

BrepGen: A B-rep Generative Diffusion Model with Structured Latent Geometry

XIANG XU, Simon Fraser University, Canada and Autodesk Research, Canada

JOSEPH G. LAMBOURNE, Autodesk Research, UK

PRADEEP KUMAR JAYARAMAN, Autodesk Research, Canada

ZHENGQING WANG, Simon Fraser University, Canada

KARL D.D. WILLIS, Autodesk Research, USA

YASUTAKA FURUKAWA, Simon Fraser University, Canada

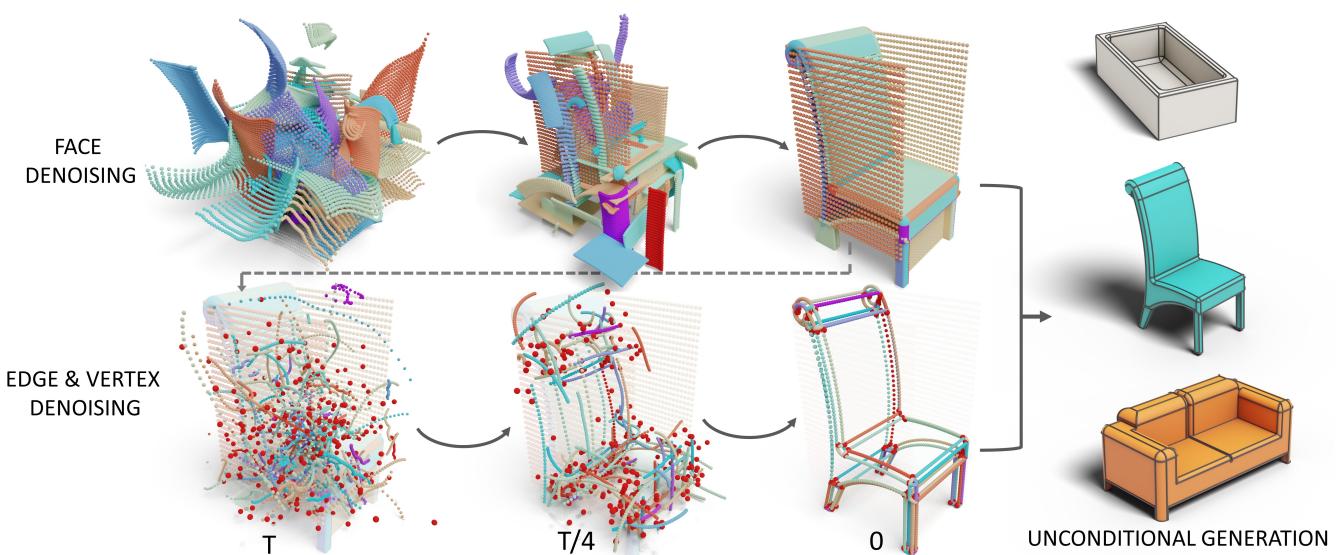


Fig. 1. *BrepGen* is a diffusion-based approach to generate 3D CAD models in B-rep format. Geometry and topology relations are progressively denoised for faces, edges, and vertices to form a water-tight solid at the end. Extensive evaluations demonstrate that *BrepGen* generates a wide range of complex B-reps.

This paper presents *BrepGen*, a diffusion-based generative approach that directly outputs a Boundary representation (B-rep) Computer-Aided Design (CAD) model. *BrepGen* represents a B-rep model as a novel structured latent geometry in a hierarchical tree. With the root node representing a whole CAD solid, each element of a B-rep model (i.e., a face, an edge, or a vertex) progressively turns into a child-node from top to bottom. B-rep geometry information goes into the nodes as the global bounding box of each primitive along with a latent code describing the local geometric shape. The B-rep topology information is implicitly represented by node duplication.

Authors' addresses: Xiang Xu, Simon Fraser University, Canada and Autodesk Research, Canada, xuxiangx@sfu.ca; Joseph G. Lambourne, Autodesk Research, UK, joseph.lambourne@autodesk.com; Pradeep Kumar Jayaraman, Autodesk Research, Canada, pradeep.kumar.jayaraman@autodesk.com; Zhengqing Wang, Simon Fraser University, Canada, zwa170@sfu.ca; Karl D.D. Willis, Autodesk Research, USA, karl.willis@autodesk.com; Yasutaka Furukawa, Simon Fraser University, Canada, furukawa@sfu.ca.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2024/7-ART119 \$15.00
https://doi.org/10.1145/3658129

When two faces share an edge, the edge curve will appear twice in the tree, and a T-junction vertex with three incident edges appears six times in the tree with identical node features. Starting from the root and progressing to the leaf, *BrepGen* employs Transformer-based diffusion models to sequentially denoise node features while duplicated nodes are detected and merged, recovering the B-Rep topology information. Extensive experiments show that *BrepGen* advances the task of CAD B-rep generation, surpassing existing methods on various benchmarks. Results on our newly collected furniture dataset further showcase its exceptional capability in generating complicated geometry. While previous methods were limited to generating simple prismatic shapes, *BrepGen* incorporates free-form and doubly-curved surfaces for the first time. Additional applications of *BrepGen* include CAD autocomplete and design interpolation. The code, pretrained models, and dataset are available at <https://github.com/samxuxiang/BrepGen>.

CCS Concepts: • Computing methodologies → Computer vision; • Applied computing → Computer-aided design.

Additional Key Words and Phrases: B-rep, Diffusion, AIGC

ACM Reference Format:

Xiang Xu, Joseph G. Lambourne, Pradeep Kumar Jayaraman, Zhengqing Wang, Karl D.D. Willis, and Yasutaka Furukawa. 2024. *BrepGen: A B-rep Generative Diffusion Model with Structured Latent Geometry*. *ACM Trans. Graph.* 43, 4, Article 119 (July 2024), 14 pages. <https://doi.org/10.1145/3658129>

1 INTRODUCTION

Nearly every man-made object begins its life as a Computer-Aided Design (CAD) model. Within the realm of CAD, the Boundary representation (B-rep) is the predominant format for describing shapes, and is widely used in freeform surface modeling to express complicated geometry. The B-rep representation [Weiler 1986] generalizes half-edge meshes [Muller and Preparata 1978], allowing parametric curves and surfaces to replace the planar facets and linear edges used in mesh modeling. A B-rep consists of sets of interconnected faces, edges, and vertices. A face is the visible region of a parametric surface, bounded by closed loops formed by its adjacent edges, while an edge is the visible region of a parametric curve, trimmed by vertices that define its start and end points. The adjacency of neighbouring edges and vertices is recorded, allowing the structure to provide a complete description of the final solid shape. A system capable of directly generating B-reps would revolutionize the CAD design workflow, reducing the extensive manual labor required from skilled designers and the reliance on professional CAD software.

However, directly generating the B-rep poses significant challenges. In contrast to triangle meshes, B-reps contain a variety of parametric surface and curve types such as arcs, toruses, and Non-Uniform Rational B-Splines (NURBS). Each geometry has a different definition and its own set of parameters, making it difficult to generate. Furthermore, the topological relations between all geometry must be correct to form a water-tight solid. Recent CAD generative models avoid direct B-rep generation and instead focus on sketch and extrude modeling operations [Li et al. 2023; Ren et al. 2022; Wu et al. 2021; Xu et al. 2023, 2022; Zhou et al. 2023] that represent only a limited range of 3D shapes. Direct B-rep generation methods, like SolidGen [Jayaraman et al. 2022], are limited to highly simplified prismatic shapes and non-freeform surfaces.

To this end, we introduce *BrepGen*, a generative approach that directly outputs diverse B-rep CAD models using Denoising Diffusion Probabilistic Models (DDPM) [Ho et al. 2020]. Key to our method is the use of a structured latent geometry representation that transforms any B-rep into a tree data structure. Concretely, *BrepGen* encodes the geometry as node features in a tree, where the root node identifies the CAD solid, and child-nodes, at the following three levels, define the global position and latent local geometry for every face, edge, and vertex. For the local geometry, we follow UV-Net [Jayaraman et al. 2021] and use a grid of points uniformly-sampled along the UV domain as a substitute for its parametric function. Mating and association topologies are encoded through the duplication of tree nodes in two ways. First, every shared edge (resp. vertex) in a B-rep model turns into multiple nodes in a tree with the same node features, establishing connections between adjacent geometries and their corresponding mating duplicates. Second, since the number of faces constituting a solid and the number of edges bounding a face vary, an additional duplication is performed to pad child nodes at each parent to a predefined length, leading to uniform width trees.

