representations for compact, continuous modeling were developed (Park et al., 2019; Mescheder et al., 2019; Chen & Zhang, 2019). Explicit mesh-based and boundary representations (BREP) have gained attention, enhancing both discriminative and generative capabilities in CAD-related applications (Hanocka et al., 2019; Chen et al., 2024b; Jayaraman et al., 2021; Lambourne et al., 2021). Recently, wavelet representations (Hui et al., 2022; Zhou et al., 2024; Hui et al., 2024) have become popular. Wavelet decompositions of SDF signals enable tractable modeling of high-resolution shapes. In this work, we extend the previous research by addressing the dimensional and computational hurdles of 3D generation. Our novel techniques for efficient shape processing enable high-quality 3D generation at scale, accommodating datasets with millions of shapes.

**3D Generative Models.** 3D generative models have evolved rapidly, initially dominated by Generative Adversarial Networks (GANs) (Goodfellow et al., 2014; Wu et al., 2016). Subsequent advancements integrated differentiable rendering with GANs, utilizing multi-view losses for enhanced fidelity (Chan et al., 2022). Parallel developments explored normalizing flows (Yang et al., 2019; Klokov et al., 2020; Sanghi et al., 2022) and Variational Autoencoders (VAEs) (Mo et al., 2019). Additionally, autoregressive models also gained traction for their sequential generation capabilities (Cheng et al., 2022; Nash et al., 2020; Sun et al., 2020; Mittal et al., 2022; Yan et al., 2022; Zhang et al., 2022; Sanghi et al., 2023a). The recent success of diffusion models in image generation has sparked a great interest in their application to 3D contexts. Most current approaches

employ a two-stage process: first, a Vector-Quantized VAE (VQ-VAE) on 3D representations such as triplanes (Shue et al., 2023b; Chou et al., 2023; Peng et al., 2020; Reddy et al., 2024; Siddiqui et al., 2024; Chen et al., 2022; Gao et al., 2022b; Shue et al., 2023a), implicit forms (Zhang et al., 2023a; Li et al., 2023b; Cheng et al., 2023), or point clouds (Jun & Nichol, 2023a; Zeng et al., 2022) is trained, and then, diffusion models are applied to the resulting latent space. Incorporating autoencoders to process latent spaces allow for the generation of complex representations like point clouds (Jun & Nichol, 2023a; Zeng et al., 2022) and implicit forms (Zhang et al., 2023a; Li et al., 2023b; Cheng et al., 2023; Zhang et al., 2024). Direct training of diffusion models on 3D representations, though less explored, has shown promise for point clouds (Nichol et al., 2022a; Zhou et al., 2021; Luo & Hu, 2021; Nakayama et al., 2023), voxels (Zheng et al., 2023), occupancy (Ren et al., 2024), and neural wavelet coefficients (Hui et al., 2022; Liu et al., 2023d; Hui et al., 2024). Our work advances this frontier by bridging the gap between compact representation and high-fidelity generation.

**Conditional 3D Models.** Two primary paradigms dominate conditional 3D generative models, each with its own approach to 3D content creation. The first paradigm ingeniously repurposes large-scale 2D conditional image generators, such as (Rombach et al., 2022a) or Imagen (Saharia et al., 2022), for 3D synthesis. This approach employs a differentiable renderer to project 3D shapes into 2D images, enabling comparison with target images or alignment with text-to-image model distributions (Jain et al., 2022; Michel et al., 2022; Poole et al., 2022). Initially focused on text-to-3D generation, this method has expanded to accommodate various input modalities, including single and multi-view images (Deng et al., 2023; Melas-Kyriazi et al., 2023; Xu et al., 2022; Liu et al., 2023c; Deitke et al., 2023; Qian et al., 2023; Shi et al., 2023; Wang et al., 2023; Liu et al., 2023b), and even sketches (Mikaeili et al., 2023). This approach, while novel, is limited by its computational demands. An alternative paradigm uses dedicated conditional 3D generative models trained on either paired datasets or through zero-shot learning. These paired models show adaptability to various input conditions, ranging from point clouds (Zhang et al., 2022; 2023b) and images (Zhang et al., 2022; Nichol et al., 2022a; Jun & Nichol, 2023a; Zhang et al., 2023b; Chen et al., 2024a; Tang et al., 2024; Li et al., 2023a; Xu et al., 2024; Zhang et al., 2024; Siddiqui et al., 2024; Bensadoun et al., 2024) to low-resolution voxels (Chen et al., 2021; 2023b), sketches (Lun et al., 2017; Guillard et al., 2021; Gao et al., 2022a; Kong et al., 2022), and textual descriptions (Nichol et al., 2022a; Jun & Nichol, 2023a; Ren et al., 2024; Yariv et al., 2024). Concurrently, zero-shot methods have gained traction, particularly in text-to-3D (Sanghi et al., 2022; 2023a; Liu et al., 2022; Xu et al., 2023a; Yan et al., 2024b) and sketch-to-3D applications (Sanghi et al., 2023b), showcasing the potential for more flexible and generalizable 3D generation. We expand on the second paired paradigm, developing a large-scale paired conditional generative model for 3D shapes. This approach enables fast generation without per-instance optimization, supports diverse inputs, and facilitates unconditional generation and zero-shot tasks like shape completion.

### 3 METHOD

Training generative models on large-scale 3D data is challenging because of the data’s complexity and size. This has driven the creation of compact representations like neural wavelets, facilitating efficient neural network training. To represent a 3D shape with wavelets, it is first converted into a Truncated Signed Distance Function (TSDF) grid. A wavelet transform is then applied to decompose this TSDF grid into coarse coefficients ( $C_0$ ) and detail coefficients at various levels ( $D_0, D_1, D_2$ ). Various wavelet transforms, such as Haar, biorthogonal, or Meyer wavelets, can be employed. Most current methods utilize the biorthogonal wavelet transform (Hui et al., 2022; Zhou et al., 2024; Hui et al., 2024). The coarse coefficients primarily capture the essential shape information, while the detail coefficients represent high-frequency details. To compress this representation, different filtering schemes can be applied to remove certain coefficients, though this involves a trade-off in reconstruction quality. In the neural wavelet representation (Hui et al., 2022), all detail coefficients are discarded during the training of the generative model and a regression network is used to predict the missing detail coefficients  $D_0$ . In contrast, the wavelet-tree representation (Hui et al., 2024) retains all coarse coefficients ( $C_0$ ), discards the third level of detail coefficients ( $D_2$ ), and selectively keeps the most significant coefficients from  $D_0$  along with their corresponding details in  $D_1$  using a subband coefficient filtering scheme. The neural wavelet representation, while modeling a smaller number of input variables, has lower reconstruction quality than the wavelet-tree representation, making latter a more attractive option.

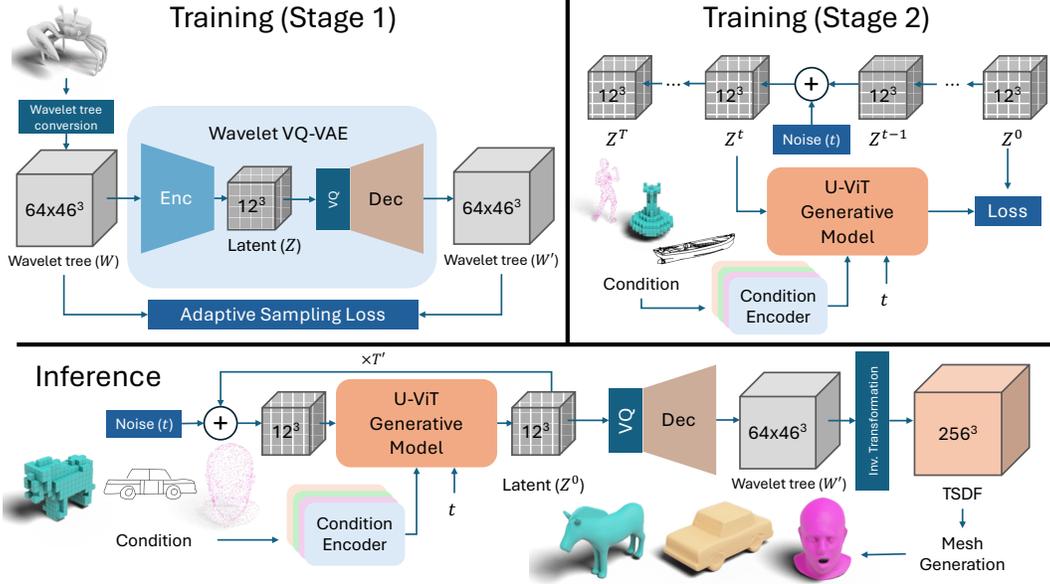


Figure 3: Overview of the WaLa network architecture and 2-stage training process and inference method. Top Left: Stage 1 autoencoder training, compressing diffusible wavelet tree ( $W$ ) shape representation into a compact latent space. Top Right: Conditional/unconditional diffusion training. Bottom: Inference pipeline, illustrating sampling from the trained diffusion model and decoding the sampled latent into a Wavelet Tree ( $W$ ), then into a mesh.

