This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Efficient Autoregressive Shape Generation via Octree-Based Adaptive Tokenization

Kangle Deng[1,2]* Hsueh-Ti Derek Liu[2] Yiheng Zhu[2] Xiaoxia Sun[2] Chong Shang[2]
Kiran S. Bhat[2] Deva Ramanan[1] Jun-Yan Zhu[1] Maneesh Agrawala[2,3] Tinghui Zhou[2]
[1]Carnegie Mellon University [2]Roblox [3]Stanford University

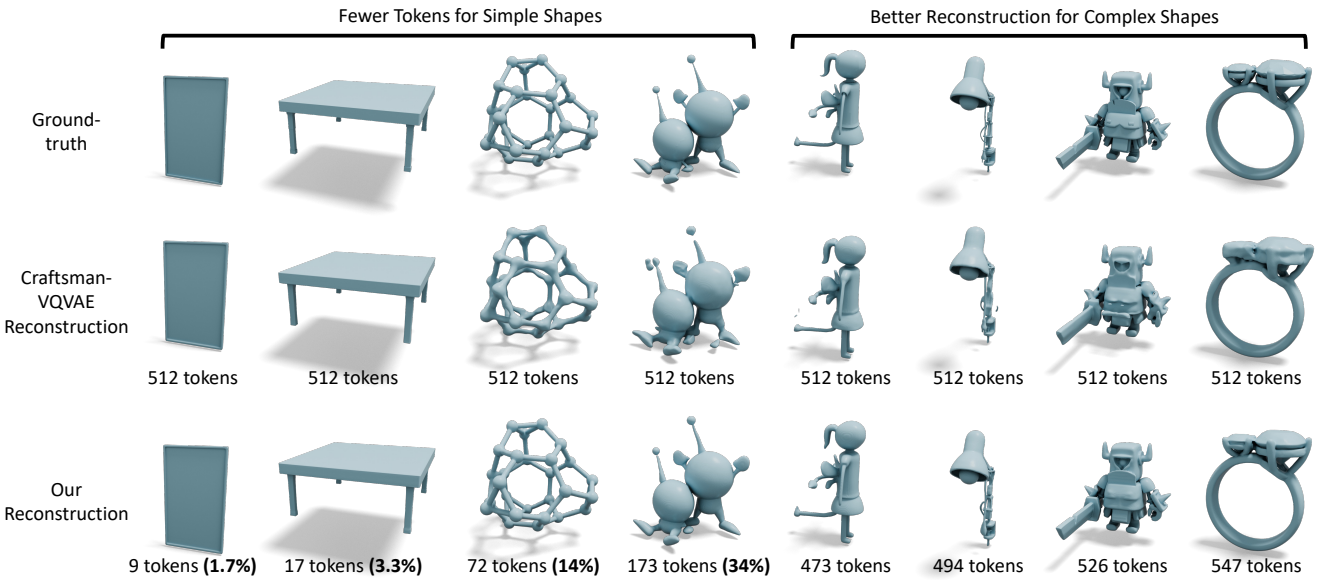

Figure 1. We propose an Octree-based Adaptive shape Tokenization (OAT) that dynamically allocates tokens based on shape complexity. Our approach achieves *better* reconstruction quality with *fewer* tokens on average (439 compared to 512 on the full test set) by intelligently distributing more tokens to complex shapes while saving on simpler ones.

Abstract

Many 3D generative models rely on variational autoencoders (VAEs) to learn compact shape representations. However, existing methods encode all shapes into a fixed-size token, disregarding the inherent variations in scale and complexity across 3D data. This leads to inefficient latent representations that can compromise downstream generation. We address this challenge by introducing Octree-based Adaptive Tokenization, a novel framework that adjusts the dimension of latent representations according to shape complexity. Our approach constructs an adaptive octree structure guided by a quadric-error-based subdivision criterion and allocates a shape latent vector to each octree cell using a query-based transformer. Building upon this tokenization, we develop an octree-based autoregressive generative model that effectively leverages these variable-sized representations in shape generation. Extensive experiments demonstrate that our approach reduces token counts by 50% compared to fixed-size methods while maintaining comparable visual quality. When using a similar token length, our method produces significantly higher-quality shapes. When incorporated with our downstream generative model, our method creates more detailed and diverse 3D content than existing approaches. Please check our project page: https://oat-3d.github.io/.

1. Introduction

Recent advances in generative models have revolutionized the field of 3D content creation, enabling diverse applications, including shape generation [29, 45, 71], text-to-3D generation [30, 52, 76], text-driven mesh texturing [4, 10],

*Work done when interning at Roblox.

single-image 3D generation [35, 38], and 3D scene editing [15, 36]. One popular paradigm among state-of-the-art methods employs 3D-native diffusion or autoregressive models [29, 55, 82, 89, 91] on top of 3D latents learned from large-scale datasets. As a result, the effectiveness of these models heavily depends on how well 3D shapes are represented and encoded as latent representations.

Effective latent representations for 3D shapes must address several fundamental challenges. First, 3D data is inherently sparse, with meaningful information concentrated primarily on surfaces rather than distributed throughout the volume. Second, real-world objects vary in geometric complexity, ranging from simple primitives to intricate structures with fine details, requiring representation structures that can adapt accordingly. Third, the encoding process must take into account capturing fine local details while preserving the global geometric structure.

Most existing shape VAEs [29, 87, 89, 91] encode shapes into fixed-size latent representations and fail to adapt to the inherent variations in geometric complexity within such shapes. As shown in Figure 1 (bottom), objects are encoded with identical latent capacity regardless of their scale, sparsity, or complexity, resulting in inefficient compression and degraded performance in downstream generative models. While some approaches [55, 82] leverage sparse voxel representations like octrees to account for sparsity, they still subdivide any cell containing surface geometry to the finest level, thus failing to adapt to shape complexity. As illustrated in Figure 2, a simple cube with only eight vertices requires similar representation capacity as a highly detailed sculpture in traditional octree structures. Ideally, hierarchical shape representations should adapt to the complexity of different regions within a shape. For instance, in the bottom right of Figure 2, complex structures like a tree canopy should require finer subdivision than simpler regions like the trunk.

To address these challenges, we propose an Octree-based Adaptive Tokenization. Our approach dynamically adjusts the latent representation based on local geometric complexity measured by quadric error, efficiently representing both simple and intricate regions with appropriate detail levels. As shown in Figure 1, our approach achieves better reconstruction quality with comparable or fewer shape tokens. By developing an Octree-based autoregressive generative model, we verify that our efficient variable-sized shape tokenization is beneficial to downstream generation tasks. Experiments show our generated results are generally better than those of existing baselines regarding FID, KID, and CLIP scores.