This node duplication is the key in our structured latent geometry representation, encoding both the B-rep geometry and topology information in a unified tree format, where continuous geometry regression implicitly recovers discrete topology. Concretely,

a Transformer-based diffusion model generates node features top to bottom (i.e., face to edge to vertex), while nodes with similar features are detected and merged 1) across different parents to recover mating and adjacency relations, and 2) within each parent to restore unique geometric elements associated with it. By merging the duplicated nodes, the topology of each face is reconstructed, first connecting edges into closed loops and then using them to trim the parent face. The resulting trimmed faces are joined together to directly output the CAD model in B-rep format. In summary, we make the following contributions:

- A structured latent geometry representation, whose hierarchical tree, with node duplication, encodes the B-rep geometry and topology information in a unified format.
- A latent diffusion module, capable of generating free-form surfaces and trimming curves.
- A newly collected *Furniture B-rep Dataset* with high-quality B-reps of indoor objects across 10 different categories.
- Direct B-rep generation with state-of-the-art performance.

2 RELATED WORK

This section reviews three CAD model representations used in the literature: constructive solid geometry, sketch and extrude, and B-rep, followed by a summary of B-rep datasets and the use of diffusion models for CAD model generation.

2.1 Constructive Solid Geometry (CSG)

CSG is a common method to create 3D shapes by combining primitives (e.g. cuboids, spheres) with Boolean operations (e.g. union, subtraction) to form a CSG tree. CSG has been used extensively in ‘shape programs’ [Ritchie et al. 2023] both with neural guidance [Ellis et al. 2019; Sharma et al. 2018; Tian et al. 2019] and without [Du et al. 2018; Nandi et al. 2017, 2018], thanks to its simplicity. Recent CSG-based approaches have focused on recovering 3D shapes as primitive assemblies without supervision from the ground truth CSG tree [Kania et al. 2020; Ren et al. 2021; Yu et al. 2023, 2022]. Even though most CSG primitives can be represented as B-reps, converting non-trivial CSG shapes into B-reps induces unwanted complexity. For example, thin sliver faces are created when CSG primitives are almost aligned but not coincident. A major advantage of our work is that we learn from the B-rep topology found in industrial quality designs, resulting in well structured B-rep output that can be edited using standard direct modeling CAD tools.

2.2 Sketch and Extrude

Sketch and extrude is a popular format, representing a CAD model as a sequence of modeling operations stored in parametric CAD files. Reconstruction [Li et al. 2023; Xu et al. 2021] and generation [Ren et al. 2022; Wu et al. 2021; Xu et al. 2023, 2022; Zhou et al. 2023] of the format have been studied in recent years. While significant progress has been made on improving the fidelity and controllability, sequential CAD generative models currently are limited to producing sketches comprised of line, arc and circle primitives, and only the extrude modeling operation. Our method focuses on direct synthesis of the B-rep data with emphasis on supporting complex curves and surfaces that cannot be achieved by previous methods.

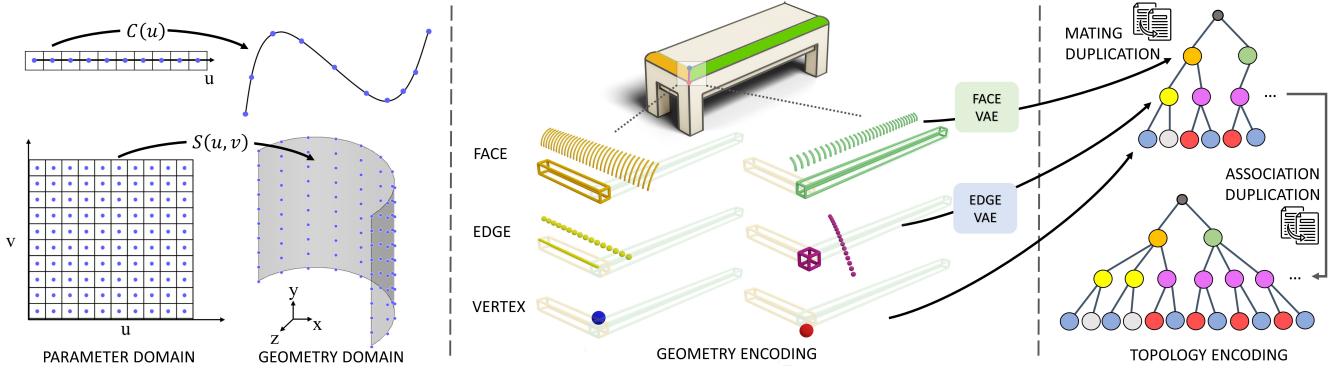


Fig. 2. Our structured latent geometry tree representation of a B-rep CAD model. Shape feature is sampled from the parametric surface or curve using grid in the parameter domain (left). Position and local geometry are extracted for the face, edge, and vertex, and then encoded as node features in the tree (mid). Topology is encoded by mating and association duplication. Node feature is represented by color values, where duplicated nodes are of the same color (right).

2.3 Boundary Representation

B-rep 3D models are represented as a graph [Ansaldi et al. 1985], consisting of geometric primitives i.e., parametric curves and surfaces, and topological primitives, i.e., vertices, edges, and faces that are used to trim and sew the surface patches into a solid model. B-rep classification and segmentation tasks have been tackled with graph neural networks [Cao et al. 2020a; Jayaraman et al. 2021; Willis et al. 2022], custom convolutions [Lambourne et al. 2021], and hierarchical graph structures [Bian et al. 2023; Jones et al. 2021, 2023]. For generation, previous approaches handled predefined template shapes [Smirnov et al. 2021] such as parametric curves [Wang et al. 2020] and surfaces [Li et al. 2019; Sharma et al. 2020]. A challenge remains for cases, where a watertight solid is formed by trimming and connecting surfaces.

PolyGen [Nash et al. 2020] uses Transformers [Vaswani et al. 2017] and pointer networks [Vinyals et al. 2015] to generate n-gon meshes, which is a special case of B-rep models with planar faces and linear edges. Other mesh generation methods also include BSP-Net [Chen et al. 2020] and MeshGPT [Siddiqui et al. 2023]. Wang *et al.* reconstructs a B-rep model from a 2D wireframe drawing by identifying planar and cylindrical faces [Wang et al. 2022]. Guo *et al.* reconstructs B-rep from point clouds using a neural network to predict the rough B-rep geometry and topology, that were refined using a combinatorial optimization [Guo et al. 2022].

Closest to our work is SolidGen [Jayaraman et al. 2022], that can generate entire B-reps, first with the synthesis of vertices, followed by conditional construction of edge topology using a pointer network, finally followed by another pointer network that conditionally connected the edges to faces. While this approach can produce plausible B-reps, it is restricted to prismatic primitives which limit the complexity of the results, and the amount of data that it can be trained on. Our method is more general and can produce prismatic-looking, as well as freeform doubly-curved geometry.

2.4 Diffusion Models for CAD Generation

Diffusion models have been successful at generating geometry when topology is given. HouseDiffusion [Shabani et al. 2023] generates 2D

floor-plan that follows a given room connectivity. CAGE [Liu et al. 2023] generates 3D articulated object conditioned on the ground-truth part connections. More recent methods like PolyDiff [Alliegro et al. 2023] do not require topology as input conditioning but are also limited to generating triangle soups of polygonal meshes without any topology. Using diffusion model to unconditionally generate well-structured geometry together with the correct topology relations remains a challenge, which is what our method aims to solve.

2.5 B-rep Datasets

B-rep 3D datasets have grown in number over the last several years. These include synthetically created datasets with class labels, such as FabWave [Angrish et al. 2019], SolidLetters [Jayaraman et al. 2021], and MFCAD [Cao et al. 2020b], as well as datasets of human designed 3D models without canonical class labels, such as ABC [Koch et al. 2019], Fusion 360 Gallery [Lambourne et al. 2021; Willis et al. 2022, 2021b], Automate [Jones et al. 2021], and DeepCAD [Wu et al. 2021]. Our *Furniture B-rep Dataset*, to the best of our knowledge, is the first dataset to contain human-designed 3D models in the B-rep format across a standard set of classes (e.g. tables, chairs etc).

3 STRUCTURED LATENT GEOMETRY

A B-rep model consists of geometric elements (faces, edges, vertices) with pairwise topological relationships (face-edge, edge-vertex adjacency matrix). The challenge lies in generating these two distinct data representations in a general graph topology. To solve this, our approach unifies the geometry and topology of a B-rep model as a hierarchical tree with a fixed graph topology, where node features encode the geometry information and duplicated nodes (i.e., nodes with near identical features during generation) implicitly encode the topology information. With the root node representing a whole CAD solid, the tree has three levels: faces, edges, and vertices from top to bottom. The unified tree representation effectively bridges the two formats and becomes the foundation for training our diffusion models. This section first explains the node features encoding geometry and then node duplication schemes encoding topology.

3.1 Geometry Encoding by Node Features

A face, an edge, or a vertex node encodes its geometry information into a node feature with 1) a global position as bounding box parameters or a point and 2) local shape details as a latent code.

