BOXSPLITGEN: A Generative Model for 3D Part Bounding Boxes in Varying Granularity

Juil Koo[1*] Wei-Tung Lin[2*] Chanho Park[1] Chanhyeok Park[1] Minhyuk Sung[1]
[1]KAIST [2]NVIDIA
{63days,charlieppark,chpark1111,mhsung}@kaist.ac.kr weitungl@nvidia.com

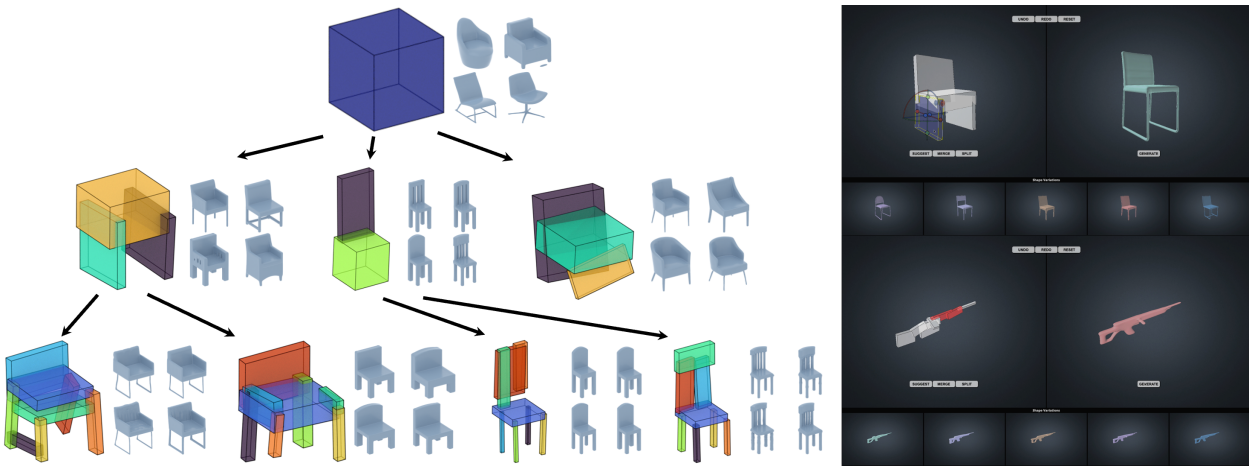


Figure 1. **An overview of box-splitting-based 3D shape generative framework.** The left shows our iterative box splitting and box-to-shape generation, where diverse shapes at the top of the tree become increasingly specific deeper in the tree. The right showcases our user-interactive box and shape editing demo.

Abstract

Human creativity follows a perceptual process, moving from abstract ideas to finer details during creation. While 3D generative models have advanced dramatically, models specifically designed to assist human imagination in 3D creation–particularly for detailing abstractions from coarse to fine–have not been explored. We propose a framework that enables intuitive and interactive 3D shape generation by iteratively splitting bounding boxes to refine the set of bounding boxes. The main technical components of our framework are two generative models: the box-splitting generative model and the box-to-shape generative model. The first model, named BOXSPLITGEN*, generates a collection of 3D part bounding boxes with varying granularity by iteratively splitting coarse bounding boxes. It utilizes part bounding boxes created through agglomerative merging and learns the reverse of the merging process—the splitting sequences. The model consists of two main components: the first learns the categorical distribution of the box to be split, and the second learns the distribution of the two new boxes, given the set of boxes and the indication of which box to split. The second model, the box-to-shape generative model, is trained by leveraging the 3D shape priors learned by an existing 3D diffusion model while adapting the model to incorporate bounding box conditioning. In our experiments, we demonstrate that the box-splitting generative model outperforms token prediction models and the inpainting approach with an unconditional diffusion model. Also, we show that our box-to-shape model, based on a state-of-the-art 3D diffusion model, provides superior results compared to a previous model.*

[*]Equal contribution.

1. Introduction

In recent years, significant progress has been made in 3D generative models, promising production-level quality in the near future. This success is largely driven by the rise of diffusion models and their applications across various 3D shape representations, including latent representations [11, 20, 33, 68], point clouds [2], voxels [29, 64], meshes [3, 32], B-reps [65], and Gaussian splats [49]. As we witness these remarkable advancements in 3D generation, the next frontier lies in enhancing the *controllability* of the generation process.

For conditional generation, text prompts have become the most popular conditioning input for both 2D image [6, 14, 16, 35, 46, 58, 62] and 3D shape generation [20, 39, 50, 71]. However, they offer limited controllability, particularly when it comes to spatial guidance. As an alternative, bounding box grounding has been explored for both 2D [25, 31] and 3D guided generation [50], offering notable benefits, such as ease of manipulation by users and its effectiveness in abstracting shapes and providing spatial guidance.

Particularly for 3D shapes, the key benefit of primitive-based abstraction lies in its ability to represent part-level structures [15, 18, 21, 26, 42, 54, 67] while effectively encoding *hierarchical* relationships across parts [26, 42, 44, 54]. Notably, foundational studies from the 1970s and 1980s [4, 36, 61] revealed that the human visual system perceives 3D objects as hierarchically organized sets of primitives, progressing systematically from coarse to fine granularity. This hierarchical framework resonates with human creativity, where the imaginative process similarly unfolds through structured, incremental levels of detail.

Building on these studies, we propose a user-interactive 3D generation framework that reflects the hierarchical nature of human imagination. Figure 1 and the **supplementary video** demonstrate the user-interactive generation examples using part bounding boxes. Our framework enables users to create 3D objects starting with a rough design represented by coarse bounding boxes, progressively adding more detail by splitting a bounding box into smaller, finer-grained boxes. This process allows users to explore diverse shapes using coarse bounding boxes (intermediate nodes in the tree shown in Figure 1) while specifying the design by increasing the level of granularity (bottom nodes). The iterative approach helps users efficiently create detailed and desired 3D shapes with minimal effort. In the interface shown on the right in Figure 1 and the **supplementary video**, users can transform bounding boxes and adjust their granularity by splitting them into finer components.

