Make-A-Shape: a Ten-Million-scale 3D Shape Model

Ka-Hei Hui [1 *] **Aditya Sanghi** [2 *] **Arianna Rampini** [2] **Kamal Rahimi Malekshan** [2] **Zhengzhe Liu** [1]
Hooman Shayani [2] **Chi-Wing Fu** [1]

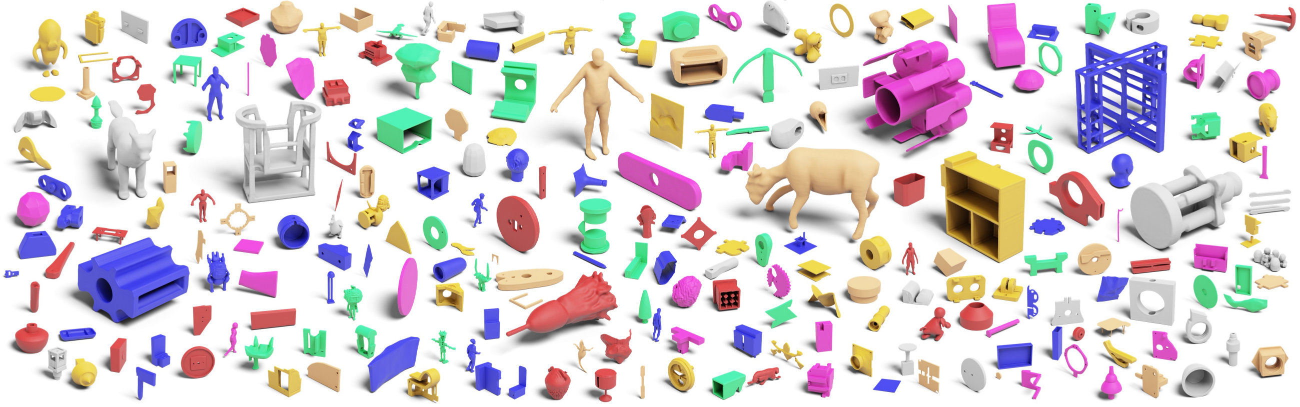

Figure 1. Make-A-Shape is a very large-scale 3D generative model trained on 10 million diverse shapes, capable of generating diverse 3D shapes with intricate geometric details, realistic structures, complex topologies, and smooth surfaces; see the generative results above.

Abstract

The progression in large-scale 3D generative models has been impeded by significant resource requirements for training and challenges like inefficient representations. This paper introduces *Make-A-Shape*, a novel 3D generative model trained on a vast scale, using 10 million publicly-available shapes. We first innovate the *wavelet-tree representation* to encode high-resolution SDF shapes with minimal loss, leveraging our newly-proposed subband coefficient filtering scheme. We then design a subband coefficient packing scheme to facilitate diffusion-based generation and a subband adaptive training strategy for effective training on the large-scale dataset. Our generative framework is versatile, capable of conditioning on various input modalities such as images, point clouds, and voxels, enabling a variety of downstream applications, *e.g.*, unconditional generation, completion, and conditional generation. Our approach clearly surpasses the existing baselines in delivering high-quality results and can efficiently generate shapes within two seconds for most conditions.

1. Introduction

Significant progress has been made in training large generative models for natural language and images (Rombach et al., 2022; Saharia et al., 2022; Ramesh et al., 2021; Yu et al., 2022). However, the advancement of 3D generative models is lagging behind. Existing models are either limited in quality, focused on small 3D datasets (Zhang et al., 2023; Shue et al., 2023; Hui et al., 2022; Mescheder et al., 2019; Gao et al., 2022b), or allowing a single condition (Nichol et al., 2022; Jun & Nichol, 2023; Liu et al., 2023a; Li et al., 2024; Hong et al., 2024; Xu et al., 2024).

Training 3D generative models introduces unique challenges. First, the extra dimension in 3D increases the number of variables, demanding more network parameters and memory in training. Particularly, U-Net-based diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song & Ermon, 2019) generate memory-intensive feature maps that exceed the processing power of GPUs, so greatly extending the training time; see (Hoogeboom et al., 2023). Second, 3D data imposes a significant input/output (IO) burden. Large model training relies on cloud services like AWS or Azure for data storage, so handling 3D data substantially increases the storage costs and prolongs the data download time for each training iteration. Third, 3D shapes are irregular and sparse, unlike 2D images. How to efficiently represent 3D shapes for large-scale training remains an open question.

Recent large generative models for 3D shapes tackle these issues by two strategies. The first employs lossy input representations to reduce the number of input variables, at the

[#] Work partially done while interning at Autodesk. [*] Equal contribution [1] The Chinese University of Hong Kong, Hong Kong SAR, China [2] Autodesk Research. Correspondence to: Chi-Wing Fu <cwfu@cse.cuhk.edu.hk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

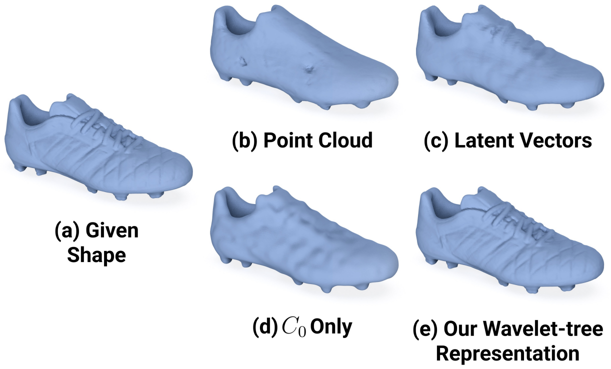

Figure 2. Reconstructing (a) a shape using (b) point clouds from Point-E (Nichol et al., 2022), (c) latent vectors from Shap-E (Jun & Nichol, 2023), (d) coarse coefficients C_0 (Hui et al., 2022), and (e) our wavelet-tree representation. See the structure and details, faithfully reconstructed in (e).

expense of compromising on detail fidelity. For instance, Point-E (Nichol et al., 2022) uses point clouds and Shap-E (Jun & Nichol, 2023) uses latent vectors; while these methods are efficient, the compromise is noticeable in the detail retention. As Table 1 clearly illustrates, reconstructions of the ground-truth signed distance function (SDF) often display a significant loss of detail. The second strategy uses multi-view images to represent the 3D geometry (Liu et al., 2023a; Li et al., 2024; Hong et al., 2024; Xu et al., 2024), where differentiable rendering is used to produce images of the generated shape for comparing with the ground truth in the training loss. However, relying on differentiable rendering can be slow and may not capture full geometry in one training example due to occlusions in the rendering.

This work aims for *efficient training* and *compact 3D representation* to enable the creation of a large model trained on ten million 3D shapes. First, we design a new 3D representation, the *wavelet-tree representation*, to compactly encode 3D shapes, while considering *both* coarse and detail coefficients. Beyond (Hui et al., 2022), which considers detail coefficients simply by network predictions in the generative process, we design a family of techniques to enable large model training that effectively and efficiently capture the high-frequency shape details: (i) *subband coefficients filtering* selectively retains coefficients rich in information, ensuring a compact yet comprehensive representation of shapes; (ii) *subband coefficients packing* reorganizes these coefficients into a compact grid format, suitable for use with diffusion models; and (iii) *subband adaptive training strategy* maintains an effective equilibrium of both coarse and detail coefficients during the training process. Lastly, we develop a range of conditioning mechanisms to accommodate various input modalities, including point clouds, voxels, and images. Hence, our new representation, while being compact, can faithfully retain most shape information and, at the same time, enables effective training of a large generative model. We named our method *Make-A-Shape*.

Table 1. Comparing 3D representations on GSO dataset (Downs et al., 2022). "IoU": Intersection Over Union; "Num. of Input Var.": number of input variables in a representation; "Add. Net.": necessity of training multiple networks to obtain SDF; and "Proc. Time": time to output the representation from a given SDF.

Representation	IoU	Num. of Input Var.	Add. Net.	Time
Ground-truth SDF (256^3)	1.0	16777216 ($\sim$ 64MB)	No	–
Point Cloud (Nichol et al., 2022)	0.8642	12288 ($\sim$ 0.05MB)	Yes	$\sim$1s
Latent Vectors (Jun & Nichol, 2023)	0.8576	1048576 ($\sim$ 4MB)	Yes	$\sim$5mins
Coarse Component (Hui et al., 2022)	0.9531	97336 ($\sim$ 0.4MB)	No	$\sim$1s
Wavelet tree (ours)	0.9956	1129528 ($\sim$ 4.3MB)	No	$\sim$1 second

Table 2. Efficiency comparison with state-of-the-art methods. For single-view, we offer inference times for both 10 and 100 iterations (iter.), with the latter determined as the quality-optimized hyperparameter through our ablation study. For multi-view, 10 iterations are chosen as the optimal option. Regarding training time, as different methods use varying numbers of GPUs, we compare their training speed by the number of training shapes they can process in a day divided by the number of GPUs used. Note that DMV3D (Xu et al., 2024), Instant3D (Li et al., 2024), and LRM (Hong et al., 2024) are concurrent works. Their codes are not publicly available. So, their numbers are derived from their original papers. While these numbers merely approximate efficiency and could be influenced by multiple factors, they do provide insight into the volume of learning steps undertaken, which is necessary for good scalability. Further, it is important to mention that we do not have access to the training time data for Point-E and Shap-E.

Method	Inference time	# Training shapes in 1 day / GPU
Point-E (Nichol et al., 2022)	$\sim$ 31 sec	–
Shape-E (Jun & Nichol, 2023)	$\sim$ 6 sec	–
One-2-3-45 (Liu et al., 2023a)	$\sim$ 45 sec	$\sim$ 50k (A10G)
DMV3D (Xu et al., 2024)	$\sim$ 30 sec	$\sim$ 110k (A100)
Instant3D (Li et al., 2024)	$\sim$ 20 sec	$\sim$ 98k (A100)
LRM (Hong et al., 2024)	$\sim$ 5 sec	$\sim$ 74k (A100)
Ours (single-view 10 iter.)	$\sim$ 2 sec	$\sim$ 290k (A10G)
Ours (single-view 100 iter.)	$\sim$ 8 sec	
Ours (multi-view 10 iter.)	$\sim$ 2 sec	$\sim$ 250k (A10G)

Make-A-Shape has several clear advantages over prior works. (i) Our representation is notably *expressive*, capable of encoding shapes with minimal loss, *e.g.*, a 256^3 grid can be bijectively encoded in one second, yet with an IoU of 99.56%. (ii) Our representation is *compact*, characterized by a low number of input variables, almost akin to lossy representations like latent vectors (Jun & Nichol, 2023), yet not necessitating additional autoencoder training, while having higher quality (Table 1). (iii) Our representation is *efficient*, enabling fast streaming and training, *e.g.*, streaming and loading a sophisticatedly compressed 256^3 SDF grid takes 266 milliseconds, while our representation takes only 184 milliseconds for the same process. On average, we can process approximately 2x to 6x more training shapes in one day than prior methods, despite using a less powerful GPU (A10G vs. A100), as detailed in Table 2. (iv) *Make-A-Shape* is versatile. It can be conditioned on different modalities such as single/multi-view images, point clouds, and low-resolution voxels, enabling various downstream applications. (v) *Make-A-Shape* enables fast generation, taking just *few seconds* to generate high-quality shapes (Table 2); see Figure 1 for a wide range of shapes generated by our approach.

2. Related Work

Neural Shape Representations. Recently, using deep learning for 3D representations has gained significant interest. Explicit representations, like point clouds (Qi et al., 2017a;b; Wang et al., 2019), meshes (Hanocka et al., 2019; Masci et al., 2015; Verma et al., 2018; Nash et al., 2020), and boundary representation (Jayaraman et al., 2021; Lambourne et al., 2021; Wu et al., 2021; Jayaraman et al., 2022) have been widely adopted for both discriminative and generative applications. Neural implicit representations, like signed distance functions (SDFs) and occupancy fields, have also gained popularity, notably explored in works, *e.g.*, (Park et al., 2019; Mescheder et al., 2019; Chen & Zhang, 2019). Some very recent works (Hui et al., 2022; Liu et al.) explored wavelets to decompose SDF signals into multi-scale coefficients. Despite enhanced quality, high-frequency details are mostly ignored at the expense of shape fidelity. In this work, we introduce a new wavelet-tree representation, encoding shapes compactly yet nearly losslessly.

