# An Object is Worth 64x64 Pixels: Generating 3D Object via Image Diffusion

Xingguang Yan[1] Han-Hung Lee[1] Ziyu Wan[2] Angel X. Chang[1,3]
[1]Simon Fraser University [2]City University of Hong Kong [3]Canada-CIFAR AI Chair, Amii

omages.github.io

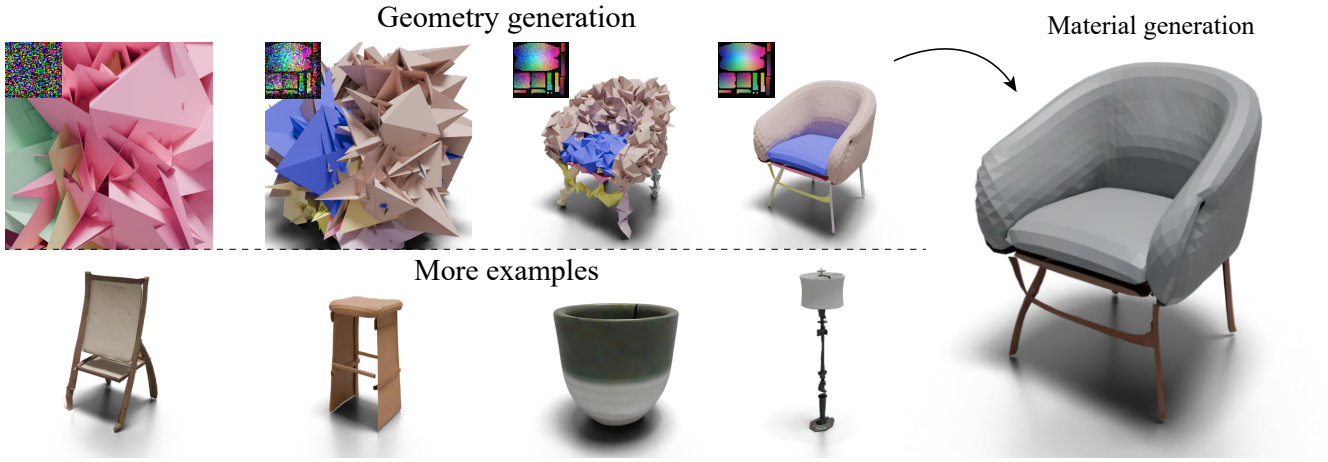


Figure 1. Visualization of geometry generation (top row) using diffusion for Object Images followed by material generation (right). The spatial coordinates (xyz) are visualized as rgb colors (see inset Object Images). The colors of the denoising mesh highlight different connected components. After generating the geometry, our model can generate PBR materials given the geometry as a condition. Other examples of generated shapes are shown in the 2nd row.

## Abstract

*We introduce a new approach for generating realistic 3D models with UV maps through a representation termed "Object Images." This approach encapsulates surface geometry, appearance, and patch structures within a 64x64 pixel image, effectively converting complex 3D shapes into a more manageable 2D format. By doing so, we address the challenges of both geometric and semantic irregularity inherent in polygonal meshes. This method allows us to use image generation models, such as Diffusion Transformers, directly for 3D shape generation. Evaluated on the ABO dataset, our generated shapes with patch structures achieve point cloud FID comparable to recent 3D generative models, while naturally supporting PBR material generation.*

## 1. Introduction

Modeling high-quality 3D shapes is vital for industries such as film, interactive entertainment, manufacturing, and robotics. However, the process can be arduous and challenging. Inspired by the success of image generation models, which have significantly enhanced the productivity of 2D content creators [48], researchers are now developing generative models for 3D shapes to streamline the synthesis of 3D assets [30, 32]. Two challenges of building generative models for 3D assets are **geometric irregularity** and **semantic irregularity**. **First**, unlike 2D images, standard 3D shape representations, such as polygonal meshes, are often highly *irregular*; their vertices and connectivity do not follow a uniform grid and vary significantly in density and arrangement. Moreover, these shapes often *possess complex topologies*, characterized by holes and multiple connected components, making it challenging to process meshes in a standardized way. These complexities pose a significant hurdle in generative modeling, as most existing techniques are designed for regular, tensorial data input. **Second**, 3D assets often possess rich *semantic sub-structures*, such as parts, patches, segmentation and so on. These are not only essential for editing, interaction, and animation, but also vital for shape understanding and 3D reasoning. However,

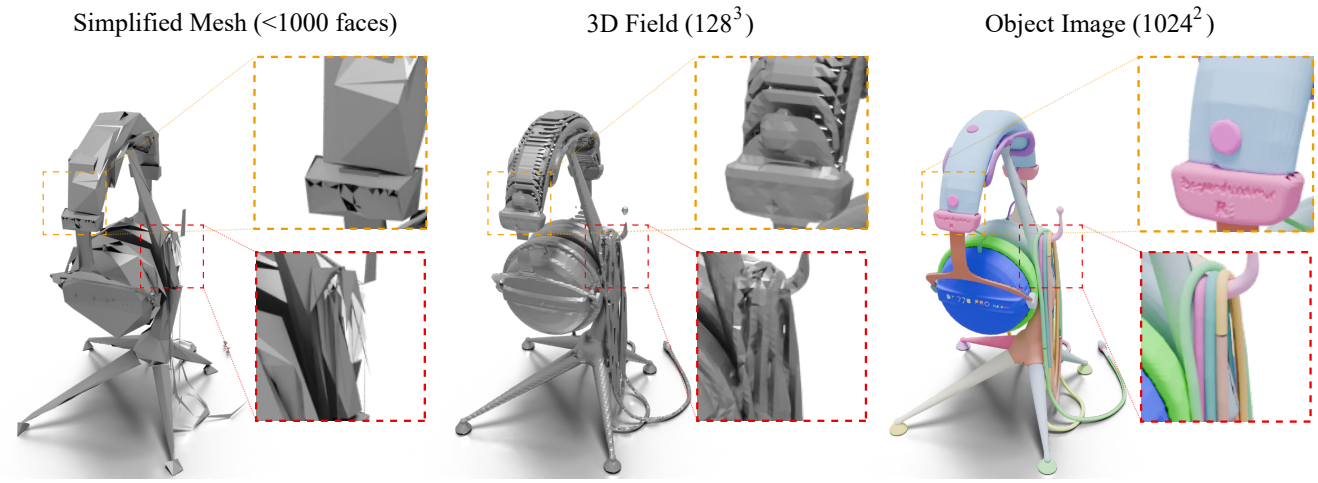


Figure 2. Comparison of different representations used for generation. Simplified meshes (left) often introduce topological errors and degenerated parts. Volumetric representations (middle) tend to merge touching parts together, struggle to model thin surfaces, and cannot handle open surfaces. In contrast, Our Object Images (right) effectively preserve the topology and structure of the original mesh.

these sub-structures also vary greatly, further hindering the design of generative models.