Creating such a framework requires two generative models: a box-splitting generative model and a box-to-shape generative model. The purpose of the first model is to generate two finer-grained bounding boxes by splitting a coarse bounding box. Training this generative model necessitates a dataset containing sequences of box-splitting results. While extensive research has been conducted on extracting part-level primitives from raw 3D shapes [10, 12, 43, 45, 57, 67], most previous methods are unable to generate bounding boxes at various granularity or capture their hierarchical relationships. Some prior works have introduced hand-crafted datasets [37], but these are limited in scalability and diversity across different shape categories. To address the lack of training data, we focus on leveraging a recent method [42] that generates bounding boxes for parts through hierarchical merging, starting from super-segments. This agglomerative merging approach naturally produces a set of bounding boxes with varying levels of granularity, down to a single bounding box, along with their two-to-one merging relationships. Our goal is to learn the reverse process of merging.

We model the reverse process of hierarchical merging, iterative splitting, using a generative model named BoxSplitGen. While iterative splitting can be viewed as a sequential process, it has unique characteristics that make it incompatible with typical sequence generation models, such as GPT [5, 48]. First, the generation of the next token (bounding box) is conditioned not only on the previous tokens but also on the selection of the box to be split. Second, and more importantly, the selected box is removed after splitting. Consequently, the set of bounding boxes at an intermediate step is not a subset of those at the final step, making GPT-like models unsuitable for this task. To address this, we propose a two-step autoregressive model that leverages a classifier and a diffusion model. In the first step, we learn the distribution of boxes to be selected for splitting using a classification network. In the second step, we use a conditional diffusion model, where the given set of boxes, along with an indication of the box to split, is used as conditioning input to generate the two split boxes.

The second generative model in our framework is a bounding-box-conditioned 3D shape generative model. To achieve this, we finetune 3DShape2VecSet [68] to incorporate bounding box conditioning using the ControlNet [35] approach, while preserving the high-quality 3D shape priors learned by 3DShape2VecSet. ControlNet requires the representations of the denoised data and the input condition to be in the same format. Previous work [50], which uses Shape-E [20] as a base 3D diffusion model, addresses this by converting bounding box representations into multi-view images and processing them with a pretrained encoder. In contrast, we propose a simpler yet effective approach: directly encoding the bounding boxes into the latent representation using a learnable encoding layer, which is jointly trained with the ControlNet branch.

In our experiments on the ShapeNet [7] dataset, we compare our method with several baselines for both box-splitting generation and bounding-box-conditioned shape generation. For box-splitting, we evaluate against a to-

ken prediction model and an inpainting approach using an unconditional diffusion model that preserves existing boxes while filling two new ones. Our conditional diffusion model achieves the best performance, while the inpainting baseline shows comparable but slightly inferior results. For bounding-box-conditioned shape generation, we compare with Spice-E [50], which uses Shape-E [50] instead of our 3DShape2VecSet, and with a finetuned version of 3DShape2VecSet [68] that replaces ControlNet with a Gated Mechanism. Our model outperforms these alternatives in both the quality of generated shapes and their alignment with the input bounding boxes.

2. Related Work

3D Shape Abstraction Methods There is a substantial body of prior work focused on abstracting raw 3D shapes into collections of basic shape representations, such as cuboids [21, 42, 55, 57, 67], superquadrics [43], primitives [24, 27, 30, 70], and implicit fields [9, 10, 12, 41, 45]. While these methods effectively extract primitives that represent parts of a given 3D shape, most do not capture the hierarchical relationships across these parts. However, a few approaches have been proposed that jointly extract both the primitives and their hierarchical structures. Sun *et al*. [55] present a method for extracting cuboids in a hierarchical structure while enforcing inclusion relationships across levels. However, the method is limited to three levels of hierarchy, which restricts the range of granularities from coarse to fine. In contrast, SMART [42] offers an iterative approach for agglomeratively merging super-segments into more abstract bounding boxes. Since SMART starts with fine-level super-segments, it allows for a much wider range of granularities. Thus, as training data for our generative model, we use the outputs of SMART while progressively merging bounding boxes until a single box remains. Our generative framework learns the reverse process of this iterative merging: splitting a bounding box into two.

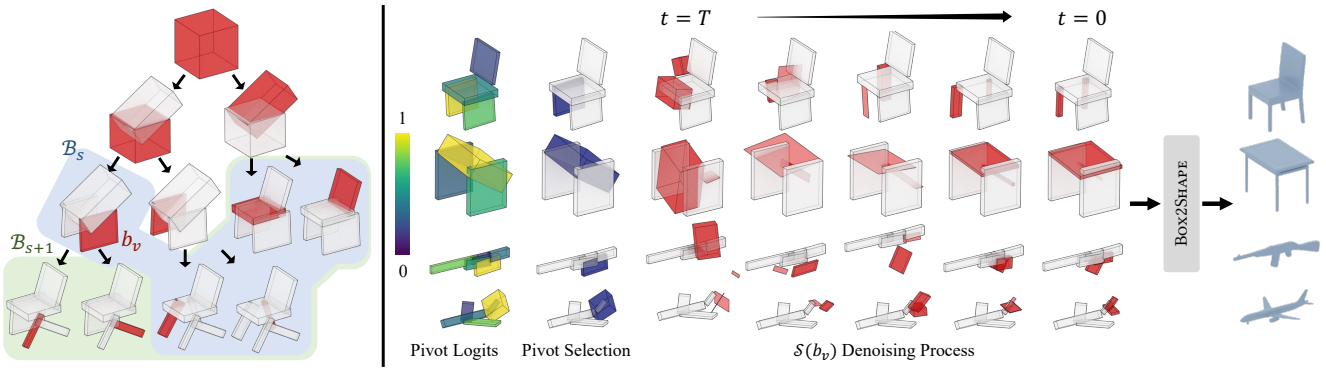
Shape Structure Generative Models Generative models for 3D shape structures have been explored in previous work, but they often face limitations in scalability or in controlling the number of primitives and the granularity of the abstraction. StructureNet [37] and GRAINS [28] are methods for generating sets of part-level bounding boxes for individual objects and object-level bounding boxes for scenes, respectively. Both approaches use recursive neural networks to train variational autoencoders (VAE) for learning N-ary hierarchies. However, these methods rely heavily on annotated datasets such as PartNet [38] and SUNCG [52], which limit their scalability. SPAGHETTI [17], SALAD [23], and DiffFacto [22] represent another line of work that does not require hand-crafted datasets for training. Instead, these methods generate part-level structures by learning from data obtained in an unsupervised manner [23], or even while jointly learning the part-level structure [17, 22]. However, these methods are limited in their ability to vary the number of parts. GRASS [26], SAGNet [63], DSG-Net [66], and PASTA [53] are notable examples that generate varying numbers of part bounding boxes without the need for hand-crafted datasets. However, these methods do not allow for adjusting the granularity of the part-level representations. To the best of our knowledge, we are the first to propose a generative model for 3D part bounding boxes that allows for controlling the granularity.

