PrimitiveAnything: Human-Crafted 3D Primitive Assembly Generation with Auto-Regressive Transformer

JINGWEN YE*, Tencent AIPD, China

YUZE HE*, Tsinghua University and Tencent AIPD, China

YANNING ZHOU†, YIQIN ZHU, and KAIWEN XIAO, Tencent AIPD, China

YONG-JIN LIU†, Tsinghua University, China

WEI YANG and XIAO HAN†, Tencent AIPD, China

Fig. 1. 3D primitive assemblies created by PrimitiveAnything span diverse shape categories, enabling versatile primitive-based 3D content creation.

Shape primitive abstraction, which decomposes complex 3D shapes into simple geometric elements, plays a crucial role in human visual cognition and has broad applications in computer vision and graphics. While recent advances in 3D content generation have shown remarkable progress, existing primitive abstraction methods either rely on geometric optimization with limited semantic understanding or learn from small-scale, category-specific datasets, struggling to generalize across diverse shape categories. We present PrimitiveAnything, a novel framework that reformulates shape primitive abstraction as a primitive assembly generation task. PrimitiveAnything includes a shape-conditioned primitive transformer for auto-regressive generation and an ambiguity-free parameterization scheme to represent multiple types of primitives in a unified manner. The proposed framework directly learns the process of primitive assembly from large-scale human-crafted abstractions, enabling it to capture how humans decompose complex shapes into primitive elements. Through extensive experiments, we demonstrate that PrimitiveAnything can generate high-quality primitive assemblies that better align with human perception while maintaining geometric fidelity across diverse shape categories. It benefits various 3D applications and
shows potential for enabling primitive-based user-generated content (UGC) in games. *Project page: https://primitiveanything.github.io*

1 INTRODUCTION

Understanding and representing 3D environments and objects has been a fundamental task in computer vision and graphics. Recent years have witnessed significant breakthroughs in 3D understanding and generation, with various representations including meshes [Chen et al. 2024a,b; Siddiqui et al. 2024], point clouds [Nichol et al. 2022; Vahdat et al. 2022], and neural fields [Hong et al. 2024; Jun and Nichol 2023; Poole et al. 2023] enabling rapid generation of high-quality 3D contents. However, these representations, while effective for visualization and rendering, often lack the semantic structure and interpretability that align with human cognitive processes. Cognitive science research has long established that the human visual system possesses a remarkable ability to decompose complex visual scenes into simple geometric primitives - a process known as perceptual organization or shape abstraction [Biederman 1985, 2005]. This cognitive mechanism not only enables efficient visual processing and understanding but also facilitates our ability to reason about object structure, function, and physical interactions.

Inspired by this human cognitive capability, the task of shape primitive abstraction and generation seeks to develop computational

methods that can similarly decompose complex 3D shapes into interpretable primitive elements [Binford 1975; Roberts 1963]. This ability is not just theoretically interesting - it enables crucial applications in robotic manipulation, scene understanding, computer-aided design, and interactive modeling systems, where high-level structural understanding is essential for downstream tasks.

The development of primitive-based shape abstraction involves two fundamental design choices: the definition of primitive types to be used, and the computational approach to extract these primitives from raw 3D data. The selection of primitive types has evolved significantly over the past years. Early approaches primarily focused on simple geometric primitives such as cuboids [Gupta et al. 2010; Tulsiani et al. 2017], geons [Biederman 1985] and cylinders [Binford 1975], which offer computational simplicity and intuitive interpretation but limited expressiveness. Later work [Paschalidou et al. 2019; Pentland 1986] introduced superquadrics, which can represent a broader range of smooth shapes through parametric equations that generalize quadric surfaces, providing a better balance between representational power and computational tractability.

For the extraction of primitives, technical approaches can be broadly categorized into two main streams: optimization-based methods and learning-based methods. The former one formulates primitive detection as a geometric fitting problem, attempting to minimize various distance metrics between the primitive representations and input geometry [Chevalier et al. 2003; Leonardis et al. 1997; Liu et al. 2022]. These methods, while mathematically principled and interpretable, primarily focus on minimizing geometric surface distance between the original shape and primitive assemblies, with limited consideration of human abstraction logic. This often results in over-segmentation of semantic parts and fails to capture meaningful structural decomposition of shapes. The latter approaches aim to learn primitive decomposition directly from data [Huang et al. 2023; Paschalidou et al. 2019; Tulsiani et al. 2017; Zou et al. 2017]. However, these learning-based methods are typically trained on small-scale, category-specific datasets, leading to limited generalization capabilities across object categories. How to effectively parameterize primitives and learn generalizable abstraction concepts across diverse categories remains an open challenge.

Recent advances in 3D content generation [Chen et al. 2024b; Hong et al. 2024; Nichol et al. 2022; Zhang et al. 2024c] have demonstrated the remarkable potential of directly learning the 3D representation from the large-scale 3D datasets, e.g. Objaverse [Deitke et al. 2023]. MeshAnything [Chen et al. 2024a,b]'s success in using an autoregressive transformer to generate human-crafted meshes that capture both geometric details and artistic intent. Drawing on this insight, we reformulate primitive abstraction as a generation task, moving away from traditional geometric fitting or direct regression approaches. Unlike previous methods that rely on hand-crafted optimization objectives or direct regression of primitive parameters, our generation-based framework learns to sequentially construct primitive abstractions in a manner similar to how humans might build up complex shapes from simple components. This fundamentally different approach allows our method to better capture the hierarchical and semantic nature of shape decomposition while maintaining geometric accuracy.

The overall design follows two core concepts: First, the *primitive representation* must achieve **high geometric fidelity while compact enough** for efficient learning. To this end, we utilize multiple types of primitives to jointly represent 3D shapes under a unified parameterization scheme. To address the inherent ambiguity in such parameterization and ensure stable training, we develop a comprehensive set of rules that uniquely define the parameter ordering and relationships between atomic elements, resulting in well-structured sequences suitable for learning. Second, the *learning framework* must possess **strong capacity** to handle complex shapes with varying numbers of primitives while remaining **primitive-agnostic** for extensibility. We address this through a shape-conditioned decoder-only transformer architecture that can generate variable-length primitive sequences. The framework's modular design treats primitive types as learnable tokens, enabling seamless integration of new primitive types without architectural changes, making it adaptable to different primitive representations.

Our main contribution can be summarized as follows: 1) We propose **PrimitiveAnything**, a novel primitive generation framework that reformulates shape abstraction as a sequence generation task, enabling the model to learn from and reproduce human-crafted shape decompositions. 2) We extend the single primitive representation to multiple primitives and design an ambiguity-free parameterization scheme, achieving high geometric fidelity while maintaining computational efficiency for learning. 3) PrimitiveAnything contains a shape-conditioned decoder-only transformer architecture that can handle variable-length primitive sequences and is easily extensible to new primitive types. 4) We demonstrate through extensive experiments that our method can generate high-quality primitive abstractions that better align with human perception compared to existing approaches, while maintaining geometric fidelity to the original shapes.

2 RELATED WORKS

2.1 3D Content Generation

Recent years have witnessed remarkable progress in 3D content generation, spanning diverse tasks from object generation [Dong et al. 2024; Petrov et al. 2024; Zhang et al. 2024c; Zhao et al. 2023] to texture synthesis [Bensadoun et al. 2024; Guerrero-Viu et al. 2024; Hu et al. 2024; Yu et al. 2024; Zhang et al. 2024b]. DreamFusion [Poole et al. 2023] and SJC [Wang et al. 2023a] pioneered the lifting of 2D diffusion models to 3D generation by optimizing neural radiance fields through score distillation sampling, with subsequent works like Magic3D [Lin et al. 2023] and VSD [Wang et al. 2023b] further refining this approach. The field has seen a shift towards data-driven large reconstruction models, starting with LRM [Hong et al. 2024] which leveraged transformers to generate triplane features from single images. This approach has been extended to handle multi-view inputs [Li et al. 2024c; Wang et al. 2024b; Xu et al. 2024] and more efficient 3D representations [Tochilkin et al. 2024; Yang et al. 2024; Zhang et al. 2024a], significantly improving generation fidelity. Concurrent development of native 3D shape generation models has also shown promising results [Hui et al. 2024; Li et al. 2024a; Zhang et al. 2024d; Zhao et al. 2023]. Notably, CLAY [Zhang et al. 2024c] introduced a two-stage approach combining a multi-resolution 3D

shape VAE (extended from 3DShape2VecSet [Zhang et al. 2023]) with a DiT-based diffusion model [Peebles and Xie 2022] for high-quality shape generation.

These methods have explored various 3D representations, including point clouds [Nichol et al. 2022; Vahdat et al. 2022], meshes [Chen et al. 2024a,b; Siddiqui et al. 2024], and neural fields [Hong et al. 2024; Jun and Nichol 2023; Poole et al. 2023]. However, while these representations excel at visualization and rendering, they typically lack the semantic abstraction and interpretability that align with human cognition. Moreover, these generated meshes pose challenges for real-time multiplayer game environments, requiring both significant bandwidth for multiple users to load new content and additional optimization steps to meet game engine performance requirements.