Most prior approaches only tried to handle either *geometric irregularity* or *semantic irregularity* [8], but not at the same time. Many works bypass the former by converting original 3D shapes into more regular representations such as point clouds [42, 67], implicit fields [70, 71], or multi-view images [53, 56, 62]. Although these formats are easier to process with neural networks, the conversions, both in forward and reverse directions, discard both geometric and semantic structures. The loss in information can significantly impact the representation accuracy and utility of the generated 3D models in applications. For example, for the headset in Fig. 2, implicit conversion fuses all the cables together, making the model difficult to use in animation. Researchers have also tried to directly model geometric irregularity [2, 41, 52], but are often restricted to simple meshes with less than 800 faces.

In our work, we explore to address the two irregularities *simultaneously* by generating 3D shapes as Multi-Chart Geometry Images (MCGIM) [49], see Fig. 1. Proposed over 20 years ago, Geometry Images [24, 49] addresses the geometric irregularity of meshes by decomposing the shape surface into one or multiple 2D patches that can be mapped and packed in a regular image. Through the irregular 2D shape packing process, MCGIM also efficiently addresses the semantic irregularity, that is, storing shapes of arbitrary number of patches in a single fixed-size image. However, its automatic patch decomposition often results in less semantically meaningful patches and boundaries. Our key observation is that many human-modeled 3D assets come with a semantically rich decomposition of patches in the form of UV-charts, which is usually only used for texturing in prior arts [68]. These UV-charts can be easily processed to MCGIMs that can then be mapped back to 3D shapes.

Thus, inspired by Geometry Images, we propose to rasterize the mesh geometry (together with texture and material maps) into a 12-channel image as a new representation for 3D generation. This approach allows us to represent 3D shapes as 2D images and provides several benefits. The representation 1) is simple and regular, 2) preserves the geometric and semantic structure together with PBR materials and 3) can be learned with image-based generative model to generate textured 3D meshes. We use the term *Object Images* (*omages* for short) for this representation, emphasizing its ability to encapsulate not just the geometry structure, but also material and semantically meaningful patch-decomposition of an object, and highlighting its potential for 3D generation by leveraging existing image-based methods. In this work, we convert the shapes of the ABO dataset [14], which contains triangle meshes with designer-made UV-maps, into 1024 resolution omages, downsample them to 64 resolution with special care and use Diffusion Transformers [44] to model their distribution. Our results show that our method generate shapes with patch structures that approaches similar geometric quality as state-of-the-art 3D generative models (in terms of point cloud FID), while naturally supporting PBR material generation.

## 2. Related Work

Our work lies in the field of surface shape generation. In this section, we present a survey of representative approaches categorized by their underlying 3D representation, with a focus on generative modeling.

**Polygonal meshes**. As the most ubiquitous 3D representation, meshes, especially those modeled by 3D designers,

are efficient and flexible, but also are well known for their difficulty to process with neural networks due to their irregularity. While various convolutional neural networks have been developed for mesh data [25, 37, 46, 50], they have predominantly focused on shape understanding tasks like classification. The complexity of developing context-free unpooling operator on meshes impedes their use for mesh generation. To avoid the challenges of directly learning meshes with their native connectivity, researchers approximate the geometry using various surrogate mesh representations like surface patches [23], predicted meshes [15], deformed cuboids [20], and binary space partitions [11]. This comes with the price of losing the details and structures of the original mesh. In contrast, PolyGen [41] directly learns the distribution of the native mesh in a vertices-then-face manner with two autoregressive transformers [57]. However, this complex two stages pipeline exhibits limited robustness during inference as described in a later work, MeshGPT [52], which avoids this complexity by first encoding meshes into sequences of graph neural networks encoded face tokens that can be easily processed with a single autoregressive transformer. MeshAnything [9] further improves MeshGPT's encoder/decoder and enables conditional mesh generation given a reference point cloud. These remarkable breakthroughs enable mesh generation with up to 800 triangular faces. However, high-quality human-designed meshes usually have many more faces. For example, in the ABO dataset [14], over $70\%$ of the shapes has more than $10^4$ triangles. The current approaches need to first decimate the meshes to less than 800 faces, which may introduce topological errors (see Fig. 2). Moreover, these meshes often are accompanied with PBR materials and patch structures that polygonal mesh can not natively represent. In contrast, our object image is not restricted by the number of faces and naturally encapsulates material and patch information.

**Multi-chart representations**. Modeling 3D shapes as single or multiple parametric patches (charts), is a prevalent approach for modeling smooth, curved shapes [21, 45]. Representing a polygonal mesh with parametric patches, commonly referred to as UV-mapping, aligns the irregular mesh with a regular 2D plane. This alignment is essential for texture mapping, which paints rich textural images onto the 3D geometry [5, 7], and for surface remeshing, editing, and many other applications [51]. To represent and store the geometry in a fully regular manner, Geometry Images [24, 26] first parameterizes a mesh onto a planar domain and resamples the geometry onto an image pixel grid. Further, Multi-charts Geometry Images (MCGIM) [6, 49, 74] proposed to pack multiple patches into a single image, achieving lower distortion and is applicable to shapes with arbitrary topology. Our proposed *Object Image* is a kind of MCGIM extended with materials built specifically for image diffusion models.

While the utility of geometry images in deep learning has been well recognized, their use has been limited to either simple topologies or with automated patch splitting, making it challenging to obtain good surface parameterizations. Sinha et al. [54] and Maron et al. [36] applied CNNs to geometry images representing parameterizations on spherical and toric domains, respectively. Later, Ben-Hamu et al. [4] and Alhaija et al. [1] (XDGAN) used GANs to generate *genus-zero* shapes as geometry images. Meanwhile, FoldingNet [65], AtlasNet [23] and its followups [3, 16–18, 31, 60] have explored learning to approximate shapes with parametric patches in an unsupervised manner. These efforts commonly employ algorithmic or approximated patch splitting, which tends to be either topologically constrained or inaccurate.