3.1.1 Face (F). Underlying each face is a parametric surface (plane, cylinder, cone, sphere, torus, Bezier, or NURBS) with a function $S(u, v) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, mapping a UV coordinate to a 3D point on the unbounded surface on which the face lies (Figure 2 left). The shape feature F_s is a 2D array of 3D points sampled on the parametric surface, and the position feature F_p is the axis-aligned bounding box enclosing the points: $F_p = [x_1, y_1, z_1, x_2, y_2, z_2]$ encoding the bottom-left and top-right corners. Following UV-Net [Jayaraman et al. 2021], let $[u_{\min}, u_{\max}] \times [v_{\min}, v_{\max}] \in \mathbb{R}^2$ be a 2D axis-aligned bounding box along the UV axes. Points are sampled from a $N \times N$ equally spaced grid along the UV axes with step size $\delta u = \frac{u_{\max} - u_{\min}}{N}$, $\delta v = \frac{v_{\max} - v_{\min}}{N}$. 3D coordinates at all grid locations are concatenated to form $F_s \in \mathbb{R}^{N \times N \times 3}$ as the shape details. Capitalizing on the success of latent diffusion models for image generation [Rombach et al. 2022], a variational autoencoder (VAE) with a UNet backbone further compresses F_s into a latent code F_z , after normalizing the 3D coordinates by an affine transformation that maps the bounding box to the canonical cube (i.e., $[-1, 1]^3$). The feature of a face node is then defined as $F = [F_p, F_z]$. Note that this feature does not include the outer trimming boundary or inner holes, which are given by the associated edges. The ‘Face’ row in Figure 2 (middle) illustrates two faces with different bounding box positions and normalized sampled points, revealing the local shape details.

3.1.2 Edge (E). Underlying each edge is a parametric curve (line, circle, elliptic, Bezier, or B-spline) with a function $C(u) : \mathbb{R} \rightarrow \mathbb{R}^3$, mapping a u -coordinate to a 3D point on the unbounded curve on which the edge lies (Figure 2 left). The shape feature E_s is a 1D array of 3D points sampled along the parametric curve. The position feature E_p is again defined as the bounding box parameters enclosing the points. Let $[u_{\min}, u_{\max}] \in \mathbb{R}$ be the minimum and the maximum U-coordinate value of an edge. Points are sampled from an N equally spaced grid along the u -axis with step size $\delta u = \frac{u_{\max} - u_{\min}}{N}$. 3D coordinates at all grid locations are concatenated to form $E_s \in \mathbb{R}^{N \times 3}$ as the shape details. Akin to the face feature construction, a VAE compresses E_s into E_z , where $E = [E_p, E_z]$ becomes the feature of an edge node. The ‘Edge’ row in Figure 2 (middle) visualizes the position and the shape details.

3.1.3 Vertex (V). A vertex is a 3D point without additional shape details. The feature of a vertex node is its point coordinate $V = (x, y, z)$. The ‘Vertex’ row in Figure 2 (middle) shows the location of two vertices.

3.2 Topology Encoding by Node Duplication

Node duplication in *BrepGen* serves two purposes: 1) encode the topological relationships between faces, edges, and vertices, 2) pad the number of faces in a B-rep and edges surrounding a face to a fixed maximum length. Two node duplication schemes achieve these objectives: *mating* and *association*.

3.2.1 Mating Duplication. Mating duplication encodes face-edge-face and edge-vertex-edge adjacency relationships geometrically. Shared edges are duplicated so that each parent face gets a copy of the edge geometry as a child node. Similarly shared vertices are duplicated as child nodes of their parent edges. This procedure turns a B-rep graph structure into a tree. The top-right tree in Figure 2 shows the resulting tree with node features represented by color values, shared blue vertices, and purple duplicated edges. Mating relations are later recovered by merging edge or vertex nodes with similar geometry across different parents.

3.2.2 Association Duplication. The number of edges bounding a face and the number of faces forming a solid vary, however the numbers of faces and edges required to build a solid will not be known at inference time. Our idea is to pick a predetermined maximum branching factor for each tree level and randomly duplicate nodes until reaching the maximum number of children. We found this ‘duplication padding’ strategy results in fewer missing faces and edges at inference time than the zero-padding strategy. Randomly selecting the duplicated nodes per training iteration also helps prevent over-fitting. The bottom-right tree in Figure 2 illustrates this process with the tree randomly padded to two faces and three edges per face. The yellow edge is duplicated once and the purple edge is duplicated twice. Association relations can be recovered by removing child nodes with the same geometry under each parent. Note that every edge is always connected to two vertices of its start and end, making the association duplication unnecessary for vertices.

4 METHOD

BrepGen consists of two VAEs that compress the face and edge shape feature into geometry latent vectors, and a latent diffusion module that sequentially denoises the node latent features in a tree from root to leaf, enabling effective training and inference.

4.1 Shape Geometry VAE

Following Stable Diffusion [Rombach et al. 2022], we use two VAEs to compress the shape feature F_s, E_s into lower-dimensional latent F_z, E_z . Shape feature is first normalized to $[-1, 1]$. A VAE with 2D convolution UNet backbone compresses the normalized face feature. A similar VAE with 1D convolution compresses the edge feature. Both VAEs are trained with MSE reconstruction loss and a KL regularization term. Concretely, the face VAE encoder compresses F_s into latent vector $F_z = E(F_s)$. A decoder then reconstructs $\hat{F}_s = D(F_z) = D(E(F_s))$. Reconstruction loss is between \hat{F}_s, F_s . In practice, we set $N = 32$ to densely sample the UV parameter domain, and downsample the input by a factor of 8 while retaining the feature depth, resulting in $F_z \in \mathbb{R}^{4 \times 4 \times 3}$ and $E_z \in \mathbb{R}^{4 \times 3}$. To further reduce computation, the two endpoint vertices are concatenated with an edge to form joint latent $E_{zv} \in \mathbb{R}^{4 \times 3+6}$.

4.2 Latent Diffusion Module

Our latent diffusion module is trained with the compressed latent geometry and global position as node features in the tree. The overall architecture of our generative latent diffusion module is illustrated in Figure 3. We follow the DDPM [Ho et al. 2020] training

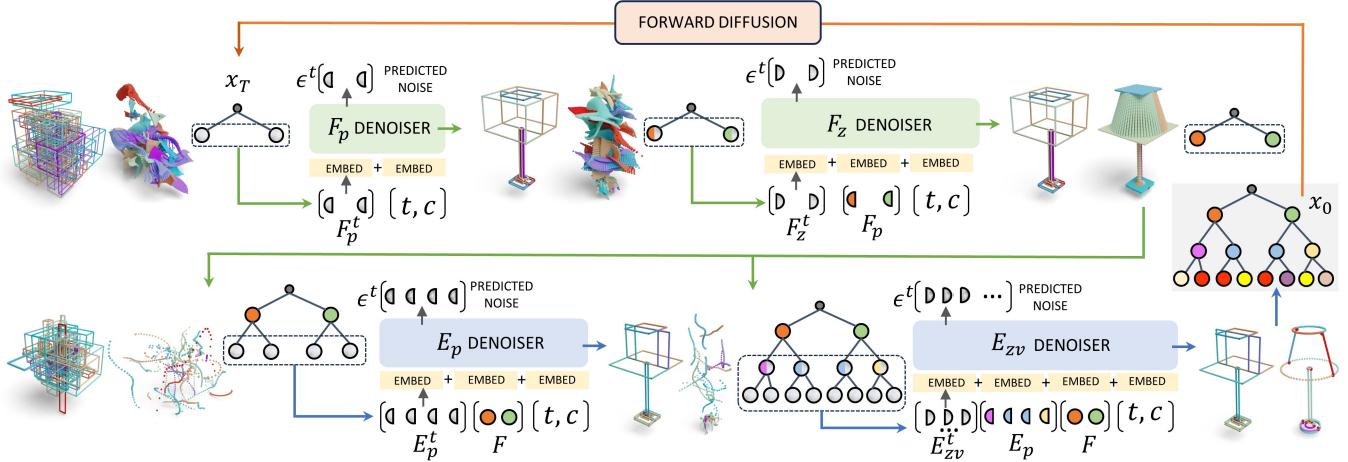


Fig. 3. Top row: Generation of the face bounding box position and face latent geometry. Nodes are split in half to represent each. t indicates the time step and c indicates the class label in the Furniture dataset. Bottom row: Generation of the edge bounding box position and edge-vertex joint latent geometry, both conditioned on the parent face. Gray color represents noisy feature values. Points are decoded from the latent geometry for visualization.

scheme and use four Transformer-based denoisers to remove the noise added to the nodes sequentially.

4.2.1 Diffusion Process. Given a tree with node features \mathbf{x}_0 , the forward diffusion process adds Gaussian noise to all the nodes in T steps. The noisy tree node features \mathbf{x}_t at time step t are sampled as:

$$\begin{aligned} q(\mathbf{x}_t | \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}), \\ \mathbf{x}_t &= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_t, \end{aligned} \quad (1)$$

where noise $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, $\alpha_t = 1 - \beta_t$. The β_t is determined by the noise variance scheduler, a linear scheduler by default. After the noise injection, the encoded B-rep geometry and topology are corrupted. For example, duplicated nodes have different features and no longer exhibit the correct topological relations.