2.2 Shape Primitive Abstraction

Shape primitive abstraction aims to represent 3D contents by "simple geometry shape", named as *primitives*. Prior approaches have used simple geometric primitives such as cuboids [Gupta et al. 2010; Li et al. 2017; Mo et al. 2019; Tulsiani et al. 2017] and cylinders [Binford 1975], which offer computational simplicity and intuitive interpretation but limited expressiveness. Super-quadrics [Paschalidou et al. 2019; Pentland 1986] provide a better balance between representational power and computational tractability by generalizing quadric surfaces. Some methods [Chen et al. 2020; Deng et al. 2020] proposed convex polytopes as primitive representations, offering different trade-offs between expressiveness and optimization complexity. Implicit primitives representing shapes through learned fields have also been explored [Chen et al. 2019; Gadelha et al. 2020; Genova et al. 2019]. Another line of works focus on Computer-aided design (CAD) modeling and define special primitives of Constructive Solid Geometry (CSG) trees [Lê et al. 2021; Li et al. 2019, 2023] via iterative boolean operators, which is beyond the scope of the paper.

To conduct shape primitive abstraction, optimization-based methods directly minimize reconstruction objectives, either through 3D supervision [Liu et al. 2022, 2023b] to ensure geometric accuracy, or 2D supervision [Gao et al. 2024; Monnier et al. 2023] from multi-view images. To overcome the local optima issues, EMS [Liu et al. 2022] models the superquadric primitive abstraction probabilistically, enhancing its robustness to outliers. However, these methods often fragment semantic parts into multiple pieces, as they primarily optimize for geometric reconstruction rather than human-like abstraction—a limitation stemming from the lack of large-scale datasets that capture human cognitive principles in shape decomposition.

Some works attempt to learn the shape distribution from data. Pioneer work [Tulsiani et al. 2017] presents a learning framework to assemble objects by predicting cuboid parameters, which was later extended to superquadrics by [Paschalidou et al. 2019]. To model the step-by-step construction process, 3D-PRNN [Zou et al. 2017] leveraged recursive neural networks (RNN) for sequential cuboid prediction. Recent work PASTA [Li et al. 2024b] employs a sequence-to-sequence model for part-aware 3D shape generation. However, it utilizes only cuboids as primitives and trains exclusively on category-specific data, limiting its geometric expressiveness and generalization capabilities across different shape categories. Similarly, other learning methods also rely on small, category-specific

datasets for training, constraining their applicability to diverse shape domains.

2.3 Auto-Regressive Model for 3D Generation

Auto-Regressive (AR) transformers have demonstrated impressive results on various tasks including language-modeling [Achiam et al. 2023; Brown et al. 2020; Radford et al. 2019; Touvron et al. 2023] and vision generation [Esser et al. 2021; Ramesh et al. 2021; Tian et al. 2024]. The core of AR models lies in their self-supervised learning strategy of predicting the next token in a sequence—a simple yet remarkably scalable and generalizable approach.

Due to their natural ability to handle variable-length outputs, AR models have been successfully applied to layout generation tasks. Sceneformer [Wang et al. 2021] pioneered this direction by introducing a transformer-based architecture to predict both categorical and geometric attributes of 3D objects for scene synthesis. This was followed by [Paschalidou et al. 2021; Zhao et al. 2024] that further improved scene generation through better object sequence modeling and shape prior integration.

Recent works have demonstrated the potential of AR models in 3D artist-created mesh generation. MeshGPT [Siddiqui et al. 2024] first introduced the paradigm of treating meshes as sequences of vertices and faces. Building on this foundation, subsequent works achieved significant improvements through various innovations: introducing shape conditioning [Chen et al. 2024a], developing more compact tokenization schemes [Chen et al. 2024b; Tang et al. 2024], and incorporating language capabilities [Wang et al. 2024a]. Particularly inspiring for our work is MeshAnything's [Chen et al. 2024a,b] approach to conditional mesh generation from point clouds, which motivates us to reformulate shape abstraction as a shape-conditioned generation task. We parameterize primitives as tokens and employ an auto-regressive model to predict the primitive sequence, effectively learning the implicit rules of shape decomposition.

3 METHOD

Our proposed **PrimitiveAnything** is a novel primitive generation framework that reformulates shape abstraction as a sequence generation task, enabling human-like shape decomposition. Our method comprises three key components: an ambiguity-free primitive parameterization scheme (Sec. 3.1), a primitive transformer architecture (Sec. 3.2), and an auto-regressive generation pipeline (Sec. 3.3). Fig. 2 illustrates the overall framework.

3.1 Primitive Parameterization

Our goal is to establish a parameterization scheme that represents 3D objects using an arbitrary number and variety of predefined primitives. Given a 3D object and a predefined 3D primitive set $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_N\}$ with N standard primitive shapes, we aim to find its approximate representation $\hat{\mathcal{M}}$ composed of n transformed primitives:

$$\hat{\mathcal{M}} = \{p_1, p_2, \ldots, p_n\} \qquad (1)$$

Each primitive p_i is defined by combining a standard primitive type $\mathcal{P}_{c_i} \in \mathcal{P}$ with its rigid transformation $\mathcal{T}(\cdot)$ in 3D space:

$$p_i = \mathcal{T}(\mathcal{P}_{c_i}; \mathbf{s}_i, \mathbf{r}_i, \mathbf{t}_i) \qquad (2)$$
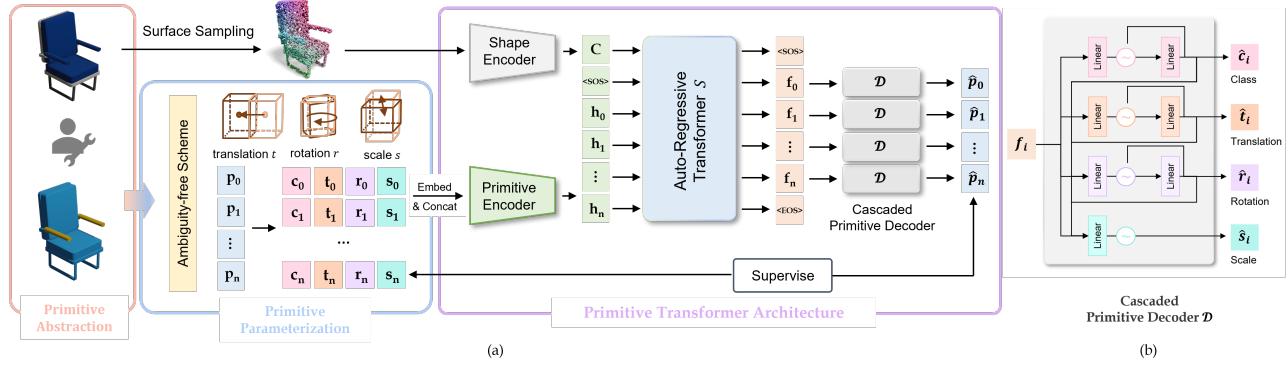
(a)   (b)

Fig. 2. Overview. We propose PrimitiveAnything to decompose complex shapes into 3D primitive assembly via the auto-regressive transformer. Given human-crafted 3D primitive abstraction contents, we first design an ambiguity-free scheme to parameterize each primitive p into class label c, translation t, rotation r and scale s, and then employ a primitive encoder to form primitive token h. Meanwhile, a shape encoder encodes 3D shape features C from sampled point clouds. Our primitive transformer S predicts the next primitive based on the input condition C and previously generated primitives. To model the dependencies among primitive attributes, we proposed a cascaded primitive decoder D that sequentially predicts primitive attributes.
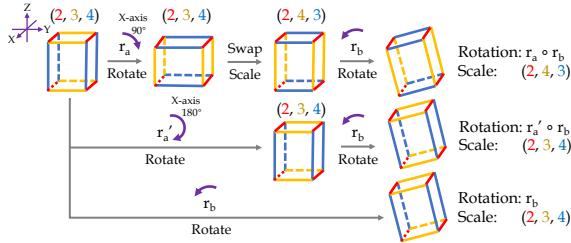

Fig. 3. Demonstration of primitive attribute ambiguity. A primitive with inherent symmetry can correspond to multiple scales and rotations through self-rotation and possible axis swapping.

where:

- c_i denotes the primitive class label
- $\mathbf{s}_i \in \mathbb{R}^3$ denotes the scale factors along three principal axes
- $\mathbf{r}_i \in \mathbb{R}^3$ specifies the rotation angles in x-y-z Euler order
- $\mathbf{t}_i \in \mathbb{R}^3$ defines the translation of the primitive center

These transformation components are sequentially applied to the standard predefined primitive to achieve the final configuration p_i.