2. Related Work

3D Generation. Recent 3D generation methods have achieved remarkable results by leveraging pre-trained large-scale 2D diffusion models [58]. Approaches like DreamFusion [52] and DreamGaussian [66] use 2D diffusion priors

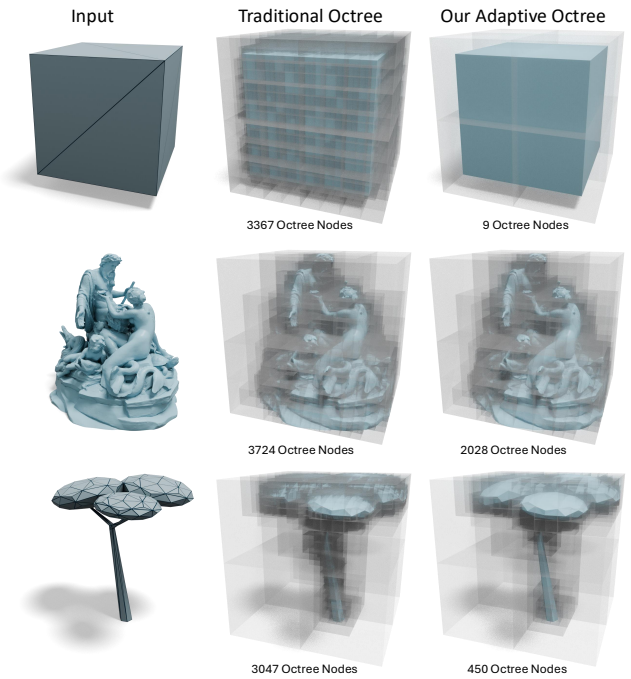

Figure 2. Traditional octree construction subdivides each octant based on whether the octant contains any mesh element. This construction always subdivides to the maximum depth (set to 6 in this example), leading to a similar amount of nodes for simple (top) and complex (middle) shapes. In contrast, our approach terminates subdivision when the local geometry is simple (e.g., a plane), leading to an adaptive octree that better reflects the shape complexity.

to optimize 3D representations, such as Neural Radiance Fields [48] and Gaussian Splats [23]. Subsequent works have improved performance with new loss functions and 3D representations [5, 22, 28, 30, 37, 40, 41, 46, 47, 62, 70, 76, 85]. However, these methods often require extensive iterative optimizations, making them impractical for real-world applications. To reduce inference time, feed-forward methods have been developed that synthesize multi-view consistent images of the same object followed by 3D reconstruction [17, 27, 31, 32, 34, 64, 69, 84, 88, 92]. Nonetheless, approaches leveraging 2D diffusion priors alone without 3D-native supervision tend to suffer from challenges in modeling refined geometric structures and complex surfaces, especially for shapes of high concavity.

More recently, a wave of 3D-native generative models [29, 33, 55, 81, 82, 89, 91] has emerged, aiming to train directly on raw 3D assets rather than relying on 2D diffusion priors. These methods have achieved superior generation quality compared to their predecessors thanks to the 3D-native architecture design. Another line of work explores auto-regressive methods for direct mesh generation with artist-like topology [7, 8, 14, 61, 65, 78]. Due to tokenization inefficiency and challenges in scaling up the context window, these methods are still struggling to model high-poly meshes with complex surfaces. In contrast, our work

aims to explore more efficient tokenization schemes that encode shapes into compact yet expressive representations for 3D-native generation.

Compact 3D latent representations. Representing 3D shapes with compact latent representations has become increasingly popular in 3D generative modeling. One line of work, pioneered by 3DShape2VecSet [50], advocates encoding 3D shapes into latent vector sets that can be decoded into diverse geometry representations such as occupancy fields [29, 50, 79, 89, 90], signed distance fields [6, 91], and meshes [65]. These methods encode all shapes into a fixed-length vector, and do not adaptively adjust the representation budget based on shape complexity. Other work [60, 80] learns latent space from triplanes, but achieving high-fidelity triplane representations remains challenging, which limits their accuracy, especially for complex shapes.

An alternative direction focuses on structured 3D latent representations to better leverage the spatial hierarchies inherent in the underlying geometry. For instance, sparse voxel grids coupled with feature-rich latents or attributes, as proposed in XCube [55], MeshFormer [33], LTS3D [44] and Trellis [81], enables more efficient training for high-resolution shapes and scenes and better preservation of high-frequency geometric details. Meanwhile, OctFusion [82] proposes to represent a 3D shape as a volumetric octree with each leaf node encoded by latent features. Although these approaches offer adaptiveness in the latent representation similar to ours, their spatial structure is determined by volumetric occupancy rather than surface complexity.

Octree-based 3D representation. Octree [42, 43] is an efficient 3D data structure that recursively divides a 3D space into eight octants. It adapts to sparsity and minimizes storage and computation in empty regions, making it both memory- and computationally efficient. Compared to dense voxel grids, octrees significantly reduce memory usage while preserving fine geometric details in complex regions. Octree has been used in a wide range of geometric processing applications, including point cloud compression [59], 3D texturing [2], multi-view scene reconstruction [63, 86], shape analysis [56, 73, 74], and shape generation [19, 67, 77, 82]. While similar adaptive octree [75] has been used for the shape classification and prediction tasks, our work is the first to explore octree representation in the context of 3D tokenization and autoregressive generation, which requires us to co-design the encoding, decoding, and generation with octree data structure. Compared to existing approaches [55, 81, 82] that use uniform tokenization schemes, our method adapts tokenization according to shape complexity.

3. Method

Figure 3 illustrates our text-based shape generation framework. Our approach comprises two components: (1) a shape tokenization method (Octree-based Adaptive Tokenization, OAT) in Section 3.1 and 3.2 that efficiently compresses 3D shapes into a compact latent space, and (2) an autoregressive generative backbone model, OctreeGPT in Section 3.3, which operates on these variable-length shape tokens.

For each 3D shape, our approach begins by sampling a point cloud $\mathbf{P}_c \in \mathbb{R}^{N\times 3}$ from the surface, along with its surface normal vectors $\mathbf{P}_n \in \mathbb{R}^{N\times 3}$ following prior work [87]. We then employ our novel adaptive octree construction algorithm that partitions 3D space based on *local geometric complexity* to obtain an sparse octree structure. We then leverage the Perceiver-based transformer architecture [20] to encode the shape into an adaptive latent tree structure. The resulting variable-length latent representation can then be decoded into an occupancy field, from which a mesh can be extracted using marching cubes [39].