Structure-Conditioned Shape Generative Models Structural 3D shape abstraction has been used as user-controllable guidance in 3D object generation. For instance, Neural Template [19] utilizes part-wise latent codes as conditions for 3D shape generation. SALAD [23] is a conditional diffusion model that takes a set of Gaussian blobs representing parts as input conditions. Note that the aforementioned methods do not allow for changes in the granularity of the abstract representation, unlike our approach. Spice-E [50] is the most recent generative model similar to ours, using a set of bounding boxes as a condition. It employs Shape-E [20] as a base unconditional 3D diffusion model and leverages the ControlNet [35] approach to transform it into a conditional model that incorporates bounding boxes. Similar to Spice-E, we propose a bounding-box-to-shape generative model but build it on a more advanced 3D diffusion model, 3DShape2VecSet [68]. While Spice-E represents bounding boxes as multi-view images and encodes them using Shape-E's pretrained encoder, we propose a simpler yet effective approach: introducing an encoder layer that directly maps input bounding boxes to the latent representation of 3DShape2VecSet, training it jointly with the ControlNet branch. Our experiments demonstrate that this novel box-to-shape generative model produces higher-quality 3D shapes than Spice-E, benefiting from the superior generative capabilities of 3DShape2VecSet.

3. BoxSplitGen: Box-Splitting Generative Model

3.1. Problem Definition and Overview

Our objective is to learn a generative model for sets of 3D bounding boxes as shape abstractions, which capture both a diverse collection of shapes and varying levels of granularity for each shape. To achieve this, we represent an arbitrary 3D shape using hierarchical shape abstractions [42] structured as a *binary tree*, as illustrated in Figure 2. The root node of the binary tree is a single unit cube that completely encloses any arbitrary shape. As we traverse deeper into the tree, each internal node splits into two child nodes through *splitting operation*. This operation refines the abstraction



(a) Binary Tree for Hierarchical Shape Abstraction (b) Overall Pipeline of BoxSplitGen (c) Box-to-Shape Generation

Figure 2. **Overview of our hierarchical bounding box splitting framework.** On the left is a binary tree for 3D shape abstraction, where red-highlighted nodes b_v are split into finer child nodes, with blue and green backgrounds showing split steps at s and $s+1$. On the right, the framework performs pivot classification, samples two red-highlighted child boxes, and generates 3D shapes using our Box2Shape model.

by subdividing a chosen box, called the pivot b_v, into two child boxes denoted by $\mathcal{C}(b_v)$. Formally, at a given split step s in the tree structure, we have a set of 3D bounding boxes $\mathcal{B}_s = \{b_i\}_{i=1}^N$. Each split step can be expressed as $\mathcal{B}_{s+1} := \mathcal{B}_s \setminus \{b_v\} \cup \mathcal{C}(b_v)$, where the coarser pivot box is replaced with two newly generated child boxes with finer details (See Figure 2). This split process provides progressively more detailed approximations of the input shape. At the finest level of detail, a collection of the leaf nodes in the tree represents the most detailed shape abstraction. This hierarchical representation thus can provide diverse shape abstractions at any level of granularity, from a single coarse cube at the root to a detailed collection of boxes at the leaves.

Given this tree structure, we design the generative process of the bounding box sets $\mathcal{B}_s$ at each split step s as a Markov process with conditional probability distributions $p(\mathcal{B}_{s+1}|\mathcal{B}_s)$, which models the splitting operation. The conditional probability distribution $p(\mathcal{B}_{s+1}|\mathcal{B}_s)$ is further decomposed into two distributions: 1) the distribution of the pivot box, and 2) the distribution of the two child boxes from the selected pivot box:

$$p(\mathcal{B}_{s+1}|\mathcal{B}_s) = p(b_v|\mathcal{B}_s)p(\mathcal{C}(b_v)|b_v, \mathcal{B}_s), \tag{1}$$

where $b_v \in \mathcal{B}_s$ is one of the bounding boxes in $\mathcal{B}_s$ selected as the pivot. In our method, named BoxSplitGen, we learn the probability distribution of the pivot bounding box $p(b_v|\mathcal{B}_s)$ using a classification network, while learning the probability distribution of the two child boxes $p(\mathcal{C}(b_v)|b_v, \mathcal{B}_s)$ using a diffusion model.

Note that while our generative process is autoregressive, it cannot be modeled using typical sequence generation models, such as GPT-based architectures (as utilized in recent mesh generation works [40, 51, 56]), for the following reasons. First, the generation of the next token (bounding box) is conditioned on the selection of a pivot box, which necessitates the use of a pivot classifier. Second, the pivot box is removed in the subsequent step, resulting in a binary tree at intermediate granularity that is not a subtree of the binary tree at the finest granularity. As a result, binary trees at arbitrary granularity cannot be generated simply by sequentializing the finest granularity binary tree and feeding it into GPT.

Below, we first describe the data preparation process for training (Section 3.2). Next, we explain how the two models in our framework, pivot classifier and child-boxes diffusion model, are implemented (Sections 3.3 and 3.4).

3.2. Data Preparation

Our training data consists of hierarchical shape abstractions generated using SMART [42]. Given an initial set of over-segmented bounding boxes $\mathcal{B}_S = \{b_i\}_{i=1}^N$, SMART iteratively performs bottom-up merging. In each iteration, it selects two boxes and merges them into a single parent box that tightly encloses the combined region. This parent box serves as a pivot b_v in our hierarchy, with the two merged boxes becoming its children $\mathcal{C}(b_v)$. SMART continues the merging process until reaching to an appropriate number of bounding boxes that best describe the given shape. These boxes are used as the leaf nodes in our binary tree; refer to Table S2 for the statistics on the number of leaf node boxes. Based on the leaf node boxes (SMART outputs), we further proceed with an iterative merging process until only a single box remains, building the binary tree up to the root. We set the root box to always be a unit cube; all raw shapes are normalized to fit within the unit cube. Each 3D bounding box $b_i = \{c_i, s_i, o_i\} \in \mathbb{R}^{15}$ is parameterized by center $c_i \in \mathbb{R}^3$, side lengths $s_i \in \mathbb{R}^3$, and a flatten orientation matrix $o_i \in \mathbb{R}^9$.