However, directly using the above parameterization is not sufficient. Upon deeper analysis, we observe that many common primitives (such as cuboids and cylinders) possess inherent symmetries. Due to these symmetries, different combinations of scale and rotation parameters can produce identical transformed primitives in 3D space, creating ambiguity in the parameter representation. Consider the cuboid example in Fig. 3: applying a 90-degree rotation around the z-axis before the original rotation, while simultaneously swapping the x and y scale factors, results in identical shapes. Such parameter ambiguity complicates the learning process, as the model encounters multiple valid parameter combinations for the same shape, which cannot be resolved through mathematical reformulation alone.

To address this issue, we propose an ambiguity-free parameterization approach. Let V denote the set of symmetry axes for the predefined primitive $\mathcal{P}_{c_i}$ corresponding to the transformed primitive

p_i, m_j represents the order of symmetry for the j-th symmetry axis $\mathbf{v}_j \in V$. Note that axis permutations that result in equivalent configurations are also counted when determining the total symmetry order. We can then define rotational symmetry set $\mathcal{R}$ as:

$$\mathcal{R} = \bigcup_{j=1}^{n} \left\{ \mathrm{Rot}(\mathbf{v}_j, \frac{2\pi k}{m_j}) \,\middle|\, k = 0, 1, \ldots, m_j - 1 \right\} \quad (3)$$

where $\mathrm{Rot}(\mathbf{v}, \theta)$ represents a rotation transformation of angle θ around axis $\mathbf{v}$. We further compose each equivalent rotation transformation $\mathbf{r}_k \in \mathcal{R}$ with the original transformation parameters $(\mathbf{s}_i, \mathbf{r}_i, \mathbf{t}_i)$, and select the combination that yields the minimal L1 norm of rotation as our new transformation $(\mathbf{s}'_i, \mathbf{r}'_i, \mathbf{t}'_i)$:

$$\mathbf{r}'_i = \arg\min_{\mathbf{r}_k \in \mathcal{R}} \|\hat{\mathbf{r}}_k\|_1, \quad \text{where} \quad (4)$$

$$\mathcal{T}(\cdot; \hat{\mathbf{s}}_k, \hat{\mathbf{r}}_k, \hat{\mathbf{t}}_k) = \mathcal{T}(\mathcal{T}(\cdot; \mathbf{s}_k, \mathbf{r}_k); \mathbf{s}_i, \mathbf{r}_i, \mathbf{t}_i) \quad (5)$$

Consequently, we reformulate the transformed primitive as:

$$p_i = \mathcal{T}(\mathcal{P}_{c_i}; \mathbf{s}'_i, \mathbf{r}'_i, \mathbf{t}'_i) \quad (6)$$

This formulation eliminates symmetry-induced ambiguity while reducing the parameter space, facilitating more effective learning and preventing mode confusion.

3.2 Primitive Transformer

Drawing inspiration from how humans sequentially compose shapes by assembling basic geometric elements, we formulate primitive abstraction as a sequential generation process. Our primitive transformer F predicts the next primitive based on the input condition C and previously generated primitives:

$$p_i = F(C; p_1, \ldots, p_{i-1}) \quad (7)$$

The architecture consists of three learnable modules: a primitive encoder $\mathcal{E}$, a decoder-only transformer model S, and a cascaded primitive decoder D. We discretize the scale, rotation, and translation parameters, and treat them along with the class label as discrete input tokens. For the i-th primitive p_i, its attributes $c_i, \mathbf{s}_i, \mathbf{r}_i, \mathbf{t}_i$ are

embedded using the learnable embeddings $\mathbf{e}_c$, $\mathbf{e}_s$, $\mathbf{e}_r$, $\mathbf{e}_t$, followed by a linear primitive encoder $\mathcal{E}$ to generate the primitive token $\mathbf{h}_i$:

$$\mathbf{h}_i = \mathcal{E}(\mathbf{e}_c(c_i), \mathbf{e}_s(s_i), \mathbf{e}_r(r_i), \mathbf{e}_t(t_i)) \tag{8}$$

The decoder-only transformer $\mathcal{S}$ receives the input condition C and tokens of all previously generated primitives as input, producing the feature representation $\mathbf{f}_i$ of the new primitive:

$$\mathbf{f}_i = \mathcal{S}(C; \mathbf{h}_1, \ldots, \mathbf{h}_{i-1}) \tag{9}$$

To generate the next primitive's attributes from $\mathbf{f}_i$, similar to previous scene generation works [Paschalidou et al. 2021; Ritchie et al. 2019; Wang et al. 2021], we utilize a cascaded primitive decoder that explicitly models the dependencies among primitive attributes:

$$\hat{c}_i = \mathcal{D}_c(\mathbf{f}_i) \tag{10}$$

$$\hat{\mathbf{t}}_i = \mathcal{D}_t(\mathbf{f}_i, \mathbf{e}_c(c_i)) \tag{11}$$

$$\hat{\mathbf{r}}_i = \mathcal{D}_r(\mathbf{f}_i, \mathbf{e}_c(c_i), \mathbf{e}_t(t_i)) \tag{12}$$

$$\hat{\mathbf{s}}_i = \mathcal{D}_s(\mathbf{f}_i, \mathbf{e}_c(c_i), \mathbf{e}_t(t_i), \mathbf{e}_r(r_i)) \tag{13}$$

where $\mathcal{D}_c$, $\mathcal{D}_t$, $\mathcal{D}_r$, and $\mathcal{D}_s$ represent the class, translation, rotation, and scale decoders respectively. Each decoder takes the concatenation of the initial feature $\mathbf{f}_i$ and the embedded representations of previously decoded attributes, and then outputs logits of the probability. This design captures the natural correlations between primitive attributes: the choice of primitive type influences its likely position, rotation, and scale parameters, and also aligns with human assembling logic: selecting type, determining position, and then adjusting rotation and scale.

3.3 Auto-Regressive Primitive Generation

Sequence Formulation. Our primitive transformer can be trained for shape-conditioned generation by taking condition features before primitive features through the carefully designed framework. We select point clouds as input conditions, leveraging their ease of extraction from various 3D representations, and utilize the Michelangelo [Zhao et al. 2023] encoder to convert the point cloud into a fixed-length token sequence. This encoded sequence is concatenated with a start token <SOS>, followed by the primitive tokens $\{h_1, ..., h_{i-1}\}$, forming the complete input sequence for the transformer. To determine when generation should terminate, we introduce an <EOS> decoder $\mathcal{D}_{eos}$ operating on the primitive feature $\mathbf{f}_i$ output by the transformer. Primitives are sorted by centroids in z-y-x order (z-axis as top), progressing from lowest to highest.

Training objective. We train the primitive transformer using next-step prediction as the primary objective, while incorporating an auxiliary 3D shape guidance term:

$$\mathcal{L} = \mathcal{L}_{eos} + \mathcal{L}_{ce} + \mathcal{L}_{cd} \tag{14}$$

Here, $\mathcal{L}_{ce}$ denotes the cross-entropy loss used to supervise the discrete primitive attributes c_i, $\mathbf{s}_i$, $\mathbf{r}_i$, $\mathbf{t}_i$, while $\mathcal{L}_{eos}$ represents the binary cross-entropy loss applied to $\mathcal{D}_{eos}(\mathbf{f}_i)$ to guide the termination prediction. To ensure precise alignment and robust control over reconstruction quality, the Chamfer Distance loss [Fan et al. 2017] $\mathcal{L}_{cd}$ is employed for each generated primitive. As the predicted primitive attributes are discrete, the Gumbel-Softmax technique [Jang et al. 2017] is applied to enable differentiable sampling for each next-token prediction, generates the predicted attributes

$\{\mathbf{s}_{i,\text{pred}}\}_{i=1}^n$, $\{\mathbf{r}_{i,\text{pred}}\}_{i=1}^n$, and $\{\mathbf{t}_{i,\text{pred}}\}_{i=1}^n$, forming the predicted next-primitive set $\{p_{i,\text{pred}}\}_{i=1}^n$. Subsequently, both $\{p_{i,\text{pred}}\}_{i=1}^n$ and the ground-truth primitive set $\{p_{i,\text{gt}}\}_{i=1}^n$ are sampled to produce the point clouds pc_{pred} and pc_{gt}, respectively. The Chamfer Distance loss is then calculated as:

$$\mathcal{L}_{cd} = CD(pc_{\text{pred}}, pc_{\text{gt}}) \tag{15}$$

where $CD(\cdot, \cdot)$ denotes the Chamfer distance [Fan et al. 2017].

Inference. Starting from an input point cloud, our primitive transformer autoregressively generates primitive features $\{\mathbf{f}_i\}_{i=1}^n$, which are subsequently decoded and assembled into the final primitive representation $\hat{\mathcal{M}}$. This process continues until the EOS judgment, signaling the completion of the primitive generation.