Unlike existing shape VAEs [29, 87, 89, 91] that learn fixed-size latent representations for every shape using Variational Autoencoders [24], we propose to encode shapes into variable-length latents based on their shape complexity. This adaptive tokenization approach results in a more compact latent space that only uses more latents by subdividing cells to finer resolution where the complexity of the shape is higher — thereby leading to better reconstruction quality and improved performance in downstream generative tasks.

3.1. Complexity-Driven Octree Construction

One of the core ingredients of our method is a sparse octree data structure which subdivides octants according to local geometry complexity, unlike existing methods subdividing cells based on occupancy.

An octree is a hierarchical spatial data structure that recursively subdivides 3D space into eight equal octants. Starting with a root node representing a bounding cube, each non-empty node can be further partitioned into eight child nodes, creating a tree-like structure $\mathcal{O} = \{\mathcal{V}, \mathcal{E}\}$. We use $\mathcal{V} = \{v_1, v_2, \dots\}$ to denote the cells in an octree hierarchy, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ defines parent-child relationships, where $(v_i, v_j) \in \mathcal{E}$ indicates that v_i is the parent of v_j. This representation is efficient to represent sparse 3D data, as it allocates higher resolution only to occupied regions.

In this paper, we consider the *sparse* octree by omitting empty child nodes, i.e., each node can have 0 to 8 child nodes, with all nodes being non-empty. This structure can be compactly encoded by an 8-bit binary code $\chi : \mathcal{V} \rightarrow \{0, 1\}^8$. For instance, $\chi(v) = (01001000)_2$ indicates that node v has two non-empty child nodes at its second and fifth slots. An octree structure can thus be *uniquely* represented as a sequence of 8-bit binary codes in breadth-first order, $[\chi(v_0), \chi(v_1), \cdots]$.

While octrees have previously been used to tokenize 3D shapes, earlier methods [55, 82] always subdivide up to the maximum depth unless empty. In contrast, we subdivide an

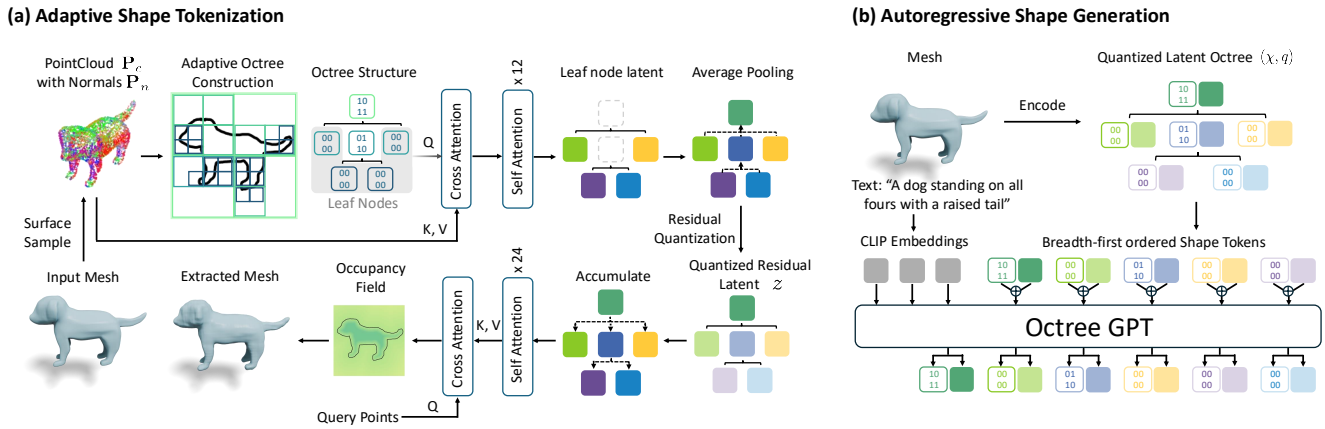

Figure 3. (a) **Adaptive Shape Tokenization.** Given an input mesh with surface point samples, we partition 3D space into a sparse octree that adapts to the local geometric complexity of the surface. We then use a Perceiver-based transformer [20] to encode the shape into a tree of latent codes, where a child node need encode only the (quantized) residual latent relative to its parent [26]. Latents can then be decoded into an occupancy field from which a mesh can be extracted. (b) **Autoregressive Shape Generation.** We define an autoregressive model for generating a tree of quantized shape tokens given a textual prompt, following a coarse-to-fine breadth-first search traversal. Similar to variable-length generation of text via end-of-sentence tokens, we make use of structural tokens to generate variable-size tree structures.

octant only when the local geometry is "complex". Inspired by the literature in mesh simplification [12] and isosurfacing [21], we use the quadric error metric to measure shape complexity and guide octree subdivision. This approach optimizes representational capacity, allocating tokens where they provide the greatest benefit for shape fidelity.

Quadric error metric was first introduced to quantify local geometric complexity for mesh simplification tasks [13]. Given a plane in $\mathbb{R}^3$, let $\mathbf{p}$ denote a point on the plane with unit normal vector $\mathbf{n}$. The plane can be defined by all points $\mathbf{x} \in \mathbb{R}^3$ satisfying

$$\mathbf{n}^\top(\mathbf{x} - \mathbf{p}) = 0. \tag{1}$$

The quadric error measures the squared point-to-plane distance between any point $\mathbf{x}$ and this plane, computed as

$$E(\mathbf{x}) = \left(\mathbf{n}^\top(\mathbf{x} - \mathbf{p})\right)^2 \doteq [\mathbf{x}^\top, 1]\, \mathbf{Q}\, [\mathbf{x}^\top, 1]^\top, \tag{2}$$

where the quadric matrix $\mathbf{Q} \in \mathbb{R}^{4\times 4}$ is defined as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{n}\mathbf{n}^\top & -\mathbf{n}\mathbf{n}^\top\mathbf{p} \\ (-\mathbf{n}\mathbf{n}^\top\mathbf{p})^\top & \mathbf{p}^\top\mathbf{n}\mathbf{n}^\top\mathbf{p} \end{bmatrix}. \tag{3}$$

As a key property, the cumulative error from a point $\mathbf{x}$ to *multiple* planes can be computed with a summed quadric,

$$E(\mathbf{x}) = \sum_i E_i(\mathbf{x}) = [\mathbf{x}^\top, 1] \left(\sum_i \mathbf{Q}_i\right) [\mathbf{x}^\top, 1]^\top. \tag{4}$$

We use the quadric error $E^* = \min_{\mathbf{x}} E(\mathbf{x})$ to measure local geometric complexity. As the energy is quadratic, the minimum E^* can be efficiently computed by solving a linear system, with details left in the appendix. Intuitively, when the planes form common intersections (e.g., an edge, a cone, or being flat), the optimal quadric error approaches zero, whereas complex regions usually yield higher quadric error values. This property makes quadric error metrics suitable for guiding adaptive geometric representations.