3D Diffusion Models. With recent advances in diffusion models for high-quality image generation, there is a growing interest in adopting diffusion models to 3D contexts. Existing approaches mostly train a Vector-Quantized-VAE (VQ-VAE) on a 3D representation like triplane (Shue et al., 2023; Chou et al., 2023; Peng et al., 2020), implicit forms (Li et al., 2023; Cheng et al., 2023), or point clouds (Jun & Nichol, 2023; Zeng et al., 2022), then employ the diffusion model in the latent space. Direct training on a 3D representation has been less explored, with some recent studies focusing on point clouds (Nichol et al., 2022; Zhou et al., 2021; Luo & Hu, 2021), voxels (Zheng et al., 2023) and neural wavelet coefficients (Hui et al., 2022; Liu et al.). In our work, we design our 3D representation in a way that it can be directly trained with a diffusion model, avoiding information loss with the VQ-VAE.

Conditional 3D Models. Existing conditional 3D models fall into two categories. The first group utilizes large 2D conditional image generative models. Initially, this area focused on text-to-3D approaches (Jain et al., 2022; Michel et al., 2022; Poole et al., 2023), and later expanded to include images (Deng et al., 2023; Melas-Kyriazi et al., 2023; Xu et al., 2023a), multi-view images (Liu et al., 2023b; Deitke et al., 2023; Qian et al., 2024; Shi et al., 2024), and additional conditions like sketches (Mikaeili et al., 2023). The second group centers on training conditional generative models using data either paired with a condition or in a zero-shot manner. Paired conditional generative models consider various conditions such as point clouds (Zhang et al., 2022; 2023), images (Zhang et al., 2022; Nichol et al., 2022; Jun & Nichol, 2023; Zhang et al., 2023), low-resolution voxels (Chen et al., 2021; 2023), sketches (Lun et al., 2017; Guillard et al., 2021; Gao et al., 2022a; Kong et al., 2022), and text (Nichol et al., 2022; Jun & Nichol, 2023). Recently, zero-shot methods have gained popularity, particularly focusing on text (Sanghi et al., 2022; 2023a; Liu et al., 2023c; Xu et al., 2023b) and sketches (Sanghi et al., 2023b). In this work, we focus on training a large, paired conditional generative model to accommodate various conditions, enabling fast generation without the need for scene optimization.

3. Method

Figure 3 overviews the *Make-A-Shape* framework, which has four components, as described from Sections 3.1 to 3.4.

3.1. Wavelet-tree representation

We formulate a novel, efficient and expressive 3D representation called *wavelet-tree representation*. We first transform a 3D shape into a truncated signed distance function (TSDF) of resolution 256^3, then we utilize a wavelet transform [1] to decompose the TSDF into coarse coefficient $C_0 \in \mathcal{R}^{46^3}$ and detail coefficients $D_0 \in \mathcal{R}^{(2^d-1)\times 46^3}, D_1 \in \mathcal{R}^{(2^d-1)\times 76^3}, D_2 \in \mathcal{R}^{(2^d-1)\times 136^3}$, where d denotes the dimension. In 3D, each D_i is a set of $2^3 - 1 = 7$ volumes, which are denoted as *subband volumes*. In Figure 3, we illustrate in 2D where $d = 2$ and each D_i has $2^2 - 1 = 3$ subband volumes. The coarse coefficient C_0 encodes the low-frequency components and represents the overall 3D topology, whereas the detail coefficients encompass high-frequency information. Importantly, this representation is lossless, allowing for bijective conversion back to a TSDF via inverse wavelet transforms.

Wavelet coefficient tree. Expanding on (Hui et al., 2022), we propose exploiting the relationships between wavelet coefficients. As illustrated in "wavelet tree" in Figure 3, each coarse coefficient in C_0 (the parent) is hierarchically connected to its associated detail coefficients in D_0 (the children). This parent-child relationship iterates through subsequent levels (such as from D_0 to D_1), creating a wavelet coefficient tree with coefficients from C_0 as the roots and those sharing the same parent as siblings.

Despite the losslessness, the multi-scale wavelet coefficients contain multiple high-resolution coefficient volumes, say

[1] Following (Hui et al., 2022), we employ biorthogonal wavelets. Additional details are in Appendix A.

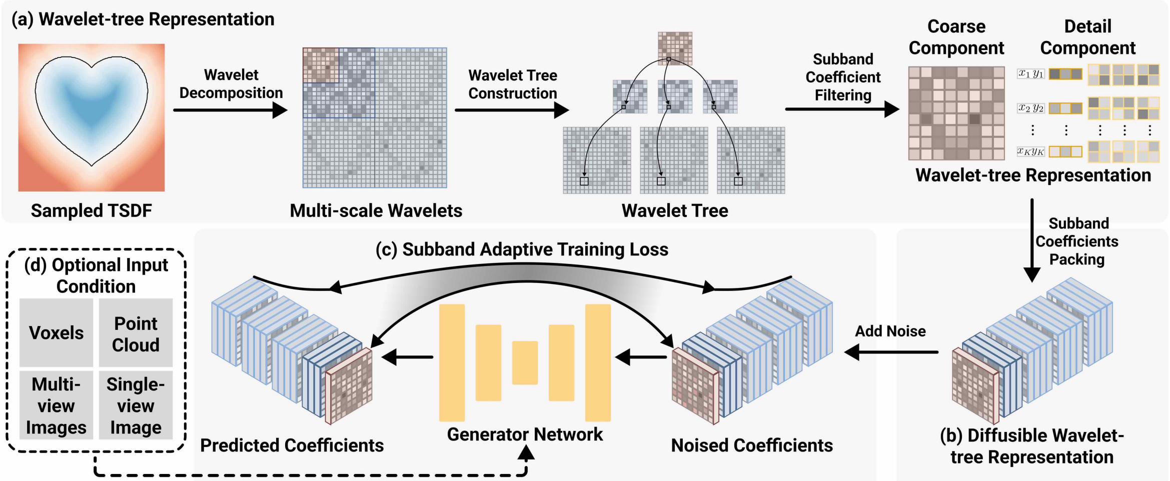
(a) Wavelet-tree Representation
Wavelet Decomposition
Wavelet Tree Construction
Subband Coefficient Filtering
Coarse Component
Detail Component
Sampled TSDF
Multi-scale Wavelets
Wavelet Tree
Wavelet-tree Representation
Subband Coefficients Packing
(d) Optional Input Condition
Voxels
Point Cloud
Multi-view Images
Single-view Image
(c) Subband Adaptive Training Loss
Add Noise
Predicted Coefficients
Generator Network
Noised Coefficients
(b) Diffusible Wavelet-tree Representation

Figure 3. Overview of our *Make-A-Shape* framework. (a) A shape, encoded in a truncated signed distance field (TSDF), is first decomposed into multi-scale wavelet coefficients in a wavelet-tree structure. We design the *subband coefficient filtering* procedure to exploit the relations among coefficients and extract information-rich coefficients to build our wavelet-tree representation. (b) We propose the *subband coefficient packing* scheme to rearrange our wavelet-tree representation into a regular grid structure of manageable spatial resolution, so that we can adopt a denoising diffusion model to effectively generate the representation. (c) Further, we formulate the *subband adaptive training* strategy to effectively balance the shape information in different subbands and address the detail coefficient sparsity. Hence, we can efficiently train our model on millions of 3D shapes. (d) Our framework can be extended to condition on various modalities.

D_1, D_2, which are far from compact and can be inefficient in both data loading and model training, making them less scalable for large-scale training. To make the representation more compact and to relieve the I/O and training burden, we conducted several empirical studies on the wavelet coefficients and identified four key observations (i)-(iv):

(i) The significance of each coefficient for shape reconstruction is positively correlated to its magnitude. If a coefficient's magnitude falls below a threshold (e.g., 1/32 of the largest coefficient within a subband), its children will likely have small magnitudes, contributing minimally to the shape. We empirically explored the D_0 subbands of 1,000 random shapes, confirming that over 96.1% of the coefficients satisfy this hypothesis.

(ii) Magnitudes of sibling coefficients are positively correlated. We evaluated the correlation coefficients between all pairs of sibling coefficients in 1,000 random shapes and found a positive correlation value of 0.35.

(iii) C_0 contains more of the shape information than siblings. Coefficients in C_0 are mostly non-zeros, with a mean magnitude of 2.2, while the mean magnitude of detail coefficients in D_0 is much closer to zero,

(iv) Most coefficients in D_2 are insignificant. By empirically setting them to zeros in inverse wavelet transforms, we can reconstruct the TSDFs faithfully for 1,000 random shapes with 99.64% IoU.

Subband Coefficient Filtering. Based on the above findings, we introduce a subband coefficient filtering approach as a pre-processing, building a wavelet-tree representation which is faithful and compact. To do this, all coefficients in C_0 are preserved, while those in D_2 are discarded, as observations (iii) and (iv) indicate. To further reduce the I/O burden, we selectively retain coefficients from D_0 and D_1 according to their information, inspired by (i) and (ii). Specifically, we evaluate each coordinate in D_0's subbands, selecting the one with the highest magnitude as a measure of 'information'. Then, we only retain the top-K information ($K = 16384 < 46^3$) coordinates X_0 and their associated sibling coefficients, denoted as D_0'. Given X_0, we then create another information-rich coefficient set D_1' by retaining the $(2^d - 1)2^d$ children in D_1 ($d = 2$ in the 2D illustrations and $d = 3$ in our 3D shapes) on each location, discarding remaining less significant coefficients. Please refer to Appendix A for a detailed 2D visualization to construct our wavelet-tree representation.

The above procedure results in our wavelet-tree representation, which contains four parts, *i.e.*, C_0, X_0, D_0', D_1', as illustrated in Figure 3. It does not only achieve both a significant compression (reducing to 1/15 in size) and an impressive mean IoU of 99.56%, but also leads to a 44.5% speed-up in streaming and loading, a critical factor for large-scale training. Notably, this process is a one-time preprocessing step, making it efficient for handling millions of 3D shapes and storing them in the cloud.

3.2. Diffusable Wavelet-tree Representation

The above representation, while efficient for data streaming, still presents challenges during the training of diffusion models (Ho et al., 2020) due to the mix of a regular structure (C_0) and the irregular structures (X_0, D'_0, D'_1). One straightforward approach would be to treat them directly as the diffusion target using a multi-branch network. However, this approach exhibited convergence issues, resulting in model training collapse. A second approach could be to load D'_0, D'_1 and reassign them to a zero-initialized volume with the same size as D_0, D_1 following their coordinates. The derived volumes $\hat{D}_0 \in \mathcal{R}^{(2^d-1)\times 46^3}, \hat{D}_1 \in \mathcal{R}^{(2^d-1)\times(2^d)\times 46^3}$ can then be naively arranged in a grid with a large spatial resolution as in Figure 3 "multi-scale wavelets". However, using the U-Net architecture, commonly employed in diffusion models, on this spatially large structure leads to memory-intensive feature maps, causing out-of-memory issues and inefficient GPU utilization.

Subband Coefficient Packing. To tackle these challenges, we take inspiration from observation (iii) and (Hoogeboom et al., 2023). Specifically, we reshape $\hat{D}_1$ to have $(2^d-1)(2^d)$ channels such that it can have the same spatial resolution as C_0 and $\hat{D}_0$, i.e., 46^3, and we concatenate C_0, $\hat{D}_0$, and the reshaped $\hat{D}_1$ in the channel dimension, inducing the regular volume as the diffusable wavelet-tree representation $x_0 \in \mathcal{R}^{2^{2d}\times 46^3}$, as depicted in the bottom row of Figure 3. This strategy can lead to an approximate cubic-order speedup and a significant reduction in GPU memory usage, estimated to be around 64x compared to when applied to the same network architecture. The technical detail is presented with an illustrative example in Appendix A

3.3. Subband Adaptive Training Strategy

Then, we train a diffusion model $f_\theta(x_t, t)$ to generate x_0, where x_t is the noised coefficient of x_0 at time step t. Directly applying MSE loss following DDPM (Ho et al., 2020) on x_0 may lead to the imbalanced loss weights, due to the imbalanced channel dimensions of C_0, $\hat{D}_0$, and $\hat{D}_1$. Moreover, our empirical observations indicate that even when assigning equal loss weights to these three volumes, the performance is still unsatisfactory.