Building upon these efficient wavelet representations, our method requires a large collection of 3D shapes. Let  $\mathcal{S} = \{(W_n, \Theta_n)\}_{n=1}^N$ , denote a dataset of  $N$  3D shapes such that each shape  $S_n \in \mathcal{S}$  is represented by a diffusible wavelet tree representation  $W_n$  (Hui et al., 2024) and an optional associated condition  $\Theta_n$ . The representation  $W_n \in \mathbb{R}^{46^3 \times 64}$  is obtained by converting a TSDF of resolution  $256^3$ . Depending on the conditional generative model, the condition  $\Theta_n$  can be a single-view image, multi-view images, a voxel representation, a point cloud, or multi-view depth maps, and may be omitted if the model is unconditional or when training the vector-quantized autoencoder (VQ-VAE). Training our model comprises two stages: First, we train a convolution-based VQ-VAE to encode the diffusible wavelet tree representation into a more compact grid latent space  $Z$  using the adaptive sampling loss. At this stage, to further enhance reconstruction accuracy, we fine-tune the VQ-VAE using a simple approach we call *balanced fine-tuning*. This VQ-VAE encodes a latent grid  $Z_n \in \mathbb{R}^{12^3 \times 4}$  for each shape  $S_n \in \mathcal{S}$ . In the second stage, we train a diffusion-based generative model on this latent grid  $Z_n$  that can be conditioned on a sequence of condition vectors derived from one of the aforementioned conditions. During inference, we initiate with a completely noisy latent vector and employ the conditional generative network to denoise it progressively through the inverse diffusion process, utilizing classifier-free guidance. The two-step process is detailed in Figure 3 and is also explained next.

### 3.1 STAGE 1: WAVELET VQ-VAE

Our primary objective is to compress the diffusible wavelet tree representation (Hui et al., 2024) into a compact latent space without significant loss of fidelity, thereby facilitating the training of a generative model directly on this latent space. Decoupling compression from generation allows for efficient scaling of a large generative model within the latent space. To this end, we employ a convolution-based VQ-VAE, known for producing sharper reconstructions and mitigating issues like posterior collapse (Van Den Oord et al., 2017; Razavi et al., 2019; Baykal et al., 2024). Specifically, the encoder  $Enc(\cdot)$  maps the input  $W_n$  to a latent representation  $Z_n = Enc(W_n)$ , which is then quantized as  $VQ(Z_n)$  via a vector quantization layer and decoded by  $Dec(\cdot)$  to reconstruct the shape  $W'_n = Dec(VQ(Z_n))$ . By integrating the vector quantization layer with the decoder, as in

(Rombach et al., 2022b), we ensure that the generative model is trained on pre-quantized latent codes. This approach leverages the robustness of the quantization layer to small perturbations by mapping generated codes to the nearest embeddings in a codebook after generation. Empirical results confirm the effectiveness of this strategy, see Ablation Section C.4.

To train the VQ-VAE, we employ a combination of three losses: a reconstruction loss to ensure fidelity between the original and reconstructed shapes, a codebook loss to encourage the codebook embeddings to adapt to the distribution of encoder outputs, and a commitment loss to align the encoder’s outputs closely with the codebook embeddings. We apply a reconstruction loss  $\mathcal{L}_{\text{rec}}(W_n, W'_n)$ , during which we adopt an adaptive sampling loss strategy (Hui et al., 2024) to focus more effectively on high-magnitude detail coefficients (i.e.,  $D_0$  and  $D_1$ ) while still considering the others. Since most detail coefficients are low in magnitude and contribute minimally to the overall shape quality, this approach identifies the significance of these coefficients in each subband based on their magnitude relative to the largest coefficient, forming a set  $P_0$  of important coordinates. By structuring the training loss to emphasize these crucial coefficients and incorporating random sampling of less important ones, the model efficiently concentrates on key information without neglecting finer details. This is formalized in the equation below:

$$\mathcal{L}_{\text{rec}} = L_{\text{MSE}}(C_0, C'_0) + \frac{1}{2} \sum_{D \in \{D_0, D_1\}} [L_{\text{MSE}}(D[P_0], D'[P_0]) + L_{\text{MSE}}(R(D[P'_0]), R(D'[P'_0]))] \quad (1)$$

In this context,  $L_{\text{MSE}}(X, Y)$  denotes the mean squared error between  $X$  and  $Y$ . The coefficients  $C_0, D_0, D_1$  extracted from  $W_n$  represent the coarse and detail components, respectively, while their reconstructed counterparts  $C'_0, D'_0, D'_1$  are derived from the reconstructed  $W'_n$ . The notation  $D[P_0]$  refers to the coefficients in  $D$  at the positions specified by the set  $P_0$ , with  $P'_0$  being its complement. The function  $R(D[P'_0])$  randomly selects coefficients from  $D[P'_0]$  such that the number of selected coefficients equals  $|P_0|$ . By balancing the number of coefficients in the last two terms of the loss function, we emphasize critical information while regularizing less significant coefficients through random sampling. This approach is also empirically validated in Ablation Section C.1.

Our model is initially trained on 10 million samples collected from 19 different datasets (see Section 4.1 for details). However, we observed that a substantial portion of this data is skewed towards simple CAD objects, introducing a bias in the training process. This imbalance can cause the model to underperform on more complex or less-represented 3D shapes. To address this issue, we fine-tune the converged VQ-VAE model using an equal number of samples from each of the 19 datasets — a process we call *balanced fine-tuning*. This approach ensures that the model is exposed uniformly to the diverse range of shapes and complexities present across all datasets, thereby reducing the bias introduced by the initial imbalance. Empirically, we find that *balanced fine-tuning* enhances reconstruction results across datasets, as demonstrated in our ablation study (Section C.2).

### 3.2 STAGE 2: LATENT DIFFUSION MODEL

In the second stage, we train a large-scale generative model with billions of parameters on the latent grid, either as an unconditioned model to capture the data distribution or conditioned on diverse modalities  $\Theta_n$  (e.g., point clouds, voxels, images). We use a diffusion model within the Denoising Diffusion Probabilistic Models (DDPM) framework (Ho et al., 2020), modeling the generative process as a Markov chain with two phases.

First, the forward diffusion process gradually adds Gaussian noise to the initial latent code  $Z_n^0$  over  $T$  steps, resulting in  $Z_n^T \sim \mathcal{N}(0, I)$ . Then, the reverse denoising process employs a generator network  $\theta$ , conditioned on  $\Theta_n$ , to systematically remove the noise and reconstruct  $Z_n^0$ . The generator predicts the original latent code  $Z_n^0$  from any intermediate noisy latent codes  $Z_n^t$  at time step  $t$ , using  $f_\theta(Z_n^t, t, \Theta_n) \approx Z_n^0$ , and is optimized using a mean-squared error loss:

$$\mathcal{L} = \mathbb{E}_t [\|f_\theta(Z_n^t, t, \Theta_n) - Z_n^0\|^2]$$

Here,  $Z_n^t$  is obtained by adding Gaussian noise  $\epsilon$  to  $Z_n^0$  at time step  $t$  using a cosine noise schedule (Dhariwal & Nichol, 2021). The condition  $\Theta_n$  is a latent set of vectors derived from various conditioning modalities, injected into the U-ViT generator (Hoogeboom et al., 2023) by using

cross-attention and by modulating the normalization parameters in the ResNet and cross-attention layers, as described in Esser et al. (2024). This is achieved via condition encoders for different modalities. During training, we apply a small dropout to the condition to implement classifier-free guidance during inference. In the case of unconditional generation, no conditioning is applied. For most input conditions (point clouds, voxels, images, multi-view images, and multi-view depth) we directly train a different conditional generative model for each condition, while for the conditioning on sketch and the single-depth, we take the image-conditioned generative model and fine-tune it with synthetic sketch data and depth data, respectively. For text-to-3D, we fine-tune MVDream (Xu et al., 2023b) to generate six multi-view depth images, as this provides better reconstruction than multi-view images (see experiments in Section 4.2.3), and then use our model during inference. Further details are provided in the appendix.

### 3.3 INFERENCE

At test time, we begin with a randomly generated noisy latent encoding  $Z_n^T \sim \mathcal{N}(0, I)$  and iteratively denoise it to reconstruct the original latent code  $Z_n^0$  through the reverse diffusion process, as described in DDPM (Ho et al., 2020). For conditional generation, we apply classifier-free guidance (Ho & Salimans, 2022) by interpolating between the unconditional and conditional denoising predictions, steering the generation process toward the desired output. This approach allows for greater control over the quality-diversity trade-off. Once the final latent code  $Z_n^0$  is obtained, we use the pre-trained decoder network of the VQ-VAE from 3.1 to generate the final 3D shape in the wavelet form. Subsequently, we apply the inverse wavelet transform to obtain the final 3D shape as an TSDF that can further be converted to a mesh using marching cubes. Notably, we can generate multiple samples for the same conditional input by using different initializations of the noisy latent grid.

## 4 RESULTS

### 4.1 EXPERIMENTAL SETUP

**Datasets.** Our training data consists of over 10 million 3D shapes, assembled from 19 publicly available datasets, including ModelNet Vishwanath et al. (2009), ShapeNet Chang et al. (2015), SMPL Loper et al. (2015), Thingi10K Zhou & Jacobson (2016), SMAL Zuffi et al. (2017), COMA Ranjan et al. (2018), House3D Wu et al. (2018), ABC Koch et al. (2019), Fusion 360 Willis et al. (2021), 3D-FUTURE Fu et al. (2021), BuildingNet Selvaraju et al. (2021), DeformingThings4D Li et al. (2021), FG3D Liu et al. (2021), Toys4K Stojanov et al. (2021), ABO Collins et al. (2022), Infinigen Raistrick et al. (2023), Objaverse Deitke et al. (2023), and two subsets of ObjaverseXL Deitke et al. (2023) (Thingiverse and GitHub). These individual datasets target specific object categories: for instance, CAD models (ABC and Fusion 360), furniture (ShapeNet, 3D-FUTURE, ModelNet, FG3D, ABO), human figures (SMPL and DeformingThings4D), animals (SMAL and Infinigen), plants (Infinigen), faces (COMA), and houses (BuildingNet, House3D). Additionally, Objaverse and ObjaverseXL cover a broader range of generic objects sourced from the internet, covering the aforementioned categories and other diverse objects. Following Hui et al. (2024), each of these 19 datasets was split into two parts for data preparation: 98% of the shapes were allocated for training, and the remaining 2% for testing. The final training and testing sets were created by merging the corresponding portions from each sub-dataset. Note that we use the entire testing dataset solely for autoencoder reconstruction validation. We also apply a 90-degree rotation augmentation along each axis, doing the same for the corresponding conditions (point clouds, voxels). We also create a balanced training set across these 19 datasets by sampling 10,000 shapes from each. If a dataset contains fewer than 10,000 shapes, we duplicate the data until the target size is reached.