Specifically, for each octree cell $v \in \mathcal{V}$, we compute the cell quadric $\mathbf{Q}_v$ by summing up quadrics for all sampled points within v,

$$\mathbf{Q}_v = \sum_{\mathbf{p} \in \mathbf{P}_c(v)} \mathbf{Q}_{\mathbf{p}}, \tag{5}$$

where $\mathbf{P}_c(v) = \{\mathbf{p} \in \mathbf{P}_c \mid \mathbf{p} \text{ is contained in cell } v\}$ denotes the subset of points that lie within cell v, and $\mathbf{Q}_p$ is the quadric matrix for point $\mathbf{p}$ with its corresponding normal vector $\mathbf{n} \in \mathbf{P}_n$. We then calculate the average quadric error

$$E_v^* = \min_{\mathbf{x}} E_v(\mathbf{x}) = \frac{1}{|\mathbf{P}_c(v)|} \min_{\mathbf{x}} [\mathbf{x}^\top, 1]\mathbf{Q}_v[\mathbf{x}^\top, 1]^\top. \tag{6}$$

We recursively subdivide v into child cells only when both of these conditions are met: (1) the maximum depth L has not been reached, and (2) the quadric error exceeds our pre-defined threshold, $E_v^* > T$. In regions with complex geometry, cells are subdivided to the maximum depth L, while subdivision stops early in areas with simpler (i.e., planar) geometry.

3.2. Adaptive shape tokenization with OAT

Following prior work [29, 87, 89–91], we adopt a Perceiver-based variational autoencoder (VAE) [20, 24] to encode the

Algorithm 1 Multi-scale octree residual quantization

Input: Octree $\mathcal{O} = \{\mathcal{V}, \mathcal{E}\}$, Latent $\phi : \mathcal{V} \to \mathbb{R}^d$.
Output: Multi-scale residual quantized latent $z : \mathcal{V} \to \mathbb{R}^d$, Quantized latent index $q : \mathcal{V} \to \mathbb{Z}$.

```
z(v_0), q(v_0) = Quantize(φ(v_0))                ▷ v_0 is the root node.
z_acc(v_0) = z(v_0)                              ▷ Initialize accumulated latent.
for d = 1, ··· , L − 1 do                        ▷ L is the max depth of O.
    for v ∈ V_d do                               ▷ V_d is the set of nodes at level d.
        Find the parent v_parent of v according to E.
        z(v), q(v) = Quantize(φ(v) − z_acc(v_parent)).
        z_acc(v) = z_acc(v_parent) + z(v).       ▷ Update z_acc.
    end for
end for
```

shape into latents. Specifically, we compute:

$$\hat{\mathbf{P}} = \text{Concat}\left(\text{PE}(\mathbf{P}_c), \mathbf{P}_n\right), \tag{7}$$

$$\hat{\mathbf{O}} = \text{Concat}\left(\text{PE}(\mathcal{V}_{\text{leaf}}), \text{SE}(\mathcal{V}_{\text{leaf}})\right), \tag{8}$$

$$\phi(\mathcal{V}_{\text{leaf}}) = \text{SelfAttn}^{(i)}(\text{CrossAttn}(\hat{\mathbf{O}}, \hat{\mathbf{P}})),\ i = 1, \cdots, L_e,$$

where the encoder ϕ outputs a latent vector $\phi(v)$ for every leaf cell $v \in \mathcal{V}_{\text{leaf}}$, where $\phi : \mathcal{V} \to \mathbb{R}^d$. Here, PE denotes the positional encoding function [72], which operates on point coordinates and octree cell centers, while SE denotes the scale encoding function on the depth of octree cells. $\mathcal{V}_{\text{leaf}}$ comprises all the leaf cells within $\mathcal{V}$, and L_e refers to the number of Self Attention layers in the shape encoder.

Notably, the cross-attention operation is global, allowing each leaf cell to attend to all points in $\hat{\mathbf{P}}$ across the entire shape, rather than just points within its local cell. This global attention enables the model to capture long-range dependencies and contextual information beyond local neighborhoods. The subsequent self-attention layers further refine these representations by allowing leaf cells to exchange information.

Finally, we propagate latent vectors from leaf cells to their ancestors bottom-up. Each non-leaf node computes its latent vector by averaging those of its child nodes.

Multi-scale octree residual quantization. The variable length of the encoded latent motivates us to adopt an autoregressive model for downstream generation in Section 3.3. This approach requires us to learn a quantization bottleneck in the VAE. To achieve this, we propose an octree-based residual quantization strategy, enabling a coarse-to-fine token ordering using residual quantization [26, 68]. Specifically, we start quantization from the root node and only process the residual latent of every latent from its parent. We use a shared codebook and quantization function for all of the nodes using vqtorch [18]. We summarize our residual quantization algorithm in Algorithm 1.

Octree decoding. Given the multi-scale octree residual latent $z : \mathcal{V} \to \mathbb{R}^d$, we recover the full latent $\hat{\phi} : \mathcal{V} \to \mathbb{R}^d$ by adding the latent to every node from all its ancestors. Motivated by prior work [29, 87, 90], we use a similar perceiver-based transformer to decode the latent to an occupancy field. Specifically, given a query 3D point $\mathbf{x} \in \mathbb{R}^3$, the decoder predicts its occupancy value:

$$\hat{\mathbf{S}} = \text{Concat}\left(\hat{\phi}(\mathcal{V}), \text{PE}(\mathcal{V}), \text{SE}(\mathcal{V})\right), \tag{9}$$

$$\hat{\mathbf{S}} = \text{SelfAttn}^{(j)}(\hat{\mathbf{S}}), \qquad j = 1, 2, \cdots, L_d, \tag{10}$$

$$\hat{\sigma}(\mathbf{x}, \hat{\phi}, \mathcal{O}) = \text{CrossAttn}\left(\text{PE}(\mathbf{x}), \hat{\mathbf{S}}\right), \tag{11}$$

where L_d is the number of Self Attention layers in the shape decoder, and $\hat{\sigma}$ is the predicted occupancy value at the query point. At inference time, we query the decoder using grid points and run marching cubes [39] to extract a mesh. During training, we sample query points using uniform and importance sampling near the mesh surface following prior work [29, 87, 90].We jointly optimize the networks and codebook via the following loss functions.