3.4 Implementation Details

For model architecture, our auto-regressive transformer has 12 layers with a hidden size of 768. The cascaded decoders are implemented as 2-layer MLPs (hidden size 768) that process the concatenation of primitive features and previously decoded attribute embeddings. All training data was normalized to lie within a unit cube. For primitive attribute discretization, rotation, scale and translation are discretized into 180, 128, and 128 levels per dimension, respectively. Attribute embeddings are 16-dimensional for rotation, scale, and translation parameters, with 48-dimensional embeddings for class labels. The training was conducted using the Adam optimizer with a learning rate of 1×10^{-3}, a batch size of 128, and gradient accumulation over 4 steps. The model was trained on 8 NVIDIA V100 GPUs for 3 days.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. We collect a large-scale 3D dataset with primitive abstraction annotations created by human annotators, named HumanPrim. HumanPrim contains 120K samples, each consisting of a 3D mesh, its surface point cloud, and manual primitive assembly. Three primitive types are utilized in the assembly: cuboids, elliptical cylinders, and ellipsoids. The primitive sequences have an average length of 30.9 primitives, with the longest sequence containing 144 primitives. To ensure a thorough evaluation of our method, we randomly select 314 high quality samples with artist-created labels to form the test set. To evaluate the proposed method's generalization capability, we additionally evaluate on data from ShapeNet [Chang et al. 2015] and Objarverse [Deitke et al. 2023].

Evaluation Metrics. For geometric evaluation, we uniformly sample point clouds on the surfaces of predicted primitives and compare them against ground-truth point clouds sampled from the original meshes. We employ four metrics for evaluation: Chamfer Distance (CD), Earth Mover's Distance (EMD), Hausdorff Distance, and Voxel-IoU. For the Voxel-IoU metric, both predicted and ground-truth point clouds are voxelized at a resolution of 32^3, after which their intersection over union is computed. Moreover, to evaluate how well the generated primitive abstractions align with human decomposition patterns, we additionally employ instance segmentation metrics. Following previous works [He et al. 2024; Xie et al. 2022], we address the geometric discrepancy between ground-truth and

predicted primitives through a label transfer process: points are first sampled on the ground-truth mesh, then matched to their nearest neighbors in the predicted primitives to transfer prediction labels. Three segmentation metrics are used: rand index (RI), variation of information (VOI), and segmentation covering (SC).

4.2 Comparisons

Comparison methods. As no existing methods can generate variable-length sequences of diverse primitive types, we compare our approach with state-of-the-art optimization-based methods: Marching-Primitives (MP) [Liu et al. 2023b] and EMS [Liu et al. 2022], both using superquadric primitives. We further compare against two learning-based methods: a cuboid-based approach [Tulsiani et al. 2017] and a superquadric-based method [Paschalidou et al. 2019], using their pre-trained models. Note that these comparisons are limited to their specifically trained categories, as these methods do not generalize to other shape classes.

Table 1. Geometric comparison with optimization-based methods on the HumanPrim test set.

Method	CD ↓	EMD ↓	Hausdorff ↓	Voxel-IoU ↑
EMS [Liu et al. 2022]	0.1062	0.0840	0.338	0.259
MP [Liu et al. 2023b]	0.0546	0.0515	**0.120**	0.201
Ours	**0.0404**	**0.0475**	0.158	**0.484**

Quantitative Comparisons. We present quantitative comparisons between our method and optimization-based approaches in Tab. 1. While EMS shows robustness in fitting single primitives, it faces challenges with multiple primitive predictions, particularly in identifying smaller primitive parts, and often fails in the absence of a main primitive. Marching-Primitives achieves progressively refined 3D contour matching through iterative optimization, as reflected in its Hausdorff distance performance (maximum distance among nearest point pairs between two point clouds). However, its results frequently deviate from human construction patterns, often decomposing regions that should be represented by a single primitive into multiple primitives. This generates erroneous occupancy within primitives, leading to notably lower Voxel-IoU scores, which measure surface coverage effectiveness, and reduced overall 3D accuracy as indicated by CD and EMD metrics. Our method demonstrates superior overall performance across metrics. The 3D instance segmentation metrics shown in Tab. 2 further validate our method's superior capability in generating human-like primitive abstractions.

We conduct additional comparisons with learning-based methods on the Chair subset of our HumanPrim test set (59 samples) and the *Chair* category from ShapeNet's test split (1,317 samples), as shown in Tab. 3, since previous learning-based approaches are limited to training on individual categories. [Tulsiani et al. 2017] predicts only cuboids and shows a limited ability to model objects effectively. [Paschalidou et al. 2019] utilizes superquadrics and offers more flexible object modeling capabilities, but its overall accuracy remains insufficient. Notably, our method demonstrates robust generalization, outperforming all benchmarked approaches across all metrics, even though it was not trained on the ShapeNet dataset—unlike the comparison methods, which were specifically

Table 2. 3D segmentation accuracy comparison with optimization-based methods on the HumanPrim test set.

Method	RI ↑	VOI ↓	SC ↑
EMS [Liu et al. 2022]	0.696	3.520	0.280
MP [Liu et al. 2023b]	0.821	3.793	0.254
Ours	**0.892**	**2.296**	**0.409**

Table 3. Geometric comparison with learning-based methods on HumanPrim test set (Chair subset) and ShapeNet test set (*Chair* category).

Method	CD ↓	EMD ↓	Hausdorff ↓	Voxel-IoU ↑
Chair subset of HumanPrim				
[Tulsiani et al. 2017]	0.2512	0.1835	0.420	0.041
[Paschalidou et al. 2019]	0.1438	0.1088	0.332	0.095
Ours	**0.0343**	**0.0458**	**0.136**	**0.550**
Chair category of ShapeNet				
[Tulsiani et al. 2017]	0.2282	0.1667	0.411	0.046
[Paschalidou et al. 2019]	0.1343	0.1038	0.285	0.094
Ours	**0.0553**	**0.0588**	**0.190**	**0.322**

Table 4. 3D segmentation accuracy comparison with learning-based methods on the HumanPrim test set (Chair subset).

Method	RI ↑	VOI ↓	SC ↑
[Tulsiani et al. 2017]	0.740	3.097	0.335
[Paschalidou et al. 2019]	0.660	3.346	0.274
Ours	**0.931**	**1.499**	**0.578**

designed for it. This superiority is further corroborated by the segmentation metrics presented in Tab. 4 (ShapeNet is not tested due to the absence of ground-truth primitive labels).

Qualitative Comparisons. Fig. 4 presents qualitative comparisons with optimization-based methods. EMS produces sparse and coarse superquadrics abstractions that lack detailed surface fidelity. Marching-Primitives achieves rough shape contours through highly overlapping primitives, its decompositions often deviate from human construction patterns. Specifically, it tends to over-segment large or elongated parts using numerous primitives and frequently overlooks fine structural details. In contrast, our method successfully identifies geometric features at various scales, achieving both human-crafted shape abstraction and faithful reproduction of the overall surfaces and global structure of the original 3D objects.

Figs. 7 and 8 illustrate visual comparisons with other learning-based methods on the Chair subset. [Tulsiani et al. 2017] produces sparse cuboid abstractions with relatively coarse geometric structures. Although [Paschalidou et al. 2019]'s multiple superquadric predictions better conform to 3D object surfaces, it still exhibits numerous imprecise and erroneous predictions. In contrast, our method demonstrates significant advantages in both accuracy and generalization capacity. Fig. 9 provides qualitative comparisons on the Objaverse dataset, further demonstrating the generalizability of our method across diverse 3D objects.

Fig. 4. Qualitative comparisons on the HumanPrim test set: In our method, colors indicate different primitive types, while in Marching Primitives and EMS, colors represent separate superquadrics. Our method achieves human-crafted primitive abstraction and faithfully reproduces the original 3D structure.

Table 5. Ablation studies on the HumanPrim test set.

Method	CD ↓	EMD ↓	Hausdorff ↓	Voxel-IoU ↑
w/o ambiguity-free param.	0.0564	0.0584	0.204	0.414
w/o cascaded decoding	0.0558	0.0586	0.243	0.458
w/o Chamfer Distance loss	0.0440	0.0514	0.174	0.475
Ours	**0.0404**	**0.0475**	**0.158**	**0.484**

4.3 Ablation Study

To validate the effectiveness of each component in our framework, we conduct ablation studies using the HumanPrim dataset while keeping the experimental and training configurations consistent with those in Sec. 4.1. We sequentially disable specific modules while leaving others unchanged.

The results in Tab. 5 show that all proposed improvements contribute effectively to the overall performance. The ambiguity-free parameterization scheme helps reduce mode confusion, as evidenced by the Voxel-IoU metric. The cascaded decoding architecture improves generation stability and prevents outlier occurrences, as reflected in a decrease of the mean Hausdorff distance. The Chamfer Distance loss allows for finer-grained control over primitive generation, leading to improved accuracy and detail. These results show that each component of our method is essential for high-quality shape abstraction.