3.3. Pivot Classifier

The first component of BOXSPLITGEN is the pivot classifier, which models the categorical distribution of the pivot bounding box given set of the bounding boxes ($p(b_v|\mathcal{B}_s)$ in Equation 1). We adopt a Transformer-based architecture [47] to handle variable-sized input sets, where each box is encoded as a token in the latent space and processed through self-attention layers [60].

3.4. Child-Boxes Diffusion

The second component of BOXSPLITGEN is the child-boxes diffusion model, a conditional diffusion model that learns the distribution of the two child bounding boxes $\mathcal{C}(b_v)$ given the input bounding boxes $\mathcal{B}_s$ and the selected pivot b_v ($p(\mathcal{C}(b_v)|b_v, \mathcal{B}_s)$ in Equation 1). The noise prediction network $\boldsymbol{\epsilon}_\theta$ of the diffusion model is trained with the condition-output pairs extracted from consecutive box sets $\mathcal{B}_s$ and $\mathcal{B}_{s+1}$ in the training dataset:

$$\mathbf{x}_0 = \mathcal{C}(b_v) \in \mathbb{R}^{2\times 15} \tag{2}$$

$$\mathcal{L} = \mathbb{E}\|\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t; b_v, \mathcal{B}_s) - \boldsymbol{\epsilon}\|, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

where $\mathbf{x}_t$ is obtained through the forward process of diffusion models: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$. The network of $\boldsymbol{\epsilon}_\theta$ consists of a Transformer encoder $\mathcal{E}_\theta$ taking the input condition $\mathcal{B}_s$ and b_v, and a decoder $\mathcal{D}_\theta$ predicting the output noise. In the encoder $\mathcal{E}_\theta$, the input bounding boxes $\mathcal{B}_s$ are augmented with an indicator bit marking the pivot box b_v, and then processed through self-attention layers to produce a latent vector $\mathbf{h} = \mathcal{E}(\mathcal{B}_s, b_v, |\mathcal{B}_s|) \in \mathbb{R}^{|\mathcal{B}_s|\times D}$; the number of input boxes $|\mathcal{B}_s|$ is also fed as an additional input. The decoder $\mathcal{D}_\theta$ then predicts the injected noise for the two noisy boxes $\mathbf{x}_t \in \mathbb{R}^{2\times 15}$ through the cross-attention between $\mathbf{x}_t$ and $\mathbf{h}$. See Figure 3 (a) for details of the architecture. More implementation details are provided in **Supplementary** S.7.

3.5. Alternative: Inpainting with Unconditional Model

The conditional generation task performed by our Child-Boxes Diffusion can also be seen as a *completion* task, where missing parts are generated while keeping the given parts fixed. Previous work [34] has demonstrated that the completion task (inpainting for images) can also be achieved using an unconditional diffusion model by combining one-step denoising outputs for the missing parts with forward process outputs for the given parts at each denoising step. We also explore this option by training another diffusion model that uses only the number of bounding boxes as its only condition. The details of this alternative approach are discussed in Section 5.1, where we evaluate it as one of the baselines. While this approach shows comparable performance, it yields slightly inferior results compared to the conditional model.

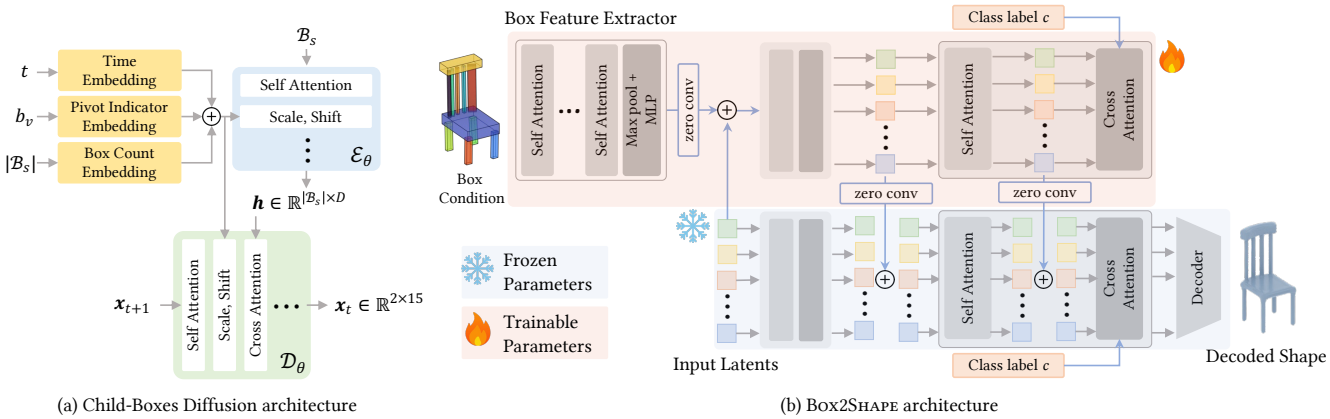
4. BOX2SHAPE: Box-to-Shape Generative Model

Next, we propose a bounding-box-conditioned 3D shape generative model named BOX2SHAPE. In contrast to the aforementioned bounding box generative model trained from scratch, here we aim to fully leverage the 3D shape priors learned by the state-of-the-art 3D diffusion model, 3DShape2VecSet [68]. By finetuning this model, we ensure high fidelity in the generated shapes while effectively incorporating bounding box conditioning across varying levels of granularity.

A crucial requirement for finetuning pretrained networks to incorporate additional input conditions is smooth adaptation to conditional generation: the network should initially retain its original generation quality while gradually adapting to the specified condition. Two representative approaches for achieving this are the Gated Mechanism [1, 25, 31, 50] and ControlNet [8, 35].