$$\mathcal{L}_{\text{VQ}} = \mathbb{E}_{v\in}||\text{sg}(\hat{\phi}(v)) - \phi(v)||^2 + ||\text{sg}(\phi(v)) - \hat{\phi}(v)||^2, \tag{12}$$

where sg() is the stop-gradient operation. Additionally, we incorporate an occupancy reconstruction loss to ensure that the latent codes accurately reconstruct the input shape:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathbf{x}}\mathcal{L}_{\text{BCE}}\left(\sigma(\mathbf{x}), \hat{\sigma}(\mathbf{x}, \hat{\phi}, \mathcal{O})\right), \tag{13}$$

where $\mathcal{L}_{\text{BCE}}$ is the binary cross-entropy loss for shape reconstruction, and $\sigma(\mathbf{x}) \in \{0, 1\}$ is the ground truth occupancy value of the query point, indicating whether it is located inside the object. Our final loss function is $\mathcal{L}_{\text{rec}} + \lambda_{\text{VQ}}\mathcal{L}_{\text{VQ}}$, where λ_{VQ} weights the vector quantization loss.

KL variant for continuous tokens. By replacing the quantization bottleneck with a KL regularization [24], our proposed OAT can learn continuous shape latent instead, which provides a fair comparison with other continuous latent baselines in Section 4.1.

3.3. OctreeGPT: Autoregressive Shape Generation

Building on our adaptive tokenization framework, we develop OctreeGPT, an autoregressive model for generating 3D shapes conditioned on text descriptions. Unlike previous approaches that operate on fixed-size representations [29, 89, 91], OctreeGPT models the joint distribution of variable-length octree tokens while maintaining a hierarchical coarse-to-fine structure.

Shape Token Sequence. To enable autoregressive modeling, we first serialize the octree structure by traversing it in a breadth-first manner as mentioned in Section 3.2. For each node v, we include both its quantized index $q(v) \in \mathbb{Z}$ and a structural code $\chi(v) \in \{0, 1\}^8$ that encodes the

presence or absence of each potential child node. A latent octree can thus be *uniquely* represented by a variable-length sequence of tokens: $[t_0, t_1, \cdots, t_N]$, where each token $t_i = (q(v_i), \chi(v_i)), \forall i \in \mathbb{N}$.

We train an autoregressive model that predicts the next token in the sequence,

$$P(t_0, t_1, \cdots, t_N | \theta) = \prod_{i=1}^{N} P(t_i | t_0, \cdots, t_{i-1}, \theta), \quad (14)$$

where θ is our learned OctreeGPT model.

Model Architecture. Our architecture builds upon decoder-only transformers similar to GPT-2 [11, 53]. Specifically, we compute the embedding for each shape token t_i as:

$$\text{Embed}(t_i) = \text{Embed}_q(q(v_i)) + \text{Embed}_\chi(\chi(v_i)) + \text{PE}_{\text{tree}}(v_i), \quad (15)$$

where $\chi(v_i)$ is interpreted as an 8-bit integer. The tree-structured positional encoding $\text{PE}_{\text{tree}}(v_i)$ captures both spatial and hierarchical information:

$$\text{PE}_{\text{tree}}(v_i) = \text{Embed}_x(x(v_i)) + \text{Embed}_y(y(v_i)) \quad (16)$$
$$+ \text{Embed}_z(z(v_i)) + \text{Embed}_d(d(v_i)), \quad (17)$$

where x, y, z are quantized coordinates of the cell center, and $d \in \{0, 1, \cdots, L-1\}$ is the depth of the octree node. This multi-dimensional positional encoding helps the model understand both spatial relationships and the hierarchical structure of the octree. Our model employs dual prediction heads for predicting quantized latent indices $\hat{q}$ and structural codes $\hat{\chi}$, allowing the model to jointly reason about geometry and tree structure. For text-conditioned generation, we prepend the sequence with 77 tokens derived from the input text's CLIP embedding [54].

Training and Inference. We train OctreeGPT using a combined loss function that balances the reconstruction of latent tokens and structural codes:

$$\mathcal{L}_{\text{GPT}} = \mathcal{L}_{\text{CE}}(q, \hat{q}) + \lambda_\chi \mathcal{L}_{\text{CE}}(\chi, \hat{\chi}), \quad (18)$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss for 2^8-way classification, and λ_χ are balancing hyperparameters. During inference, we employ sampling with temperature τ to control the diversity and quality of generated shapes. We process the predicted structural code $\chi(v_i)$ on the fly to determine the octree topology, which dynamically establishes the final length of the token sequence. Further implementation details and hyperparameter settings are provided in the appendix.

4. Experiments

We evaluate our method on shape tokenization and generation. We perform qualitative and quantitative comparisons with existing baselines and conduct an ablation study on the significance of each major component.

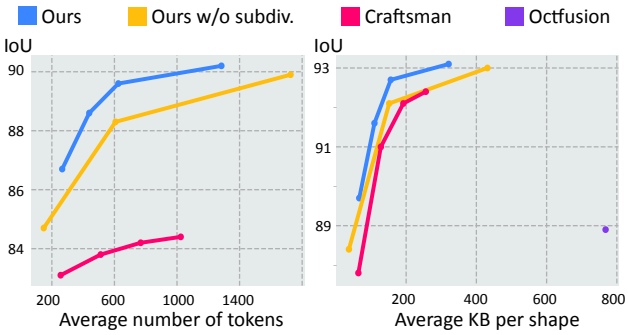

Figure 4. We plot reconstruction quality (IoU) against latent size in both discrete (left) and continuous (right) scenarios. We use KiloBytes (KB) for continuous latent representations for a fair comparison. Our method consistently outperforms baseline approaches at equivalent latent sizes and achieves comparable reconstruction quality with much smaller latent representations.

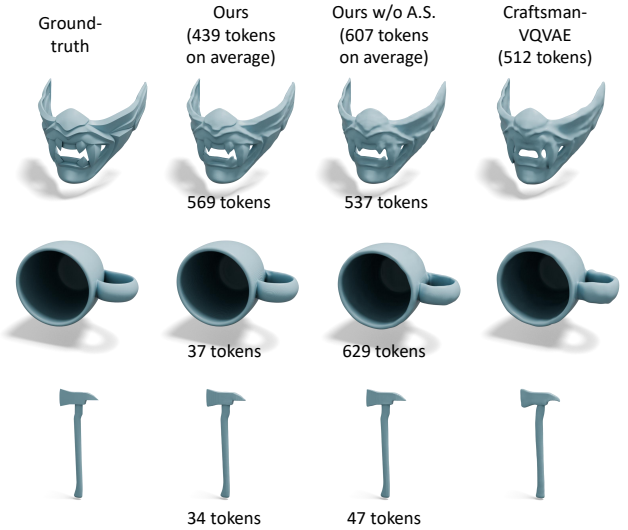

Figure 5. **Shape reconstruction with discrete latent.** We compare our full method against Craftsman-VQ [29] as well as an ablation without Adaptive Subdivision (A.S.). With comparable or lower token budget, our method generally outperforms the baseline regarding reconstruction fidelity. Meanwhile, without adaptive subdivision, the vanilla octree only allocates the token budget efficiently for objects of small volume (bottom) but wastes tokens on geometrically simple objects that occupy large space (middle).