In contrast, we recognize that human-authored UV-atlases can be easily processed into MCGIMs, supporting arbitrary patch topology, and can be easily generated with image diffusion models. While UV-atlases are widely used in recent learning-based mesh texturing methods [68], they serve primarily as auxiliary information. In contrast, we note that UV-atlases effectively transform a mesh into parametric surfaces, providing a valuable representation for both geometry and material generation. More recently, BrepGen [63] synthesizes CAD B-Rep models by generating their patches and edges with diffusion models. However, it is still restricted to simple genus-zero patches and can only be applied to B-Rep models.

**3D fields and multi-view images**. 3D shapes can be implicitly represented as a level-set of a spatial field. In this way, the irregularity challenge is circumvented, although important structural and topological information is inevitably gone along the way (See Fig. 2). Instead of generating a field directly as a 3D grid (voxels) [61], the recent trend is to first parameterize the field with a neural network (neural field). Seminal works utilize auto-encoders [10, 38] or auto-decoders [43] to compress the neural field as a single latent vector, which can be easily generated via methods like GANs [22] or VAEs [29]. Later works represent and generate neural fields using multiple latent vectors to enhance spatial reasoning [12, 13, 19, 27, 40, 72, 73]. In particular, ShapeFormer [64], 3DILG [69], 3DShape2VecSet [70] and Mosaic-SDF [66] utilize the sparsity of the 3D shape to further compress the field and enables generating higher-resolution results.

Another line of work represents and generates shapes as multi-view images [33, 34, 55, 56, 62]. They adopt diffusion models to generate multiple 3D consistent images of different views. Meshes can then be reconstructed via neural field methods like NeRF [39] or NeuS [58]. To enhance

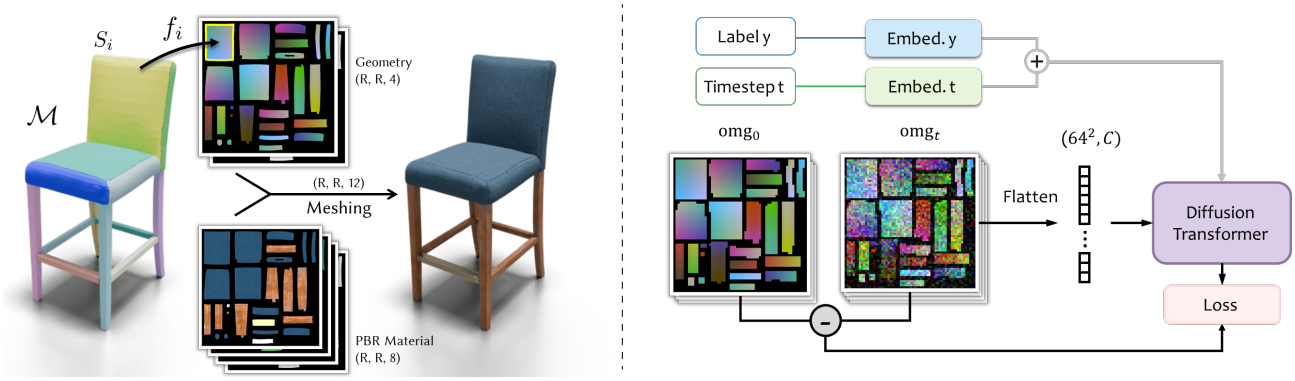


Figure 3. Method overview. **Left**: We assume the mesh $\mathcal{M}$ has patch decomposition $\{S_i\}$, and has single-valued uv-map $f_i$ that flattens patch $S_i$ into the 2D uv-domain. Together with the material maps, Object Images can represent high-quality photo-realistic object. **Right**: We train the image diffusion generative model with Diffusion Transformer. The input noised Object Image, omg, is first flattened into a sequence before passing into the transformer to predict the clean omg$_0$.

understanding of the shape structure, especially the interior, Slice3D [59] proposes using images of shape slices instead. Since most of these multi-image methods obtain geometry through neural fields, they share similar advantages and disadvantages with 3D field generation methods.

Our object image can be seen as a combination of neural field and mesh representations. It preserves the topology and patch structure of the original mesh while functioning as a specialized form of 2D neural field, making them highly suitable for neural network processing due to their regular structure.

## 3. Method

In this section, we first present the mathematical formulation of the Object Image (*omage* for short) representation (Section 3.1). Next, we describe how to utilize image-based generative models, specifically Diffusion Transfomer [44], to generate these omages (Section 3.2). Finally, we describe how to obtain omages from a 3D asset (Section 3.3).

### 3.1. Object Images

Given a 3D shape $\mathcal{M}$ that is a 2D manifold embedded in 3D space, we consider it as a disjoint union of a set of $N$ surface patches $\{S_i\}$. By disjoint, we mean any two distinct patches $S_i$ and $S_j$ only overlap on their boundaries, i.e., $S_i \cap S_j = \partial S_i \cap \partial S_j$. We assume each $S_i$ is assigned a injective *UV-mapping*, $f_i : S_i \to [0,1]^2$, where $f_i(p) = (u, v)$ and $[0,1]^2$ is the *UV-space*. The domain and image of $f_i$ together are called an *UV-island*, or *UV-chart*: $I_i = (S_i, f_i(S_i))$. We denote the set of the $N$ UV-islands, $I := \{I_i\}$, as the *UV-atlas* of $\mathcal{M}$.