**Training Details.** For optimization and training, we use the Adam optimizer Kingma & Ba (2014) with a learning rate of 0.0001 and a gradient clipping value of 1. For VQ-VAE training, we use a batch size of 256 with 1024 codebook embeddings of dimension 4. We train the network until convergence and then fine-tune the VQ-VAE using a more balanced dataset until it converges again. For the base generative model, we use a batch size of 64 and train it for 2 to 4 million iterations for each modality. Each generative model is trained on a single H100 GPU per condition. We train our model on six conditions: point clouds with 2,500 points, voxels at  $16^3$  resolution, single-view RGB, multi-view RGB with 4 views, multi-view depth with 4 views, and multi-view depth with 6 views. We also fine-tune the single-view model with synthetic sketch data and single-depth data to obtain two more conditions. Additionally, we train an unconditional model beyond these. Finally,

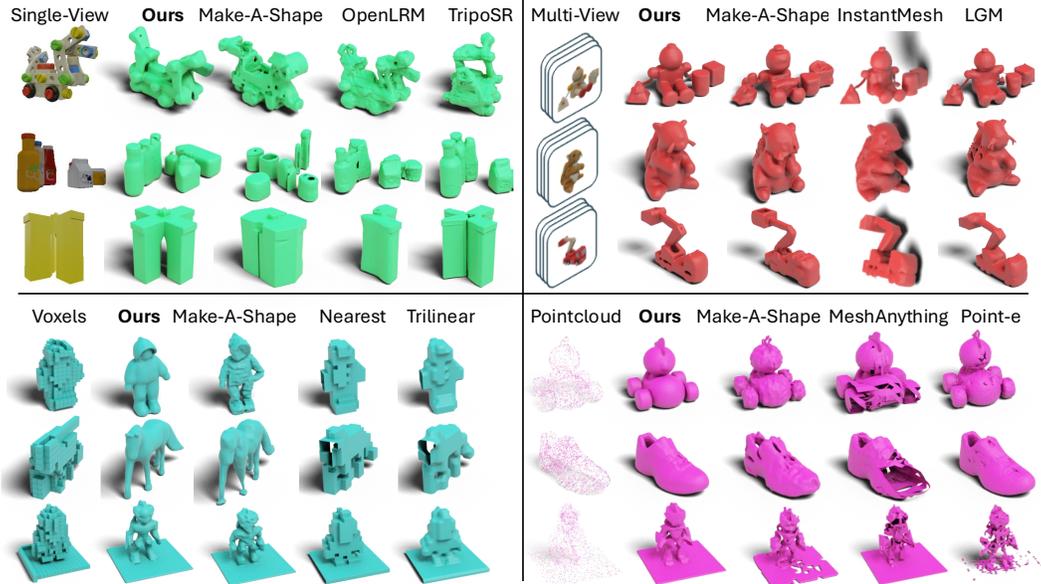


Figure 4: Qualitative comparison with other methods for single-view (top-left), multi-view (top-right), voxels (bottom-left), and point cloud (bottom-right) conditional input modalities. Hui et al. (2024); He & Wang (2024); Tochilkin et al. (2024); Xu et al. (2024); Tang et al. (2024); Chen et al. (2024b); Nichol et al. (2022c)

we train a large single-view RGB model with 1.4 billion parameters, which we call the WaLa Large model, using 8 H100 GPUs and a batch size of 256. Once this model has converged, we fine-tune it with depth data, using the same number of GPUs and batch size, to obtain the WaLa Depth Large model.

**Evaluations Dataset.** We perform qualitative and quantitative evaluation of our method on Google Scanned Objects (GSO) (Downs et al., 2022) and MAS validation data (Hui et al., 2024). Importantly, Google Scanned Objects (GSO) is not a part of the dataset detailed in Section 4.1 used to train our model. Consequently, evaluating on Google Scanned Objects (GSO) data assesses the cross-domain generalization of our method. We include all validation objects from the GSO dataset to ensure a broad evaluation. The MAS validation data is an unseen test set consisting of 50 randomly selected shapes from each of the 19 large-scale datasets mentioned in Section 4.1. This ensures that validation data contains all the subcategories like CAD models, human figures, faces, houses, and others, thereby enabling a comprehensive evaluation. We present three metrics for each method on both datasets, the metrics being: (i) Light Field Distance (LFD) (Chen et al., 2003) which evaluates how alike two 3D models appear when viewed from multiple angles, (ii) Intersection over Union (IoU) ratio, which compares the intersection volume to the total volume of two voxelized 3D objects, and (iii) Chamfer Distance (CD), which measures the similarity between two shapes based on the minimum distance between corresponding points on their surfaces. Note that among these three metrics, for generated shapes that are not aligned (that can occur during generation from conditions such as single images), the most reliable metric is LFD as it is rotation invariant.

## 4.2 EXPERIMENTS

We conducted a comprehensive study across various modalities, quantitatively evaluating our method against baselines using four distinct input types: point clouds (Section 4.2.1), voxels (Section 4.2.2), single-view images, and multi-view images (Section 4.2.3). For qualitative analysis, we present the results of all our models, showcasing visual outcomes in Figure 1 and Figure 2, and provide additional examples in the appendix as well as on our website: <https://autodeskaiab.github.io/WaLaProject>. We also report a detailed ablation study in the appendix.

Table 2: Quantitative comparison between different methods of point cloud to mesh generation. We present LFD, IOU and CD metrics. Our method, WaLa, outperforms the other methods on both GSO and MAS Validation datasets.

<i>Method</i>	GSO Dataset			MAS Dataset		
	LFD ↓	IoU ↑	CD ↓	LFD ↓	IoU ↑	CD ↓
Poisson surface reconstruction (Kazhdan et al., 2006)	3306.66	0.3838	0.0055	4565.56	0.2258	0.0085
Point-E SDF model (Nichol et al., 2022c)	2301.96	0.6006	0.0037	4378.51	0.4899	0.0158
MeshAnything (Chen et al., 2024b)	2228.62	0.3731	0.0064	2892.13	0.3378	0.0091
Make-A-Shape (Hui et al., 2024)	2274.92	0.7769	0.0019	1857.84	0.7595	0.0036
WaLa(Ours)	<b>1114.01</b>	<b>0.9389</b>	<b>0.0011</b>	<b>1467.55</b>	<b>0.8625</b>	<b>0.0014</b>

Table 3: Quantitative evaluation on lower resolution voxel data ( $16^3$  resolution) to mesh generation task. Our method, WaLa, surpasses traditional Nearest neighbour and Trilinear upsampling as well as data-centric method like Make-a-Shape.

<i>Method</i>	GSO Dataset			MAS Dataset		
	LFD ↓	IoU ↑	CD ↓	LFD ↓	IoU ↑	CD ↓
Nearest Neighbour Interpolation	5158.63	0.1773	0.0225	5401.12	0.1724	0.0217
Trilinear Interpolation	4666.85	0.1902	0.0361	4599.97	0.1935	0.0371
Make-A-Shape (Hui et al., 2024)	1913.69	0.7682	0.0029	2566.22	0.6631	0.0051
WaLa(Ours)	<b>1544.67</b>	<b>0.8285</b>	<b>0.0020</b>	<b>1874.41</b>	<b>0.75739</b>	<b>0.0020</b>

#### 4.2.1 POINT CLOUD-TO-MESH

In this study, we aim to evaluate the generation of a mesh from an input point cloud containing 2,500 points. We present qualitative results of this task in the bottom right of Figure 4 and in rows 1–2 of Figure 2. To quantitatively assess WaLa’s performance, we compare it against both traditional and data-driven techniques, as shown in Table 2. For the traditional approach, we benchmark against Poisson surface reconstruction, which uses heuristic methods to create smooth meshes from point clouds. For Poisson reconstruction, we need normals, so we estimate them using the five nearest neighbors via O3D (Zhou et al., 2018). After performing Poisson surface reconstruction, we remove vertices whose density values fall below the 20th percentile to avoid spurious faces. Additionally, we evaluate our method alongside data-driven generative models such as Point-E (Nichol et al., 2022b), MeshAnything (Chen et al., 2024b), and Make-A-Shape (Hui et al., 2024). For Point-E (Nichol et al., 2022b), we utilize its SDF network to estimate the distance field from the point cloud. We also compare our method with MeshAnything (Chen et al., 2024b), a recent transformer-based neural network designed for meshing point clouds. In this case, we use 2,500 input points and follow their hyperparameters and procedure. Finally, we compare against Make-A-Shape (Hui et al., 2024), which also generates meshes conditioned on point clouds and has its model open-sourced.

The quantitative results in Table 2 demonstrate that our method significantly outperforms existing point cloud to mesh generation techniques on both the GSO and MAS validation datasets. These results are despite us not needing normals as in Poisson reconstruction and MeshAnything. Our method can also scale well with data compared to methods like MeshAnything which do not scale well with large face counts. Moreover, our method does not require many surface points to reconstruct a 3D shape, whereas methods like MeshAnything require 8k points (as mentioned in their work) and Point-E requires 4k points. Qualitatively, our method also outperforms the baselines, as shown in the bottom right of Figure 4, and creates smoother shapes with complex geometry, as demonstrated in rows 1–2 of Figure 2.