Dataset. We use the Objaverse [9] dataset, which contains around 800K 3D models, as our training and test data. To ensure high-quality training and evaluation, we filter out low-quality meshes, such as those with point clouds, thin structures, or holes. This results in a curated dataset of around 207K objects for training and 22K objects for testing.

For preprocessing, each mesh is normalized to a unit cube. For each mesh, we sample 1M points with their normals from the surface as the input point cloud. To generate ground-truth occupancy values, we uniformly sample 500K points within the unit volume and an additional 1M points near the mesh surface to capture fine details and obtain the occupancy based on visibility following prior work [57]. We then construct an

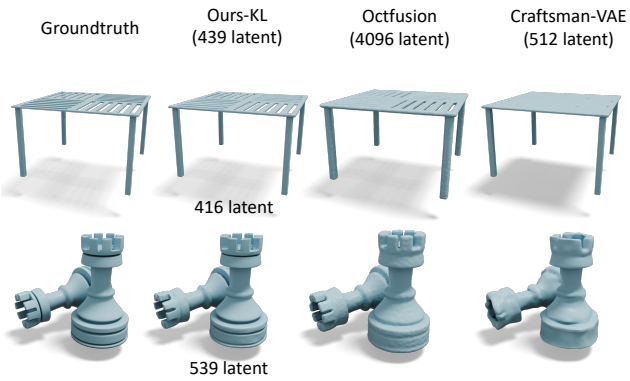

Figure 6. **Shape reconstruction with continuous latent.** We include the visual comparison between our continuous VAE (OAT-KL) and other baselines. In general, our reconstruction preserves more details using similar or smaller number of latent vectors.

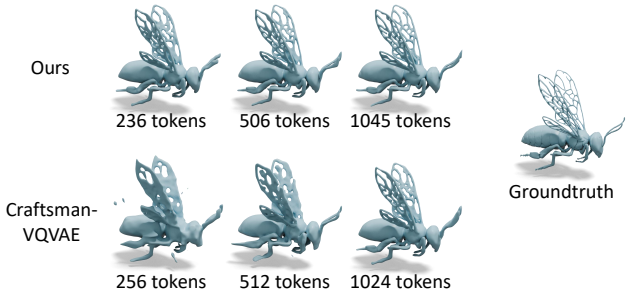

Figure 7. **Ablation study on token length.** With an increasing number of tokens, our method achieves better quality while consistently outperforming the baseline at a comparable token length.

adaptive octree for each shape based on the sampled point cloud using a pre-defined quadric error threshold T, which guides the subdivision process according to local geometric complexity. To enable text conditioning, we render nine views of each object under random rotations and use GPT-4o [1] to generate descriptive captions from these renderings.

4.1. Shape Reconstruction

Baselines. We compare OAT with latent vector sets from Craftsman3D [29]. For a fair comparison, we train both methods under identical conditions, using both quantization for discrete tokenization and KL regularization for continuous latent space. Additionally, we evaluate against two other recent approaches, XCube [55] and Octfusion [82]. Due to computational resource constraints, we use publicly available pre-trained models for these two baselines rather than retraining them on our dataset. We exclude VAE models from Direct3D [79], CLAY [89], and LTS3D [44] as their implementations are not available.

Results. We evaluate shape reconstruction quality using volume Intersection over Union (IoU) and Chamfer Distance (CD) with 10K sampled surface points in Table 1 and Table 2. Note that XCube [55] outputs an Unsigned Distance Function (UDF), which cannot be evaluated with IoU metrics.

Method	Avg Token Cnt	IoU ↑	CD ($\times 10^{-3}$) ↓
Craftsman-VQ [29]	256	83.1	2.31
	512	83.8	1.94
	768	84.2	1.88
	1024	84.4	1.80
Ours (OAT) w/o A.S.	148	84.7	2.19
	607	88.3	1.85
	1726	89.9	1.37
Ours (OAT)	266	86.7	1.94
	439	88.6	1.78
	625	89.7	1.53
	1284	**90.2**	**1.27**

Table 1. **Quantitative analysis of shape reconstruction with discrete latent.** We compare our method against Craftsman-VQ [29] and ablation without Adaptive Subdivision (A.S.). With comparable token counts, our approach outperforms both baselines, showing the effectiveness of our proposed adaptive tokenization.

Method	Avg Latent Len	IoU ↑	CD ($\times 10^{-3}$) ↓
Craftsman [29]	256	87.8	1.96
	512	91.0	1.83
	768	92.1	1.33
	1024	92.4	1.29
Octfusion† [82]	4096	88.9	1.87
XCube† [55]	4096	-	1.26
Ours (OAT-KL) w/o A.S.	148	88.4	1.89
	607	92.1	1.29
	1726	93.0	1.01
Ours (OAT-KL)	266	89.7	1.81
	439	91.6	1.29
	625	92.7	1.08
	1284	**93.1**	**0.97**

Table 2. **Quantitative analysis of shape reconstruction with continuous latent.** We replace the quantization with a KL regularization to learn continuous latent (OAT-KL) as mentioned in Section 3.2. Our method outperforms all the baselines with comparable or shorter latent code lengths. † indicates off-the-shelf models that are pre-trained on different data sources than ours.

Visual comparisons in Figure 5 and Figure 6 demonstrate our approach outperforms all baselines.