To address these issues, we introduce a *subband adaptive training strategy* that prioritizes high-magnitude detail coefficients while maintaining balance with the remaining detail coefficients. Let $\hat{D}_{0,j}$ denote one of the subbands in $\hat{D}_0$, we select an information-rich coordinate set $Y_{0,j} = \{y | y \in \hat{D}_{0,j}, \hat{D}_{0,j}[y] > max(\hat{D}_{0,j})/32\}$. We then union coordinate sets of all subbands into $Y_0 = \bigcup_{j=0}^{2^d-1} Y_{0,j}$, which records the locations of important detail coefficients. For $\hat{D}_1$, we still adopt Y_0 as the most informative coordinates due to our finding (i). We thus formulate the loss:

$$L_{\text{MSE}}(C_0)+\frac{1}{2}\sum_{i\in\{0,1\}}\sum_{j=0}^{2^d-1}\Big[L_{\text{MSE}}(\hat{D}_{i,j}[Y_0])+L_{\text{MSE}}(\hat{D}_{i,j}[Z_0])\Big], \tag{1}$$

where $\hat{D}_{i,j}[Y_0]$ denotes the information-rich coefficients of $\hat{D}_{i,j}$; and Z_0 is a subset randomly sampled from the complement set $\hat{D}_0 \setminus Y_0$. We ensure that the size of Z_0 equals to $|Y_0|$ to provide supervision for network predictions at these coordinates, where this approach guarantees that less important coefficients receive an equal amount of supervision.

Efficient Loss Computation. For efficient code compilation in PyTorch, we utilize a fixed-size binary mask to represent the coordinate set. This allows us to calculate the MSE loss by masking both the generation target and network prediction, eliminating the need for irregular operations.

3.4. Conditional Generation

Our framework is versatile and can be extended beyond unconditional generation to accommodate conditional generation across various modalities. To achieve this, we adopt a different encoder for each modality that transforms a given condition into a sequence of latent vectors. Subsequently, these vectors are injected into the generator using multiple conditioning mechanisms. We also use a classifier-free guidance mechanism (Ho & Salimans, 2021), which has empirically demonstrated greater effectiveness in conditional settings. Note that our designed generator with architecture details, together with our formulated condition encoders and conditioning mechanisms, are reported in Appendix B.

It is worthwhile to note that due to space limit, please refer to Appendices A and B for the full technical details of the methodology, presented with various illustrative figures.

4. Results

4.1. Experimental Setup

Dataset. We compile a new, extensive dataset that features over 10 million 3D shapes aggregated from 18 different publicly-available sub-datasets: 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).

For the data division, we randomly split each sub-dataset into two segments: a training set, which includes 98% of the shapes, and a testing set, which contains the remaining 2%. We then assembled the ultimate training and testing datasets by merging these segmented sets from each sub-dataset.

For every shape, we produced both a Truncated Signed Distance Function (TSDF) and its corresponding wavelet-tree representation to facilitate model training. Note that this data pre-processing is one-time and highly parallelizable, allowing for the conversion of 10 million data points in less than one day using 40,000 CPUs, which is highly efficient for large-scale datasets. Also, for tasks that require conditional generation, we prepared a range of additional inputs to support these activities. Specifically, (i) Image inputs. We randomly sampled 55 pre-defined camera poses and rendered 55 images for each object according to these poses, using the scripts provided by (Jun & Nichol, 2023). (ii) Voxel inputs. We prepared two voxel grids (16^3 and 32^3) per 3D object. (iii) Point cloud input. We randomly sampled 25,000 points on the surface of each 3D shape.

Training Details. We train our shape model *Make-A-Shape* using the Adam Optimizer (Kingma & Ba, 2014) with a learning rate of $1e^{-4}$ and a batch size of 96. To stabilize the training, we employ an exponential moving average with a decay rate of 0.9999, in line with existing 2D large-scale diffusion models (Rombach et al., 2022). Our model is trained on 48 × A10G with 2M-4M iterations, depending on the input condition. Altogether, six different models were trained from scratch, each on a unique type of input: single-view images, multi-view images, voxels (16^3), voxels (32^3), point clouds, and an unconditional model that does not require any specific condition. Each model is trained over 20 days, amounting to around 23,000 GPU hours.

Evaluation Dataset. For qualitative evaluation, we utilized the testing set shapes to provide the visual results ($\sim 2\%$ of shapes). Due to computational constraints, for quantitative evaluation, we randomly selected 50 shapes from the test set of each sub-dataset. This set-aside collection is denoted as the “Our Val” dataset and will be used throughout the remainder of the paper. To further evaluate the cross-domain generalization capability of our model to new and unseen data, we also conducted evaluation on the entire Google Scanned Objects (GSO) dataset that is not part of our model’s training data.

Evaluation Metrics. To assess our model’s performance on conditional tasks, we use these three metrics: (i) Intersection over Union (IoU), a metric that quantifies the volumetric overlap by calculating the ratio of the intersection to the union of voxelized volumes; (ii) Light Field Distance (LFD) (Chen et al., 2003), a metric that evaluates the resemblance between two sets of images rendered from diverse viewpoints; and (iii) Chamfer Distance (CD), a metric that quantifies the bidirectional distances between two sets of points, which are sampled from the generated shape and the corresponding ground-truth shape. For the unconditional task, we ablate the generative quality (Table 5) using the Frechet Inception Distance (FID) (Heusel et al., 2017), following (Zhang et al., 2023), to compare the rendered images of the generated shapes against a reference set to evaluate the geometric plausibility of these shapes.

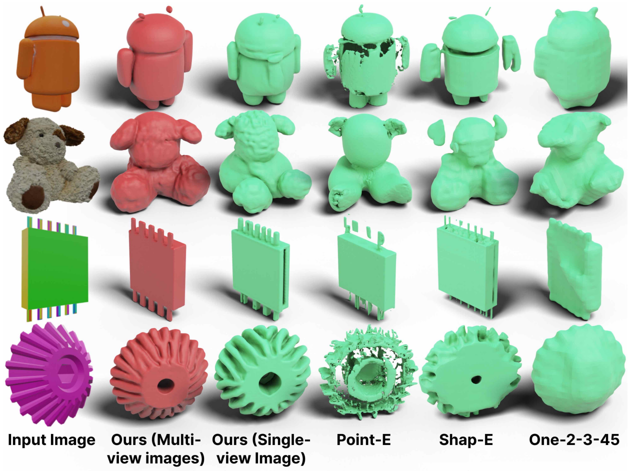

Figure 4. Visual comparisons for the Image-to-3D generation task clearly reveal that our model largely outperforms three major generative models; see the high-frequency geometric details (row 2) and patterns (row 4) captured by our model. More results are illustrated in Figure 5.

4.2. Quantitative Comparison with Existing Large Generative Models

In this experiment, we compare our method against existing large image-to-3D generative models (Nichol et al., 2022; Jun & Nichol, 2023; Liu et al., 2023b) conditioned on single-view image and multi-view images. Also, we compare with OpenLRM (He & Wang, 2024), an open-sourced implementation of a concurrent work, LRM (Hong et al., 2024).

The results in Table 3 reveal that our single-view model surpasses all the existing baselines (Point-E (Nichol et al., 2022), Shap-E (Jun & Nichol, 2023), and One-2-3-45 (Liu et al., 2023a)) by a significant margin for all three metrics (IoU, LFD, and CD). Note that LFD is a rotation-insensitive metric, indicating that the effectiveness of our approach does not depend on how the generated shapes are aligned with the ground-truth shapes. For the concurrent work OpenLRM (He & Wang, 2024), our model demonstrates similar or better performance for different metrics, despite that it has only one tenth of the model parameters (25M vs 260M, see Table 4), highlighting its high efficiency and

Table 3. Quantitative evaluation of the Image-to-3D task shows that our single-view model surpasses all the existing baselines (Point-E, Shap-E, and One-2-3-45), achieving the highest IoU, lowest LFD, and lowest CD. At the same time, our framework achieves a comparable performance with OpenLRM (He & Wang, 2024), an open-source implementation of the concurrent work, LRM (Hong et al., 2024). Also, our multi-view model further enhances the performance, when more views are available. See also Table 4 for the model size comparison.

Method	GSO Dataset			Our Val Dataset		
	LFD ↓	IoU ↑	CD ↓	LFD ↓	IoU ↑	CD ↓
Point-E (Nichol et al., 2022)	5018.73	0.1948	0.02231	6181.97	0.2154	0.03536
Shap-E (Jun & Nichol, 2023)	3824.48	0.3488	0.01905	4858.92	0.2656	0.02480
One-2-3-45 (Liu et al., 2023a)	4397.18	0.4159	0.04422	5094.11	0.2900	0.04036
OpenLRM (He & Wang, 2024)	**3198.28**	**0.5748**	**0.01303**	4348.20	0.4091	**0.01499**
Ours (single-view)	3406.61	0.5004	0.01748	**4071.33**	**0.4285**	0.01851
Ours (multi-view)	1890.85	0.7460	0.00337	2217.25	0.6707	0.00350

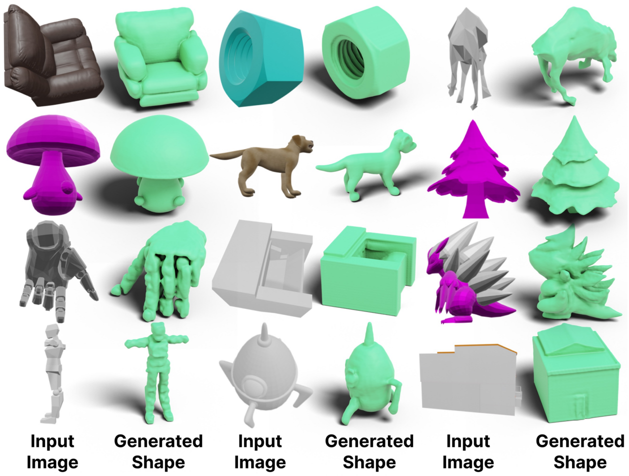

Figure 5. Our single-view conditioned generation model is able to yield a wide variety of shapes with higher fidelity than others (Figure 4). It adeptly generates both CAD objects (*e.g.*, screws and chairs) and organic forms (*e.g.*, humans, animals, and plants).

effectiveness. Further, Figure 4 illustrates that our approach effectively captures both the global structures and the fine local details, along with more complex geometric patterns, outperforming the baseline methods by a large margin.

Upon incorporating three additional views, our multi-view model demonstrates notable enhancements in performance, as evidenced in Table 3 and Figure 4. The improvement in performance can be attributed to the augmented informational input derived from multiple views. However, it is crucial to acknowledge that, even with the inclusion of four images, the visual information accessible remains somewhat restricted for achieving a fully detailed and comprehensive reconstruction of a shape.

Both the qualitative and quantitative comparisons demonstrate the effectiveness of our wavelet-tree representation and adaptive training scheme. Additionally, we provide a comparison of the parameter counts of different image-to-3D generative models. As illustrated in Table 4, our proposed framework demands significantly less parameters than most of the compared baselines, while achieving a comparable or superior performance. Overall, these comparisons highlight the effectiveness and the generation capability of Make-A-Shape over the existing works.

Table 4. Comparison on the number of model parameters for different methods. Note that "M" stands for million and the parameter count of DMV3D (Xu et al., 2024) is not available.

Methods	# of parameters
Point-E (Nichol et al., 2022)	∼40M
Shap-E (Jun & Nichol, 2023)	∼300M
One-2-3-45 (Liu et al., 2023a)	∼0.5M
Instant3D (Li et al., 2024)	∼500M
LRM (Hong et al., 2024)	∼500M
OpenLRM (He & Wang, 2024)	∼260M
Ours (single-view/multi-view)	∼25M

4.3. Conditional generation

Image-to-3D. The qualitative results for Image-to-3D Generation tasks are showcased in Figure 5, demonstrating our method's capability to generate objects across a broad spectrum of categories. . In addition, Figure 6 shows that our method can generate diverse objects in the multi-view setting. The produced results exhibit a noticeable alignment with the images, which is more pronounced compared to the single-view approach.