4.2.2 Sequential Denoising. Generating all geometry at once is difficult. We use sequential generation to denoise the face, edge, and vertex progressively. Concretely, we factored the distribution of node features \mathbf{x} into a product of conditional distributions:

$$p(\mathbf{x}) = p(F, E, V) = p(E_{zv}|E_p, F)p(E_p|F)p(F_z|F_p)p(F_p|\emptyset). \quad (2)$$

The ordering reflects a top-down generation process with edge conditioned on face, and latent geometry conditioned on global position. Figure 3 shows the conditional denoising process with a denoising network for each distribution.

The four denoising networks in our latent diffusion module share a common Transformer backbone. Without loss of generality, we will use edge position denoiser (Figure 3 bottom-left) as an example. Input to the denoiser are edge tokens with noisy bounding box parameters. Network is conditioned on the previously denoised face tokens from Figure 3 (top-row), which are embedded as:

$$F \leftarrow \text{MLP}(W_p F_p) + \text{MLP}(W_z F_z). \quad (3)$$

$F_p \in \mathbb{R}^6$ is the face bounding box coordinates and $F_z \in \mathbb{R}^{48}$ is the flattened latent geometry vector. $W_p \in \mathbb{R}^{d \times 6}$, $W_z \in \mathbb{R}^{d \times 48}$ are the

two d -dimension embedding matrices. MLP is fully connected layers with SiLU activation. The noisy edge bounding boxes are encoded in a similar manner as $E_p = \text{MLP}(W_p E_p)$.

Rather than using cross-attention, the predefined parent-child relations from the tree are used to directly inject the face condition. Let edge j be the child node of face i . Embedding from the i th face token is added to the j th edge bounding box position token as:

$$\hat{E}_{p,j} \leftarrow E_{p,j} + F_i. \quad (4)$$

During top-down generation, a known number of child nodes are added for each parent at the next stage, leading to simple token addition for the condition. Edge embedding E_{zv} is similarly embedded as $E_{zv} = \text{MLP}(W_{zv} E_{zv})$ with $W_{zv} \in \mathbb{R}^{d \times 18}$. The previously denoised edge position and parent face condition are added to every j th edge token as $E_{zv,j} + \hat{E}_{p,j}$ following Equation 4. Final input to the denoiser also adds the time and condition embedding as $f(t, c)$. Note that we do not use learnable positional encoding.

4.2.3 Loss Function. The four Transformer-based denoisers are trained separately to predict the L2-norm regression loss of the added sampled noise as in DDPM [Ho et al. 2020]. Loss term is:

$$L = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}_t} \left[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_t, t)\|^2 \right], \quad (5)$$

where $\boldsymbol{\epsilon}_t$ is the Gaussian noise added at time t in the forward pass, \mathbf{x}_t is the corrupted data following Equation 1. In our case, \mathbf{x}_0 is the clean node features F_p, F_z, E_p, E_{zv} , and \mathbf{x}_t is the noisy node features after forward pass. During sampling, the predicted noise at every time step is used to denoise the data from random Gaussian noise.

4.3 B-rep Post-Processing

A set of heuristics is used to find duplicated nodes in the tree and explicitly decode the generated B-rep topology. For efficiency, we use early pruning to recover association as the tree is being generated. Given the face nodes, we detect duplicates as those with bounding box corners within 0.08 Euclidean distance of each other

and averaged point-wise difference less than 0.2 for the decoded points from the latent shape feature. Duplicate faces are removed to recover the unique faces belonging to a solid. We avoid regenerating edges for the duplicated faces in the next stage.

A similar procedure recovers the unique edges associated under every face, the same 0.08 edge bounding box threshold and 0.2 threshold for the decoded shape feature points are used to merge edges under the same parent. The vertex/edge association is always two and does not require deduplication. Finally, we traverse the tree again from leaf to root to find duplicated child nodes at the same level but associated with different parents, which are the shared edges and vertices for determining the final mating relations.

After recovering the mating and association relations, we perform an extra step of fine-tuning the geometry based on the generated topology. The vertex positions are averaged across its associated duplicates. Edge shape geometry points are scaled and translated to align with the associated start and end vertices. An edge is flipped if it is reversed with regard to its vertices. The face shape geometry points are also scaled and translated so that they tightly fit all the associated edges with the minimum Chamfer Distance.



Fig. 4. *Furniture B-rep Dataset* overview colored by category.

5 FURNITURE B-REP DATASET

To promote future B-rep generation research, we introduce the *Furniture B-rep Dataset* containing 6,171 B-rep CAD models across 10 common furniture categories. To our knowledge, this is the first dataset in the B-rep format containing 3D models with freeform surfaces in addition to canonical category labels. We reached out to OnShape, a web-based CAD tool and repository [Onshape 2012], and obtained an agreement to export their publicly available design assets into our B-rep dataset. We used keywords to identify 3D models in each category and searched inside both partstudio and assembly documents. We perform a number of manual filtering steps on the raw data to remove duplicates, verify correct categories, remove low quality data, and rotate the 3D models in each category into a canonical orientation. A visual overview of the CAD B-rep models in the *Furniture B-rep Dataset* is provided in Figure 4.

Additional statistics for the *Furniture B-rep Dataset* are provided in Figure 5. Figure 5a shows the category names in the dataset along with the number of B-rep models per category. Figure 5b provides



Fig. 5. Statistics for the *Furniture B-rep Dataset*.

the distribution for the number of solid bodies per B-rep in the dataset. Similarly, Figure 5c and Figure 5d plot the distributions of the face and edge counts in every B-rep model.

6 EXPERIMENTS

This section presents unconditional and conditional generation results. Extensive analysis demonstrates that *BrepGen* consistently produces high quality B-reps with complex topology and geometry, while providing enhanced control over the generation process.

6.1 Experiment Setup

6.1.1 Datasets. We evaluate generation performance on 1) the DeepCAD dataset [Wu et al. 2021] of mechanical parts made from sketch and extrude operations, 2) the *Furniture B-rep Dataset* with more complicated furniture models, and 3) the ABC dataset [Koch et al. 2019] containing a wide variety of parts from industrial designs. We use the original train/val/test split from DeepCAD and remove duplicated models in the training set following [Willis et al. 2021a]. Closed faces (cylinders, etc.) are split on the seams following SolidGen [Jayaraman et al. 2022]. B-reps with more than 30 faces or 20 edges per face, and made from multiple bodies are removed. After filtering, a total of 87,815 B-reps are used for training the VAEs and the latent diffusion module. We also randomly split the ABC and *Furniture B-rep Dataset* into 90%-5%-5% for train/val/test. In total, 259,597 ABC B-reps and 1,198 furniture B-reps with a maximum of 50 faces and 30 edges per face are used for training. Latent diffusion model is trained on all 10 categories for the *Furniture B-rep Dataset*.

6.1.2 Training. We implement *BrepGen* in PyTorch and trained it on 4 NVIDIA RTX A5000 GPUs. Half-precision is used to speed up the training. We use AdamW [Loshchilov and Hutter 2018] with $5e - 4$ learning rate. Gradient clipping of 5 are used for the VAE optimization. Weight for KL regularization is set to $1e - 6$. Latent diffusion module uses 1,000 diffusion steps and a linear beta schedule

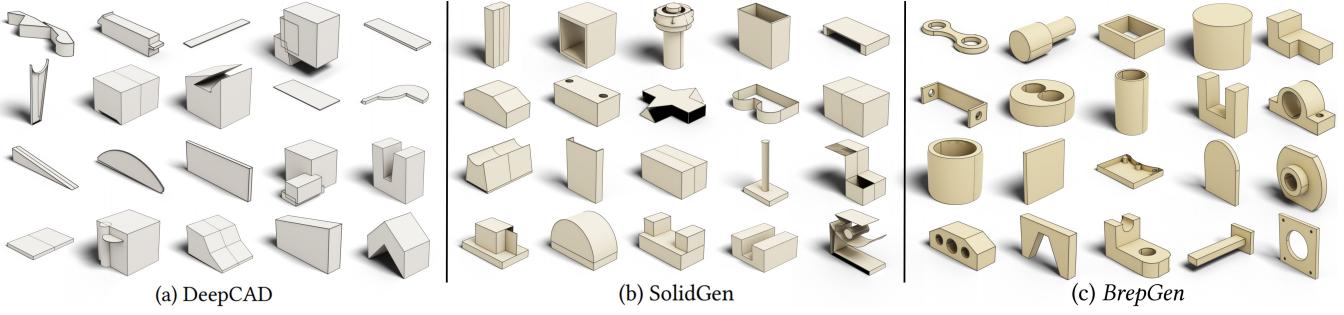


Fig. 6. Unconditional generation results on DeepCAD mechanical parts by (a) DeepCAD, (b) SolidGen and (c) our method *BrepGen*. Our method generates more realistic-looking CAD models with fewer broken geometry. Topological connections are also correct even on complicated objects.

from $1e - 4$ to 0.02. The face and edge VAEs are trained for 400 epochs on DeepCAD (or 200 epochs on ABC) with a batch size of 512. Weights are further fine-tuned for 200 epochs for the *Furniture B-rep Dataset*. The two face denoisers in the latent diffusion model are trained for 3,000 epochs with a batch size of 512, and the two edge denoisers are trained for 1,000 epochs with a batch size of 64. For the ABC dataset, we reduce the training epochs of the edge denoisers to 300, and the surface latent geometry denoiser to 1,000.