#### 4.2.2 VOXEL-TO-MESH

In this experiment, we evaluate our proposed method, WaLa, against several baseline approaches for generating 3D shapes from low-resolution voxels with a resolution of  $16^3$ . Quantitative results are presented in Table 3, while qualitative comparisons are illustrated in the bottom left of Figure 4 and in rows 3 and 4 of Figure 2. We evaluate using the GSO and MAS datasets. As detailed in Table 3, WaLa is benchmarked against traditional upsampling techniques (nearest neighbor and trilinear interpolation) and a data-driven approach, Make-A-Shape (Hui et al., 2024). For the traditional upsampling baselines, we apply nearest neighbor and trilinear interpolation methods to the

Table 4: Comparison between different methods on Image-to-3D task (Top) and Multiview-to-3D task (Bottom). Quantitative evaluation shows that our single-view model excels the baselines, achieving the highest IoU and lowest LFD metrics. Our multi-view model further enhances performance by incorporating additional information. RGB 4, Depth 4, and Depth 6 represents conditioning using RGB images from 4 different views, and depth estimates from 4 and 6 views respectively. Inference time is measured on A100 GPU.

	Method	Inference Time↓	GSO Dataset			MAS Val Dataset		
			LFD ↓	IoU ↑	CD ↓	LFD ↓	IoU ↑	CD ↓
Single-view	Point-E (Nichol et al., 2022a)	~31 Sec	5018.73	0.1948	0.02231	6181.97	0.2154	0.03536
	Shap-E (Jun & Nichol, 2023a)	~6 Sec	3824.48	0.3488	0.01905	4858.92	0.2656	0.02480
	One-2-3-45 (Liu et al., 2023a)	~45 Sec	4397.18	0.4159	0.04422	5094.11	0.2900	0.04036
	OpenLRM (He & Wang, 2024)	~5 Sec	3198.28	0.5748	0.01303	4348.20	0.4091	0.01668
	TripoSR(Tochilkin et al., 2024)	~1 Sec	3750.65	0.4524	0.01388	4551.29	0.3521	0.03339
	InstantMesh(Xu et al., 2024)	~10 Sec	3833.20	0.4587	0.03275	5339.98	0.2809	0.05730
	LGM(Tang et al., 2024)	~37 Sec	4391.68	0.3488	0.05483	5701.92	0.2368	0.07276
	Make-A-Shape(Hui et al., 2024)	~2 Sec	3406.61	0.5004	0.01748	4071.33	0.4285	0.01851
	WaLa (RGB)	~2.5 Sec	2509.20	0.6154	0.02150	2920.74	0.6056	0.01530
	WaLa Large (RGB)	~2.6 Sec	2473.35	0.5984	0.02175	2562.70	0.6610	<b>0.00575</b>
	WaLa (depth)	~2.5 Sec	2172.52	0.6927	<b>0.01301</b>	2544.56	0.6358	0.01213
	WaLa Large (depth)	~2.6 Sec	<b>2076.50</b>	<b>0.7043</b>	0.01344	<b>2322.75</b>	<b>0.6758</b>	0.00756
Multi-view	InstantMesh(Xu et al., 2024)	~1.5 Sec	3009.19	0.5579	0.01560	4001.09	0.4074	0.02855
	LGM(Tang et al., 2024)	~35 Sec	1772.98	0.6842	0.00783	2712.30	0.5418	0.00867
	Make-A-Shape(Hui et al., 2024)	~2 Sec	1890.85	0.7460	0.00337	2217.25	0.6707	0.00350
	WaLa(RGB 4)	~2.5 Sec	1260.64	0.8500	0.00182	1540.22	0.8175	0.00208
	WaLa(Depth 4)	~2.5 Sec	1185.39	0.87884	0.00164	1417.40	0.83313	0.00160
	WaLa(Depth 6)	~4 Sec	<b>1122.61</b>	<b>0.91245</b>	<b>0.00125</b>	<b>1358.82</b>	<b>0.85986</b>	<b>0.00129</b>

$16^3$  voxel grids, followed by the marching cubes algorithm (Lorenson & Cline, 1998) to generate the corresponding meshes. In contrast, the data-driven method utilizes the pre-trained voxel-to-mesh model provided by Make-A-Shape.

The results in Table 3 demonstrate that WaLa consistently outperforms all baseline methods across various metrics and datasets. Notably, our approach achieves significantly lower LFD and CD values, alongside higher IoU scores, compared to both traditional and data-driven techniques. These quantitative findings suggest that WaLa not only effectively upsamples 3D shapes to higher resolutions but also produces smoother surfaces and higher-quality meshes by accurately filling in missing details. This holds true even for ambiguous shapes (see Figure 2, third row, columns 3–4) and those with disjoint components (see Figure 2, fourth row, columns 3–4). Furthermore, qualitative assessments in Figure 4 corroborate our quantitative results, demonstrating that WaLa reconstructs finer geometric features and more precise details than both traditional interpolation methods and existing data-driven approaches.

#### 4.2.3 IMAGE-TO-MESH

In this section, we compare WaLa with other state-of-the-art image-to-3D generative models, focusing on both single-view and multi-view scenarios. In the single-view setting, our model generates 3D shapes from a single input image or depth map. For multi-view generation, we utilize four RGB images or four to six depth images along with their corresponding camera parameters. This approach allows us to evaluate the model’s performance under varying conditions, demonstrating the versatility and effectiveness of our generative model in different image-to-3D generation contexts. Qualitative results for single-view RGB are shown in the top left of Figure 4 and in rows 5–6 of Figure 2. Conversely, qualitative results for multi-view RGB are displayed in the top right of Figure 4 and in rows 7–8 of Figure 2. Additional details and results can be found in the appendix and on our website. Our quantitative results, which assess both quality and inference time on the GSO and MAS validation datasets, are presented in Table 4, with the Image-to-3D task results at the top and the multiview-to-3D task at the bottom. We attempted to perform an extensive comparison; however, this proved challenging as many methods are not available as open-source implementations or utilize subsets of the GSO dataset for which the sample lists are not publicly available (Zhang et al., 2024; Siddiqui et al., 2024; Bensadoun et al., 2024). Consequently, we chose to use the entire GSO dataset and run open-source models whose code is available for both GSO and MAS.

As demonstrated in Table 4, our method consistently outperforms other 3D generation techniques across both tasks. For the single image-to-3D task, our base RGB model surpasses all baseline

methods by a wide margin on most metrics, except for OpenLRM on the CD metric within the GSO dataset. We believe this exception is primarily due to CD’s sensitivity to rotation, as many generated shapes may not be perfectly aligned with the ground truth. In contrast, the LFD, which is a rotation-invariant metric, clearly shows significant improvement from WaLa as compared to LRM. Another noteworthy observation is that the WaLa Large model significantly outperforms the WaLa Base model on the MAS test set but does not show a notable improvement on the GSO dataset. This outcome is expected since we trained on the MAS training set, indicating that increasing the number of parameters may not necessarily enhance generalization to the GSO dataset. Additionally, the single-depth model outperforms the RGB base model, which is intuitive as depth maps provide more comprehensive information about the 3D structure. Finally, it is important to highlight that our model either outperforms or is comparable to most methods in terms of inference time while delivering significantly better quality. A similar trend is observed in the multi-view image-to-3D task, where our model significantly outperforms baseline methods in quality while maintaining similar or better inference times. Another interesting observation is that increasing the number of depth map images improves performance, which again intuitively makes sense as we have more shape information.

Qualitatively, our method generates 3D shapes with complex geometry (see Figure 2, row 8, column 7-8), multiple disjoint components (see Figure 2, row 6, column 3-4), and intrinsic geometric features (see Figure 2, row 5, column 3-4). Additionally, it produces diverse shapes across various object categories, including organic forms (see Figure 2, row 5, column 5-6) and CAD models (see Figure 2, row 7 column 1-2). Furthermore, our multi-view model outperforms the single-view model, which is intuitively expected due to the additional information provided by multiple perspectives. Our method also visually surpasses other baselines, as demonstrated in Figure 4, by capturing more details and creating more complex geometries. We also apply our multi-view approach to text-to-3D generation, as shown in Figure 2, rows 11–12. For these experiments, we utilize the six-view depth model, which achieves the best reconstruction performance. Additionally, we present visual sketch-to-3D results in Figure 2, rows 9–10. The sketch-to-3D models are obtained by fine-tuning the image-to-mesh model with synthetic sketch data. We also fine-tune the image-to-mesh model with single-view depth data and present visual results in Figure 2, rows 13–14. Further details about these models are provided in the appendix.

## 5 CONCLUSION

In this work, we introduce Wavelet Latent Diffusion (WaLa), a novel approach to 3D generation that tackles the challenges of high-dimensional data representation and computational efficiency. Our method compresses 3D shapes into a wavelet-based latent space, enabling highly efficient compression while preserving intricate details. WaLa marks a significant leap forward in 3D shape generation, with our billion-parameter model capable of generating high-quality shapes in just 2–4 seconds, outperforming current state-of-the-art methods. Its versatility allows it to handle diverse input modalities, including single and multi-view images, voxels, point clouds, depth maps, sketches, and text descriptions, making it adaptable to a wide range of 3D modeling tasks. We believe WaLa sets a new benchmark in 3D generative modeling by combining efficiency, speed, and flexibility. Finally, we release our code and model across multiple modalities to promote further research and support reproducibility within the community.

## 6 ACKNOWLEDGEMENT

We are deeply grateful to Shyam Sudhakaran and Martynas Pocius for their generous code release, which provided a valuable foundation for this work. We also wish to thank Hilmar Koch for insightful discussions that guided our approach. Special thanks to Justin Matejka and Kendra Wannamaker for their invaluable support in developing the UI interface and advancing our understanding of the model through hands-on experimentation. Dan Ahren’s significant contributions to PR were instrumental, and we are especially thankful for his efforts. Finally, we extend our sincere appreciation to Anthony Ruto, Tonya Custis, Daron Green, and Mike Haley for their steadfast support and encouragement throughout this project.