Point-cloud-to-3D. In this experiment, we aim to process a point cloud input and generate a Truncated Signed Distance Function (TSDF) that faithfully follows its geometry. To do this, we utilize PointNet (Qi et al., 2017a) combined with a Permutation Invariant Set Attention (PMA) block (Lee et al., 2019) as our encoder, which takes 25,000 points during the training phase and can accommodate an arbitrary number of points during inference. The additional visual results in Figure 7 further underscore our method's consistent performance across a diverse range of object categories. Also, a comprehensive quantitative and qualitative analysis of the

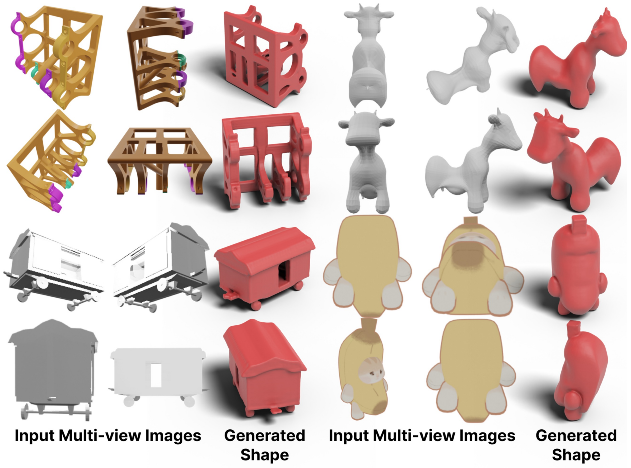

Figure 6. Our multi-view conditioned generation model can effectively utilize the available multi-view information to create diverse and coherent shapes with nontrivial topologies, exemplified by the CAD objects shown in the first two rows.

number of input points is detailed in Appendix C.

Voxel-to-3D. Next, we explore creating a high-resolution Signed Distance Function using low-resolution voxel inputs at resolutions of 16^3 and 32^3. Figure 8 showcases the outcomes achieved by our model, emphasizing its capability to generate outputs with smooth, clean surfaces and enhanced geometric precision. This capability is particularly remarkable when dealing with complex cases, such as disjoint objects and comprehensive scene-level inputs. To underline the superiority of our method, we conduct a detailed comparison against conventional techniques used for transforming low-resolution voxels into meshes, both quantitatively and qualitatively; please refer to Appendix C.

4.4. 3D Shape Completion

Further, our model can be adapted for the zero-shot shape completion task, i.e., to create a complete 3D shape conditioned on an incomplete 3D input. To do this, we adopt the approach (Lugmayr et al., 2022) with the missing region as the input mask. Figure 9 demonstrates the semantic meaningful (first and third rows), highly consistent (body region of the animal in the fifth row), and diversified shape completion capability of our approach.

4.5. Ablation Studies

In this section, we present ablation studies, with further detailed investigations available in Appendix D.

Ablation of Wavelet Tree Representation. We begin by assessing the efficacy of our wavelet tree representation. We do this by comparing it to a baseline representation that relies exclusively on the coarse coefficient C_0 as described in (Hui et al., 2022), omitting the inclusion of the detail coefficients in D_0 and D_1. The results presented in the first and last rows of Table 5 underscore the improved representational capacity of our wavelet tree representation compared to the baseline that uses only the C_0 coefficients. This demonstrates the significant advantage of integrating detail coefficients, which allows for a more effective capture of the data's intricacies and complexities.

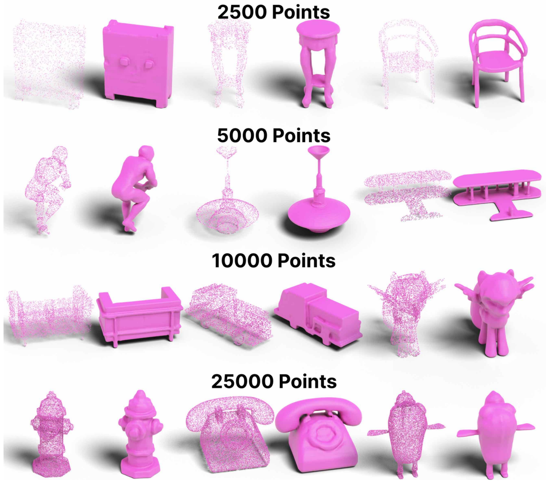

Figure 7. Point-cloud-conditioned generation. Our model is capable of producing shapes with intricate geometric details while preserving the complex topologies given in the input point clouds.

Table 5. Ablation studies conducted on the "Our Val" dataset. Our approach with the wavelet tree representation alongside an adaptive training strategy outperforms the following baselines (i) the use of only the coarse coefficient for representation, and (ii) two distinct MSE training strategies.

Settings	Metrics	
	LFD ↓	IoU ↑
Coarse component only (Hui et al., 2022)	2855.41	0.5919
MSE	3191.49	0.5474
subband-based MSE	2824.28	0.5898
Ours	**2611.60**	**0.6105**

Ablation of Adaptive Training Strategy. Further, to show the effectiveness of our subband adaptive training strategy, we conduct a comparison against two baseline training objectives. Note that we adopt our wavelet tree representation in these two baselines for a fair comparison. (i) MSE: we apply an MSE (Mean Squared Error) loss to simultaneously optimize the entire coefficients volume, which includes $\{C_0, D_0, D_1\}$, aligning with the objective used in standard diffusion models (Ho et al., 2020). (ii) Subband-based MSE, we individually compute MSE losses on C_0, D_0, and D_1 respectively, and take an average of the three terms. As indicated in Table 5, both of the two strategies using the MSE loss lead to a notable performance drop, demonstrat-

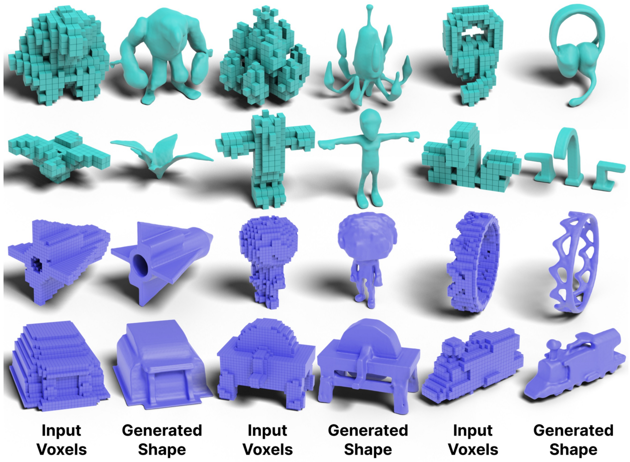

Figure 8. Our voxel-conditioned generative model excels in creating high-quality outputs from low-resolution inputs, imaginatively introducing various plausible geometric patterns. This is exemplified by the creation of holes in the crown, which are obviously not available in the initial inputs.

ing the effectiveness of our adaptive training strategy. Note that in (Hui et al., 2022), an additional detail predictor is adopted to regress (predict) the detail coefficients D_0 based on the coarse coefficients. We empirically find that this strategy does not converge well even in a subset of our dataset (all categories of ShapeNet (Chang et al., 2015)), thus not adopting it in our setting.

5. Limitations and Future Works

Our approach exhibits the following limitations:

(i) While our unconditional model is capable of generating a diverse variety of shapes from various sub-datasets, it can not ensure a balanced representation of objects across different categories during sampling. Hence, the learned 3D shape distribution is inherently imbalanced, evident in the disproportionate representation of CAD models. We can utilize large zero-shot language models like ChatGPT for annotating object categories, enabling the application of data augmentation methods to balance training data according to these categories.

(ii) Our generation network, trained on a mix of sub-datasets, does not utilize the category label as an extra condition. For this reason, our unconditional model may occasionally generate implausible shapes (mixing shapes from different sub-datasets). Identifying and mitigating these undesired shapes represents an interesting direction for future research. It is particularly intriguing to consider the development of data-driven metrics for assessing the visual plausibility of generated 3D shapes, especially in the context with multiple categories.

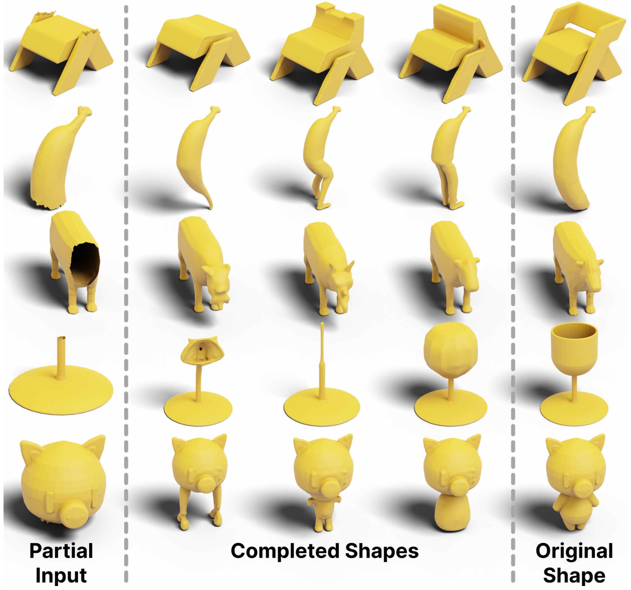

Figure 9. From partial inputs (left), our generative model completes the missing regions in a coherent and semantically meaningful way. Also, it can generate multiple variations of the completed shapes (middle), many of which are significantly different from the inputs (right), highlighting the diverse shape distribution learned.

(iii) At present, our primary focus lies in the direct generation of 3D geometry. To generate additional texture, our approach can work with existing texture generation approaches, such as (Richardson et al., 2023), to first create the geometry and then generate the textures on the geometry. Yet, an interesting direction for future exploration involves generating textures together on the geometry, with the aim of achieving this without relying on computationally-expensive optimizations.

6. Conclusion

In summary, this paper introduces Make-A-Shape, a novel 3D generative framework trained on an extensive dataset of over 10 million publicly-available 3D shapes. It can efficiently generate a wide range of high-quality 3D shapes with superior details and plausible structures in just 2 seconds for most conditions. Our comprehensive experiments showcase the model's superiority in synthesizing high-quality 3D shapes across various challenging conditions, including single/multi-view images, point clouds, and low-resolution voxels, all while demanding minimal resources during training. Notably, our model not only quantitatively outperforms existing baselines but also demonstrates zero-shot applications like partial shape completion. We acknowledge the limitations of our model and outline potential future work in Appendix 5. We believe Make-A-Shape will set the stage for future research on large-scale 3D model training.

Impact statement

While Make-A-Shape represents a significant advancement in the field of 3D modeling, enabling the creation of detailed shapes with unprecedented speed and quality, it also inherits and potentially amplifies ethical challenges, common to the existing generative models.

First, the use of a vast dataset of 10 million publicly-available shapes, while instrumental in achieving high-quality generation, introduces the risk of propagating existing biases or inadvertently, including sensitive or potentially copyrighted material. Though our dataset selection process aimed to be comprehensive and responsible, the sheer scale and diversity of the data mean that unintended biases could be encoded within the model, affecting the fairness and neutrality of the generated outputs.

Moreover, the capability of Make-A-Shape to produce detailed 3D models based on a variety of input modalities raises concerns regarding the potential misuse of the technology. In the wrong hands, such technology could be utilized to create counterfeit products or other harmful objects with relative ease. The fidelity of the generated shapes could also contribute to creating highly realistic or deceptive materials, posing challenges in areas such as copyright infringement, privacy, and security.

Finally, while our model can serve as a powerful tool for designers, architects, and artists, facilitating creativity and innovation, it also poses the risk of automating tasks traditionally performed by human professionals. However, it is also anticipated that the technology will democratize access to high-quality 3D modeling, enabling growth and improving accessibility for the creative industry.

Acknowledgement

We thank the anonymous reviewers for their valuable comments. This work is supported by the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No.: CUHK 14201921].

References

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Chen, D.-Y., Tian, X.-P., Shen, Y.-T., and Ouhyoung, M. On visual similarity based 3D model retrieval. In *Computer graphics forum*, volume 22, pp. 223–232. Wiley Online Library, 2003.

Chen, Q., Chen, Z., Zhou, H., and Zhang, H. ShaDDR: Real-time example-based geometry and texture generation via 3D shape detailization and differentiable rendering. *SIGGRAPH Asia 2023*, pp. 58: 1–11, 2023.

Chen, Z. and Zhang, H. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5939–5948, 2019.

Chen, Z., Kim, V. G., Fisher, M., Aigerman, N., Zhang, H., and Chaudhuri, S. 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.

Cheng, Y.-C., Lee, H.-Y., Tulyakov, S., Schwing, A. G., and Gui, L.-Y. SDFusion: Multimodal 3D shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4456–4465, 2023.

Chou, G., Bahat, Y., and Heide, F. Diffusion-SDF: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2262–2272, 2023.

Cohen, A. Biorthogonal wavelets. *Wavelets: A Tutorial in Theory and Applications*, 2:123–152, 1992.