4.4 Primitive-based 3D content generation

Our framework enables versatile primitive-based 3D content generation through its flexible design, which can interface with various 3D generative models to create customized primitive-based 3D content from diverse user inputs, as demonstrated in Fig. 5 (TRELLIS [Xiang et al. 2024] for image-conditioning and SDXL [Podell et al. 2023] + Rembg + TRELLIS for text-conditioning). This approach offers several key advantages over conventional mesh-based representations. First, since each primitive is directly represented by a predefined primitive type with associated scale, rotation, and translation parameters, users can easily modify the geometric structure through

Fig. 5. PrimitiveAnything interfaces with state-of-the-art 3D shape generation models to enable text- and image-conditioned primitive-based 3D content generation.

common graphics interfaces while maintaining high modeling capabilities. This accessibility particularly benefits non-expert users in fine-tuning generated results. Additionally, our primitive-based representation achieves significant storage efficiency, reducing space requirements by over 95% compared to traditional mesh representations while preserving geometric fidelity. These characteristics make our method particularly suitable for applications requiring both user interactivity and resource efficiency in 3D content generation.

5 CONCLUSION

In this work, we presented PrimitiveAnything, a novel framework that reformulates 3D shape abstraction as a sequence generation task. Our framework learns directly from human-crafted primitive assemblies, enabling it to capture and reproduce intuitive shape decomposition patterns. PrimitiveAnything demonstrates strong generalization capability, successfully generating high-quality primitive assemblies across diverse shape categories, enabling versatile primitive-based 3D content creation. Moreover, the lightweight and efficient nature of primitive representation shows promise for enabling primitive-based user-generated content (UGC) in games.
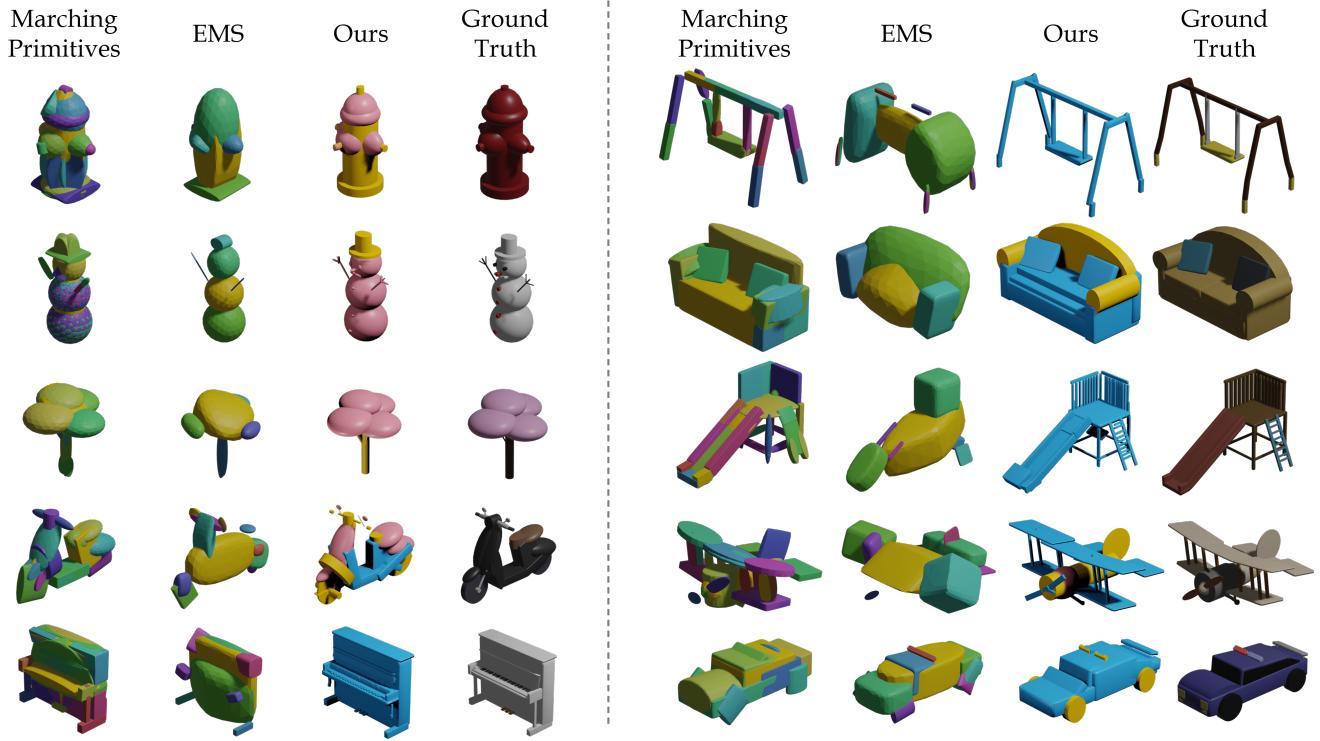
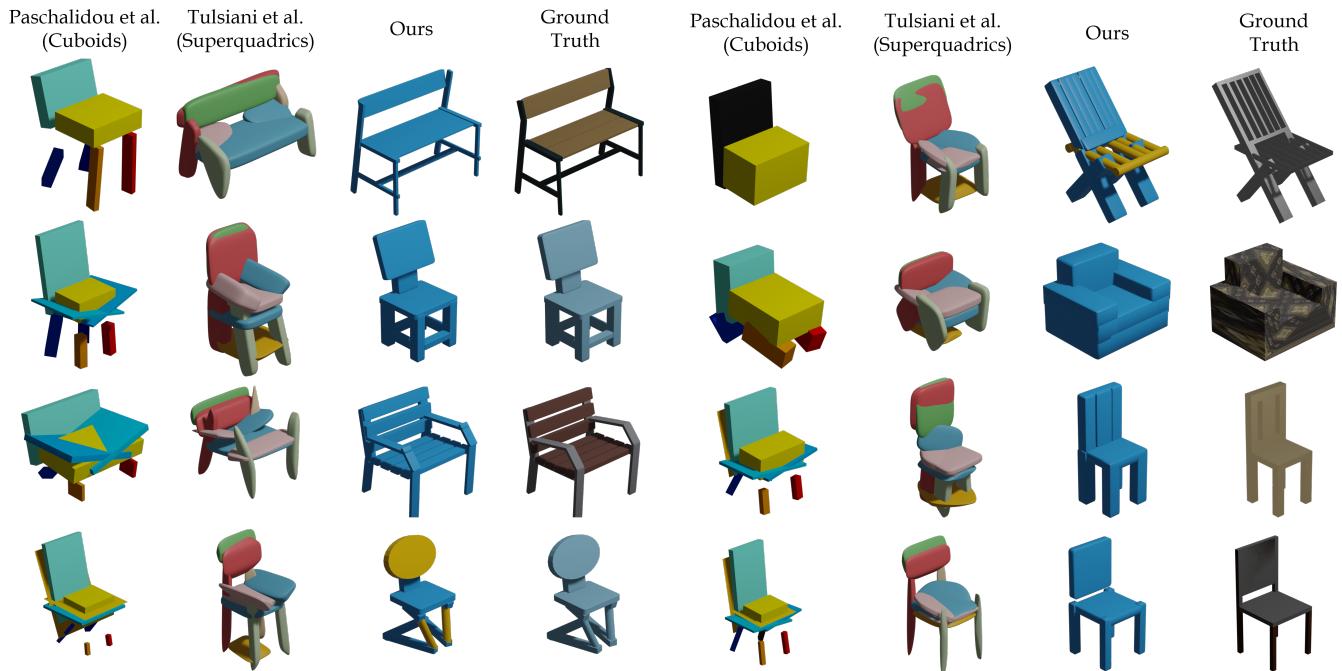Fig. 6. More qualitative comparisons with optimization-based methods on the HumanPrim dataset.

Fig. 7. Comparisons on the HumanPrim test set (Chair subset).