We indicate if a point in UV-space is inside an island by defining an occupancy function $\alpha$ as follows:

$$\alpha(u,v) = \begin{cases} 1 & \text{if } \exists\, i \text{ such that } (u,v) \in f_i(S_i) \\ 0 & \text{otherwise} \end{cases}$$

We then define the position map $\pi$ of $M$ as follows:

$$\pi(u,v) = \begin{cases} f_i^{-1}(u,v) & \text{if } (u,v) \in f_i(S_i) \text{ for some } i \\ \text{undefined} & \text{otherwise} \end{cases}$$

By *packing* the UV-islands through translation and scaling, we ensure that the set family $\{f_i(S_i)\}$ is disjoint, making $\pi$ and $\alpha$ strictly deterministic (single-valued). Hence, we can always map the UV-domain back to the original shape $\mathcal{M}$ easily. Therefore, $(\pi, \alpha)$ is an equivalent representation to $\mathcal{M}$. By rasterizing $(\pi, \alpha)$ to an image $O \in \mathbb{R}^{R\times R\times 4}$, where $O[i,j] = (\pi[i,j], \alpha[i,j])$, we can approximate $\mathcal{M}$ with $\mathcal{M}^*$, where $\mathcal{M}^*$ is the triangular mesh reconstructed from $O$ via *remeshing*, where we connect $\pi[i,j], \pi[i,j+1], \pi[i+1,j]$ and $\pi[i+1,j+1], \pi[i,j+1], \pi[i+1,j]$ to form triangles if the occupancy of the triplet is all 1. In theory, as $R \to \infty$, $\mathcal{M}^*$ will be infinitely close to $\mathcal{M}$. This forms the geometry part of the *omage* representation. The material part of an *omage* consists of albedo (3 channels), normal (3 channels), metalness (1 channel), and roughness (1 channel) maps. Together, we obtain a 12 channel omage $O^*$ which can be meshed back to a photo-realistic 3D object, as shown in Fig. 3.

With the 3D objects encoded as omages, we aim to train an image diffusion model to model the distribution of the 3D objects. In the next subsections, we will first discuss our design choice for the generative model, and then show how the omages are obtained.

### 3.2. Generative modeling for omages

We observe that generating Object Images (omages) combines aspects of ordinary image generation and set gener-

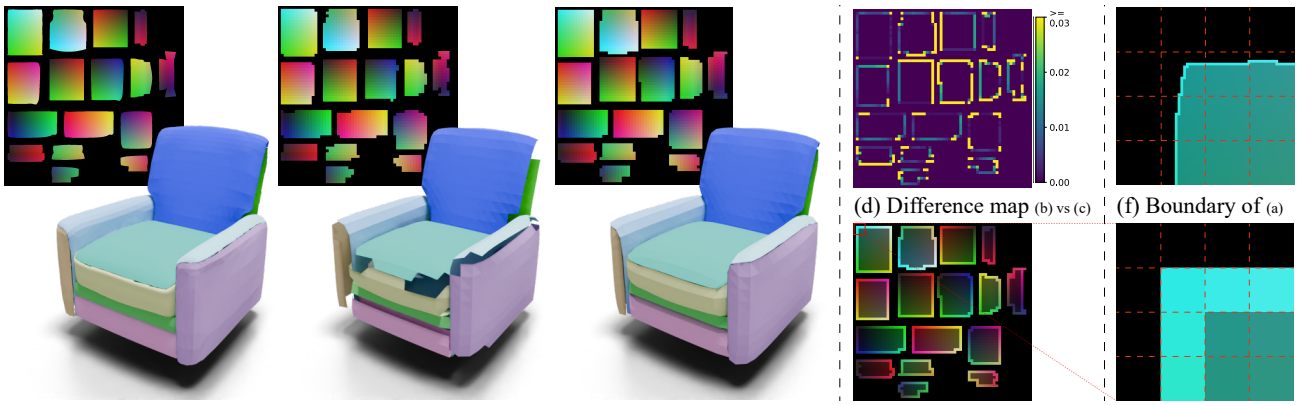


Figure 4. Direct downscaling an omage from high-resolution (a) to lower resolution (b) usually leads to significant gaps between patches. By snapping the boundary vertices of the high resolution omage (f) into lower resolution via sparse pooling (e)(g), the gaps are significantly reduced (c)(d).

ation. Within each patch, the generation process resembles standard image generation due to regular connectivity. However, among the patches, the problem behaves more like set generation: the patch's location in 2D does not affect the 3D shape. Patches can be swapped and moved around without altering the 3D geometry. Additionally, touching boundaries in 3D between two patches often sit far apart in 2D, requiring long-range dependency modeling. Since transformers excel at learning sets and modeling long-range dependencies, and diffusion models are well-known for their image generation capabilities, we use the Diffusion Transformer [44] as our architecture. Unlike the original method, we set the patch size to 1 to avoid jagged edges in the generation results.

Given the importance of geometry in omages, we first train a model to generate the four geometric channels. We then train a second model to generate the remaining eight channels. In the second stage, the input has 12 channels, using the first four channels as conditions and excluding them from noise addition and loss computation.

### 3.3. Obtaining object images

3D objects with UV maps cannot be directly converted into images due to issues such as overlapping regions, out-of-boundary UVs, touching boundaries, or excessive patches. To address this, we use a UV-atlas repacking method with special care to pack patches with material maps into a $(1024, 1024, 12)$ omage. To avoid large number of patches, we merge vertices with the same 3D and 2D UV coordinates, and keep a maximum of $K$ largest patches. Detailed descriptions of this process are provided in Sec. A of the supplement. For efficient learning, we downsample the images with sparse pooling, which snaps the boundaries and eliminates gaps. Further details are provided below.

**Downsample object images and boundary snapping**. Operating within the image domain offers the intrinsic benefit of multi-resolution support. By simply rescaling the omage, object resolution can be adjusted accordingly. For training, we downscale high-resolution omages from 1024 to 64 pixels, enabling efficient processing by transformer models. As illustrated in Fig. 4, standard rescaling methods often fail to preserve boundary information, leading to notable gaps between patches. Inspired by MCGIM [49], we address this challenge through boundary snapping, where boundary pixels are adjusted based on the contours of the high-resolution image. While this approach is less accurate than using the ground truth mesh boundaries as MCGIM does, it offers greater convenience. Assuming the higher resolution is divisible by the lower, each pixel in the low-resolution image corresponds to a block of pixels. In our case, the block is 16x16. We determine the value of each pixel in the lower resolution image via sparse pooling, averaging only the boundary pixels within each 16x16 block while ignoring other values. This process is illustrated in Fig. 4 (f) and (g).

## 4. Experiments

### 4.1. Implementation Details

**Dataset**. We conduct experiments on the Amazon Berkeley Objects (ABO) [14] dataset (license CC BY 4.0) which consists of roughly 8000 high-quality designer-made 3D models with UV-atlases across 63 categories. All of these objects are textured meshes accompanied with initial unprocessed UV-atlases and PBR materials. We convert the glb format shapes to 12 channel 1024 resolution omages with Blender 4.0[1] using the method described in Sec. 3.3. The

[1] https://www.blender.org/

Figure 5. Examples of label-conditioned Omage-64 generation results. The left side displays results for 'ottoman', 'bed', 'exercise equipment', 'painting', 'lamp' 'vanity', 'plant pot', 'chair', 'pillow' and 'lamp'. Even at this resolution, thin structures are successfully generated. On the right, a scene with three objects generated by our method is shown, highlighting our capability in material generation.