Collins, J., Goel, S., Deng, K., Luthra, A., Xu, L., Gundogdu, E., Zhang, X., Vicente, T. F. Y., Dideriksen, T., Arora, H., 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.

Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., and Farhadi, A. Objaverse: A universe of annotated 3D objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023.

Deng, C., Jiang, C., Qi, C. R., Yan, X., Zhou, Y., Guibas, L., Anguelov, D., 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.

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. *Neural Information Processing Systems*, 34:8780–8794, 2021.

Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T. B., and Vanhoucke, V. 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.

Fu, H., Jia, R., Gao, L., Gong, M., Zhao, B., Maybank, S., and Tao, D. 3D-FUTURE: 3D furniture shape with texture. *International Journal of Computer Vision*, 129: 3313–3337, 2021.

Gao, C., Yu, Q., Sheng, L., Song, Y.-Z., and Xu, D. 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.

Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., and Fidler, S. Get3D: A generative model of high quality 3D textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35: 31841–31854, 2022b.

Guillard, B., Remelli, E., Yvernay, P., and Fua, P. Sketch2Mesh: Reconstructing and editing 3D shapes from sketches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13023–13032, 2021.

Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., and Cohen-Or, D. MeshCNN: a network with an edge. *ACM Transactions on Graphics (ToG)*, 38(4):1–12, 2019.

He, Z. and Wang, T. OpenLRM: Open-source large reconstruction models. https://github.com/3DTopia/OpenLRM, 2024.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in neural information processing systems*, 30:6629—-6640, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *Neural Information Processing Systems Workshop*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., and Tan, H. LRM: Large reconstruction model for single image to 3D. *International Conference on Learning Representations*, 2024.

Hoogeboom, E., Heek, J., and Salimans, T. Simple diffusion: End-to-end diffusion for high resolution images. *The International Conference on Machine Learning*, 2023.

Huang, Z., Zhou, P., Yan, S., and Lin, L. ScaleLong: Towards more stable training of diffusion model via scaling network long skip connection. *Neural information processing systems*, 2023.

Hui, K.-H., Li, R., Hu, J., and Fu, C.-W. Neural wavelet-domain diffusion for 3D shape generation. In *ACM SIGGRAPH Asia*, pp. 24: 1–9, 2022.

Jain, A., Mildenhall, B., Barron, J. T., Abbeel, P., and Poole, B. 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.

Jayaraman, P. K., Sanghi, A., Lambourne, J. G., Willis, K. D., Davies, T., Shayani, H., and Morris, N. UV-Net: Learning from boundary representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11703–11712, 2021.

Jayaraman, P. K., Lambourne, J. G., Desai, N., Willis, K. D., Sanghi, A., and Morris, N. J. SolidGen: An autoregressive model for direct B-rep synthesis. *Transactions on Machine Learning Research*, 2022.

Jun, H. and Nichol, A. Shap-E: Generating conditional 3D implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., and Panozzo, D. 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.

Kong, D., Wang, Q., and Qi, Y. A diffusion-refinement model for sketch-to-point modeling. In *Proceedings of the Asian Conference on Computer Vision*, pp. 1522–1538, 2022.

Lambourne, J. G., Willis, K. D., Jayaraman, P. K., Sanghi, A., Meltzer, P., and Shayani, H. 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.

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., and Bi, S. Instant3D: Fast text-to-3D with sparse-view generation and large reconstruction model. *International Conference on Learning Representations*, 2024.

Li, M., Duan, Y., Zhou, J., and Lu, J. Diffusion-SDF: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12642–12651, 2023.

Li, Y., Takehara, H., Taketomi, T., Zheng, B., and Nießner, M. 4DComplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12706–12716, 2021.

Liu, M., Xu, C., Jin, H., Chen, L., Xu, Z., Su, H., 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.

Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., and Vondrick, C. 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, 2023b.

Liu, X., Han, Z., Liu, Y.-S., and Zwicker, M. Fine-grained 3D shape classification with hierarchical part-view attention. *IEEE Transactions on Image Processing*, 30: 1744–1758, 2021.

Liu, Z., Hu, J., Hui, K.-H., Qi, X., Cohen-Or, D., and Fu, C.-W. EXIM: A hybrid explicit-implicit representation for text-guided 3D shape generation. *ACM Transactions on Graphics (SIGGRAPH).*

Liu, Z., Dai, P., Li, R., Qi, X., and Fu, C.-W. ISS: Image as stepping stone for text-guided 3D shape generation. *International Conference on Learning Representations*, 2023c.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16, October 2015.

Lorensen, W. E. and Cline, H. E. Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471, 2022.

Lun, Z., Gadelha, M., Kalogerakis, E., Maji, S., and Wang, R. 3D shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, pp. 67–77. IEEE, 2017.

Luo, S. and Hu, W. 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.

Masci, J., Boscaini, D., Bronstein, M., and Vandergheynst, P. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 37–45, 2015.

Melas-Kyriazi, L., Laina, I., Rupprecht, C., and Vedaldi, A. 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.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. 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.

Michel, O., Bar-On, R., Liu, R., Benaim, S., and Hanocka, R. Text2Mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13492–13502, 2022.

Mikaeili, A., Perel, O., Safaee, M., Cohen-Or, D., and Mahdavi-Amiri, A. SKED: Sketch-guided text-based 3D editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14607–14619, 2023.

Nash, C., Ganin, Y., Eslami, S. A., and Battaglia, P. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pp. 7220–7229. PMLR, 2020.

Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., and Chen, M. Point-E: A system for generating 3D point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. 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.

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. 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.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. DreamFusion: Text-to-3D using 2D diffusion. *International Conference on Learning Representations*, 2023.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. 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.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Qian, G., Mai, J., Hamdi, A., Ren, J., Siarohin, A., Li, B., Lee, H.-Y., Skorokhodov, I., Wonka, P., Tulyakov, S., et al. Magic123: One image to high-quality 3D object generation using both 2D and 3D diffusion priors. *International Conference on Learning Representations*, 2024.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Raistrick, A., Lipson, L., Ma, Z., Mei, L., Wang, M., Zuo, Y., Kayan, K., Wen, H., Han, B., Wang, Y., 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.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

Ranjan, A., Bolkart, T., Sanyal, S., and Black, M. J. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pp. 725–741, 2018.

Richardson, E., Metzer, G., Alaluf, Y., Giryes, R., and Cohen-Or, D. Texture: Text-guided texturing of 3D shapes. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.

Sanghi, A., Chu, H., Lambourne, J. G., Wang, Y., Cheng, C.-Y., Fumero, M., and Malekshan, K. R. CLIP-Forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 18603–18613, 2022.

Sanghi, A., Fu, R., Liu, V., Willis, K. D., Shayani, H., Khasahmadi, A. H., Sridhar, S., and Ritchie, D. 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.

Sanghi, A., Jayaraman, P. K., Rampini, A., Lambourne, J., Shayani, H., Atherton, E., and Taghanaki, S. A. Sketch-a-shape: Zero-shot sketch-to-3D shape generation. *arXiv preprint arXiv:2307.03869*, 2023b.

Selvaraju, P., Nabail, M., Loizou, M., Maslioukova, M., Averkiou, M., Andreou, A., Chaudhuri, S., and Kalogerakis, E. BuildingNet: Learning to label 3D buildings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10397–10407, 2021.

Shi, Y., Wang, P., Ye, J., Long, M., Li, K., and Yang, X. MVDream: Multi-view diffusion for 3D generation. *International Conference on Learning Representations*, 2024.

Shue, J. R., Chan, E. R., Po, R., Ankner, Z., Wu, J., and Wetzstein, G. 3D neural field generation using triplane diffusion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 20875–20886, 2023.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, pp. 11918—-11930, 2019.

Stojanov, S., Thai, A., and Rehg, J. M. 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.

Verma, N., Boyer, E., and Verbeek, J. FeaStNet: Feature-steered graph convolutions for 3D shape analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2598–2606, 2018.

Vishwanath, K. V., Gupta, D., Vahdat, A., and Yocum, K. ModelNet: Towards a datacenter emulation environment. In *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*, pp. 81–82. IEEE, 2009.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38 (5):1–12, 2019.

Willis, K. D., Pu, Y., Luo, J., Chu, H., Du, T., Lambourne, J. G., Solar-Lezama, A., and Matusik, W. 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.

Wu, R., Xiao, C., and Zheng, C. DeepCAD: A deep generative network for computer-aided design models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6772–6782, 2021.

Wu, Y., Wu, Y., Gkioxari, G., and Tian, Y. Building generalizable agents with a realistic and rich 3D environment. *arXiv preprint arXiv:1801.02209*, 2018.

Xu, D., Jiang, Y., Wang, P., Fan, Z., Wang, Y., and Wang, Z. NeuralLift-360: Lifting an in-the-wild 2D photo to a 3D object with 360 views. *IEEE Conference on Computer Vision and Pattern Recognition*, 2023a.

Xu, J., Wang, X., Cheng, W., Cao, Y.-P., Shan, Y., Qie, X., and Gao, S. 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, 2023b.

Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., et al. DMV3D: Denoising multi-view diffusion using 3D large reconstruction model. *International Conference on Learning Representations*, 2024.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022.

Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., and Kreis, K. LION: Latent point diffusion models for 3D shape generation. *Neural Information Processing Systems*, 2022.

Zhang, B., Nießner, M., and Wonka, P. 3DILG: Irregular latent grids for 3D generative modeling. *Advances in Neural Information Processing Systems*, 35:21871–21885, 2022.

Zhang, B., Tang, J., Niessner, M., and Wonka, P. 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023.

Zheng, X.-Y., Pan, H., Wang, P.-S., Tong, X., Liu, Y., and Shum, H.-Y. Locally attentional SDF diffusion for controllable 3D shape generation. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023.

Zhou, L., Du, Y., and Wu, J. 3D shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5826–5835, 2021.

Zhou, Q. and Jacobson, A. Thingi10k: A dataset of 10,000 3D-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

Zuffi, S., Kanazawa, A., Jacobs, D., and Black, M. J. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

A. Wavelet-tree representation

To build our representation, we transform every 3D shape in our dataset into a truncated signed distance function (TSDF) with a resolution of 256^3. Following the approach in (Hui et al., 2022), we decompose the TSDF using a discrete wavelet transform. In particular, we use biorthogonal wavelets with 6 and 8 moments (Cohen, 1992).

Initially, we compute coarse coefficients C_0 and detail coefficients $\{D_0, D_1, D_2\}$. This process, depicted in Figure 10, involves three subsequent wavelet decomposition. In general, each D_i includes $2^d - 1$ *subband volumes*, where d is the dimension of the original data. For simplicity, we present our method using 2D illustrations, yet the actual computation is performed in 3D with seven *subband volumes* (instead of three *subband images*, in the 2D case) of detail coefficients in each decomposition. We start by converting the TSDF into C_2 and its associated detail coefficients $D_2 = \{D_{2,0}, D_{2,1}, D_{2,2}\}$. Then, we decompose C_2 into C_1 and $D_1 = \{D_{1,0}, D_{1,1}, D_{1,2}\}$. Finally, C_1 is decomposed into C_0 and $D_0 = \{D_{0,0}, D_{0,1}, D_{0,2}\}$.

Up to this point, the resulting representation is lossless, in the sense that we can convert it back to a TSDF via inverse wavelet transforms without loss of information.

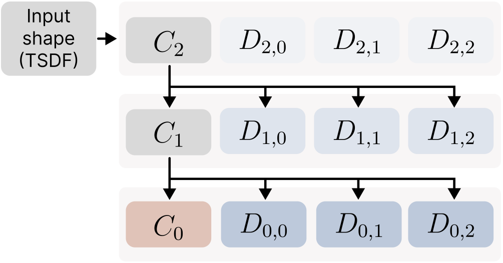

Figure 10. Wavelet decomposition of the input shape, represented as a TSDF, recursively into coarse coefficients C_i and detail coefficients $\{D_{i,0}, D_{i,1}, D_{i,2}\}$. Note that in the 3D case, there will be seven subbands of detail coefficients in each decomposition.

Figure 11 further illustrates how the relationships between wavelet coefficients are exploited. Generally, each coarse coefficient in C_0, referred to as a *parent*, and its associated detail coefficients in D_0, known as *children*, reconstruct the corresponding coefficients in C_1 through the inverse wavelet transform. This *parent-child relation* extends between D_0 and D_1, and so forth, as shown by the arrows leading from D_0 to D_1 in Figure 11. Additionally, coefficients sharing the same parent are termed *siblings*. By aggregating all descendants of a coefficient in C_0, we can construct a *wavelet coefficient tree* or simply a wavelet tree, with the coefficient in C_0 serving as its root.