To alleviate train and test time difference, we perform cross-model augmentation where input to the conditional denoisers are randomly augmented following the scheme used in [Ho et al. 2022]. Association duplication randomly selects and pads the edges and faces in a B-rep (with repetition) until reaching the predefined maximum length, which is 30/600 for the face/edge denoisers of DeepCAD, and 50/1500 for the face/edge denoisers of the Furniture B-Rep and ABC dataset. All bounding boxes are normalized to [-3, 3] range.

6.1.3 Inference. At inference time, we use PNDM [Liu et al. 2022] of 200 forward passes for fast sampling, with the exception of the face and edge position denoisers where we switch to a slower DDPM without strided sampling from $T = 250$ to 0. Empirically, this coarse-to-fine denoising generates a more precise bounding box position. When sampling from random Gaussian noise, increasing the number of tokens beyond the maximum threshold used in training is helpful. This avoids noisy nodes being incorrectly merged early in the denoising process, leading to a lack of available primitives to fill in missing regions. For the DeepCAD dataset we denoise B-reps with a maximum of 50 faces and 30 edges per face. For the *Furniture B-rep Dataset* and ABC, 80 faces and 40 edges per face are denoised. After denoising and post-processing, we use the OpenCascade functions *GeomAPI_PointsToBSplineSurface* and *GeomAPI_PointsToBSpline* to approximate the face and edge points with B-Spline surfaces and curves. The connected loops bound the surface, and the trimmed faces are sewn together to form the final B-rep solid. On a RTX A5000 GPU, the four denoisers in total require an average of around 5 seconds to generate a B-rep trained on DeepCAD data, and around 10 seconds for the more complicated Furniture and ABC data.

6.1.4 Network Architecture. The face VAE network has four down-sampling and four up-sampling blocks. Each block contains two

layers of 2D-ResNet blocks with skip connections. The feature dimensions of the downsampling and upsampling blocks are 512-512-256-128 and 128-256-512-512, respectively. The edge VAE has similar ResNet blocks but uses 1D convolution instead. The diffusion model is a standard Transformer module with pre-layer normalization, 12 self-attention layers with 12 heads, a hidden dimension of 1024, feature dimension of 768, and a 0.1 dropout rate.

6.1.5 Evaluation Metrics. We use two sets of metrics to quantitatively measure generation quality: *Distribution Metrics* and *CAD Metrics*. For *Distribution Metrics* we use 3,000 B-reps randomly-sampled from the generated data and 1,000 B-reps from the reference test set. For each B-rep, we sample 2,000 points from the solid surface and compute the following metrics:

- *Coverage* (COV) is the percentage of reference data with at least one match after assigning every generated data to its closest neighbor in the reference set based on Chamfer Distance (CD).
- *Minimum Matching Distance* (MMD) is the averaged CD between a reference set data and its nearest neighbor in the generated data.
- *Jensen-Shannon Divergence* (JSD) measures the distribution distance between reference and generated data after converting point clouds into 28^3 discrete voxels.

For *CAD Metrics* the same 3,000 B-reps are used to compute the following metrics, as used by [Jayaraman et al. 2022]:

- *Novel* percentage of data that do not appear in training set.
- *Unique* percentage of data that appears only once in generation.
- *Valid* percentage of B-rep data that are watertight solids.

6.2 Unconditional B-rep Generation

We compare with DeepCAD [Wu et al. 2021] and SolidGen [Jayaraman et al. 2022] for unconditional generation of mechanical parts. DeepCAD is evaluated based on the reconstructed B-reps from its generated sketch and extrude sequences. We also evaluate *BrepGen* on the large-scale ABC dataset [Koch et al. 2019].

6.2.1 Quantitative Evaluation. The first three rows in Table 1 report the averaged scores on DeepCAD data from 10 different runs. *BrepGen* consistently outperforms both baselines with better COV, MMD scores and a substantially lower JSD; demonstrating improvements in generation quality and a closer match to the ground-truth distribution. Generated data from *BrepGen* is also novel and different

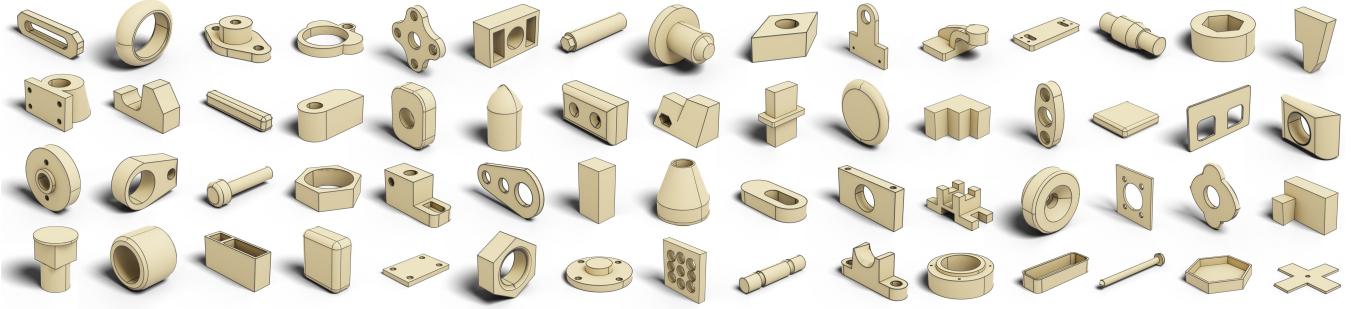


Fig. 7. Unconditional generation results on the ABC dataset. *BrepGen* generates realistic CAD B-reps with complicated geometry and topology.

Table 1. Quantitative evaluations for DeepCAD unconditional generation based on the *Coverage* (COV) percentage, *Minimum Matching Distance* (MMD), *Jensen-Shannon Divergence* (JSD), and *Unique*, *Novel*, *Valid* ratio. The bottom row reports *BrepGen* results on the ABC dataset.

Method	COV	MMD	JSD	Novel	Unique	Valid
	% ↑	↓	↓	% ↑	% ↑	% ↑
DeepCAD	65.46	1.29	1.67	87.4	89.3	46.1
SolidGen	71.03	1.08	1.31	99.1	96.2	60.3
<i>BrepGen</i>	71.26	1.04	0.09	99.8	99.7	62.9
<i>BrepGen</i> (ABC)	57.92	1.35	3.69	99.7	99.4	48.2

from the training set, as indicated by the high novel and unique scores. Here, we consider two B-reps identical if the topology connections are the same, and the shape geometry points from the faces are equal after 4 bit quantization. The valid ratio of *BrepGen* is also better compared to the SolidGen direct B-rep generation baseline. Our criteria for validity enforces watertight CAD models with no broken topology. Novel and unique scores are calculated using the valid B-reps. Last row in Table 1 also reports the *BrepGen* quantitative results for ABC unconditional generation.

6.2.2 Qualitative Evaluation. Figure 6 illustrates qualitative results for unconditional DeepCAD generation. We see that *BrepGen* generates appealing B-reps made of a wide range of topologically connected faces, showing better generation diversity than the baselines. The structural complexity is also higher with fewer unbounded open regions or self-intersecting edges. Additional results are provided in Figure 16. Figure 7 also shows the unconditional ABC generation results by *BrepGen*. We observe that our method effectively trains on the large-scale ABC dataset and generates realistic, diverse CAD B-reps of complicated topology and geometry. Figure 8 further compares the distribution of B-reps generated using *BrepGen* for the number of faces, edges, and vertices, with the training set and other baselines. The plot shows that our method has the best matching curves to the ground-truth DeepCAD training data.

6.2.3 Shape Novelty Analysis. To further evaluate the ability of *BrepGen* to generate novel shapes, we adopt the settings from Neural Wavelet [Hui et al. 2022] and retrieve the most similar shapes from the DeepCAD training set using Chamfer Distance (CD) and

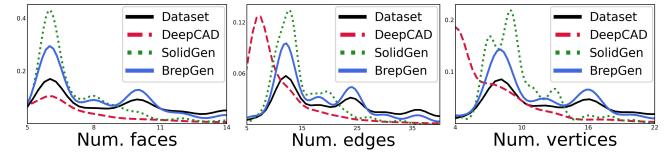


Fig. 8. Generated B-rep data distributions, for the number of faces, edges, and vertices, compared with the DeepCAD training set and other baselines.