Table 1. Evaluation on class conditional generation. We measure the point cloud FID (p-FID) and KID (p-KID) for uncolored points sampled from the generated mesh. In geometry generation, with 64 resolution images, we outperform MeshGPT (mGPT) [52] and slightly underperform 3DShape2VecSec (S2VS) [70].

| | Chair | | | Sofa | | | Table | | | Lamp | | | Mean | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S2VS | mGPT | Ours | S2VS | mGPT | Ours | S2VS | mGPT | Ours | S2VS | mGPT | Ours | S2VS | mGPT | Ours |
| p-FID ↓ | 15.9 | 31.2 | 18.9 | 20.6 | 24.9 | 23.3 | 11.9 | 20.3 | 22.4 | 33.0 | 51.2 | 43.6 | 20.4 | 31.9 | 27.0 |
| p-KID ↓ | 7.31 | 17.3 | 7.83 | 9.22 | 10.7 | 9.69 | 2.43 | 7.12 | 6.75 | 14.3 | 31.4 | 26.4 | 8.32 | 16.6 | 12.7 |

1024 omages are downsampled to 64 resolution with edge snapping. Unlike volumetric representations, the processing of omages is highly efficient and robust. We can obtain 1024 omage from a single raw glb file within 6 seconds.

**Diffusion Transformer architecture and training**. We use DiT-B/1 [44] model which has 12 layers of Transformer blocks. We set the patch size to 1 to avoid results with jaggies. This essentially removes the patchify layer, resulting in a full 4096 sequence length. With the help of mixed-16 bit precision training, we train our model with 4 NVIDIA 3090 GPUs for 3 days. We use AdamW [35] optimizer with learning rate set to 1e-4. The effective batch size is 32. For generation, we use a classifier-free guidance scale of 4, and 250 sampling steps.

### 4.2. Class conditional generation

In Fig. 5, we present generated results from our model trained on all categories of the dataset. The geometry and material are generated in an autoregressive manner. With a single representation, our method is able to generate challenging materials such as mirrors (see Fig. 5 right). For evaluation and comparison, we focus on a subset of the four largest categories ('chair', 'sofa', 'table', and 'lamp'), comprising approximately 3800 shapes. We train both our method and the baseline methods on this subset.

**Evaluation metrics**. Following previous works [42, 66, 70], We use point cloud FID (p-FID) and KID (p-KID) to measure the quality of the generation results. We adopt the pretrained PointNet++ [47] feature extractor provided by Point-E [42] for calculating FID and KID. We randomly generate 512 shapes using each model and calculate the metrics for these 512 shapes versus the training set of the categories.

**Baselines**. We compare to 3DShape2VecSet [70], which is one of the state-of-the-art neural implicit-based 3D generative models. Its representation module encodes a 3D occupancy field into a set of latent vectors. We also compare to MeshGPT [52], which uses graph convolutional autoencoder to turn triangle mesh generation into a sequence generation problem. We refer to our model for comparison as 'omage64-DiT'.