In Figure 12, we show our filtering procedure. For each coefficient location, we examine the sibling coefficients in the subbands $D_{0,0}$, $D_{0,1}$, and $D_{0,2}$, selecting the one with the largest magnitude. We consider its magnitude value as the measure of *information* for that coefficient location. Next, we filter the top K coefficient locations with the highest information content, as illustrated on the left of Figure 12, and store their location coordinates (denoted as X_0) and associated coefficient values in D_0 (denoted as D'_0), along with their children's coefficient values in D_1 (denoted as D'_1). These three quantities (X_0, D'_0, D'_1) form the *detail component* in our wavelet-tree representation (Figure 12 on the right).

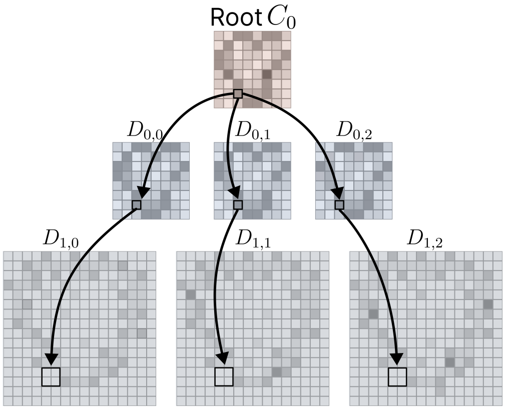

Figure 11. Overview of Parent-child relation. A wavelet tree is formed for each coefficient in C_0 as the root, with a coarser-level coefficient as parent and the finer-level coefficients as children.

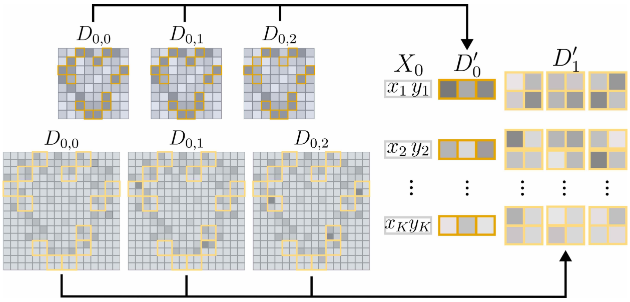

Figure 12. The detail component part of our representation. We extract and pack informative coefficients from D_0 and D_1, indicated in yellow boxes, along with their spatial locations to form our representation's detail component.

Finally, we need to effectively pack the irregular structure of the detailed component for training a diffusion model. Flattening the extracted coefficients and arranging C_0, $\hat{D}_0$, and $\hat{D}_1$ in a 2D grid naively creates a spatially large representation, as shown in Figure 13 (left). We thus rearrange sibling subband coefficients with their children in a channel-wise manner, reshaping the $\hat{D}_1$ children from $2 \times 2 \times 1$ to $1 \times 1 \times 4$ format, as shown in Figure 13.

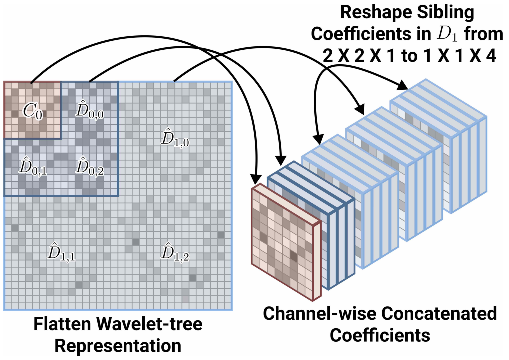
Reshape Sibling Coefficients in D_1 from 2 X 2 X 1 to 1 X 1 X 4
C_0
$\hat{D}_{0,0}$
$\hat{D}_{1,0}$
$\hat{D}_{0,1}$
$\hat{D}_{0,2}$
$\hat{D}_{1,1}$
$\hat{D}_{1,2}$
Flatten Wavelet-tree Representation
Channel-wise Concatenated Coefficients

Figure 13. Diffusible wavelet representation. First, we unpack and flatten the coefficients in our wavelet-tree representation (left). Then, we channel-wise concatenate sibling coefficients to reduce the spatial resolution (right). Here we concatenate each coefficient in C_0 with its three children in $\hat{D}_0$ and the reshaped descendants in $\hat{D}_1$ (each of size $1\times1\times4$).

B. Architecture details

Figure 14 illustrates the network architecture of our generator. The main branch, highlighted by yellow boxes, adopts a U-ViT architecture (Hoogeboom et al., 2023). The network uses multiple ResNet convolution layers for downsampling our noised coefficients into a feature bottleneck volume, as shown in the middle part of Figure 14. Following this step, we apply a series of attention layers to the volume. The volume is then upscaled using various deconvolution layers to produce the denoised coefficients. A key feature of our architecture is the inclusion of learnable skip-connections between the convolution and deconvolution blocks, which have been found to enhance stability and facilitate more effective information sharing (Huang et al., 2023).

Condition Latent Vectors. We deliberately convert all input conditions into a sequence of latent vectors, which we call *condition latent vectors*, to preserve the generality of our conditioning framework. This approach eliminates the need to devise new specific condition mechanisms to diffusion model for each modality, thereby enabling our framework to function seamlessly across various modalities. Our encoder for different modalities are described below:

1. *Single-view image.* Given a rendered image of a 3D model, we utilize the pre-trained CLIP L-14 image encoder (Radford et al., 2021) to process the image. The latent vectors extracted from just before the pooling layer of this encoder are then used as the conditional latent vectors.

2. *Multi-view images.* We are provided with four images of a 3D model, each rendered from one of 55 predefined camera poses (selected randomly). To generate the conditional latent vectors, we first use the CLIP L-14 image encoder to process each rendered image individually to produce an image latent vector. Considering the camera poses, we maintain 55 trainable camera latent vectors, each corresponding to one camera pose and matching the dimensionality of the latent vectors encoded by the CLIP image encoder. For each encoded image latent vector, we retrieve the corresponding trainable camera latent vector based on the camera pose of the image. This camera vector is then added to each image latent vector in the sequence in an element-wise fashion. Finally, the four processed sequences of latent vectors are concatenated to form the conditional latent vectors.

3. *3D point cloud.* We utilize three Multi-Layer Perceptron (MLP) layers to first transform the given point cloud into feature vectors like PointNet (Qi et al., 2017a). These vectors are then aggregated using the PMA block from the Set Transformer layer (Lee et al., 2019), resulting in a sequence of latent vectors that serve as the condition.

4. *Voxels.* We utilize two 3D convolution layers to progressively downsample the input 3D voxels into a 3D feature volume. This volume is subsequently flattened to form the desired conditional latent vectors.

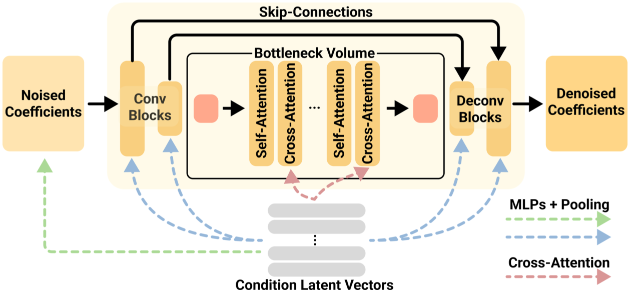

Figure 14. Our generator network progressively downsamples input coefficients to a bottleneck feature volume (middle). This volume goes through attention layers and deconvolution for upsampling to predict the denoised coefficients. If the condition latent vectors are available, we simultaneously transform these vectors and adopt them at three locations in our architecture: (i) concatenating with the input noised coefficients (the green arrow); (ii) conditioning the convolution and deconvolution blocks (the blue arrows); and (iii) cross-attention with the bottleneck volume (the red arrows).

Conditioning mechanism. When condition latent vectors are available, we integrate them into our generation network at three distinct locations in the U-ViT architecture, as depicted in the bottom part of Figure 14. Initially, these latent vectors are processed through MLP layers and a pooling layer to yield a single latent vector (highlighted by the green arrow in the left section of Figure 14). This vector is

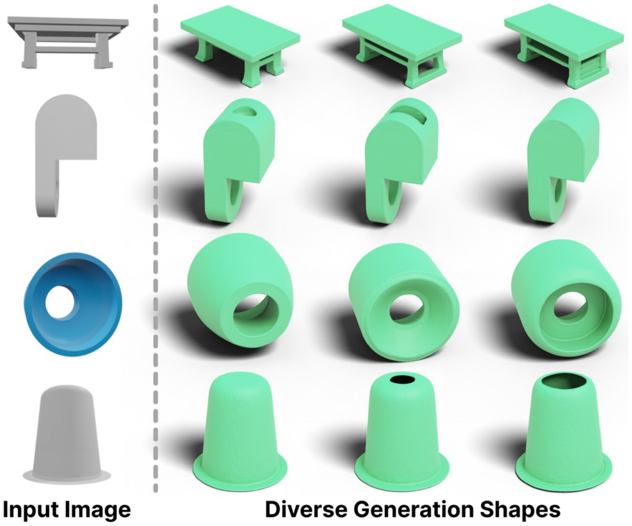

Figure 15. Our model demonstrates the capability to generate varied results from a single input image, accurately resembling the visible portions while offering diversity in unseen areas.

subsequently concatenated as additional channels of the input noise coefficients. Second, following a similar process, we convert condition latent vectors to another latent vector. However, this vector is utilized to condition the convolution and deconvolution layers via modulating the affine parameters of group normalization layers (Dhariwal & Nichol, 2021). This integration is represented by the blue arrows in Figure 14. Lastly, to condition the bottleneck volume, an additional positional encoding is applied to the condition latent vectors in an element-wise fashion. These vectors are then used in a cross-attention operation with the bottleneck volume, as indicated by the red arrows in Figure 14.

C. Additional results

A gallery of our generation results from diverse input modalities can be found in Figures 18, 19, 20, 21, 22.

Our model can produce multiple variations from a given condition, as demonstrated in Figure 15 using single-view images. In addition to the correct reconstruction of the visible input regions, our model can also create diverse and faithful results in the occluded or invisible areas, as evidenced in rows 2 and 4 of Figure 15. Note that all the quantitative comparisons are conducted in "Our Val" dataset.

C.1. Number of Points.

We conduct an ablation study to assess how the quality of generation is influenced by different sets of points, as detailed in Table 6. Our findings reveal that an increase in the number of points leads to improved IoU results on our val set. Notably, even with sparse point clouds with as few as 5,000 points, our model achieves a reasonable IoU, demonstrating the robustness of our method.

Table 6. The quantitative evaluation (on Our Val dataset) reveals that our model's performance is not significantly impacted by the number of points of inputs. Even with inputs of 5000 points, it manages to deliver reasonable reconstructions, though trained on 25000-point inputs.

Metrics	Number of Points			
	2500	5000	10000	25000
LFD ↓	1857.84	1472.02	1397.39	1368.90
IoU ↑	0.7595	0.8338	0.8493	0.8535

This analysis is also visually illustrated in Figure 16. Here, we observe that certain details are lost when fewer points are used, as evident in row 2. However, it's worth mentioning that, in general, our method performs well even with fewer points.

C.2. Voxel Comparisons.

We further compare our method with traditional approaches for converting low-resolution voxels to meshes. For the baselines, we first employ interpolation techniques such as nearest neighbor and trilinear interpolation, followed by the use of marching cubes (Lorensen & Cline, 1998) to derive the meshes. Importantly, our approach is the first large-scale generative model to tackle this task. The quantitative and qualitative results of this comparison are presented in Table 7 and Figure 17. It is evident that, among the baseline methods, trilinear interpolation outperforms nearest neighbor, which is intuitively reasonable. Our method easily surpasses both of these traditional methods in terms of IoU and LFD metrics.

D. Ablation Study

Classifier-free Guidance. As previously mentioned, we employ classifier-free guidance, as detailed in (Ho & Sali-

Table 7. Our method is quantitatively compared with traditional voxel upsampling techniques, specifically nearest neighbour upsampling and trilinear interpolation, followed by marching cubes (Lorensen & Cline, 1998) for mesh extraction. Our generative model significantly surpasses these two baselines in both Light Field Distance (LFD) and Intersection Over Union (IoU) metrics.