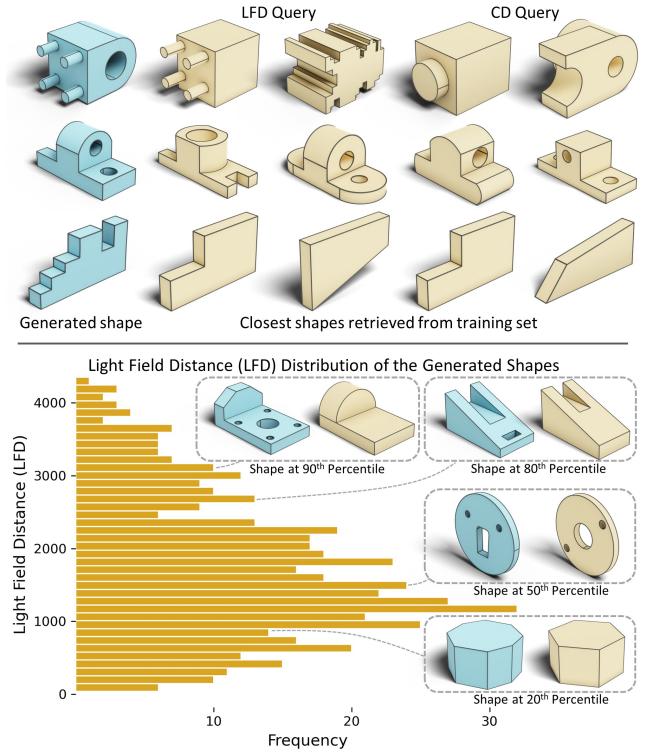


Fig. 9. Novelty analysis of the generated DeepCAD shapes. Top: Given generated shapes (blue), we visualize the top-two most similar shapes (yellow) retrieved from the training set using LFD or CD. Bottom: LFD distribution between 500 randomly generated data and their most-similar retrieved training shapes. Visual results at different percentiles are shown, where generated shapes can be both realistic and novel.

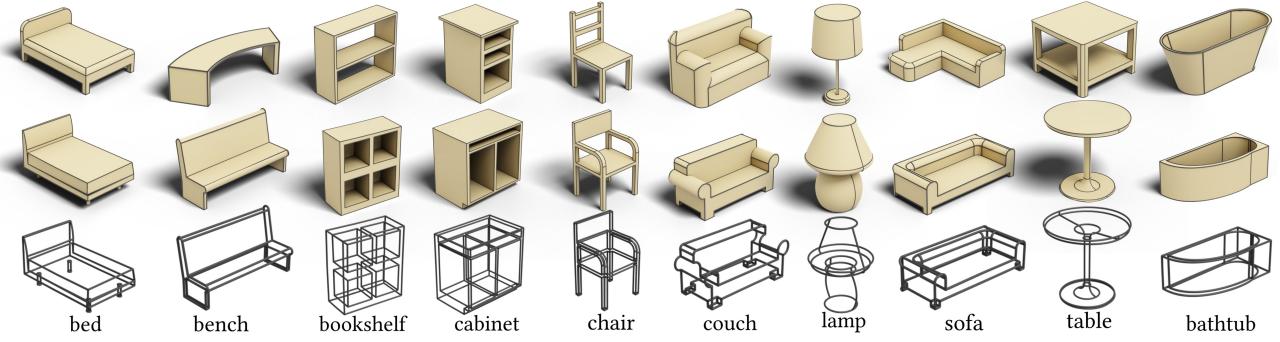


Fig. 10. *BrepGen* generation results conditioned on the class label. Last row shows the wireframe connected by the generated edges and vertices. The results clearly show that *BrepGen* excels in generating well-structured CAD models with free-form surfaces and curvatures.

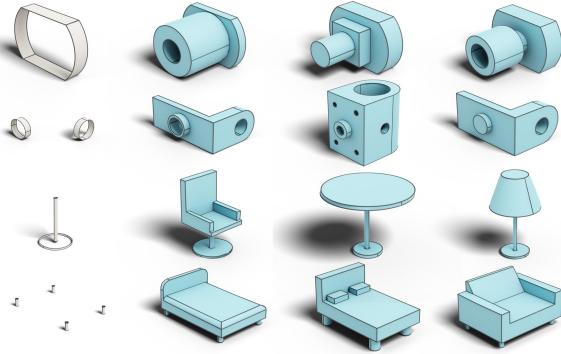


Fig. 11. Given partial faces (gray), *BrepGen* auto-completes the full B-rep geometry and topology, generating a diverse set of CAD models (blue).

Light Field Distance (LFD) [Chen et al. 2003]. Figure 9 (top) shows the generated shapes (blue), together with the top-two most similar training shapes (yellow) retrieved using LFD or CD. Additionally, Figure 9 (bottom) plots the LFD distribution between 500 randomly generated samples and their closest retrieved results from the training set. Results show that *BrepGen* generates realistic and visually accurate CAD models, while featuring novel topology and geometry different from the training data.

6.3 Controllable B-rep Generation

We demonstrate *BrepGen* results on *Furniture B-rep Dataset* and show two applications for CAD design. With no positional encoding and random shuffling at training, *BrepGen* is permutation invariant with respect to the input tokens, eliminating the need for a specific sequence order between existing and to-be-generated components.

6.3.1 Class-conditioned Generation. A class-conditioned *BrepGen* is trained on the *Furniture B-rep Dataset* using classifier-free guidance [Ho and Salimans 2022]. Class embeddings are added to every input token and an extra label indicates the unconditional case, which occurs with a 10% probability at training. Figure 10 shows the generation results for all 10 categories. *BrepGen* generates the correct topological connections across different components and outputs

free-form surfaces not possible with SolidGen. The generated edges form connected wireframes with loops precisely trimming the faces into water-tight solids. Additional results are provided in Figure 17.

6.3.2 CAD Autocompletion. We use a pretrained *BrepGen* to autocomplete the full B-rep from partial faces provided by the user. Motivated by RePaint [Lugmayr et al. 2022], a random subset of face tokens are replaced with the provided face geometry diffused to that time step during face denoising. The replacement is discontinued in the final 50 steps to allow minor adjustments to the provided faces. Subsequent edges and vertices are also regenerated. Figure 11 shows diverse autocompleted results from different Gaussian noise. Disjoint parts are connected and transformed into water-tight solids with faces seamlessly generated between, below, or above the given input. The inpainting process can make slight adjustments to user input so that it connects with the rest of the geometry. For example, the radius of the legs slightly increase for the bed in the last row.

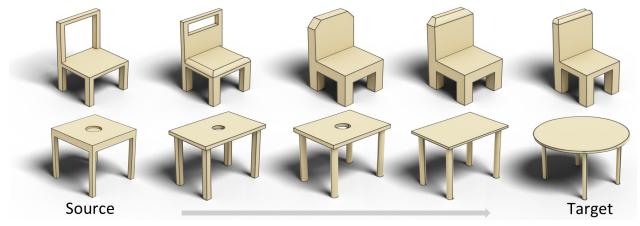


Fig. 12. Design interpolation from source to target. The geometry and topology of the generated B-rep undergo progressive changes.

6.3.3 Design Interpolation. Given a pair of CAD models, face tokens from the two shapes are concatenated, diffused for 150 steps, repeatedly padded and passed to the face denoiser. Edges and vertices are also regenerated. This is different from blending the latent value as mixing two geometry tokens will not lead to valid shapes. Figure 12 shows the interpolated results. From left to right, face tokens from the target are increasingly added to the source for face generation, after which the source tokens are removed until only the target tokens are left. The topology and geometry changes are smooth, with interpolated design directly generated in B-rep format.

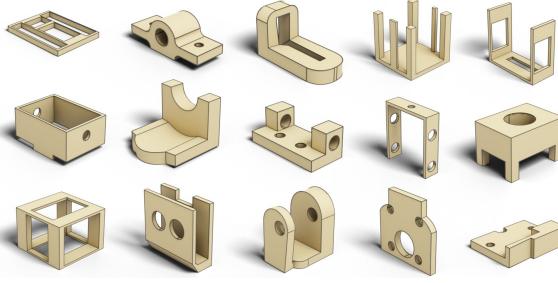


Fig. 13. DeepCAD generation results conditioned on ground-truth topology.

Table 2. Quantitative evaluations for DeepCAD generation with ground-truth topology shown as *BrepGen**. Second row reports the unconditional topology generation using multinomial diffusion.

Method	COV % ↑	MMD ↓	JSD ↓	Valid		
				Novel % ↑	Unique % ↑	Valid % ↑
<i>BrepGen*</i>	78.16	1.02	0.09	99.9	97.6	79.8
Topology	-	-	-	95.1	64.5	6.2

6.4 Ablation Studies

6.4.1 Two-stage Generation. To demonstrate the effectiveness of our unified generation approach, we conduct an ablation study where topology and geometry are separately generated in two stages. In the second stage, we modify *BrepGen* to generate geometry conditioned on the given ground-truth topology. The node duplication process is removed and replaced by the graph attention used in HouseDiffusion [Shabani et al. 2023] to encode the topology relations. The first row in Table 2 shows the conditional generation results trained on DeepCAD data, with visual results shown in Figure 13. Noticeably, the coverage and valid ratios increase, and the generated B-reps exhibit more intricate structures. We attribute this improvement primarily to knowing the ground-truth topology.