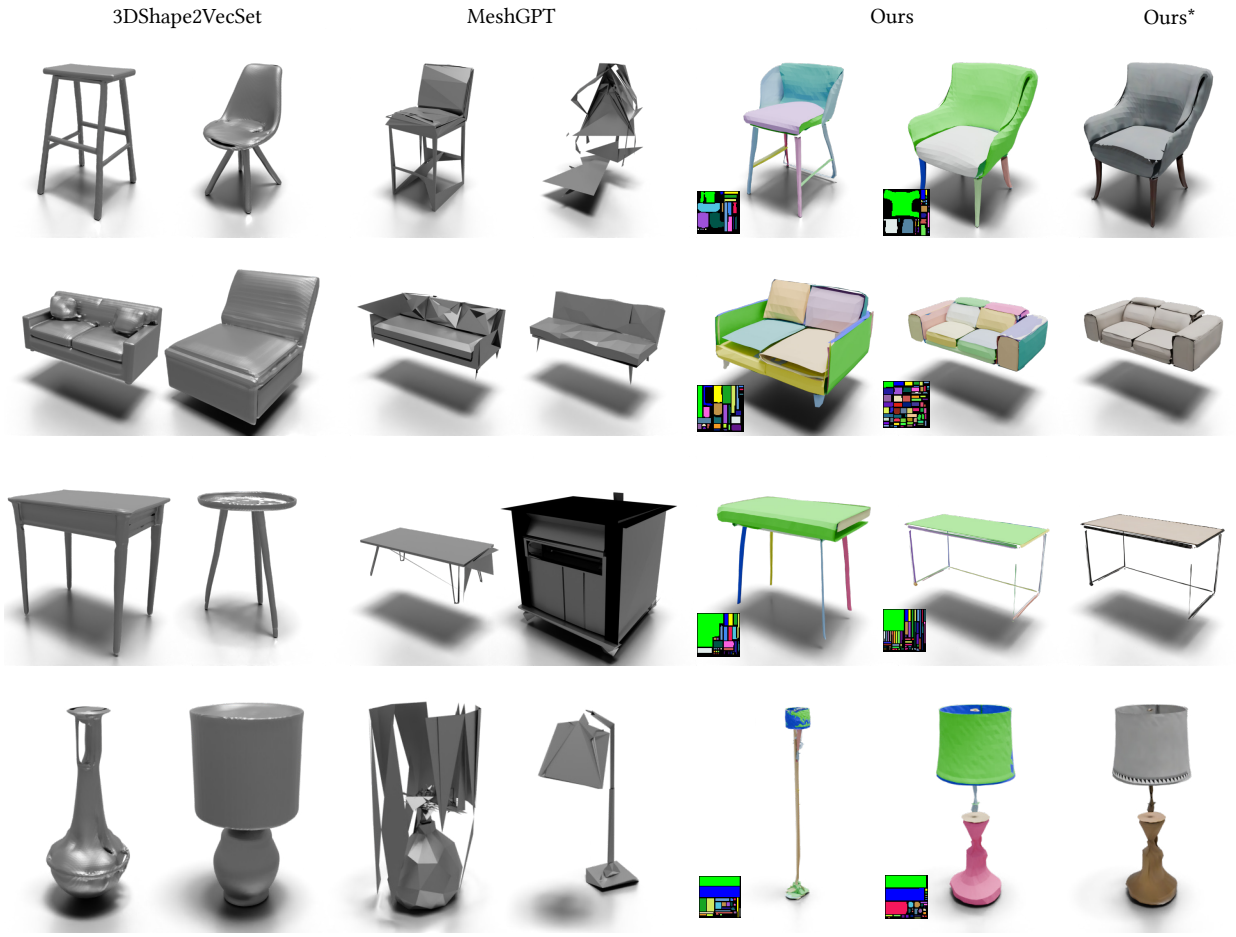


Figure 6. Label conditioned generation results for *chair*, *sofa*, *table*, and *lamp*. For our method, we show generated patches in different colors, and with generated material. Using Object Images, we are able to generate fine detailed geometry with material information. In contrast, MeshGPT [52] often fails to generate coherent geometry. 3DShape2VecSet [70] generates cleaner geometry but is not able to generate material and patch decomposition.

For 3DShape2VecSet, we adopt the official implementation from the authors. More specifically, we directly use their autoencoder without additional training and finetune their pretrained diffusion model on ABO dataset. For MeshGPT, we use a third-party implementation[2], which can be trained on shapes decimated to 400 faces. We finetune both its autoencoder and auto-regressive transformer on the ABO dataset.

As shown in Fig. 6, 3DShape2VecSet can generate good quality shapes, but may fail to generate reasonable thin structures (the lamp's wire). Also, the generated shape has very dense triangles due to its implicit nature. Meanwhile, MeshGPT can obtain very compact results (table and sofa), but is prone to have messy triangles. MeshGPT may also generate flipped triangles (see the second table). In contrast, our method can directly generate thin structures and open surfaces. Table 1 shows that despite the difficulty of structured geometry generation, our method can still achieve similar p-FID and p-KID scores to 3DShape2VecSet. In addition, our method generates realistic PBR materials and semantically meaningful patch decomposition.

### 4.3. Shape Novelty

In Fig. 8, we check if our method can generate novel samples by comparing generated examples with their closest ground-truth examples in the dataset. We retrieve the nearest neighbour by directly computing the mean square errors between the generated omage and the omages in the dataset. Our generated result has non-trivial differences toward the nearest neighbours in the training set, showing that our method is not overfitting. However, maybe due to the challenge of the combinatorial nature of omages, the layout of the 2D patches is similar. The third sample of Fig. 8 can be regarded as a failure case of our method: It occasionally connects wrong side of the patch boundary. However, this

[2] https://github.com/MarcusLoppe/meshgpt-pytorch

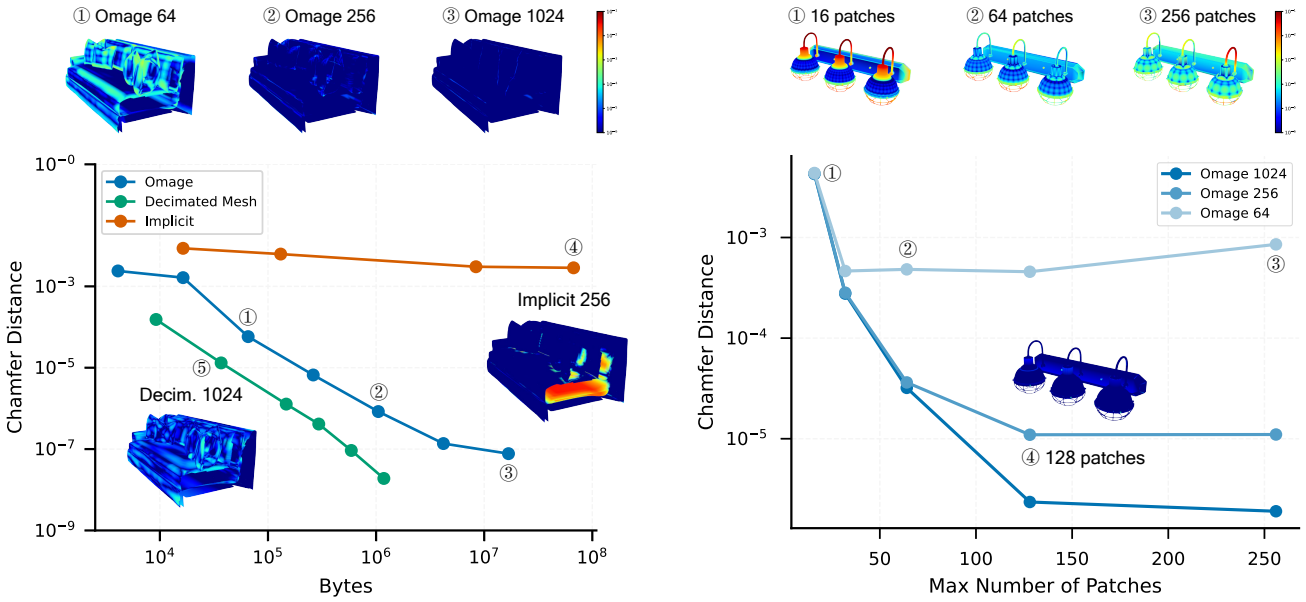


Figure 7. Representation analysis. Left: Chamfer Distance (CD) vs. byte size. A sectional view of the sofa example highlights the accuracy for both exterior and interior structures. Right: The effect of the maximum number of patches on the accuracy of Omage representations, demonstrating the trade-offs of this parameter. Note that the color map is displayed in log-scale.

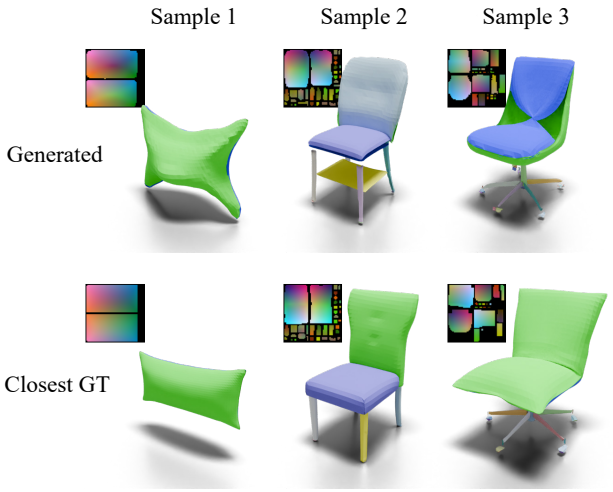


Figure 8. Our generated results compared with its nearest neighbour in the dataset.

example still demonstrates an interesting patch alignment.