The Gated Mechanism introduces trainable layers gated by a gate parameter initialized to zero, ensuring unchanged output at the start of finetuning. However, it updates relatively few parameters, leading to slower convergence. In contrast, ControlNet duplicates existing layers, prepends a 1×1 zero-convolution layer with weights and biases initialized to zero, and merges outputs via residual connections. This approach updates more parameters, enabling faster and more effective convergence compared to the Gated Mechanism. See Section 5.2 for a detailed comparison of these two approaches. A notable example of adapting ControlNet to a 3D generative model is Spice-E [50], which uses Shape-E [20] as its backbone diffusion model whose generation quality is inferior to 3DShape2VecSet [68].

Adapting ControlNet [8, 35] to our setup presents a challenge due to a mismatch in data representation between the input and the condition, as ControlNet requires these to be aligned. In our case, the backbone diffusion model, 3DShape2VecSet [68], represents 3D shapes as unordered latent sets $\mathbf{z} \in \mathbb{R}^{M\times D}$, while the input condition is a set of bounding boxes. Spice-E [50] addresses this issue by representing input bounding boxes in the same format as their 3D shapes—multi-view images in their case—and encoding them using a pretrained encoder. For 3DShape2VecSet, we find that a simpler approach performs effectively: incorporating a trainable encoding layer, $\mathcal{F}(\cdot)$, which directly processes the input bounding boxes and outputs latent sets without relying on a pretrained encoder. This module is trained jointly with the ControlNet branch. The architecture is illustrated in Figure 3 (b). Our experiments detailed in Section 5 demonstrate that our box-to-shape generative model based on 3DShape2VecSet outperforms Spice-E [50] due to its strong priors for generating high-fidelity shapes.



(a) Child-Boxes Diffusion architecture

(b) BOX2SHAPE architecture

Figure 3. **Diagram of network architectures.** (a) Child-Boxes Diffusion. (b) BOX2SHAPE. Starting from a unit cube, we iteratively split boxes using Child-Boxes Diffusion to obtain the box condition with desired granularity, which then guides BOX2SHAPE to generate aligned 3D shapes.

5. Experiment Results

We evaluate both components of our framework—the box-splitting model and the box-conditioned shape generation model—through qualitative and quantitative results using the ShapeNet [7] dataset. Please see **Supplementary** S.2 and the **supplementary video** for our user-interactive shape generation demo and shape editing results. Additional experiment details are provided in **Supplementary** S.4 and S.5.

5.1. Box-Splitting Generation

Baselines. As discussed in Section 3.1, our sequential generative process exhibits a unique characteristic. As outlined in Equation 1, each step can be decomposed into (1) selecting and removing one existing element ($p(b_v|\mathcal{B}_s)$) and (2) producing two new elements at a time ($p(\mathcal{C}(b_v)|b_v, \mathcal{B}_s)$). Given the novelty of this problem, we extensively explore potential generative approaches. Specifically, we use the same pivot classifier to model $p(b_v|\mathcal{B}_s)$, while exploring different methods for modeling $p(\mathcal{C}(b_v)|b_v, \mathcal{B}_s)$. This includes our Child-Boxes Diffusion, a conditional diffusion model, as well as two baselines:

- **Conditional Token Prediction Model**: We propose an approach based on sequence generation models that emulates the splitting process, departing from the typical sequential prediction of a single token at a time. The architecture is similar to the conditional diffusion model, taking $\mathcal{B}_s$ and b_v as conditions to generate two child boxes $\mathcal{C}(b_v)$). However, the input and output representations are replaced with discretized representations, as is typical in GPT-like models. To enable the network to model a categorical distribution over tokens, we first train a VQ-VAE [59] to encode continuous 15-dimensional box features into a discrete token space with a vocabulary size of $|V| = 16K$. Unlike the conditional diffusion model, which predicts noise $\boldsymbol{\epsilon} \in \mathbb{R}^{2\times 15}$, this model outputs logits for two elements $\{l_1, l_2\} \in \mathbb{R}^{2\times|V|}$. The tokens sampled from the predicted categorical distributions are then decoded into the original continuous box parameters using the pre-trained VQ-VAE.
- **Unconditional Diffusion Model with Inpainting**: This model is trained to generate a varying number of complete bounding boxes without the pivot box as a condition but only the box count. During inference, we perform an inpainting technique. Specifically, given the set of input bounding boxes $\mathcal{B}_s$ and the pivot box $b_v \in \mathcal{B}_s$, we first duplicate b_v, increasing the total number of boxes to $|\mathcal{B}_s| + 1$, and then perform the DDIM inversion [13], obtaining the standard normal sample $\mathbf{x}_T$ from the input boxes. Next, we reset the portion of $\mathbf{x}_T$ corresponding to the duplicates of b_v to random standard normal samples. Then, we perform RePaint [34] while treating $\mathcal{B}_s \setminus \{b_v\}$ as the background to guide the inpainting process. Unlike our conditional diffusion model, which explicitly uses the pivot box as a condition, the limitation of this approach is that it cannot leverage information about the pivot box, as it performs inpainting after the pivot box is removed.

For further details on the baselines, please refer to **Supplementary** S.6.

Quantitative Evaluation. To quantitatively evaluate the quality and diversity of the generated bounding boxes, we measure Coverage (COV), Minimum Matching Distance (MMD), and 1-Nearest Neighbor Accuracy (1-NNA) between the set of generated bounding boxes and the reference bounding box set. We use the training bounding box set as the reference set and generate 2,000 bounding boxes for each class by iteratively splitting them. We then compare the resulting boxes at various split levels to the reference set.

For a comprehensive analysis, we evaluate the metrics separately at two split steps, $s = 5$ (coarser) and $s = 8$ (finer), and then take their average, since these steps fall

Table 1. **Quantitative comparison of shape abstraction generation with different pivot selection strategies.** MMD-CD scores and MMD-EMD scores are scaled by 10^3 and 10^2, respectively. The best results are highlighted in **bold**. The numbers are the averages across split levels $s = 5$ and $s = 8$.