Fig. 8. Comparisons on the ShapeNet test set (Chair category).
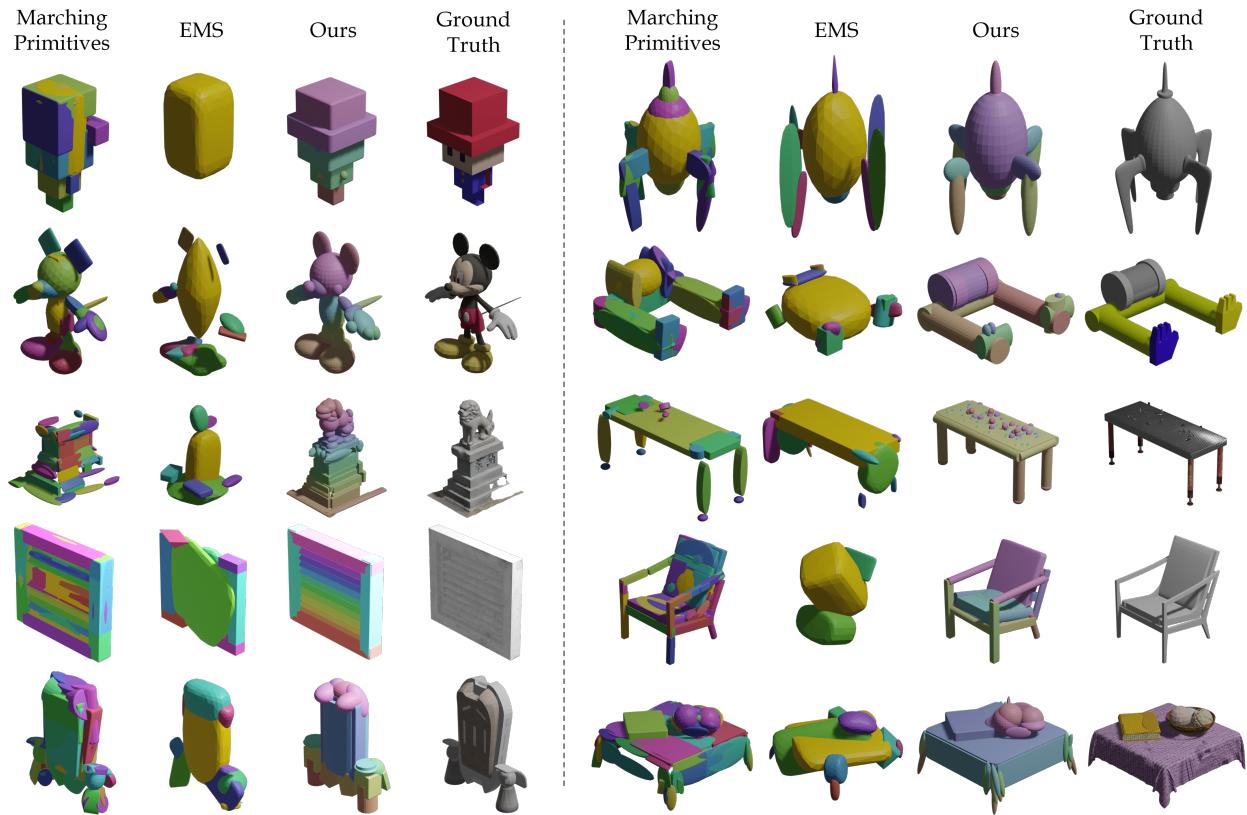
Fig. 9. Comparisons on the Objaverse dataset.

Fig. 10. More visualizations of primitive-based 3D content generation on text and image conditions.

ACKNOWLEDGMENTS

REFERENCES

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

Raphael Bensadoun, Tom Monnier, Yanir Kleiman, Filippos Kokkinos, Yawar Siddiqui, Mahendra Kariya, Omri Harosh, Roman Shapovalov, Benjamin Graham, Emilien Garreau, et al. 2024. Meta 3d gen. *arXiv preprint arXiv:2407.02599* (2024).

Irving Biederman. 1985. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing* 32, 1 (Oct 1985), 29–73. https://doi.org/10.1016/0734-189x(85)90002-7

Irving Biederman. 2005. Recognition-by-components: A theory of human image understanding. *Psychological Review* (Sep 2005), 115–147. https://doi.org/10.1037/0033-295x.94.2.115

Thomas Binford. 1975. Visual perception by computer. In *Proc. IEEE Conf. on Systems and Control, 1975*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. *ShapeNet: An Information-Rich 3D Model Repository*. Technical Report arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. 2024a. MeshAnything: Artist-Created Mesh Generation with Autoregressive Transformers. *arXiv preprint arXiv:2406.10163* (2024).

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

Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. 2020. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 45–54.

Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. 2019. Bae-net: Branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8490–8499.

Laurent Chevalier, Fabrice Jaillet, and Atilla Baskurt. 2003. Segmentation and superquadric modeling of 3D objects. (2003).

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2023. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13142–13153.

Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. 2020. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 31–44.

Wenqi Dong, Bangbang Yang, Lin Ma, Xiao Liu, Liyuan Cui, Hujun Bao, Yuewen Ma, and Zhaopeng Cui. 2024. Coin3d: Controllable and interactive 3d assets generation with proxy-guided conditioning. In *ACM SIGGRAPH 2024 Conference Papers*. 1–10.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12873–12883.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 605–613.

Matheus Gadelha, Giorgio Gori, Duygu Ceylan, Radomir Mech, Nathan Carr, Tamy Boubekeur, Rui Wang, and Subhransu Maji. 2020. Learning generative models of shape handles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 402–411.

Zhirui Gao, Renjiao Yi, Yuhang Huang, Wei Chen, Chenyang Zhu, and Kai Xu. 2024. Learning Part-aware 3D Representations by Fusing 2D Gaussians and Superquadrics. *arXiv preprint arXiv:2408.10789* (2024).

Michael Garland and Paul S Heckbert. 1997. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 209–216.

Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. 2019. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF international conference on computer vision*. 7154–7164.

Julia Guerrero-Viu, Milos Hasan, Arthur Roullier, Midhun Harikumar, Yiwei Hu, Paul Guerrero, Diego Gutierrez, Belen Masia, and Valentin Deschaintre. 2024. Texsliders: Diffusion-based texture editing in clip space. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

Minghao Guo, Bohan Wang, and Wojciech Matusik. 2024. Medial Skeletal Diagram: A Generalized Medial Axis Approach for Compact 3D Shape Representation. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–23.

Abhinav Gupta, Alexei A Efros, and Martial Hebert. 2010. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer, 482–496.

Yuze He, Wang Zhao, Shaohui Liu, Yubin Hu, Yushi Bai, Yu-Hui Wen, and Yong-Jin Liu. 2024. AlphaTablets: A Generic Plane Representation for 3D Planar Reconstruction from Monocular Videos. *arXiv preprint arXiv:2411.19950* (2024).

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2024. LRM: Large Reconstruction Model for Single Image to 3D. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=sllU8vvsFF

Anita Hu, Nishkrit Desai, Hassan Abu Alhaija, Seung Wook Kim, and Maria Shugrina. 2024. Diffusion texture painting. In *ACM SIGGRAPH 2024 Conference Papers*. 1–12.

Xiaoyang Huang, Yi Zhang, Kai Chen, Teng Li, Wenjun Zhang, and Bingbing Ni. 2023. Learning shape primitives via implicit convexity regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3642–3651.

Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. 2024. Make-a-shape: a ten-million-scale 3d shape model. In *Forty-first International Conference on Machine Learning*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkE3y85ee

Heewoo Jun and Alex Nichol. 2023. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463* (2023).

Eric-Tuan Lê, Minhyuk Sung, Duygu Ceylan, Radomir Mech, Tamy Boubekeur, and Niloy J Mitra. 2021. Cpfn: Cascaded primitive fitting networks for high-resolution point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7457–7466.

Ales Leonardis, Ales Jaklic, and Franc Solina. 1997. Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 11 (1997), 1289–1295.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. 2024c. Instant3D: Fast Text-to-3D with Sparse-view Generation and Large Reconstruction Model. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=2lDQLiH1W4

Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. 2017. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.

Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. 2019. Supervised fitting of geometric primitives to 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2652–2660.

Songlin Li, Despoina Paschalidou, and Leonidas Guibas. 2024b. PASTA: Controllable Part-Aware Shape Generation with Autoregressive Transformers. *arXiv preprint arXiv:2407.13677* (2024).

Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. 2024a. CraftsMan3D: High-fidelity Mesh Generation with 3D Native Generation and Interactive Geometry Refiner. *arXiv preprint arXiv:2405.14979* (2024).

Yuanqi Li, Shun Liu, Xinran Yang, Jianwei Guo, Jie Guo, and Yanwen Guo. 2023. Surface and edge detection for primitive fitting of point clouds. In *ACM SIGGRAPH 2023 conference proceedings*. 1–10.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 300–309.

Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. 2023a. Openshape: Scaling up 3d shape representation towards open-world understanding. *Advances in neural information processing systems* 36 (2023), 44860–44879.

Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. 2022. Robust and accurate superquadric recovery: A probabilistic approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2676–2685.

Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. 2023b. Marching-primitives: Shape abstraction from signed distance function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8771–8780.

Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. 2019. StructureNet: Hierarchical Graph Networks for 3D Shape Generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019* 38, 6 (2019), Article 242.

Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei Efros, and Mathieu Aubry. 2023. Differentiable blocks world: Qualitative 3d decomposition by rendering primitives. *Advances in Neural Information Processing Systems* 36 (2023), 5791–5807.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751* (2022).

Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. 2021. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 12013–12026.

Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. 2019. Superquadrics Revisited: Learning 3D Shape Parsing beyond Cuboids. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

William Peebles and Saining Xie. 2022. Scalable Diffusion Models with Transformers. *arXiv preprint arXiv:2212.09748* (2022).