### 4.4. Representation Analysis

We also analyze the the ability of our representation to capture details of the shape geometry at different resolutions. The left side of Fig. 7 illustrates a key drawback of implicit representation: it fails to differentiate touching parts, considering all nearby regions as inside, thus failing to reconstruct surfaces like the cushion of a sofa resting on the seat. In contrast, omage preserves such structures effectively and is more efficient, achieving comparable performance to decimated triangle mesh. On the right side of the figure, we show that the maximum number of patches is also a critical parameter. If too low, large patches are removed, causing significant errors. If too high, particularly for complex shapes with many intricate parts, the gap ratio increases, and pixel density per patch decreases, leading to reduced accuracy in those regions. We choose $K = 64$ for 64-resolution omages generation since it strikes a good balance between patch coverage and per-patch accuracy.

## 5. Conclusion

In this paper, we introduced a new paradigm for generating photo-realistic 3D objects with patch structures. We show the possibility of generating 3D object with materials by only denoising a small 64x64 2D image with an image diffusion model. This new paradigm also has limitations: It can not guarantee to generate watertight meshes, requires 3D shapes for training to have good quality UV atlases, and the current resolution is only limited to 64. In the future, we will continue to explore how to address these problems to fully utilize the benefits of this regular representation for high-quality structured 3D assets generation.

**Acknowledgements**. We thank Xueqi Ma and Biao Zhang for their advice and guidance in training the baseline methods. This work was supported by a CIFAR AI Chair, an NSERC Discovery grant, and a CFI/BCKDF JELF grant. *Mesh credits*: Fig. 2 Headphone [28].

## References


[1] Hassan Abu Alhaija, Alara Dirik, André Knörig, Sanja Fidler, and Maria Shugrina. XDGAN: Multi-modal 3D shape generation in 2D space. In *British Machine Vision Conference*, 2022, arXiv:2210.03007.

[2] Antonio Alliegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. PolyDiff: Generating 3D polygonal meshes with diffusion models. *Advances in neural information processing systems*, 2023, arXiv:2312.11417.

[3] Jan Bednarík, Shaifali Parashar, Erhan Gundogdu, Mathieu Salzmann, and Pascal Fua. Shape reconstruction by learning differentiable surface representations. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4715–4724, 2020, arXiv:1911.11227.

[4] Heli Ben-Hamu, Haggai Maron, Itay Kezurer, Gal Avineri, and Yaron Lipman. Multi-chart generative surface modeling. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018, arXiv:1806.02143.

[5] James F. Blinn and Martin E. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–547, 1976, doi:10.1145/360349.360353.

[6] Nathan A Carr, Jared Hoberock, Keenan Crane, and John C Hart. Rectangular multi-chart geometry images. In *Symposium on geometry processing*, pages 181–190, 2006, doi:10.5555/1281957.1281981.

[7] Edwin E. Catmull. *A subdivision algorithm for computer display of curved surfaces*. PhD thesis, The University of Utah, 1974.

[8] Siddhartha Chaudhuri, Daniel Ritchie, Jiajun Wu, Kai Xu, and Hao Zhang. Learning generative models of 3D structures. *Computer Graphics Forum*, 39(2):643–666, 2020, doi:10.1111/cgf.14020.

[9] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, and Chi Zhang. MeshAnything: Artist-created mesh generation with autoregressive transformers, 2024, arXiv:2406.10163.

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 5939–5948, 2019, arXiv:1812.02822.

[11] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. BSP-Net: Generating compact meshes via binary space partitioning. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 45–54, 2020, arXiv:1911.06971.

[12] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G. Schwing, and Liangyan Gui. SDFusion: Multimodal 3D shape completion, reconstruction, and generation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4456–4465, 2023, arXiv:2212.04493.

[13] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-SDF: Conditional generative modeling of signed distance functions. In *Proc. Int. Conf. on Computer Vision*, pages 2262–2272, 2023, arXiv:2211.13757.

[14] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. ABO: Dataset and benchmarks for real-world 3D object understanding. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 21126–21136, 2022, arXiv:2110.06199.

[15] Angela Dai and Matthias Nießner. Scan2mesh: From unstructured range scans to 3D meshes. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 5574–5583, 2019, arXiv:1811.10464.

[16] Zhantao Deng, Jan Bednarík, Mathieu Salzmann, and P. Fua. Better patch stitching for parametric surface reconstruction. In *International Conference on 3D Vision (3DV)*, pages 593–602, 2020, arXiv:2010.07021.

[17] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Learning elementary structures for 3D shape generation and matching. *Advances in neural information processing systems*, 2019, arXiv:1908.04725.

[18] Theo Deprelle, Thibault Groueix, Noam Aigerman, Vladimir G. Kim, and Mathieu Aubry. Learning joint surface atlases. In *ECCV Workshop on Learning to Generate 3D Shapes and Scenes*, 2022, arXiv:2206.06273.

[19] 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*, 2022, arXiv:2209.11163.

[20] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. SDM-NET: Deep generative network for structured deformable mesh, 2019, arXiv:1908.04520.

[21] Ronald Goldman. *An integrated introduction to computer graphics and geometric modeling*. CRC Press, 2009, doi:10.5555/1594365.

[22] 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*, pages 2672–2680, 2014, arXiv:1406.2661.

[23] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2018, arXiv:1802.05384.

[24] Xianfeng Gu, Steven J Gortler, and Hugues Hoppe. Geometry images. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, pages 355–361, 2002, doi:10.1145/566654.566589.

[25] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: a network with an edge. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 38(4):1–12, 2019, arXiv:1809.05910.

[26] Hugues Hoppe. Overview of recent work on geometry images. In *Proceedings of Geometric Modeling and Processing*, 2004, doi:10.1109/GMAP.2004.1290021.

[27] Moritz Ibing, Isaak Lim, and Leif Kobbelt. 3D shape generation with grid-based implicit functions. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 13554–13563, 2021, arXiv:2107.10607.

[28] Halil Kantarci. Headphone with stand. https://sketchfab.com/3d-models/headphone-with-stand-4ffedc9bffad4a549f6e0a46b0f92b05, 2018.

[29] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. Int. Conf. on Learning Representations*, 2014, arXiv:1312.6114.