Setting	Methods	LFD ↓	IoU ↑
Voxel (16^3)	Ours	2266.41	0.687
	Nearest	6408.82	0.2331
	Trilinear	6132.99	0.2373
Voxel (32^3)	Ours	1580.98	0.7942
	Nearest	3970.49	0.4677
	Trilinear	3682.83	0.4719

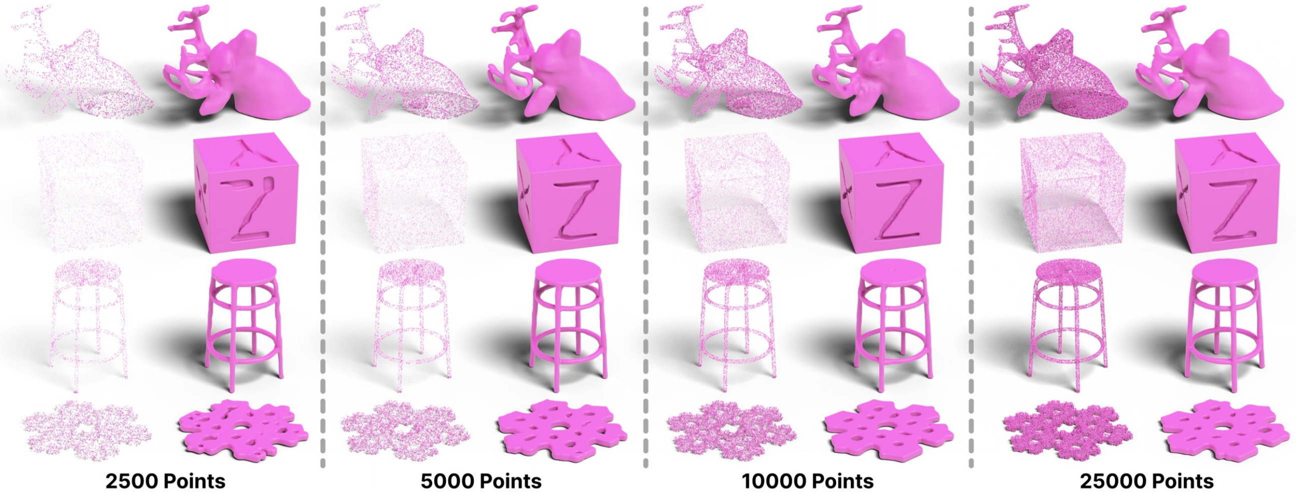

Figure 16. Visual comparisons, based on the number of input points, highlight our model's ability to robustly generate thin structures, like the deer horn or the chair leg, with a reasonable number of points (≥ 5000).

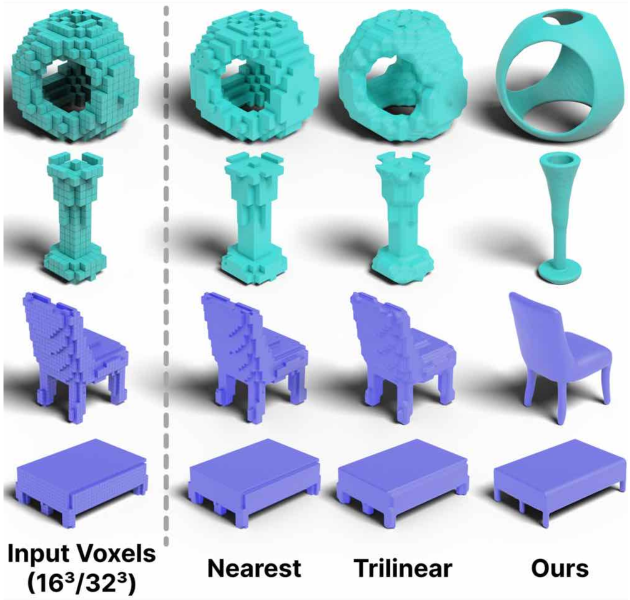

Figure 17. In comparison with meshes generated from interpolation using nearest neighbor upsampling and trilinear interpolation, our generation results display notably smoother surfaces.

mans, 2021), to enhance the quality of conditioned samples. A crucial hyperparameter in this classifier-free guidance, during inference, is the scale parameter or guidance weight, denoted as w. This parameter plays a key role in managing the trade-off between the generation's fidelity to the input conditions and the diversity of the generated output.

We experiment to explore the effect of the guidance weight parameter on the quality of samples generated by various conditional models. The guidance weight parameter was systematically adjusted in a linear progression from 1.0 to 5.0. It is important to note that, for efficient evaluation, an inference timestep of 100 was consistently employed across all experiments. The results of this study are presented in Table 8.

Empirically, we observe that a guidance weight of 2.0 is optimal for most conditional generation tasks. However, when the model is conditioned on point cloud data, a lower guidance weight of 1.0 yields better results. This contrasts with the text-to-image scenarios, which typically require a larger value for the guidance weight. We suspect this difference is attributable to the nature of the input conditions we use, such as images and point clouds, which contain more information and thus make it more challenging to generate diverse samples compared to text-based inputs. Note that we adopt these identified optimal values as fixed hyperparameters for all subsequent inferences in the remainder of our experiments, as well as for the generation of qualitative results.

Inference Time Step Analysis. Furthermore, we also provide a detailed analysis of the inference timesteps for both our conditional and unconditional models. Specifically, we evaluate the generation models under the same settings as above but with varying timesteps, namely 10, 100, 500, and 1000.

Table 9 presents the quantitative results for our different generative models using various time steps during inference. Specifically, we empirically find that a small time step (10) suffices for conditions with minimal ambiguity, such as multi-view images, voxels, and point clouds. As ambiguity rises, the required number of time steps to achieve satisfactory sample quality also increases. For the unconditional

Table 8. Quantitative analysis of the performance of our conditional generation models on different guidance weights.

Model	Metrics	Guidance Weight (w)				
		1	2	3	4	5
Single-view	LFD ↓	4395.15	**4071.33**	4121.14	4192.30	4295.28
	IoU ↑	0.3706	0.4285	**0.4348**	0.4289	0.4202
Multi-view	LFD ↓	2378.48	**2310.30**	2413.18	2522.03	2639.69
	IoU ↑	0.6322	**0.6595**	0.6488	0.6317	0.6148
Voxels (32)	LFD ↓	1701.17	**1683.95**	1769.93	1900.48	2029.59
	IoU ↑	0.7636	**0.7771**	0.7659	0.7483	0.7323
Voxels (16)	LFD ↓	2453.69	**2347.04**	2426.40	2556.62	2724.72
	IoU ↑	0.6424	**0.6726**	0.6614	0.6452	0.6289
Points	LFD ↓	**1429.37**	1432.95	1521.55	1658.03	1830.78
	IoU ↑	**0.8380**	0.8379	0.8207	0.8002	0.7781

Table 9. We quantitatively evaluate the performances of generation models with different inference time steps.

Model	Metrics	Inference Time step (t)			
		10	100	500	1000
Single-view	LFD ↓	4312.23	**4071.33**	4136.14	4113.10
	IoU ↑	**0.4477**	0.4285	0.4186	0.4144
Multi-view	LFD ↓	**2217.25**	2310.30	2369.15	2394.17
	IoU ↑	**0.6707**	0.6595	0.6514	0.6445
Voxels (32^3)	LFD ↓	**1580.98**	1683.95	1744.48	1763.91
	IoU ↑	**0.7943**	0.7771	0.7700	0.7667
Voxels (16^3)	LFD ↓	**2266.41**	2347.04	2375.89	2373.42
	IoU ↑	**0.6870**	0.6726	0.6620	0.6616
Point Cloud	LFD ↓	**1368.90**	1429.37	1457.89	1468.91
	IoU ↑	**0.8535**	0.8380	0.8283	0.8287
Unconditional	FID ↓	371.32	85.25	74.60	**68.54**

model, which has no condition, the optimal time step is the maximum one (1000). Similarly to the guidance weight, we consider the optimal time step as a hyper-parameter, which is utilized in all experiments.

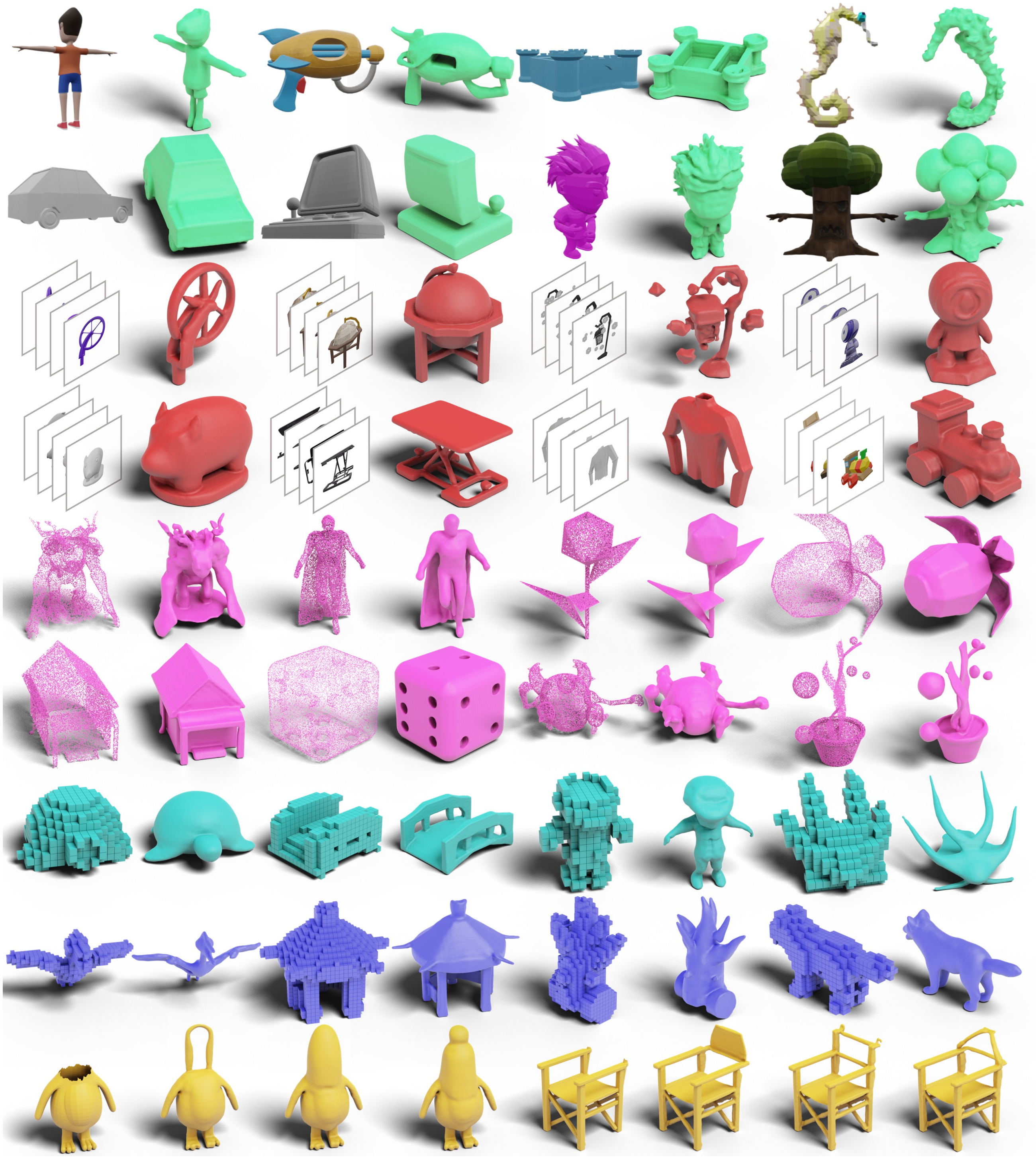

Figure 18. Make-A-Shape is able to generate a large variety of shapes for diverse input modalities: single-view images (rows 1 & 2), multi-view images (rows 3 & 4), point clouds (rows 5 & 6), voxels (rows 7 & 8), and incomplete inputs (last row). The resolution of the voxels in rows 7 & 8 are 16^3 and 32^3, respectively. In the top eight rows, odd columns show the inputs whereas even columns show the generated shapes. In the last row, columns 1 & 4 show the partial input whereas the remaining columns show the diverse completed shapes.