Alex Pentland. 1986. Parts: Structured Descriptions of Shape.. In *AAAI*. 695–701.

Dmitry Petrov, Pradyumn Goyal, Vikas Thamizharasan, Vladimir Kim, Matheus Gadelha, Melinos Averkiou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. 2024. Gem3d: Generative medial abstractions for 3d shape synthesis. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* (2023).

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=FjNys5c7VyY

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

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*. Pmlr, 8821–8831.

Daniel Ritchie, Kai Wang, and Yu-an Lin. 2019. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6182–6190.

Lawrence G Roberts. 1963. *Machine perception of three-dimensional solids*. Ph. D. Dissertation. Massachusetts Institute of Technology.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2024. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19615–19625.

Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. 2024. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation. *arXiv preprint arXiv:2409.18114* (2024).

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. 2024. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905* (2024).

Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, , Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. 2024. TripoSR: Fast 3D Object Reconstruction from a Single Image. *arXiv preprint arXiv:2403.02151* (2024).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. 2017. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2635–2643.

Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. 2022. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems* 35 (2022), 10021–10039.

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. 2023a. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12619–12629.

Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. 2021. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 106–115.

Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. 2024a. LLaMA-Mesh: Unifying 3D Mesh Generation with Language Models. arXiv:2411.09595 [cs.LG] https://arxiv.org/abs/2411.09595

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. 2023b. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=ppJuFSOAnM

Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. 2024b. CRM: Single Image to 3D Textured Mesh with Convolutional Reconstruction Model. *arXiv preprint arXiv:2403.05034* (2024).

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2024. Structured 3D Latents for Scalable and Versatile 3D Generation. *arXiv preprint arXiv:2412.01506* (2024).

Yiming Xie, Matheus Gadelha, Fengting Yang, Xiaowei Zhou, and Huaizu Jiang. 2022. Planarrecon: Real-time 3d plane detection and reconstruction from posed monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. 2024. InstantMesh: Efficient 3D Mesh Generation from a Single Image with Sparse-view Large Reconstruction Models. *arXiv preprint arXiv:2404.07191* (2024).

Xianghui Yang, Huiwen Shi, Bowen Zhang, Fan Yang, Jiacheng Wang, Hongxu Zhao, Xinhai Liu, Xinzhou Wang, Qingxiang Lin, Jiaao Yu, Lifu Wang, Zhuo Chen, Sicong Liu, Yuhong Liu, Yong Yang, Di Wang, Jie Jiang, and Chunchao Guo. 2024. Tencent Hunyuan3D-1.0: A Unified Framework for Text-to-3D and Image-to-3D Generation. arXiv:2411.02293 [cs.CV]

Xin Yu, Ze Yuan, Yuan-Chen Guo, Ying-Tian Liu, Jianhui Liu, Yangguang Li, Yan-Pei Cao, Ding Liang, and Xiaojuan Qi. 2024. TEXGen: a Generative Diffusion Model for Mesh Textures. *ACM Trans. Graph.* 43, 6, Article 213 (2024), 14 pages. https://doi.org/10.1145/3687909

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 2023. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–16.

Jinzhi Zhang, Feng Xiong, and Mu Xu. 2024d. 3D representation in 512-Byte: Variational tokenizer is the key for autoregressive 3D generation. *arXiv preprint arXiv:2412.02202* (2024).

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. 2024a. GS-LRM: Large Reconstruction Model for 3D Gaussian Splatting. *European Conference on Computer Vision* (2024).

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024c. CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

Shangzhan Zhang, Sida Peng, Tao Xu, Yuanbo Yang, Tianrun Chen, Nan Xue, Yujun Shen, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. 2024b. Mapa: Text-driven photo-realistic material painting for 3d shapes. In *ACM SIGGRAPH 2024 Conference Papers*. 1–12.

Yiqun Zhao, Zibo Zhao, Jing Li, Sixun Dong, and Shenghua Gao. 2024. Roomdesigner: Encoding anchor-latents for style-consistent and shape-compatible indoor scene generation. In *2024 International Conference on 3D Vision (3DV)*. IEEE, 1413–1423.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, BIN FU, Tao Chen, Gang YU, and Shenghua Gao. 2023. Michelangelo: Conditional 3D Shape Generation based on Shape-Image-Text Aligned Latent Representation. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=xmxgMij3LY

Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 2017. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 900–909.

A MORE RESULTS

User Study. To address the human-centric aspects of our method, we conducted a comprehensive user study involving 30 participants (15 female, 15 male) who evaluated 20 randomly selected shapes from the Objaverse dataset. The evaluation focused on three key criteria: (1) geometric similarity, measuring how well the abstraction preserves the original 3D surface structure; (2) anthropomorphism, assessing alignment with human intuition in shape abstraction; and (3) editability, gauging usefulness for interactive editing tasks. Participants rated each abstraction on a 5-point Likert scale (1: poor, 5: excellent). As shown in Tab. 6, our method achieved superior average scores across all three metrics compared to EMS and Marching-Primitives. These results validate that our primitive-based shape abstraction scheme not only maintains geometric fidelity but also produces structures that better align with human perception and facilitate easier manipulation for editing tasks.

Table 6. User study results comparing our method with EMS and Marching-Primitives. Each score represents the average rating (on a 5-point scale) from 30 participants evaluating 20 randomly selected Objaverse shapes across three criteria. Our method consistently outperforms baseline approaches.

Method	Geometric Similarity	Anthropomorphism	Editability
EMS	2.16	2.18	2.17
MP	3.55	3.09	3.23
Ours	**4.17**	**4.18**	**4.22**

Choices of rotation representation. We opted for Euler angles as the rotation representation in our framework. This decision was driven by several considerations. SVD-based parameterizations, while mathematically elegant, are unsuitable for our application as they lack direct interpolation capabilities, which is crucial for learning-based frameworks. Quaternions, despite their popularity in computer graphics, present challenges including their non-intuitive physical meaning and the requirement for additional constraints (e.g., $w^2 + x^2 + y^2 + z^2 = 1$) to prevent numerical drift, which can complicate implementation and optimization.

Euler angles provide a more intuitive and interpretable representation with their straightforward Euclidean parameter space. Importantly, given that many practical cases in our dataset are nearly gravity-aligned, Euler angles exhibit minimal variance in their values, making the learning process more stable and efficient. To validate our choice, we conducted an empirical comparison between different rotation representations (Quaternions, Rotation Vector, and Euler Angles) as shown in Tab. 7. The results demonstrate that quaternions yield notably inferior results. Rotation vectors perform reasonably well due to their continuity and partial compatibility with gravity-aligned cases, yet still demonstrate a performance gap compared to Euler angles, which consistently deliver superior decomposition quality across our evaluation metrics.

Qualitative results of ablation study. We provide additional qualitative comparisons in Fig. 11 to demonstrate the effectiveness of our ambiguity-free scheme, cascade decoding, and Chamfer distance loss in the ablation study.

Generalization Analysis. A key contribution of our method is its ability to perform semantic-aware primitive decomposition on

Table 7. Experiments on different choices of rotation representations.

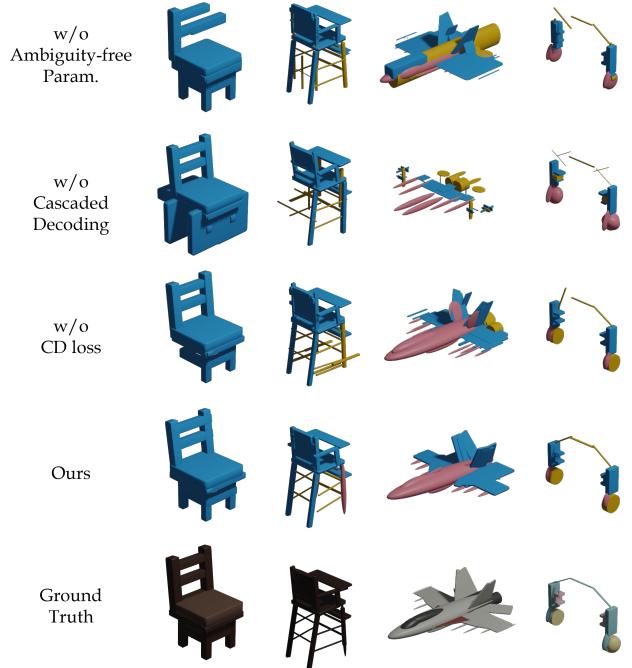| Representation	CD ↓	EMD ↓	Hausdorff ↓	Voxel-IoU ↑
Quaternions	0.0704	0.0684	0.280	0.383
Rotation Vector	0.0426	0.0494	0.163	0.477
Euler Angles	**0.0404**	**0.0475**	**0.158**	**0.484**

Fig. 11. Qualitative comparisons of ablation study.

shapes that differ significantly from those in the training data. To empirically validate this claim, we conducted a comparative analysis between test shapes and their geometrically closest counterparts in the training dataset. We employs the pointbert-vitg14 from the OpenShape [Liu et al. 2023a] for point cloud feature extraction. We then perform similarity-based object retrieval by measuring the cosine similarity between these extracted feature representations Fig. 12 illustrates this comparison. The top row displays test case shapes that were not seen during training. The middle row shows our method's primitive decomposition results, while the bottom row presents the shapes most similar to those of our training dataset. As evident from the visualization, our method successfully decomposes test shapes into semantically meaningful primitives despite substantial geometric differences from training examples.