Pivot Selection	Models	COV ↑		MMD ↓		1-NNA ↓		COV ↑		MMD ↓		1-NNA ↓	
		CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD
		Chair						Airplane					
Random	Token Pred. Model	22.05	25.76	27.960	21.335	94.13	93.02	55.25	58.80	10.645	15.680	90.27	90.08
	Uncond. Diffusion	29.70	30.97	18.784	18.128	88.68	87.93	63.82	68.21	7.207	12.953	88.50	87.90
	Cond. Diffusion	**32.84**	**33.97**	**16.896**	**16.910**	**85.28**	**83.43**	**77.75**	**77.38**	**6.842**	**12.599**	**86.35**	**85.64**
Classifier	Token Pred. Model	27.41	30.87	24.615	20.235	91.39	90.19	66.87	70.41	8.185	13.870	87.16	86.49
	Uncond. Diffusion	33.03	35.68	18.174	17.598	88.53	85.77	71.76	77.75	6.811	12.505	86.15	85.55
	Cond. Diffusion	**46.08**	**47.08**	**14.166**	**15.580**	**75.87**	**73.33**	**82.89**	**80.56**	**6.478**	**12.188**	**85.62**	**84.81**
		Table						Rifle					
Random	Token Pred. Model	18.77	17.24	32.175	22.580	90.77	92.59	57.42	58.67	3.400	9.475	87.18	86.72
	Uncond. Diffusion	25.05	25.89	21.840	18.784	83.79	84.94	**66.46**	**70.48**	3.015	8.813	**84.88**	**84.86**
	Cond. Diffusion	**30.17**	**30.58**	**14.229**	**15.132**	**78.22**	**79.22**	64.95	68.97	**2.716**	**8.285**	87.28	85.65
Classifier	Token Pred. Model	25.45	24.47	25.895	19.680	87.47	88.72	68.72	71.86	3.495	9.230	84.28	82.67
	Uncond. Diffusion	30.36	31.68	17.726	17.104	81.96	82.26	**75.38**	**79.02**	3.110	8.717	**79.17**	**79.96**
	Cond. Diffusion	**36.79**	**37.96**	**12.314**	**14.218**	**71.38**	**72.82**	74.75	75.25	**2.607**	**8.076**	82.86	81.96

Table 2. **Quantitative comparison of box-conditioned shape generation.** MMD-CD scores and MMD-EMD scores are scaled by 10^3 and 10^2, respectively. The best results are highlighted in **bold**.

Models	COV ↑		MMD ↓		1-NNA ↓		Box Alignment			
	CD	EMD	CD	EMD	CD	EMD	Box-CD ↓	Box-EMD ↓	TOV ↓	VIoU ↑
Spice-E [50]	38.33	40.93	14.762	15.308	87.09	85.98	0.043	0.255	2.44	0.27
Gated 3DS2V [68]	**54.55**	**57.42**	13.067	14.991	78.93	77.14	0.012	0.143	1.39	0.17
Box2Shape (Ours)	51.06	50.99	**10.369**	**12.324**	**72.75**	**72.16**	**0.006**	**0.098**	**1.08**	**0.31**

within the average bounding box count. More detailed results for each split step and evaluation setups can be found in **Supplementary** S.4.

Table 1 shows the effectiveness of both our pivot classifier and our Child-Boxes Diffusion. Compared to random pivot selection, our pivot classifier improves overall box-splitting performance, enhancing both the quality and diversity of the produced abstractions. Under the same pivot selection strategy, BOXSPLITGEN consistently outperforms them in most metrics, demonstrating its robustness in modeling conditional distribution $p(\mathcal{C}(b_v)|\mathcal{B}_s, b_v)$.

Qualitative Comparison. Figure 4 presents a qualitative comparison across different classes. Refer to **Supplementary** S.12 for more qualitative results. As illustrated, our conditional diffusion model produces diverse bounding boxes that capture fine details of the 3D structure across various classes. On the other hand, the conditional token prediction model often struggles to produce plausible shape abstractions, highlighting the difficulty of modeling the splitting process using sequential token prediction models. Similarly, the inpainting approach with unconditional diffusion models often generates boxes that miss parts of the shape. It demonstrates that enforcing the other remaining boxes to remain fixed during the denoising process can easily cause deviations from the learned data manifold of diffusion models.

5.2. Box-Conditioned Shape Generation

Baselines. We compare our method with two baselines:

- **Spice-E** [50]: A recent box-conditioned shape generative model. We finetune a pre-trained model on our dataset to process bounding boxes at varying granularity.
- **Gated 3DShape2VecSet** [68]: A variant of our model finetuned with the Gated Mechanism [1, 25, 31] instead of ControlNet. Cross-attention layers gated by box features are injected into each Transformer block, while the rest of othe model is frozen.

Quantitative Evaluation. For quantitative evaluation, we use the same metrics as in the box-splitting stage—COV, MMD, and 1-NNA—to assess the fidelity and the diversity of generated meshes, but using 3D shapes as the reference set. Additionally, following SMART [42], we evaluate the alignment between the input shape abstraction and its generated 3D shape using the following metrics: Total Outside Volume (TOV), Volumetric Intersection over Union (VIoU), Box-CD and Box-EMD. TOV and VIoU measure box alignment based on mesh volumes, while Box-CD and Box-EMD assess geometric alignment using CD and EMD. See **Supplementary** S.5 for more details on the evaluation metrics

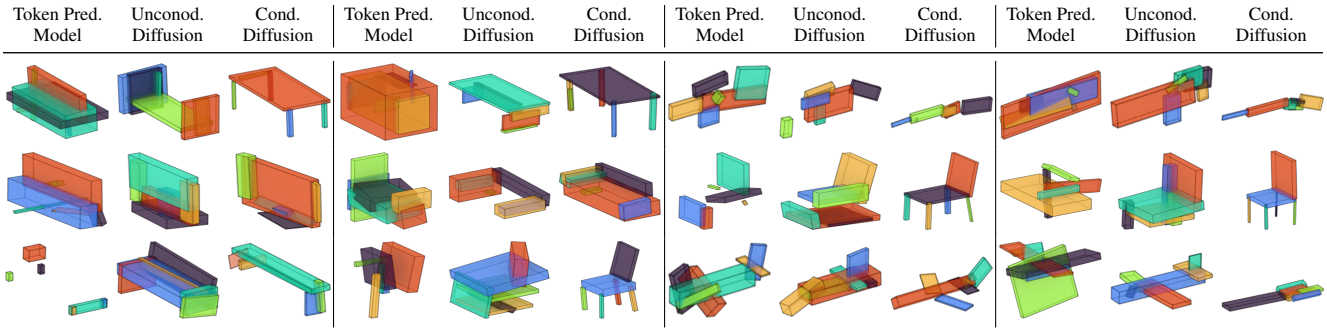


Figure 4. **Qualitative comparison of shape abstraction generation.** For each pair of columns, we query the ground truth shape and retrieve the closest generated boxes measured with chamfer distance. Our method demonstrates higher-fidelity boxes.