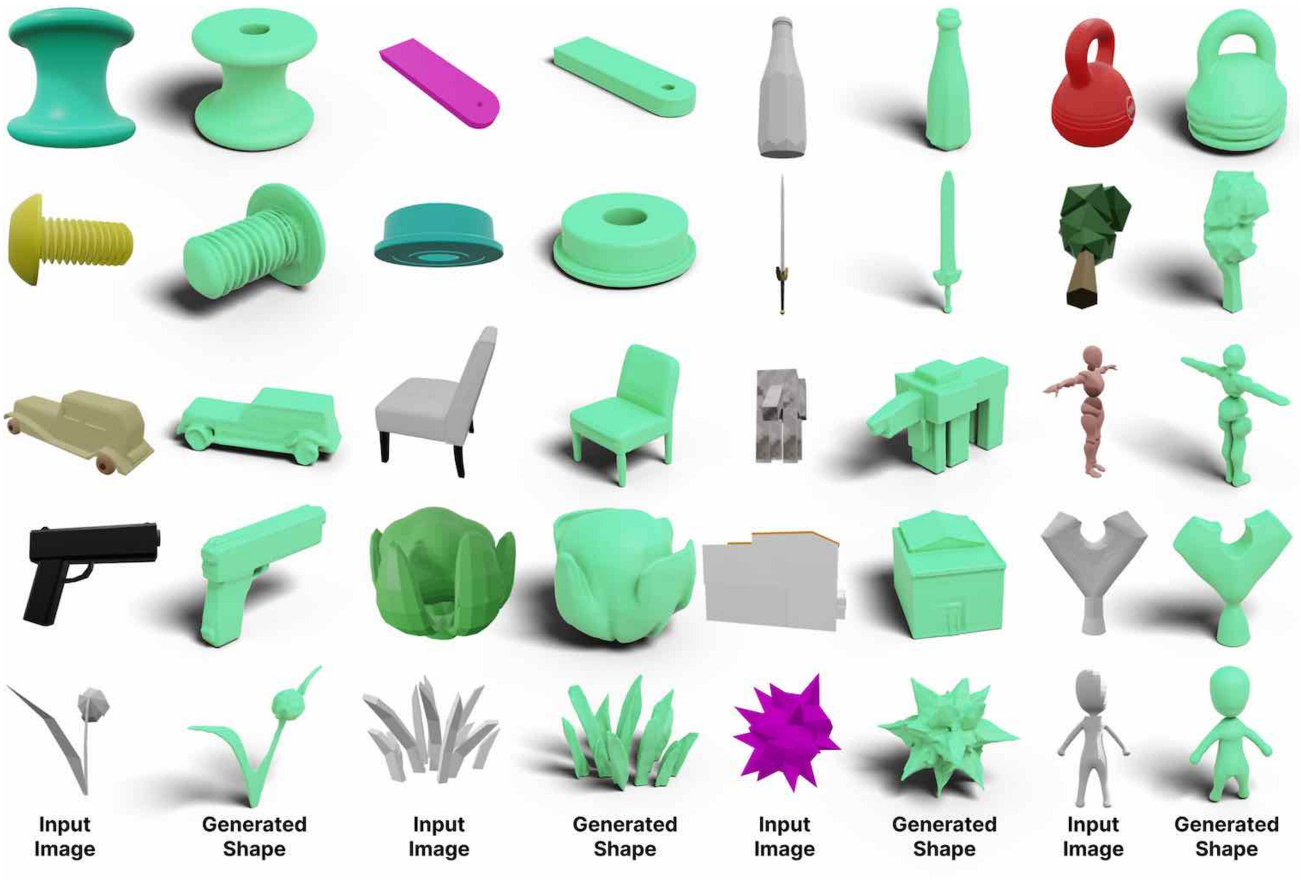

Figure 19. Shape generation conditioned on a single image.

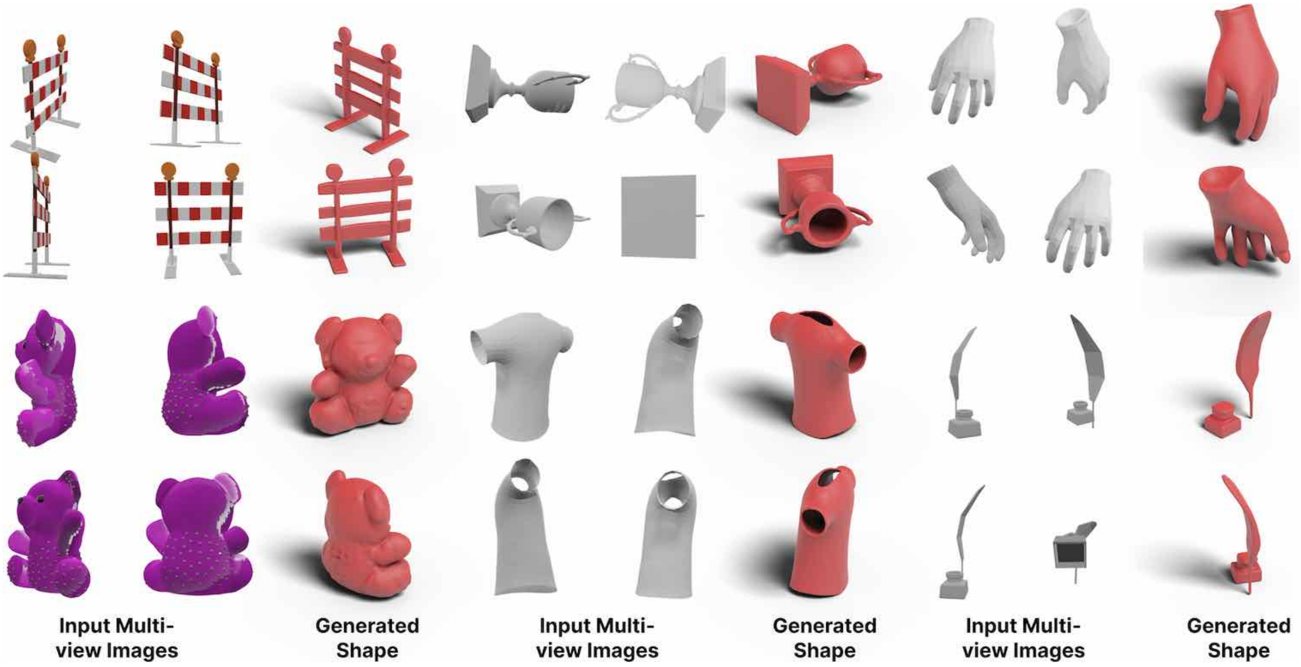

Figure 20. Shape generation conditioned on multiple views.

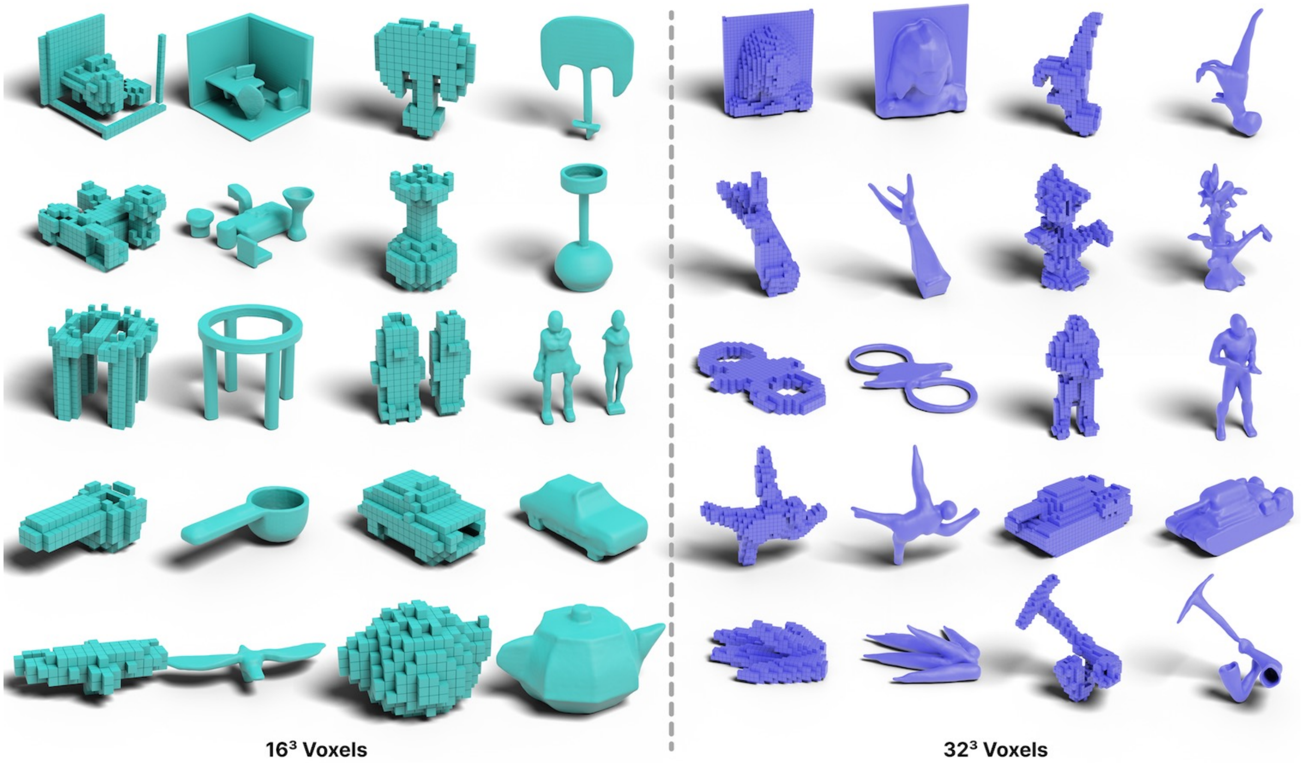

Figure 21. Additional visual results for the voxel-to-3D application.

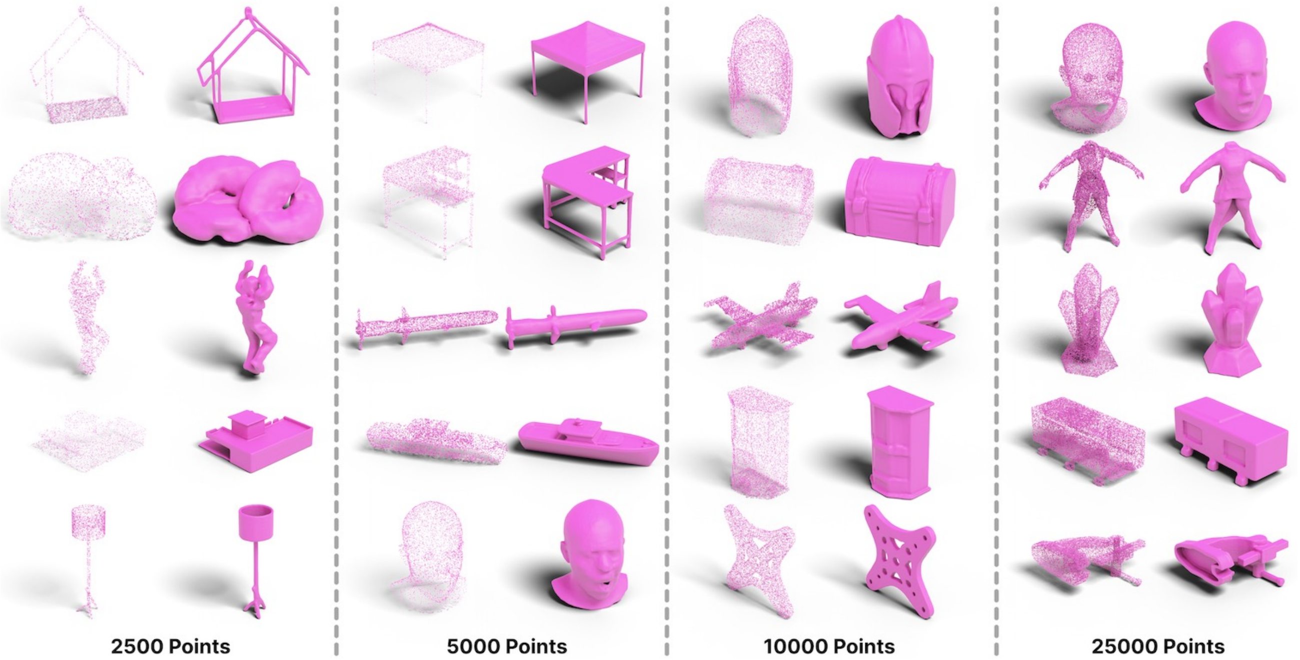

Figure 22. Additional visual results for the pointcloud-to-3D application.