B MORE IMPLEMENTATION DETAILS

Dataset. To address the need for high-quality shape abstractions with semantic primitives, we carefully constructed a comprehensive dataset through a systematic annotation process. Our annotators were provided with a custom 3D engine featuring an intuitive graphical user interface that enabled precise primitive selection and

Fig. 12. Generalization capability of our method. Top row: unseen test shapes; Middle row: our primitive decomposition results; Bottom row: geometrically closest shapes from the training dataset.

manipulation (including scale, rotation, and translation operations). Annotators were explicitly instructed to follow two key principles: ensure complete contour coverage of the original shapes and create abstractions that align with human perception. Fig. 12 (lower part) showcases representative samples from our dataset.
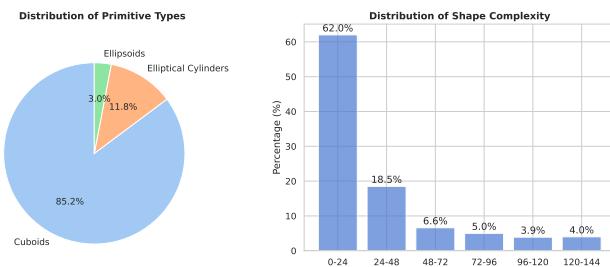

Fig. 13. Statistical characteristics of our HumanPrim dataset.

Our HumanPrim dataset exhibits diverse primitive composition characteristics, as shown in Fig. 13. Analysis reveals that 85.2% of all primitives are cuboids, 11.8% are elliptical cylinders, and 3.0% are ellipsoids. This distribution reflects the predominance of box-like structures in common objects while still incorporating curved surfaces where appropriate. In terms of complexity, our dataset shows varied primitive counts: data with 0-24, 24-48, 48-72, 72-96, 96-120, and 120-144 primitives account for 62.0%, 18.5%, 6.6%, 5.0%, 3.9%, and 4.0%, respectively. This distribution demonstrates our dataset's

balance between simple and complex shape abstractions, supporting robust learning across varying levels of geometric complexity.

Symmetry Order Calculation. To determine the total symmetry order of an axis, we account for both rotational symmetry and axis permutations that result in equivalent configurations. This approach captures all possible symmetric transformations of a primitive and is crucial for achieving ambiguity-free parameterization. Specifically, if swapping two axes after rotation achieves alignment with the original configuration, we include that rotational angle in the symmetry order count. Fig. 14 illustrates this calculation process for the x-axis of a cuboid. When considering pure rotational symmetry without axis permutations, only the 180° rotation produces a configuration that perfectly aligns with the original state, yielding a symmetry order of 2. However, our method also recognizes that after 90° and 270° rotations, swapping the y and z axes produces configurations equivalent to the original. By incorporating these axis-permutation-enabled symmetries, the total symmetry order for a cuboid around its x-axis increases to 4. The explicit inclusion of axis permutations in symmetry calculations parameterizes all possible self-symmetry cases and can apply to all primitives.

C MORE DISCUSSIONS

Limitations. Our approach exhibits several limitations despite its effectiveness. Our method struggles with certain out-of-distribution objects, particularly those with topological structures rarely seen in our training data (e.g., ring shapes with holes), as demonstrated in Fig. 15. This challenge stems partly from our current primitive types and could be addressed by expanding our primitive vocabulary and
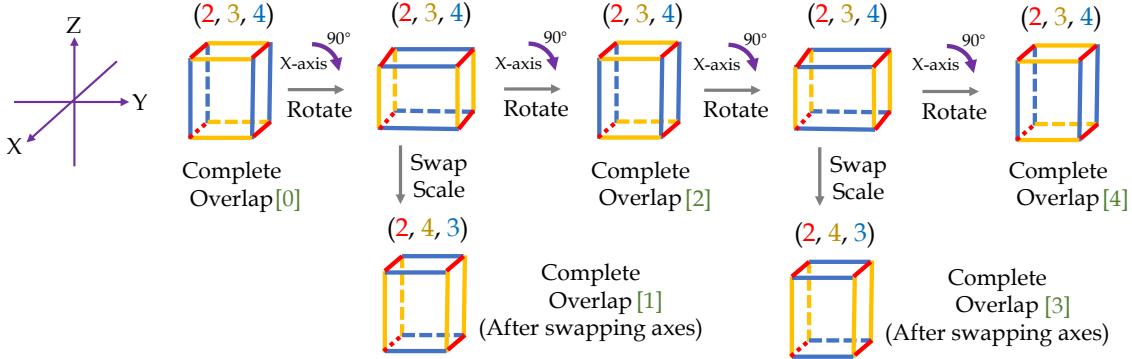
Fig. 14. Symmetry order calculation for the x-axis of a cuboid. Axis permutation (swapping y and z axes) after 90° and 270° rotations creates configurations equivalent to the original, increasing the symmetry order from 2 to 4.
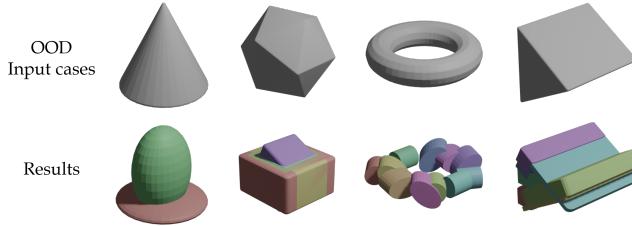

Fig. 15. Failure cases with out-of-distribution inputs.

enriching the training dataset with more diverse examples, as our framework is inherently generalizable to such extensions.

We also observe diversity in annotation styles. Some annotators tend to use more primitives than necessary to fully cover the original object, resulting in over-segmentation in certain cases. Regarding design choices of primitive attribute prediction, our discretization scheme focuses learning on larger errors and helps model convergence. However, we acknowledge that this approach introduces trade-offs, potentially causing precision loss and connectivity issues between adjacent primitives.

Symmetry constraints are also excluded to provide greater design freedom, though this sometimes results in asymmetric decompositions of inherently symmetric objects. In addition, our focus remains on geometric shape abstraction rather than appearance modeling. While textures can be handled via back-projection or nearest-neighbour matching with the original 3D object, direct texture generation is not addressed in our pipeline. Both symmetry integration and native texture synthesis represent promising directions for future work that could enhance the practical utility of our method.

Abstraction level of annotations. The question of appropriate abstraction levels is central to semantic shape decomposition. Annotators were instructed to ensure complete surface coverage while adhering to human-aligned construction principles, which naturally produces varying primitive counts across different 3D objects. This variation reflects inherent complexity rather than enforcing arbitrary consistency. Unlike optimization-based methods that often fragment semantic parts into multiple pieces with significant overlaps, our approach preserves semantic coherence while maintaining geometric accuracy. This balance stems from our annotation guidelines that prioritize human interpretability alongside geometric fidelity.

Different applications, however, may require different abstraction levels. While detailed decompositions might benefit precise editing tasks, coarser representations could better serve classification or retrieval applications. Though explicitly instructing annotators to provide multiple abstraction levels is challenging, future work could explore inferring these levels based on primitive counts, potentially enabling adaptive abstractions tailored to downstream tasks.

Difference with other abstraction paradigms. Shape abstraction has evolved along several distinct approaches in the literature, each offering unique perspectives on how to represent 3D objects efficiently and meaningfully. Our work contributes to this field through primitive-based shape abstraction, while other paradigms exist, such as hierarchical representations, skeletal abstractions, and surface simplification techniques.

Hierarchical representations like GRASS [Li et al. 2017] organize shapes into structured trees capturing part relationships and symmetries. While these approaches excel at representing organization, our method offers more direct geometric interpretability through explicit primitive decomposition. Medial Skeletal Diagram [Guo et al. 2024] similarly seeks sparse representations, but replaces discrete skeletal elements with continuous primitives, whereas we maintain a clearer separation between structural abstraction (through primitive arrangement) and geometric representation. Compared to mesh simplification techniques [Garland and Heckbert 1997] that preserve surface details through vertex/edge removal, our primitive-based abstraction operates at a higher semantic level. We don't just simplify geometry - we reconstruct shapes using fundamental building blocks that naturally align with how humans perceive object structure. This also differs from optimisation-based fitting methods that may over-segment shapes.