## REFERENCES

Gulcin Baykal, Melih Kandemir, and Gozde Unal. Edvae: Mitigating codebook collapse with evidential discrete variational autoencoders. *Pattern Recognition*, 156:110792, 2024.

- Raphael Bensadoun, Tom Monnier, Yanir Kleiman, Filippos Kokkinos, Yawar Siddiqui, Mahendra Kariya, Omri Harosh, Roman Shapovalov, Benjamin Graham, Emilien Garreau, et al. Meta 3d gen. *arXiv preprint arXiv:2407.02599*, 2024.
- 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, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022.
- Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
- Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022.
- Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *European Conference on Computer Vision (ECCV)*, 2024a.
- Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. In *Computer graphics forum*, volume 22, pp. 223–232. Wiley Online Library, 2003.
- Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023a.
- Qimin Chen, Zhiqin Chen, Hang Zhou, and Hao Zhang. Shaddr: Real-time example-based geometry and texture generation via 3d shape detailization and differentiable rendering. *arXiv preprint arXiv:2306.04889*, 2023b.
- Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. *arXiv preprint arXiv:2408.02555*, 2024b.
- Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024c.
- 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*, pp. 5939–5948, 2019.
- Zhiqin Chen, Vladimir G Kim, Matthew Fisher, Noam Aigerman, Hao Zhang, and Siddhartha Chaudhuri. Decor-gan: 3d shape detailization by conditional refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15740–15749, 2021.
- An-Chieh Cheng, Xueting Li, Sifei Liu, Min Sun, and Ming-Hsuan Yang. Autoregressive 3d shape generation via canonical mapping. In *European Conference on Computer Vision*, pp. 89–104. Springer, 2022.
- Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sd-fusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4456–4465, 2023.
- 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*, pp. 2262–2272, 2023.
- 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*, pp. 21126–21136, 2022.

- 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*, pp. 13142–13153, 2023.
- Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchun Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20637–20647, 2023.
- Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021.
- 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)*, pp. 2553–2560. IEEE, 2022.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.
- Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129: 3313–3337, 2021.
- Chenjian Gao, Qian Yu, Lu Sheng, Yi-Zhe Song, and Dong Xu. Sketchsampler: Sketch-based 3d reconstruction via view-dependent depth sampling. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pp. 464–479. Springer, 2022a.
- Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022b.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- Benoit Guillard, Edoardo Remelli, Pierre Yvernav, and Pascal Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13023–13032, 2021.
- Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (ToG)*, 38(4):1–12, 2019.
- Zexin He and Tengfei Wang. OpenLRM: Open-source large reconstruction models. <https://github.com/3DTopia/OpenLRM>, 2024.
- Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- 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.

- Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. *arXiv preprint arXiv:2301.11093*, 2023.
- Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3D shape generation. In *ACM SIGGRAPH Asia*, pp. 1–9, 2022.
- Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. Make-a-shape: a ten-million-scale 3d shape model. In *Forty-first International Conference on Machine Learning*, 2024.
- Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 867–876, 2022.
- Pradeep Kumar Jayaraman, Aditya Sanghi, Joseph G Lambourne, Karl DD Willis, Thomas Davies, Hooman Shayani, and Nigel Morris. Uv-net: Learning from boundary representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11703–11712, 2021.
- Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3D implicit functions. *arXiv preprint arXiv:2305.02463*, 2023a.
- Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023b.
- Yash Kant, Ziyi Wu, Michael Vasilkovsky, Guocheng Qian, Jian Ren, Riza Alp Guler, Bernard Ghanem, Sergey Tulyakov, Igor Gilitschenski, and Aliaksandr Siarohin. Spad : Spatially aware multiview diffusers, 2024.
- Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson Surface Reconstruction. In Alla Sheffer and Konrad Polthier (eds.), *Symposium on Geometry Processing*. The Eurographics Association, 2006. ISBN 3-905673-24-X. doi: /10.2312/SGP/SGP06/061-070.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Roman Klokov, Edmond Boyer, and Jakob Verbeek. Discrete point flow networks for efficient point cloud generation. In *European Conference on Computer Vision*, pp. 694–710. Springer, 2020.
- Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9601–9611, 2019.
- Di Kong, Qiang Wang, and Yonggang Qi. A diffusion-refinement model for sketch-to-point modeling. In *Proceedings of the Asian Conference on Computer Vision*, pp. 1522–1538, 2022.
- Joseph G Lambourne, Karl DD Willis, Pradeep Kumar Jayaraman, Aditya Sanghi, Peter Meltzer, and Hooman Shayani. Brepnet: A topological message passing system for solid models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12773–12782, 2021.
- Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosior, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.
- 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*, 2023a.
- 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*, pp. 12642–12651, 2023b.

- Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12706–12716, 2021.
- Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023a.
- 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. *Advances in Neural Information Processing Systems*, 36, 2024.
- 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, 2023b.
- 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*, pp. 9298–9309, 2023c.
- Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Fine-grained 3d shape classification with hierarchical part-view attention. *IEEE Transactions on Image Processing*, 30:1744–1758, 2021.
- Zhengzhe Liu, Peng Dai, Ruihui Li, Xiaojuan Qi, and Chi-Wing Fu. Iss: Image as setting stone for text-guided 3d shape generation. *arXiv preprint arXiv:2209.04145*, 2022.
- Zhengzhe Liu, Jingyu Hu, Ka-Hei Hui, Xiaojuan Qi, Daniel Cohen-Or, and Chi-Wing Fu. Exim: A hybrid explicit-implicit representation for text-guided 3d shape generation. *ACM Transactions on Graphics (TOG)*, 42(6):1–12, 2023d.
- Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.
- William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. ACM, 1998.
- Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, pp. 67–77. IEEE, 2017.
- 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*, pp. 2837–2845, 2021.
- Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 922–928. IEEE, 2015.
- Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8446–8455, 2023.
- 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*, pp. 4460–4470, 2019.
- 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*, pp. 13492–13502, 2022.

- Aryan Mikaeili, Or Perel, Mehdi Safae, Daniel Cohen-Or, and Ali Mahdavi-Amiri. Sked: Sketch-guided text-based 3d editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14607–14619, 2023.
- 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*, pp. 306–315, 2022.
- Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. StructureNet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019.
- George Kiyohiro Nakayama, Mikaela Angelina Uy, Jiahui Huang, Shi-Min Hu, Ke Li, and Leonidas Guibas. Diffacto: Controllable part-based 3d point cloud generation with cross diffusion. In *ICCV*, 2023.
- Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pp. 7220–7229. PMLR, 2020.
- 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*, 2022a.
- Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts, 2022b.
- Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *CoRR*, abs/2212.08751, 2022c.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- 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*, pp. 165–174, 2019.
- William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.
- Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pp. 523–540. Springer, 2020.
- Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view CNNs for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5648–5656, 2016.
- 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*, pp. 652–660, 2017a.
- Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.
- 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.

- Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, et al. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12630–12641, 2023.
- Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pp. 725–741, 2018.
- Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.
- Pradyumna Reddy, Ismail Elezi, and Jiankang Deng. G3dr: Generative 3d reconstruction in imagenet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9655–9665, 2024.
- Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- 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*, pp. 10684–10695, 2022a.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022b.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.
- Aditya Sanghi, Hang Chu, Joseph G. Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. CLIP-Forge: Towards zero-shot text-to-shape generation. In *CVPR*, pp. 18603–18613, 2022.
- Aditya Sanghi, Rao Fu, Vivian Liu, Karl DD Willis, Hooman Shayani, Amir H Khasahmadi, Srinath Sridhar, and Daniel Ritchie. Clip-sculptor: Zero-shot generation of high-fidelity and diverse shapes from natural language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18339–18348, 2023a.
- Aditya Sanghi, Pradeep Kumar Jayaraman, Arianna Rampini, Joseph Lambourne, Hooman Shayani, Evan Atherton, and Saeid Asgari Taghanaki. Sketch-a-shape: Zero-shot sketch-to-3d shape generation. *arXiv preprint arXiv:2307.03869*, 2023b.
- Pratheba Selvaraju, Mohamed Nabail, Marios Loizou, Maria Maslioukova, Melinos Averkiou, Andreas Andreou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. Buildingnet: Learning to label 3d buildings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10397–10407, 2021.
- Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.
- J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *CVPR*, 2023a.
- J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3D neural field generation using triplane diffusion. In *CVPR*, pp. 20875–20886, 2023b.
- Yawar Siddiqui, Tom Monnier, Filippos Kokkinos, Mahendra Kariya, Yanir Kleiman, Emilien Garreau, Oran Gafni, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, et al. Meta 3d assetgen: Text-to-mesh generation with high-quality geometry, texture, and pbr materials. *arXiv preprint arXiv:2407.02445*, 2024.

- Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1798–1808, 2021.
- Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 945–953, 2015.
- 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*, pp. 61–70, 2020.
- 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.
- 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.
- Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.
- Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022.
- Kashi Venkatesh Vishwanath, Diwaker Gupta, Amin Vahdat, and Ken Yocum. Modelnet: Towards a datacenter emulation environment. In *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*, pp. 81–82. IEEE, 2009.
- Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5): 1–12, 2019.
- Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023.
- Karl DD Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. *ACM Transactions on Graphics (TOG)*, 40(4): 1–24, 2021.
- 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.
- Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018.
- Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.
- Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.
- Bojun Xiong, Si-Tong Wei, Xin-Yang Zheng, Yan-Pei Cao, Zhouhui Lian, and Peng-Shuai Wang. Octofusion: Octree-based diffusion models for 3d shape generation. *arXiv preprint arXiv:2408.14732*, 2024.

- Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360 views. *arXiv e-prints*, pp. arXiv-2211, 2022.
- Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20908–20918, 2023a.
- 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.
- Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023b.
- Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6239–6249, 2022.
- Xingguang Yan, Han-Hung Lee, Ziyu Wan, and Angel X Chang. An object is worth 64x64 pixels: Generating 3d object via image diffusion. *arXiv preprint arXiv:2408.03178*, 2024a.
- Xingguang Yan, Han-Hung Lee, Ziyu Wan, and Angel X. Chang. An object is worth 64x64 pixels: Generating 3d object via image diffusion, 2024b. URL <https://arxiv.org/abs/2408.03178>.
- 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*, pp. 4541–4550, 2019.
- Lior Yariv, Omri Puny, Oran Gafni, and Yaron Lipman. Mosaic-sdf for 3d generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4630–4639, 2024.
- Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022.
- Biao Zhang, Matthias Nießner, and Peter Wonka. 3dilg: Irregular latent grids for 3d generative modeling. *Advances in Neural Information Processing Systems*, 35:21871–21885, 2022.
- Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *arXiv preprint arXiv:2301.11445*, 2023a.
- Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *TOGSIG*, 42(4), 2023b.
- Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets, 2024.
- Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional sdf diffusion for controllable 3d shape generation. *arXiv preprint arXiv:2305.04461*, 2023.
- Junsheng Zhou, Weiqi Zhang, Baorui Ma, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Udiff: Generating conditional unsigned distance fields with optimal wavelet diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21496–21506, 2024.

- 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*, pp. 5826–5835, 2021.
- Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.
- Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016.
- Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

## A ADDITIONAL RESULTS AND DETAILS

For more visual results and detailed information about our model, please visit <https://autodeskailab.github.io/WaLaProject>. The code is available at <https://github.com/AutodeskAILab/WaLa>.

## B ARCHITECTURE DETAILS

In the first stage, we train a convolution-based VQ-VAE using a codebook size of 1024 with a dimension of 4. We downsample the input wavelet tree representation to a  $12^3 \times 4$  latent space. Our generative model operates within this latent space by utilizing the U-ViT architecture Hooeboom et al. (2023), incorporating two notable modifications. Firstly, we do not perform any additional downsampling since our latent space is already quite small. Instead, the model comprises multiple ResNet blocks followed by attention blocks, and then more ResNet blocks at the end, with a skip connection from the initial ResNet block. The attention blocks include both self-attention and cross-attention mechanisms, as described in (Chen et al., 2023a). Secondly, we modulate the layer normalization parameters in both the ResNet and attention layers, following the approach detailed in (Esser et al., 2024). This tailored architecture enables our generative model to effectively operate within the compact latent space, enhancing both performance and efficiency.

In this section, we describe the details of the various conditions utilized in our model:

1. **Point Cloud Model:** During training, we randomly select 2,500 points from the pre-computed point cloud dataset, which was generated from our large-scale dataset comprising 10 million shapes. These points are encoded into feature vectors using the PointNet encoder Qi et al. (2017a). To aggregate these feature vectors into condition latent vectors, we apply attention pooling as described in Lee et al. (2019). This process converts the individual points into a latent set vector. Finally, we pass this latent set vector through additional Multi-Layer Perceptron (MLP) layers to obtain the final condition latent vectors.
2. **Voxel  $16^3$  Model:** For voxel-based conditions, we employ a ResNet-based convolutional encoder to process the  $16^3$  voxel grid. After applying multiple ResNet layers, the voxel volume is downsampled to reduce its dimensionality to  $8^3$ . This downsampled volume is then processed with additional ResNet layers, ultimately resulting in the conditional latent vectors. This approach leverages the spatial hierarchy captured by the ResNet architecture to effectively encode volumetric data.
3. **Single View Image Model:** Our dataset consists of a predetermined set of views for each object. During training, we randomly select one view from this set. The selected view is then processed by the DINO v2 encoder (Oquab et al., 2023) to extract feature representations. The encoder’s output serves as the conditional latent vectors, encapsulating the visual information from the single view. It is important to note that we do not train the DINO v2 encoder; instead, we freeze its weights and utilize only the conditional latent vectors.
4. **Single View Depth Model:** We begin by selecting a checkpoint from a pre-trained Single View Image Model once it has converged and initialize the depth conditioned generative model using the same architecture described in the single-view section. We then fine-tune the model using pre-computed depth data. Throughout this process, we utilize the DINO v2 encoder (Oquab et al., 2023) to obtain the conditional latent vectors while keeping the encoder’s weights frozen.
5. **Sketch Model:** We initialize the model using the architecture described in the single-view section. After the base model converges, we fine-tune it with sketch data. This fine-tuning process involves training the model on sketch representations to adapt the latent vectors, enabling them to capture the abstract and simplified features characteristic of sketches. As in previous cases, the DINO v2 encoder (Oquab et al., 2023) remains frozen. Further details about the sketch data are provided in Appendix D.
6. **Multi-View Image/Depth Model:** For multi-view scenarios, we select four viewpoints for the multi-view RGB image model and use configurations with four and six views for the multi-view depth model. These views are carefully chosen from pre-defined angles to ensure comprehensive coverage of the object. Each view is independently processed through

the DINO v2 encoder (Oquab et al., 2023), generating a latent vector for each viewpoint. The latent vectors from all views are then concatenated sequentially, forming a final conditional latent representation structured as a sequence of latent vectors with dimensions corresponding to the number of views and the condition vector size. This approach effectively integrates information from multiple perspectives. It’s also important to note that we keep the DINO v2 encoder frozen in this setup.

7. **Text to 3D Model:** In this case, we use the six-view multi-view depth model for 3D generation and the MVDream model for six-view generation from text. The MVDream model is fine-tuned using six-view depth maps, and details are provided in Appendix E.
8. **Unconditional Model:** For the unconditional model, we use the base U-ViT architecture without any conditioning. We only use time to modulate the normalization parameters of the network. Additionally, we do not apply classifier-free guidance.

## C ABLATION STUDIES

### C.1 VQ-VAE ADAPTIVE SAMPLING LOSS ANALYSIS

In this section, we evaluate the importance of adaptive sampling loss by training two autoencoder models for up to 200,000 iterations: one incorporating the adaptive sampling loss and one without it. The results are presented in the first two rows of Table 5 . We use Intersection over Union (IoU) and Mean Squared Error (MSE) to measure the average reconstruction quality across all data points. Additionally, we introduce D-IoU and D-MSE metrics, which assess the average reconstruction performance by weighting each dataset equally. This approach ensures that any data imbalance is appropriately addressed during evaluation.

As shown in the table, even after approximately 200,000 iterations, the model utilizing adaptive sampling loss significantly outperforms the one without it. Specifically, the adaptive sampling loss leads to higher IoU and lower MSE values, indicating more accurate and reliable reconstructions. These results clearly demonstrate the substantial benefits of using adaptive sampling loss in enhancing the performance and robustness of autoencoder models.

### C.2 VQ-VAE ANALYSIS AND FINETUNING ANALYSIS

In this section, we examine the benefits of performing *balanced fine-tuning*, as described in the main section of the paper. We conduct an ablation study to determine the optimal amount of finetuning data required per dataset to achieve the best results. The results are presented in the rows following the first two in Table 6 , utilizing the metrics described above.

Our observations indicate that even a small amount of fine-tuning data improves the IoU and MSE. Specifically, incorporating as few as 2,500 samples per dataset leads to noticeable enhancements in reconstruction accuracy. However, we found that increasing the finetuning data to 10,000 samples per dataset provides optimal performance. At this level, both IOU and Mean Squared Error (MSE) metrics reach their best values, demonstrating the effectiveness of *balanced fine-tuning* in enhancing model performance.

Moreover, the D-IoU and D-MSE metrics confirm that using 10,000 samples per dataset effectively mitigates data imbalance to a certain degree. Based on these findings, all subsequent results in this study are based on using 10,000 finetuning samples per dataset. We believe that an interesting area for future work is to improve data curation to further enhance reconstruction accuracy.

### C.3 ARCHITECTURE ANALYSIS OF GENERATIVE MODEL

In this section, we conduct an extensive study on the architectural design choices of the generative model. Given the high computational cost of training large-scale generative models, we implement early stopping after 400,000 iterations. The results are presented in Table 6 . First, we examine the importance of the hidden dimension in the attention layer. It is clearly observed that increasing the dimension enhances performance. A similar trend is noted when additional layers of attention

---

<sup>1</sup>Results for the first two rows are based on 200k iterations.

Table 5: Ablation study on adaptive sampling as well finetuning of the VQ-VAE model.

Sampling Loss	Amount of finetune data	IOU $\uparrow$	MSE $\downarrow$	D-IOU $\uparrow$	D-MSE $\downarrow$
No <sup>1</sup>	-	0.91597	0.00270	0.91597	0.00270
Yes <sup>1</sup>	-	<b>0.92619</b>	<b>0.00136</b>	<b>0.91754</b>	<b>0.00229</b>
Yes	-	0.95479	0.00090	0.94093	0.00169
Yes	2500	0.95966	0.00078	0.94808	0.00149
Yes	5000	0.95873	0.00078	0.94793	0.00149
Yes	10000	<b>0.95979</b>	<b>0.00078</b>	<b>0.94820</b>	<b>0.00148</b>
Yes	20000	0.95707	0.00079	0.94659	0.00150

Table 6: Ablation study on the generative model design choices.

Architecture	hidden dim	No. of layers	post or pre	LFD $\downarrow$	IoU $\uparrow$	CD $\downarrow$
U-ViT	384	32	pre	1523.74	0.8211	0.001544
U-ViT	768	32	pre	1618.73	0.7966	0.001540
U-ViT	1152	8	pre	1596.88	0.8020	0.001561
U-ViT	1152	16	pre	1521.81	<b>0.8237</b>	0.001573
U-ViT	1152	32	pre	<b>1507.43</b>	0.8199	<b>0.001482</b>
DiT	1152	32	pre	1527.16	0.8145	0.001602
U-ViT	1152	32	post	1576.07	0.8176	0.001695

blocks are incorporated. Although the improvement is not pronounced, it is important to mention that these observations are based on only 400,000 iterations. Finally, we compare the DiT (Peebles & Xie, 2023) architecture to the U-ViT architecture (Hooeboom et al., 2023) and find that U-ViT outperforms DiT. This comparison highlights the superior performance of the U-ViT architecture in our generative model framework.