For the initial topology stage, we design a multinomial diffusion baseline [Hoogeboom et al. 2021] that unconditionally generates the face-edge, edge-vertex incidence matrix as a 128×128 binary image. We observe that it is challenging to unconditionally generate the correct incidence matrix. Last row in Table 2 reports a very low valid ratio. Topology is deemed valid only if every edge is connected to two faces and two vertices, and edges on each face form closed loops. Broken topology makes it difficult to integrate the two modules and output valid B-reps. This further shows the advantage of our structured latent geometry, which unifies topology and geometry and jointly generates rather than in two separate stages.

6.4.2 Post-process Thresholds. To study the impact of different thresholds on B-rep post-processing, we perform a grid search of the bounding box and decoded shape feature threshold values from subsection 4.3. Concretely, we trained a latent diffusion model on the DeepCAD validation set and report the ratio of valid B-reps that are watertight from 100 randomly generated samples. Figure 14 shows the valid ratios obtained under various threshold combinations. From the figure, we see that the best ratio is achieved when

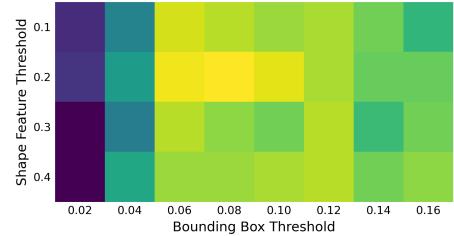
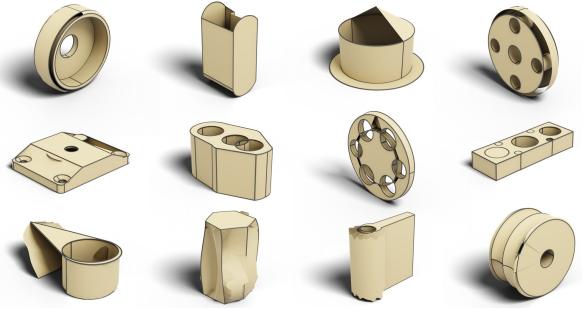


Fig. 14. Valid ratio under different threshold values. Brighter color indicates higher ratio value. Result are averaged over 100 randomly generated samples from a model trained on the DeepCAD validation set.

the bounding box threshold is within the range of 0.06 to 0.1, and the shape feature threshold is around 0.2. A large bounding box threshold tends to incorrectly merge unique edges or faces, whereas a very small threshold fails to effectively identify duplicated geometry.

6.5 Failure Cases

Figure 15 shows B-reps generated by *BrepGen* from three common failure case categories. 1) Missing faces resulting in non-watertight solids (top row). This failure is anticipated given *BrepGen* can not guarantee watertight outputs. A potential solution involves detecting and denoising additional faces in the open regions. 2) Self-intersection of edges or faces leading to broken geometry after trimming (middle row). Future integration of a self-intersection loss might reduce such occurrences. 3) Wobbly and broken geometry (bottom row). This is mostly caused by noise in the decoded points, or inconsistent geometry between a surface and its connected edges.

Fig. 15. Three common failure cases categories for *BrepGen*. Top row: non-watertight solids. Middle row: self-intersecting edges or faces. Bottom row: wobbly and broken geometry.

7 LIMITATIONS AND FUTURE WORK

BrepGen enables the generation of 3D models in the Boundary representation but has several limitations that warrant future work. *BrepGen* supports only single body solids; more complicated CAD models with multiple assembled bodies are left to future work. If edges or faces are too close to one another they will be merged and identified as one after deduplication. In practice, this implies a minimum threshold, determining how close two similar surfaces or edges

can be. We train on data with a minimum threshold of 0.05 after normalizing the CAD geometry to the $[-3, 3]$ range. This limitation is similar to other sketch-and-extrude generation approaches, e.g. [Xu et al. 2022], that quantize geometry to a set bit range resulting in vertex merging. The chosen threshold of 0.05 is roughly equivalent to 1 bin difference after 7 bit quantization. *BrepGen* does not guarantee watertight solids. Small gaps may exist due to the slow convergence of the denoising process. Occasionally entire faces are also missing. Experimentally we find that the valid ratio for *BrepGen*, which includes watertightness, is higher than for other architectures. Finally, while the heuristic post-processing used to generate the B-rep is simple, fast, and can handle complicated data, to achieve better results future work on a learning-based post-processing module may provide more robust handling of invalid shapes.

8 CONCLUSION

We introduced *BrepGen*, a generative diffusion model for direct B-rep generation. Extensive experiments showed that *BrepGen* surpasses existing methods and establishes state-of-the-art results for B-rep generation. We outlined our newly collected *Furniture B-rep Dataset* and results that demonstrate the ability of *BrepGen* to generate complicated B-rep 3D models with free-form and doubly-curved surfaces for the first time. We believe *BrepGen* moves us one step closer towards an automatic system capable of directly generating B-reps to reduce the extensive manual labor required from skilled designers using today’s professional CAD software.

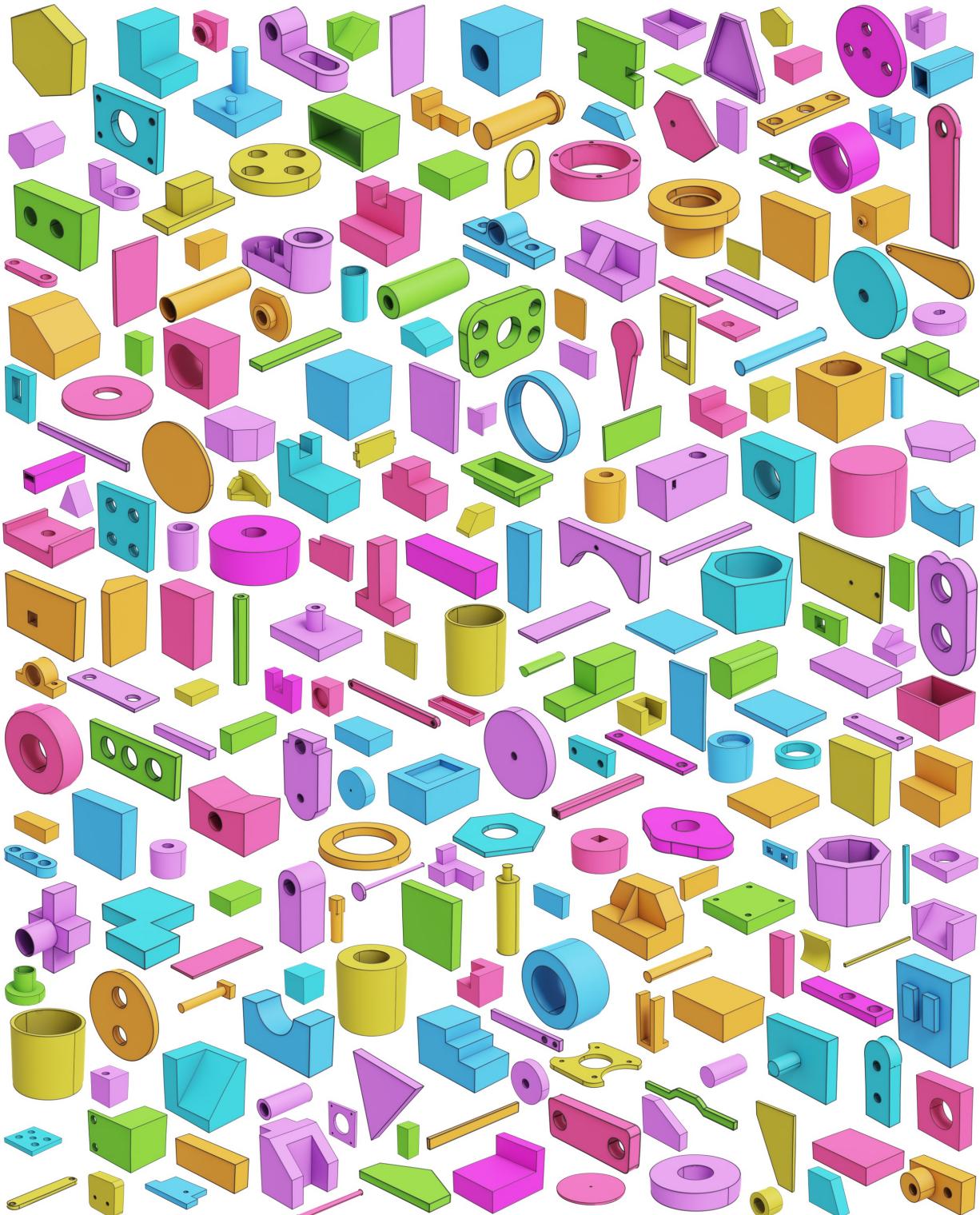
ACKNOWLEDGMENTS

This research is partially supported by NSERC Discovery Grants with Accelerator Supplements and DND/NSERC Discovery Grant Supplement, NSERC Alliance Grants, and John R. Evans Leaders Fund (JELF). We also thank Onshape for their support and access of the publicly available CAD models.