[30] Hanhung Lee, Manolis Savva, and Angel X Chang. Text-to-3D shape generation. *Computer Graphics Forum*, 43(2): e15061, 2024, arXiv:2403.13289.

[31] Jiahui Lei, Srinath Sridhar, Paul Guerrero, Minhyuk Sung, Niloy Jyoti Mitra, and Leonidas J. Guibas. Pix2Surf: Learning parametric 3D surface models of objects from images. In *Proc. Euro. Conf. on Computer Vision*, 2020, arXiv:2008.07760.

[32] Xiaoyu Li, Qi Zhang, Di Kang, Weihao Cheng, Yiming Gao, Jingbo Zhang, Zhihao Liang, Jing Liao, Yan-Pei Cao, and Ying Shan. Advances in 3d generation: A survey, 2024, arXiv:2401.17807.

[33] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. SyncDreamer: Generating multiview-consistent images from a single-view image. In *Proc. Int. Conf. on Learning Representations*, 2024, arXiv:2309.03453.

[34] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3D: Single image to 3D using cross-domain diffusion. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2024, arXiv:2310.15008.

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. Int. Conf. on Learning Representations*, 2019, arXiv:1711.05101.

[36] Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir G. Kim, and Yaron Lipman. Convolutional neural networks on surfaces via seamless toric covers. *ACM Transactions on Graphics (TOG)*, 36: 1 – 10, 2017, doi:10.1145/3072959.3073616.

[37] Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on Riemannian manifolds. In *ICCV workshop on 3D Representation and Recognition*, 2015, arXiv:1501.06297.

[38] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4460–4470, 2019, arXiv:1812.03828.

[39] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021, arXiv:2003.08934.

[40] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. AutoSDF: Shape priors for 3D completion, reconstruction and generation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2022, arXiv:2203.09516.

[41] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. PolyGen: An autoregressive generative model of 3D meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020, arXiv:2002.10880.

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

[43] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 165–174, 2019, arXiv:1901.05103.

[44] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proc. Int. Conf. on Computer Vision*, 2023, arXiv:2212.09748.

[45] Les A. Piegl and Wayne Tiller. The NURBS book. In *Monographs in Visual Communication*. 1995, doi:10.1007/978-3-642-59223-2.

[46] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution, 2018, arXiv:1810.02303.

[47] 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, 2017, arXiv:1706.02413.

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 10684–10695, 2022, arXiv:2112.10752.

[49] Pedro V Sander, Zoë J Wood, Steven Gortler, John Snyder, and Hugues Hoppe. Multi-chart geometry images. In *Eurographics Symposium on Geometry Processing*. Eurographics Association/Association for Computing Machinery, 2003, doi:10.5555/882370.882390.

[50] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. DiffusionNet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.*, 01(1), 2022, arXiv:2012.00888.

[51] Alla Sheffer, Emil Praun, and Kenneth Rose. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision*, 2(2):105–171, 2007, doi:10.1561/0600000011.

[52] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. MeshGPT: Generating triangle meshes with decoder-only transformers. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2024, arXiv:2311.15475.

[53] Yawar Siddiqui, Tom Monnier, Filippos Kokkinos, Mahendra Kariya, Yanir Kleiman, Emilien Garreau, Oran Gafni, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, and David Novotny. Meta 3D AssetGen: Text-to-mesh generation with high-quality geometry, texture, and PBR materials, 2024, arXiv:2407.02445.

[54] Ayan Sinha, Jing Bai, and Karthik Ramani. Deep learning 3D shape surfaces using geometry images. In *Proc. Euro. Conf. on Computer Vision*, 2016, doi:10.1007/978-3-319-46466-4_14.

[55] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. MVDiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *Advances in neural information processing systems*, 2023, arXiv:2307.01097.

[56] Shitao Tang, Jiacheng Chen, Dilin Wang, Chengzhou Tang, Fuyang Zhang, Yuchen Fan, Vikas Chandra, Yasutaka Furukawa, and Rakesh Ranjan. MVDiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3D object reconstruction. In *Proc. Euro. Conf. on Computer Vision*, 2024, arXiv:2402.12712.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017, arXiv:1706.03762.

[58] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in neural information processing systems*, 2021, arXiv:2106.10689.

[59] Yizhi Wang, Wallace Lira, Wenqi Wang, Ali Mahdavi-Amiri, and Hao Zhang. Slice3D: Multi-slice, occlusion-revealing, single view 3D reconstruction. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2024, arXiv:2312.02221.

[60] Francis Williams, Teseo Schneider, Cláudio T. Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 10122–10131, 2019, arXiv:1811.10943.

[61] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. *Advances in neural information processing systems*, 2016, arXiv:1610.07584.

[62] 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, arXiv:2404.07191.

[63] Xiang Xu, Joseph G Lambourne, Pradeep Kumar Jayaraman, Zhengqing Wang, Karl D. D. Willis, and Yasutaka Furukawa. BrepGen: A B-rep generative diffusion model with structured latent geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 2024, arXiv:2401.15563.

[64] 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*, pages 6239–6249, 2022, arXiv:2201.10326.

[65] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2018, arXiv:1712.07262.

[66] Lior Yariv, Omri Puny, Natalia Neverova, Oran Gafni, and Yaron Lipman. Mosaic-SDF for 3D generative models. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2024, arXiv:2312.09222.

[67] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: Latent point diffusion models for 3D shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, arXiv:2210.06978.

[68] Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. Paint3D: Paint anything 3D with lighting-less texture diffusion models. *ArXiv*, abs/2312.13913, 2023, arXiv:2312.13913.

[69] 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, arXiv:2205.13914.

[70] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023, arXiv:2301.11445.

[71] 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. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024, arXiv:2406.13897.

[72] Xin Zheng, Yang Liu, Peng-Shuai Wang, and Xin Tong. SDF-StyleGAN: Implicit SDF-based StyleGAN for 3D shape generation. *Computer Graphics Forum*, 41, 2022, arXiv:2206.12055.

[73] Xin Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional SDF diffusion for controllable 3D shape generation. *ACM Transactions on Graphics (TOG)*, 42:1 – 13, 2023, arXiv:2305.04461.

[74] Kun Zhou, John Synder, Baining Guo, and Heung-Yeung Shum. Iso-charts: stretch-driven mesh parameterization using spectral analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 45–54, 2004, doi:10.1145/1057432.1057439.