#### C.4 PRE-QUANT VS POST-QUANT

In this section, we compare whether it is better to apply the generative model to the grid before or after quantization. We conduct this comparison over 400,000 iterations. The results are shown in Table 6. These results indicate that pre-quantization performs better.

## D SKETCH DATA GENERATION



Figure 5: The 6 different sketch types. From left to right: Grease Pencil, Canny, HED, HED+potrace, HED+scribble, CLIPaasso, and a depth map for reference. Mesh taken from (Fu et al., 2021).

We generate sketches using 6 different techniques. In the first technique, we use Blender to perform non-photorealistic rendering of the meshes using a Grease Pencil Line Art modifier. The modifier is configured to use a line thickness of 2 with a crease threshold of 140°. Since disconnected faces can cause spurious lines using this method, we automatically merge vertices by distance using a threshold of 1e-6 before rendering. The second technique takes previously generated depth maps and produces sketches using Canny edge detection. We apply the Canny edge filter built into `imagemagick` using a value of 1 for both the blur radius and sigma and a value of 5% for both the low and high threshold. We then clean the output by running it through the `potrace` program with the flags `--turdsize=10` and `--opttolerance=1`. The third technique uses HED (Xie & Tu, 2015) in its default configuration, also on depth maps. The fourth technique applies `potrace` on top of default HED, and the fifth applies HED’s “scribble” filter instead. The sixth and final

technique uses CLIPasso (Vinker et al., 2022) on previously rendered color images. We configure CLIPasso to use 16 paths, a width of 0.875, and up to 2,000 iterations, with early stopping if the difference in loss is less than  $1e-5$ .

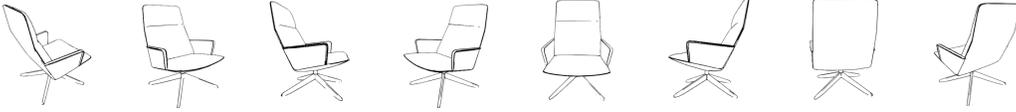


Figure 6: The 8 different views for which sketches were generated. Images created using the Grease Pencil technique on a mesh taken from Fu et al. (2021). The CLIPasso technique was only used on the first, fifth, and sixth views from the left.

For the first 5 techniques, sketches are generated from a total of 8 views: the 4 views used for multi-view to 3D, plus views from the front, right side, left side, and back. For CLIPasso we only generate sketches from the front, right side, and left back. Additionally, we only generate sketches from a subset of the 10 million shapes which we constructed by taking up to 10,000 shapes from each of the 20 datasets.

During training we augment the sketches by adding random translation, rotation, and scale in order to improve the model’s over-sensitivity to line thickness and padding. We also add random positional noise to the shapes in the SVG drawings produced by CLIPasso and `potrace`. Finally, we add a non-affine cage transformation by dividing the image into 9 squares of equal size. We treat the four corners of the central square as control points and move each one independently, warping the image.

## E TEXT-TO-3D DETAILS

The dataset used for this part contains 330,000 objects, comprising 3D-FUTURE, House3D, Toy4K, ShapeNet-v2, and a filtered subset of Objaverse datasets (filtered by (Kant et al., 2024)). We began by generating captions for this dataset using the Internvl 2.0 model (Chen et al., 2024c). For each object, we provided the model with four renderings and created two versions of captions by applying two distinct prompts. These initial captions were then augmented using LLaMA 3.1 (Dubey et al., 2024) to enhance their diversity and richness.

Next, we fine-tuned the Stable Diffusion model, initializing it with weights from MVDream (Shi et al., 2023). Utilizing the depth map-text paired data we had collected, we generated six depth maps for each object. To ensure consistency, we identified a uniform cropping box around each object across all depth maps and applied this cropping uniformly to all 6 images. Following the MVDream methodology, we resized the cropped images to  $256 \times 256$  pixels and employed `bfloat16` precision for processing.

During the inference phase, we input text prompts to generate six corresponding depth maps. These depth maps were then used to condition our multi-view depth model, which successfully generated the 3D shape of each object.

## F MODEL SIZES

Table 7 lists the number of parameters for each of our models.

## G SCALE AND TIMESTEPS FOR DIFFERENT MODELS

Table 8 lists the classifier-free guidance scales and timesteps used in this paper. These parameters were determined through an extensive grid search on the MAS dataset’s validation set. We find that, for most conditions, fewer than 10 timesteps are sufficient, except for the unconditional model. This finding aligns with the results from Make-A-Shape (Hui et al., 2024), indicating that if the conditioning information is substantial, the diffusion model requires very few timesteps to generate

Table 7: Number of Parameters for Different Models

Method	Number of Parameters
Autoencoder Model	12.9 million
Uncondition Model	1.1 billion
Single View Model	956 million
Single View Model Large	1.4 billion
Depth View Model	956 million
Depth View Model Large	1.4 billion
Pointcloud Model	966.7 million
Multi View Model (Depth and Image)	956 million
6 view Depth Model	898 million
Voxel Model	906.9 million

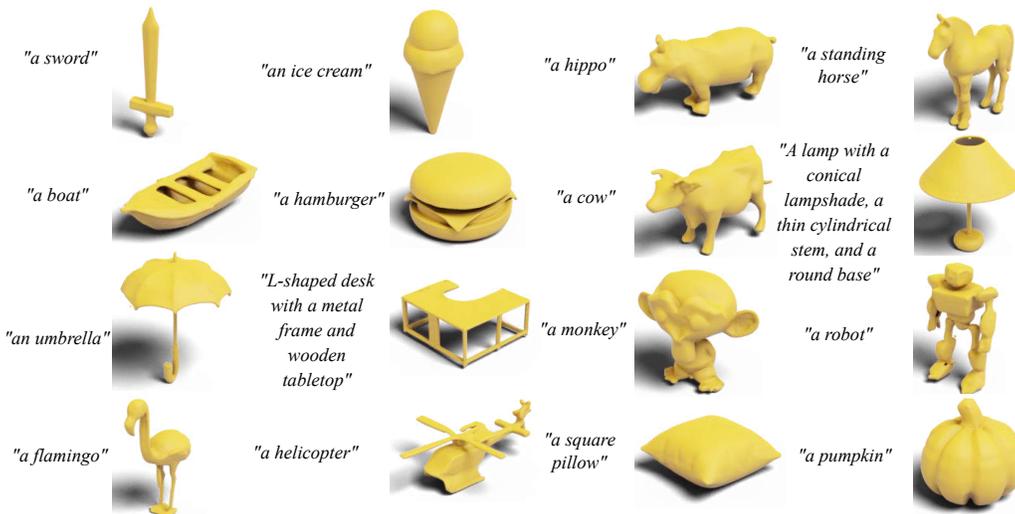


Figure 7: This figure presents more results from the text-to-3D generation task. Each row corresponds to a unique text prompt, with the resulting 3D renderings highlighting the model’s capability to produce detailed and varied shapes from these inputs.

the 3D shape. This is particularly evident in the unconditional setting, where, lacking any shape information hint, the best results are obtained using 1000 timesteps.

Table 8: Classifier free scale and timestep used in the paper

Model	Scale	Timestep
Voxel	1.5	5
Pointcloud	1.3	8
Single-View RGB	1.8	5
Single-View Depth	1.8	5
Multi-View RGB	1.3	5
Multi-View Depth	1.3	5
6 Multi-View Depth	1.5	10
Unconditional	-	1000

## H MORE VISUAL RESULTS

In Figure 7, we present additional text-to-3D generation results, showcasing the diversity and quality of outputs produced by our model. Each result highlights the model’s ability to capture various

object details and structures based solely on text prompts. In Figure 8 and Figure 9, we illustrate the variety in generation for each caption. For each given caption, we display four different generated outputs, demonstrating the model’s capacity to create diverse yet semantically consistent results based on the same input description. These figures collectively emphasize the robustness and versatility of our approach in generating 3D content from textual inputs.

## I CONTRIBUTIONS

Aditya Sanghi: I am the lead author of this paper and contributed significantly to its development. I was responsible for formulating the main research idea, overseeing the dataset generation, and conducting experiments across various modalities. In addition to leading the research team, I coordinated the integration of different sections, ensured cohesion among contributors, and provided guidance throughout the writing process. I also contributed to the drafting and editing of the manuscript and played a key role in creating the figures and visualizations included in the paper.

Aliasghar Khani: In this paper, I contributed by using vision-language models (VLMs) and large language models (LLMs) to generate captions for over 10 million 3D objects based on their four renderings. I also trained the text-to-multi-view depth generation model utilizing a subset of our dataset. Additionally, I helped create key figures, specifically figures 2, 7, 8, and 9, which visualize and support the paper’s findings.

Pradyumna Reddy: Drafting and refining the paper(Introduction, Related Work, Results), along with assistance with visualizations(Fig 4. Single-View and Multi-View), ensuring clear and concise communication. Running multiple baselines (Tab 1. Poisson Surface Reconstruction. Tab 2. Nearest Neighbour Interpolation and Trilinear Interpolation. Tab 3. Single-View, Multi-View InstantMesh and LGM) for quantitative evaluation with current state of the art models. Designing and implementing the project website to enable sharing of a large number of results for each conditioning variable. Active participation in research discussions, focusing on strategies to improve the model’s resource efficiency.