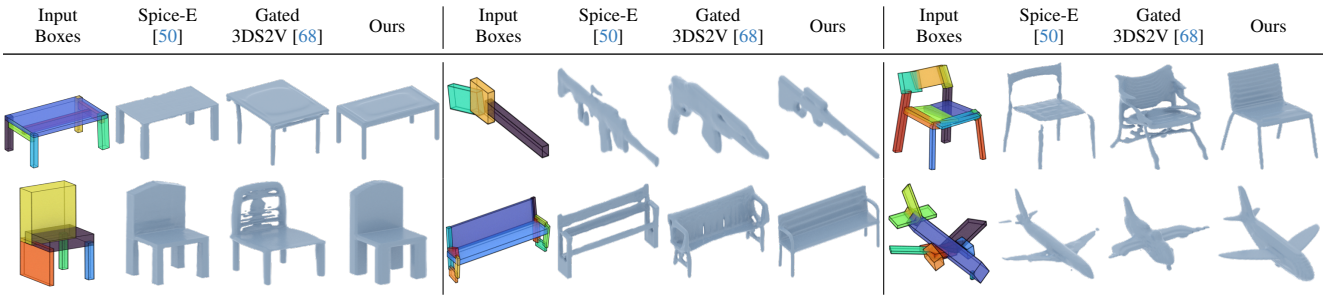


Figure 5. **Gallery of our generated bounding boxes and their final generated 3D shapes by box-conditioned shape generation models.** Each pair of columns shows the input bounding boxes (left) and their corresponding generated 3D shapes (right).

and setups.

Table 2 presents a quantitative comparison between our method and the baselines. Our method BOX2SHAPE outperforms others in shape fidelity and diversity metrics—COV, MMD, and 1-NNA—with a significant advantage in 1-NNA. While Gated 3DShape2VecSet achieves higher COV, this is due to its lack of adaptation to box-conditioned generation, often producing shapes deviated from the input boxes. This limitation is further reflected in its lacking box alignment performance across all box alignment metrics: Box-CD, Box-EMD, TOV, VIoU. Spice-E [50] demonstrates suboptimal performance compared to BOX2SHAPE, largely due to its backbone model's limited shape prior. These results demonstrate that our conditioning approach achieves superior box alignment while preserving the original model's superior performance to generate high-fidelity and diverse shapes.

Qualitative Results. Figure 5 presents a qualitative comparison, where each pair of columns shows the input bounding boxes and their generated 3D shapes by different approaches. Refer to **Supplementary** S.13 for more results. Given an input set of bounding boxes, ours produces a plausible 3D shape while being well-aligned with the input boxes. In contrast, Spice-E [50] often fails to capture fine-grained details in the 3D shapes due to its backbone's suboptimal performance. Compared to Gated 3DShape2VecSet, which is based on the Gated Mechanism, it often struggles to produce box-aligned 3D shapes, highlighting the limitations of the Gated Mechanism in modeling the conditional probability distribution with our diverse conditioning bounding boxes.

6. Conclusion

We presented a box-splitting-based interactive 3D shape generation framework composed of two generative models. The first model, BOXSPLITGEN, is an autoregressive model that enables the progressive refinement of bounding boxes via splitting. We introduce a pivot classifier and a child-box diffusion model to select which box to split and to generate the two new boxes, respectively. The second model is a box-to-shape generative model that effectively adapts a pre-trained unconditional 3D diffusion model. We demonstrate that the proposed framework facilitates intuitive 3D generation by mimicking the human imagination process from abstract concepts to detailed structures. Users can split and manipulate bounding boxes to generate aligned 3D shapes, with diversity naturally decreasing as the bounding boxes become fine-grained.

As future work, we plan to improve our user-interactive 3D shape generation framework to incorporate additional spatial guidance, including other primitives [17, 23, 57] and 3D sketches [69].

Acknowledgements. This work was supported by the IITP grants (RS-2022-00156435, RS-2024-00399817, RS-2025-25441313, RS-2025-25443318, RS-2025-02653113); and the Industrial Technology Innovation Program (RS-2025-02317326), all funded by the Korean government (MSIT and MOTIE), as well as by the DRB-KAIST SketchTheFuture Research Center.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 5, 7

[2] Nichol Alex, Jun Heewoo, Dhariwal Prafulla, Mishkin Pamela, and Chen Mark. Point-E: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[3] Antonio Alliegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. Polydiff: Generating 3d polygonal meshes with diffusion models. *arXiv preprint arXiv:2312.11417*, 2023. 2

[4] Thomas O. Binford. Visual perception by computer. In *Proceedings of the IEEE Conference on Systems and Control (Miami, FL)*, 1971. 2

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 2

[6] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *ICCV*, 2023. 2

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 6

[8] Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. Pixart-{\delta}: Fast and controllable image generation with latent consistency models. *arXiv preprint arXiv:2401.05252*, 2024. 5

[9] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. Bae-net: Branched autoencoder for shape co-segmentation. In *ICCV*, 2019. 3

[10] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *CVPR*, 2020. 2, 3

[11] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-SDF: Conditional generative modeling of signed distance functions. In *ICCV*, 2023. 2

[12] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *CVPR*, 2020. 2, 3

[13] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 6

[14] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arxiv:2307.10373*, 2023. 2

[15] Stefan Gottschalk, Ming C Lin, and Dinesh Manocha. OBBTree: a structure for rapid interference detection. 1996. 2

[16] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2

[17] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. SPAGHETTI: Editing implicit shapes through part aware generation. *ACM TOG*, 2022. 3, 8

[18] Kai Huebner, Steffen Ruthotto, and Danica Kragic. Minimum volume bounding box decomposition for shape approximation in robot grasping. In *ICRA*, 2008. 2

[19] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural template: Topology-aware reconstruction and disentangled generation of 3d meshes. In *CVPR*, 2022. 3