Ablation Study. We ablate our proposed adaptive subdivision in Figure 5. Without quadric-error-based adaptive subdivision, the octree representation subdivides to the deepest level unless empty, wasting tokens on simple objects of large volumetric occupancy (middle row). Figure 4 shows reconstruction quality (IoU) versus latent size in both discrete and continuous scenarios, confirming our method achieves better quality at equivalent latent sizes and requires significantly smaller latent representations for comparable reconstruction quality. Figure 7 further shows a qualitative comparison be-

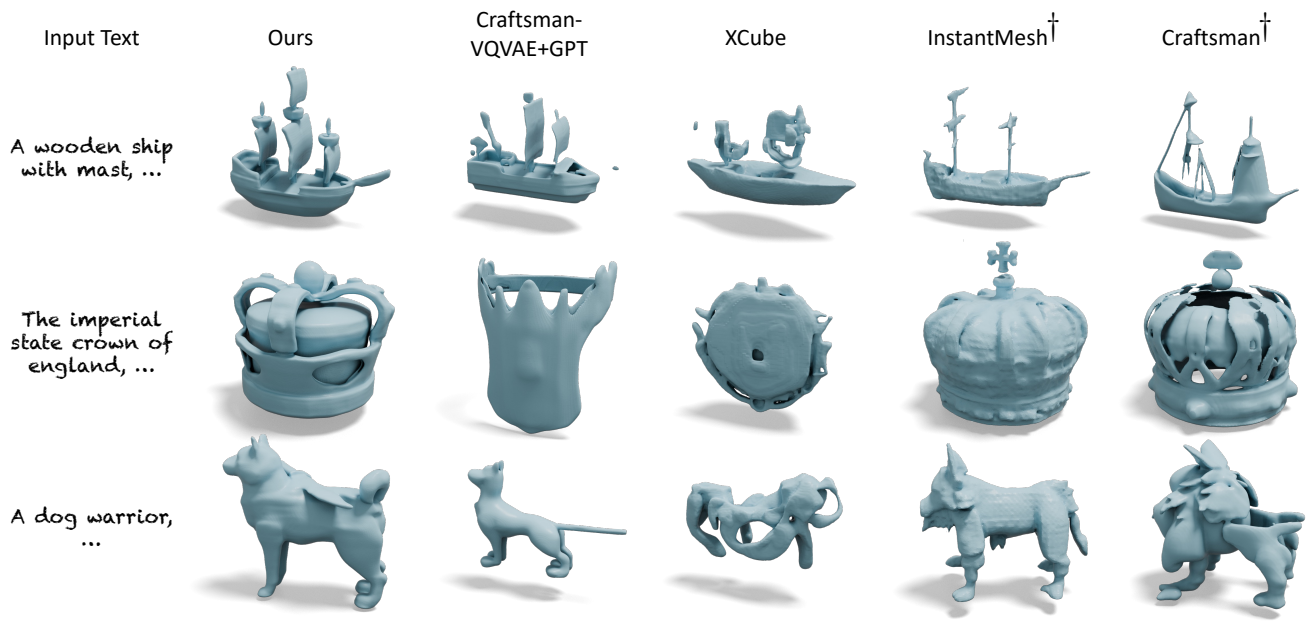

Figure 8. **Shape Generation Results.** We compare OctreeGPT with a GPT baseline trained on Craftsman-VQVAE (Section 4.1), text-to-3D model XCube [55], and image-to-3D methods InstantMesh [83] and Craftsman [29]. Our results have smoother surfaces, finer details, and fewer artifacts than baselines. For image-conditioned methods†, we use FLUX.1 [25] to generate condition images from input text.

Method	FID↓	KID↓ ($\times 10^{-3}$)	CLIP-score↑	Runtime↓ (secs)
Craftsman† [29]	65.18	6.42	0.27	54.8
InstantMesh† [49]	67.93	7.23	0.31	21.5
XCube [55]	132.56	9.83	0.23	32.3
Craftsman-VQ + GPT	85.10	7.49	0.26	15.4
Ours (OctreeGPT)	**56.88**	**5.79**	**0.34**	**11.3**

Table 3. **Quantitative analysis of shape generation.** We compare OctreeGPT with a GPT baseline trained on Craftsman-VQ (Section 4.1), XCube [55], and image-to-3D methods InstantMesh [83] and Craftsman [29]. We compute FID [16], KID [3], and CLIP-score on the renderings of generated shapes, and report the average generation time. Our method outperforms all the baselines, showing higher quality and better consistency with the input text while achieving the fastest runtime due to our efficient tokenization.

tween our method and the baseline in reconstruction quality with respect to the number of tokens used.

4.2. Shape Generation

This section evaluates our text-to-shape generation quality against multiple baselines. We train our OctreeGPT on top of OAT using 439 tokens on average, and for comparison, train a GPT model on Craftsman-VQ with 512 tokens. We include XCube [55]'s pre-trained Objaverse model as a native text-to-3D baseline. We also compare against two image-to-3D methods, InstantMesh [83] and Craftsman [29], using FLUX.1 [25] to generate condition images from input text.

Results. We quantitatively evaluate generation quality in Table 3 by rendering generated shapes and computing FID [16, 51] and KID [3] against groundtruth renderings. We also report CLIP-score [54] to evaluate text-shape consistency, and average generation time to evaluate efficiency. In addition, we also provide qualitative comparisons in Figure 8. Overall, thanks to a more compact and representative latent space, our OctreeGPT produces finer details with fewer artifacts compared to Craftsman-VQ with GPT, while also outperforming other 3D generation baselines in both geometry quality and prompt adherence, with a faster runtime.

5. Discussion

In this work, we propose an octree-based Adaptive Shape Tokenization, OAT, a framework that dynamically adjusts latent representations according to shape complexity. At its core, OAT constructs an adaptive octree structure guided by a quadric-error-based subdivision criterion, allocating more tokens to complicated parts and objects while saving on simpler ones. Extensive experiments show that OAT reduces token counts by 50% compared to previous fixed-size approaches while maintaining comparable visual quality. Alternatively, with a similar number of tokens, OAT produces much higher-quality shapes. Building upon this tokenization, we develop an octree-based Autoregressive generative model, OctreeGPT that effectively leverages these variable-sized representations, outperforming existing baselines.

Limitations. Our framework only addresses geometric shape reconstruction and generation without incorporating texture information. We leave modeling both shape and texture properties jointly for future work.