Arianna Rampini: Implemented fine-tuning for MVDream, later used in text-to-3D applications. Ran baselines (OpenLRM, TripoSR, Point-E) and computed metrics for quantitative evaluation against state-of-the-art models (Tables 2,3,4). Contributed to presenting the work through paper writing (Image to 3D), proof-reading, and figures creation (Fig 2, 4).

Derek Cheung: Researched and implemented techniques for generating synthetic sketches. Performed fine-tuning experiments using sketches; improved lack of robustness in earlier models by adding additional sketch styles and sketch augmentations. Rewrote data processing scripts to perform sketch generation. Demonstrated that fine-tuning on a balanced subset of the 10m dataset was effective for sketch-to-3d; wrote code for creating and managing balanced dataset subsets. Rewrote distributed inference and evaluation scripts.

Kamal Rahimi Malekshan: In this project, I contributed by creating the infrastructure for large-scale training and data processing, ensuring smooth workflows. I also prepared the code for release, resolving key issues related to data preprocessing and post-processing. My work focused on enabling efficient model training and large-scale data handling.

Kanika Madan: Contributed towards running of multiple baselines for comparisons with the relevant state of the art methods (Table 1: MeshAnything with different point cloud resolutions, Make-A-Shape; Table 2: Make-A-Shape). Helped with creating the figures (Fig 1, 2, 4, 7, 8 and 9), as well as helped with paper writing and proof-reading.

Hooman Shayani: Co-managed the project alongside Aditya, contributed to writing the paper, and developed Figure 3. Guided the project throughout its development, including idea generation and strategic planning. Collaborated with Derek on sketch generation and provided valuable feedback on various techniques and methodologies. Furthermore, assisted in testing and validating the models and was heavily involved in the code and model releases.

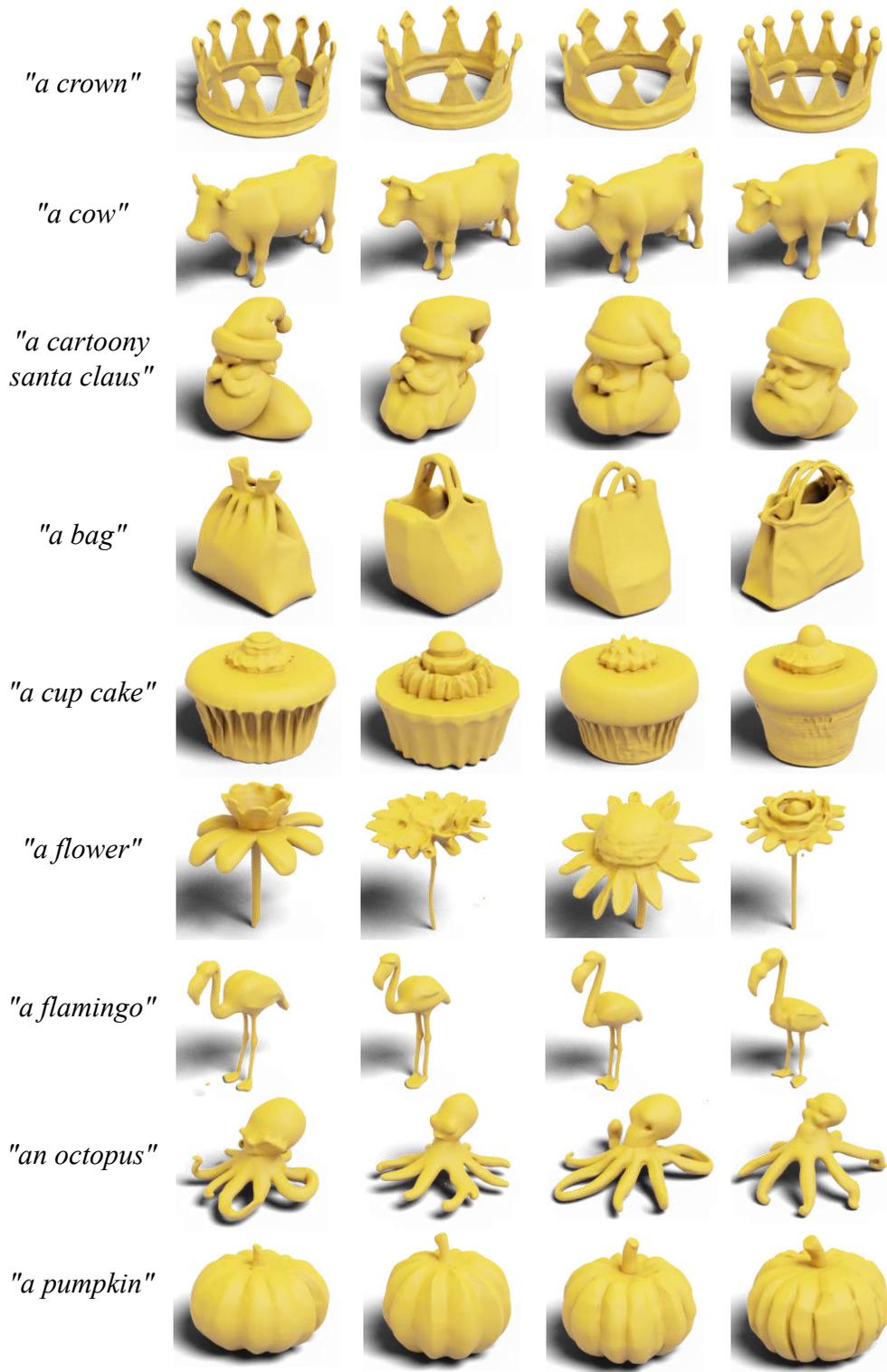


Figure 8: Here, for each caption, four different 3D variations are displayed. This figure emphasizes the model's flexibility in generating multiple distinct outputs for the same text description while maintaining thematic consistency.

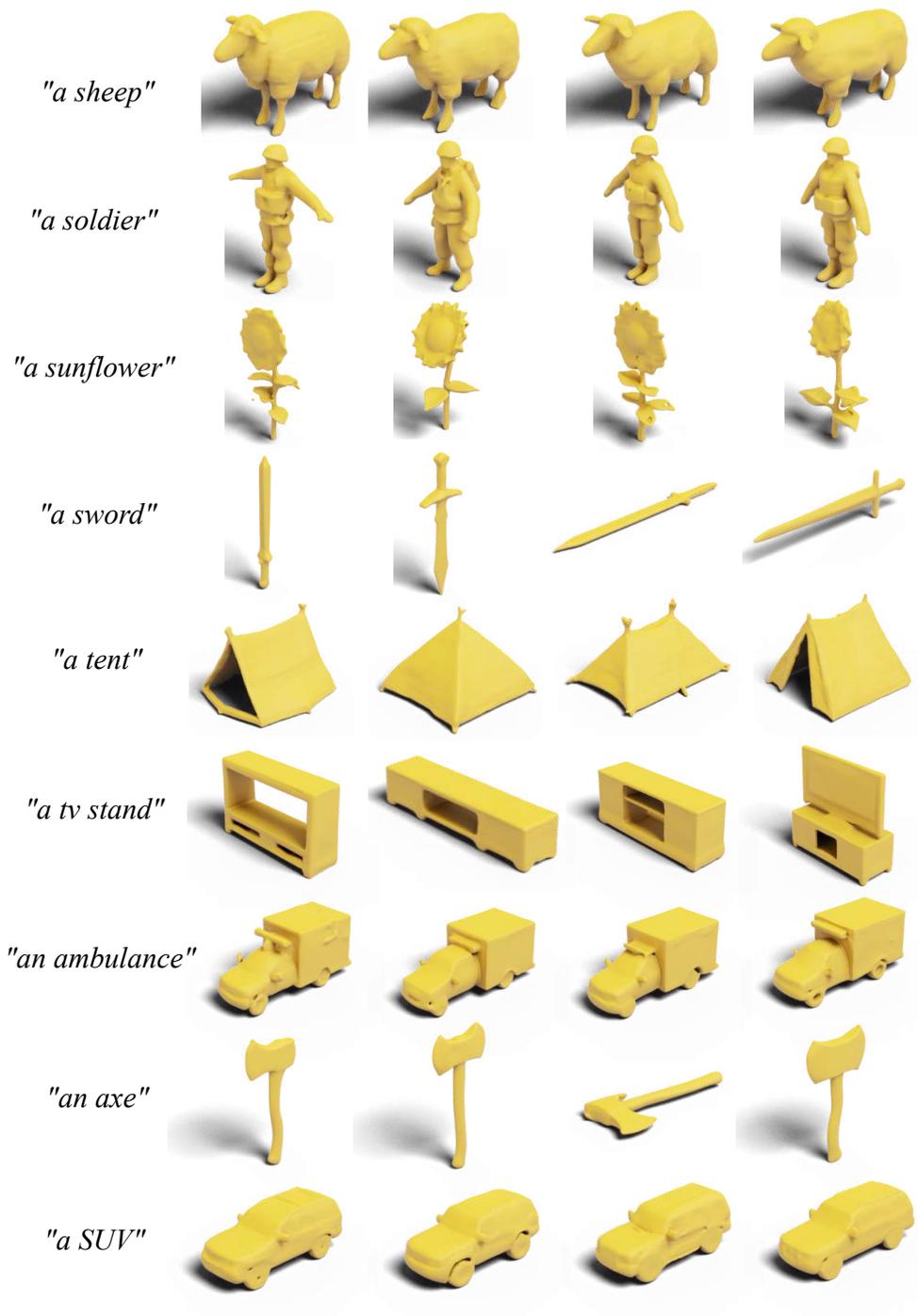


Figure 9: Here, for each caption, four different 3D variations are displayed. This figure emphasizes the model's flexibility in generating multiple distinct outputs for the same text description while maintaining thematic consistency.