[20] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2, 3, 5

[21] Jeonghyun Kim, Kaichun Mo, Minhyuk Sung, and Woontack Woo. Seg&Struct: The interplay between part segmentation and structure inference for 3d shape parsing. 2023. 2, 3

[22] Nakayama Kiyohiro, Angelina Uy Mikaela, Huang Jiahui, Hu Shi-Min, Li Ke, and Guibas Leonidas. DiffFacto: Controllable part-based 3d point cloud generation with cross diffusion. In *ICCV*, 2023. 3

[23] Juil Koo, Seungwoo Yoo, Minh Hieu Nguyen, and Minhyuk Sung. SALAD: Part-level latent diffusion for 3d shape generation and manipulation. In *ICCV*, 2023. 3, 8

[24] Eric-Tuan Lê, Minhyuk Sung, Duygu Ceylan, Radomir Mech, Tamy Boubekeur, and Niloy J Mitra. CPFN: Cascaded primitive fitting networks for high-resolution point clouds. In *ICCV*, 2021. 3

[25] Phillip Y Lee and Minhyuk Sung. Reground: Improving textual and spatial grounding at no cost. In *ECCV*, 2024. 2, 5, 7

[26] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. GRASS: Generative recursive autoencoders for shape structures. *ACM TOG*, 2017. 2, 3

[27] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, L. Yi, and Leonidas J. Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *CVPR*, 2019. 3

[28] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. Grains: Generative recursive autoencoders for indoor scenes. *ACM TOG*, 2018. 3

[29] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-SDF: Text-to-shape via voxelized diffusion. In *CVPR*, 2023. 2

[30] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J. Mitra. GlobFit: consistently fitting primitives by discovering global relations. *ACM TOG*, 2011. 3

[31] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-set grounded text-to-image generation. In *CVPR*, 2023. 2, 5, 7

[32] Zhen Liu, Yao Feng, Michael J. Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. MeshDiffusion: Score-based generative 3d mesh modeling. In *ICLR*, 2023. 2

[33] Zhang Longwen, Wang Ziyu, Zhang Qixuan, Qiu Qiwei, Pang Anqi, Jiang Haoran, Yang Wei, Xu Lan, and Yu Jingyi. CLAY: A controllable large-scale generative model for creating high-quality 3d assets. *arXiv preprint arXiv:2406.13897*, 2024. 2

[34] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 5, 6

[35] Zhang Lvmin, Rao Anyi, and Agrawala Maneesh. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 3, 5

[36] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200:269–294, 1978. 2

[37] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. StructureNet: Hierarchical graph networks for 3d shape generation. *ACM TOG*, 2019. 2, 3

[38] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, 2019. 3

[39] Gimin Nam, Mariem Khlifi, Andrew Rodriguez, Alberto Tono, Linqi Zhou, and Paul Guerrero. 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. *arXiv preprint arXiv:2212.00842*, 2022. 2

[40] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *ICML*, 2020. 4

[41] Chengjie Niu, Manyi Li, Kai Xu, and Hao Zhang. Rim-net: Recursive implicit fields for unsupervised learning of hierarchical shape structures. In *CVPR*, 2022. 3

[42] Chanhyeok Park and Minhyuk Sung. Split, merge, and refine: Fitting tight bounding boxes via over-segmentation and iterative search. In *3DV*, 2024. 2, 3, 4, 7

[43] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *CVPR*, 2019. 2, 3

[44] Despoina Paschalidou, Luc van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *CVPR*, 2020. 2

[45] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *CVPR*, 2021. 2, 3

[46] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *ICCV*, 2023. 2

[47] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *CVPR*, 2023. 5

[48] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 2

[49] Barbara Roessle, Norman Müller, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, Angela Dai, and Matthias Nießner. L3DG: Latent 3d gaussian diffusion. 2024. 2

[50] Etai Sella, Gal Fiebelman, Noam Atia, and Hadar Averbuch-Elor. Spic-e: Structural priors in 3d diffusion models using cross-entity attention. 2024. 2, 3, 5, 7, 8

[51] 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 *CVPR*, 2024. 4

[52] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 3

[53] Li Songlin, Paschalidou Despoina, and Guibas Leonidas. Pasta: Controllable part-aware shape generation with autoregressive transformers. *arXiv preprint arXiv:2407.13677*, 2024. 3

[54] Chunyu Sun, Qianfang Zou, Xin Tong, and Yang Liu. Learning adaptive hierarchical cuboid abstractions of 3d shape collections. 2019. 2

[55] Chun-Yu Sun, Qian-Fang Zou, Xin Tong, and Yang Liu. Learning adaptive hierarchical cuboid abstractions of 3d shape collections. *ACM TOG*, 2019. 3

[56] 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. 4

[57] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, 2017. 2, 3, 8

[58] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023. 2

[59] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017. 6

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[61] R. J. (Roger J.) Watt. *Visual processing : computational, psychophysical, and cognitive research*. L. Erlbaum Associates, Hove, UK ;, 1988. 2

[62] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023. 2

[63] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. SAGNet: structure-aware generative network for 3d-shape modeling. *ACM TOG*, 2019. 3

[64] 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

[65] Xiang Xu, Joseph G Lambourne, Pradeep Kumar Jayaraman, Zhengqing Wang, Karl DD Willis, and Yasutaka Furukawa. BrepGen: A b-rep generative diffusion model with structured latent geometry. *ACM TOG*, 2024. 2

[66] Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J. Guibas, and Lin Gao. DSG-Net: Learning disentangled structure and geometry for 3d shape generation. *ACM TOG*, 2022. 3

[67] Kaizhi Yang and Xuejin Chen. Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. *ACM TOG*, 2021. 2, 3

[68] Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3DShape2VecSet: A 3d shape representation for neural fields and generative diffusion models. *ACM TOG*, 2023. 2, 3, 5, 7, 8

[69] Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional sdf diffusion for controllable 3d shape generation. *ACM TOG*, 2023. 8

[70] Yang Zhou, Kangxue Yin, Hui Huang, Hao Zhang, Minglun Gong, and Daniel Cohen-Or. Generalized cylinder decomposition. *ACM TOG*, 2015. 3

[71] Zhao Zibo, Liu Wen, Chen Xin, Zeng Xianfang, Wang Rui, Cheng Pei, FU BIN, Chen Tao, YU Gang, and Gao Shenghua. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *NeurIPS*, 2023. 2