Acknowledgement

We thank Akash Garg, Daiqing Li, Alexander Weiss, Alejandro Peláez, Kayvon Fatahalian, Sheng-Yu Wang, Gaurav Parmer, Ruihan Gao, Nupur Kumari, and Maxwell Jones for their discussion and help. This work was done when KD was an intern at Roblox. The project is partly supported by Roblox. JYZ is partly supported by the Packard Fellowship. The Microsoft Research PhD Fellowship supports KD.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 7

[2] David Benson and Joel Davis. Octree textures. *ACM Transactions on Graphics (TOG)*, 21(3):785–790, 2002. 3

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations (ICLR)*, 2018. 8

[4] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 1

[5] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 2

[6] Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[7] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 2

[8] Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. *arXiv preprint arXiv:2408.02555*, 2024. 2

[9] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6

[10] Kangle Deng, Timothy Omernick, Alexander Weiss, Deva Ramanan, Jun-Yan Zhu, Tinghui Zhou, and Maneesh Agrawala. Flashtex: Fast relightable mesh texturing with lightcontrolnet. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[11] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[12] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *ACM SIGGRAPH*, 1997. 4

[13] Michael Garland and Yuan Zhou. Quadric-based simplification in any dimension. In *ACM Transactions on Graphics (TOG)*, 2005. 4

[14] Zekun Hao, David W Romero, Tsung-Yi Lin, and Ming-Yu Liu. Meshtron: High-fidelity, artist-like 3d mesh generation at scale. *arXiv preprint arXiv:2412.09548*, 2024. 2

[15] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 2

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 8

[17] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *International Conference on Learning Representations (ICLR)*, 2024. 2

[18] Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *International Conference on Machine Learning (ICML)*, 2023. 5

[19] Moritz Ibing, Gregor Kobsik, and Leif Kobbelt. Octree transformer: Autoregressive 3d shape generation on hierarchically structured sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[20] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, 2021. 3, 4

[21] Tao Ju, Frank Losasso, Scott Schaefer, and Joe D. Warren. Dual contouring of hermite data. In *ACM Transactions on Graphics (TOG)*, 2002. 4

[22] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. In *International Conference on Learning Representations (ICLR)*, 2024. 2

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM Transactions on Graphics (TOG)*, 2023. 2

[24] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013. 3, 4, 5

[25] Black Forest Labs. Flux. `https://huggingface.co/black-forest-labs/FLUX.1-schnell`, 2024. 8

[26] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4, 5

[27] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *In-*

ternational Conference on Learning Representations (ICLR), 2024. 2

[28] Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. *arxiv:2310.02596*, 2023. 2

[29] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2, 3, 4, 5, 6, 7, 8

[30] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[31] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2

[32] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[33] Minghua Liu, Chong Zeng, Xinyue Wei, Ruoxi Shi, Linghao Chen, Chao Xu, Mengqi Zhang, Zhaoning Wang, Xiaoshuai Zhang, Isabella Liu, et al. Meshformer: High-quality mesh generation with 3d-guided reconstruction model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 2, 3

[34] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 2

[35] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 2

[36] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[37] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[38] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[39] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, 1998. 3, 5

[40] Artem Lukoianov, Haitz Sáez de Ocáriz Borde, Kristjan Greenewald, Vitor Campagnolo Guizilini, Timur Bagautdinov, Vincent Sitzmann, and Justin Solomon. Score distillation via reparametrized ddim. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[41] David McAllister, Songwei Ge, Jia-Bin Huang, David W. Jacobs, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Rethinking score distillation as a bridge between image distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[42] Donald Meagher. Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2):129–147, 1982. 3

[43] Donald JR Meagher. *Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer*. Electrical and Systems Engineering Department Rensseiaer Polytechnic . . . , 1980. 3

[44] Quan Meng, Lei Li, Matthias Nießner, and Angela Dai. Lt3sd: Latent trees for 3d scene diffusion. *arXiv preprint arXiv:2409.08215*, 2024. 3, 7

[45] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[46] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[47] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[48] 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. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[49] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. In *ACM SIGGRAPH*, 2022. 8

[50] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M. Seitz. Photoshape: Photorealistic materials for large-scale shape collections. 2018. 3

[51] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

[52] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 2

[53] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 6

[54] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 6, 8

[55] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 7, 8

[56] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[57] Foundation AI Team Roblox. Cube: A roblox view of 3d intelligence. *arXiv preprint arXiv:2503.15475*, 2025. 6

[58] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[59] Ruwen Schnabel and Reinhard Klein. Octree-based point-cloud compression. *PBG@ SIGGRAPH*, 3(3), 2006. 3

[60] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[61] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[62] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior, 2023. 2

[63] Richard Szeliski. Rapid octree construction from image sequences. *CVGIP: Image understanding*, 58(1):23–32, 1993. 3

[64] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[65] Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. Edgerunner: Autoregressive auto-encoder for artistic mesh generation. *arXiv preprint arXiv:2409.18114*, 2024. 2, 3

[66] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations (ICLR)*, 2024. 2

[67] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[68] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 5

[69] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, , Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 2

[70] Uy Dieu Tran, Minh Luu, Phong Ha Nguyen, Khoi Nguyen, and Binh-Son Hua. Diverse text-to-3d synthesis with augmented text embedding. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[71] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 5

[73] Peng-Shuai Wang. Octformer: Octree-based transformers for 3D point clouds. In *ACM Transactions on Graphics (TOG)*, 2023. 3

[74] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. In *ACM Transactions on Graphics (TOG)*, 2017. 3

[75] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive o-cnn: A patch-based deep representation of 3d shapes. In *ACM Transactions on Graphics (TOG)*, 2018. 3

[76] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2

[77] Si-Tong Wei, Rui-Huan Wang, Chuan-Zhi Zhou, Baoquan Chen, and Peng-Shuai Wang. Octgpt: Octree-based multiscale autoregressive models for 3d shape generation. In *ACM SIGGRAPH*, 2025. 3

[78] Haohan Weng, Zibo Zhao, Biwen Lei, Xianghui Yang, Jian Liu, Zeqiang Lai, Zhuo Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, et al. Scaling mesh generation via compressive tokenization. *arXiv preprint arXiv:2411.07025*, 2024. 2

[79] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *arXiv preprint arXiv:2405.14832*, 2024. 3, 7

[80] Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, et al. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. In *ACM Transactions on Graphics (TOG)*, 2024. 3

[81] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3

[82] Bojun Xiong, Si-Tong Wei, Xin-Yang Zheng, Yan-Pei Cao,

Zhouhui Lian, and Peng-Shuai Wang. Octfusion: Octree-based diffusion models for 3d shape generation. *arXiv preprint arXiv:2408.14732*, 2024. 2, 3, 7

[83] 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. 8

[84] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. In *International Conference on Learning Representations (ICLR)*, 2024. 2

[85] Junliang Ye, Fangfu Liu, Qixiu Li, Zhengyi Wang, Yikai Wang, Xinzhou Wang, Yueqi Duan, and Jun Zhu. Dreamreward: Text-to-3d generation with human preference. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[86] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 3

[87] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. In *ACM Transactions on Graphics (TOG)*, 2023. 2, 3, 4, 5

[88] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[89] 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. In *ACM Transactions on Graphics (TOG)*, 2024. 2, 3, 4, 5, 7

[90] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3, 5

[91] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 2, 3, 4, 5

[92] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2