REFERENCES

- Antonio Allegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. 2023. Poly-Diff: Generating 3D Polygonal Meshes with Diffusion Models. *arXiv preprint arXiv:2312.11417* (2023).
- Atin Angrish, Benjamin Craver, and Binil Starly. 2019. “FabSearch”: A 3D CAD Model-Based Search Engine for Sourcing Manufacturing Services. *Journal of Computing and Information Science in Engineering* 19, 4 (2019). <https://doi.org/10.1115/1.4043211> *arXiv:https://asmedigitalcollection.asme.org/computingengineering/article-pdf/19/4/041006/5999469/jcise_19_4_041006.pdf* 041006.
- Silvia Ansaldi, Leila De Floriani, and Bianca Falcidieno. 1985. Geometric modeling of solid objects by using a face adjacency graph representation. *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* 19, 3 (1985), 131–139.
- Shijie Bian, Daniele Grandi, Tianyang Liu, Pradeep Kumar Jayaraman, Karl Willis, Elliot Sadler, Bodia Borjin, Thomas Lu, Richard Otis, Nhut Ho, and Bingbing Li. 2023. HG-CAD: Hierarchical Graph Learning for Material Prediction and Recommendation in Computer-Aided Design. *Journal of Computing and Information Science in Engineering* 24, 1 (10 2023), 011007.
- Weijuan Cao, Trevor Robinson, Yang Hua, Flavien Boussuge, Andrew R. Colligan, and Wanbin Pan. 2020a. Graph Representation of 3D CAD Models for Machining Feature Recognition With Deep Learning. In *Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE)*. ASME.
- Weijuan Cao, Trevor Robinson, Yang Hua, Flavien Boussuge, Andrew R. Colligan, and Wanbin Pan. 2020b. Graph Representation of 3D CAD Models for Machining Feature Recognition With Deep Learning. In *Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE)*. ASME.
- Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. 2003. On visual similarity based 3D model retrieval. In *Computer graphics forum*, Vol. 22. Wiley Online Library, 223–232.
- Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. 2020. BSP-Net: Generating Compact Meshes via Binary Space Partitioning. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2020.
- Tao Du, Jeevana Priya Inala, Yewen Pu, Andrew Spielberg, Adriana Schulz, Daniela Rus, Armando Solar-Lezama, and Wojciech Matusik. 2018. Inversecsg: Automatic conversion of 3d models to csg trees. *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* 37, 6 (2018), 1–16.
- Kevin Ellis, Maxwell Nye, Yewen Pu, Felix Sosa, Josh Tenenbaum, and Armando Solar-Lezama. 2019. Write, execute, assess: Program synthesis with a repl. In *Advances in Neural Information Processing Systems (NeurIPS)*. 9169–9178.
- Haoxiang Guo, Shilin Liu, Hao Pan, Yang Liu, Xin Tong, and Baining Guo. 2022. ComplexGen: CAD Reconstruction by B-Rep Chain Complex Generation. *ACM Trans. Graph. (SIGGRAPH)* 41, 4, Article 129 (July 2022), 18 pages. <https://doi.org/10.1145/3528223.3530078>
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. 2022. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research* 23, 1 (2022), 2249–2281.
- Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).
- Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems* 34 (2021), 12454–12465.
- Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. 2022. Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.
- Pradeep Kumar Jayaraman, Joseph G. Lambourne, Nishkrit Desai, Karl D. D. Willis, Aditya Sanghi, and Nigel J. W. Morris. 2022. SolidGen: An Autoregressive Model for Direct B-rep Synthesis. *“arXiv Preprint”* (2022). <https://doi.org/10.48550/ARXIV.2203.13944>
- Pradeep Kumar Jayaraman, Aditya Sanghi, Joseph G. Lambourne, Karl D.D. Willis, Thomas Davies, Hooman Shayani, and Nigel Morris. 2021. UV-Net: Learning From Boundary Representations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11703–11712.
- Benjamin Jones, Dalton Hildreth, Duowen Chen, Ilya Baran, Vladimir G Kim, and Adriana Schulz. 2021. Automate: A dataset and learning approach for automatic mating of cad assemblies. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–18.
- Benjamin T. Jones, Michael Hu, Milin Kodnongbu, Vladimir G Kim, and Adriana Schulz. 2023. Self-Supervised Representation Learning for CAD. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21327–21336.
- Kacper Kania, Maciej Zięba, and Tomasz Kajdanowicz. 2020. UCSG-Net–Unsupervised Discovering of Constructive Solid Geometry Tree. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. 2019. ABC: A Big CAD Model Dataset For Geometric Deep Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Joseph G. Lambourne, Karl D.D. Willis, Pradeep Kumar Jayaraman, Aditya Sanghi, Peter Meltzer, and Hooman Shayani. 2021. BRepNet: A Topological Message Passing System for Solid Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12773–12782.
- 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.
- P. Li, J. Guo, X. Zhang, and D. Yan. 2023. SECAD-Net: Self-Supervised CAD Reconstruction by Learning Sketch-Extrude Operations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 16816–16826. <https://doi.org/10.1109/CVPR52729.2023.01613>
- Jiayi Liu, Hou In Ivan Tam, Ali Mahdavi-Amiri, and Manolis Savva. 2023. CAGE: Controllable Articulation GEneration. *arXiv preprint arXiv:2312.09570* (2023).
- Luping Liu, Yi Ren, Zhipjie Lin, and Zhou Zhao. 2022. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778* (2022).
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.
- Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11461–11471.
- D.E. Muller and F.P. Preparata. 1978. Finding the intersection of two convex polyhedra. *Theoretical Computer Science* 7, 2 (1978), 217–236. [https://doi.org/10.1016/0304-3975\(78\)90051-8](https://doi.org/10.1016/0304-3975(78)90051-8)
- Chandrakana Nandi, Anat Caspi, Dan Grossman, and Zachary Tatlock. 2017. Programming language tools and techniques for 3D printing. In *2nd Summit on Advances in Programming Languages*.

- in Programming Languages (SNAPL 2017).* Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Chandrakana Nandi, James R Wilcox, Pavel Panekha, Taylor Blau, Dan Grossman, and Zachary Tatlock. 2018. Functional programming for compiling and decompiling computer-aided design. *Proceedings of the ACM on Programming Languages* 2, ICFP (2018), 1–31.
- Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. 2020. Polygen: An autoregressive generative model of 3d meshes. In *International Conference on Machine Learning (ICML)*. PMLR, 7220–7229.
- Onshape. 2012. <http://www.onshape.com>.
- Daxuan Ren, Jianmin Zheng, Jianfei Cai, Jiatong Li, Haiyong Jiang, Zhongang Cai, Junzhe Zhang, Liang Pan, Mingyuan Zhang, Haiyu Zhao, et al. 2021. Csg-stump: A learning friendly csg-like representation for interpretable shape parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12478–12487.
- Daxuan Ren, Jianmin Zheng, Jianfei Cai, Jiatong Li, and Junzhe Zhang. 2022. ExtrudeNet: Unsupervised Inverse Sketch-and-Extrude for Shape Parsing. In *ECCV*.
- Daniel Ritchie, Paul Guerrero, R Kenny Jones, Niloy J Mitra, Adriana Schulz, Karl DD Willis, and Jiajun Wu. 2023. Neurosymbolic Models for Computer Graphics. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, 545–568.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
- Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. 2023. Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5466–5475.
- Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. 2018. CSGNet: Neural Shape Parser for Constructive Solid Geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Gopal Sharma, Difan Liu, Evangelos Kalogerakis, Subhransu Maji, Siddhartha Chaudhuri, and Radomír Měch. 2020. ParSeNet: A Parametric Surface Fitting Network for 3D Point Clouds. arXiv:2003.12181 [cs.CV].
- Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2023. Meshgpt: Generating triangle meshes with decoder-only transformers. *arXiv preprint arXiv:2311.15475* (2023).
- Dmitriy Smirnov, Mikhail Bessmeltsev, and Justin Solomon. 2021. Learning Manifold Patch-Based Representations of Man-Made Shapes. In *International Conference on Learning Representations (ICLR)*.
- Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. 2019. Learning to Infer and Execute 3D Shape Programs. In *International Conference on Learning Representations (ICLR)*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) (NIPS’17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in Neural Information Processing Systems (NeurIPS)* (2015).
- Kehan Wang, Jia Zheng, and Zihan Zhou. 2022. Neural Face Identification in a 2D Wireframe Projection of a Manifold Object. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1612–1621. <https://doi.org/10.1109/CVPR52688.2022.00167>
- Xiaogang Wang, Yuelang Xu, Kai Xu, Andrea Tagliasacchi, Bin Zhou, Ali Mahdavi-Amiri, and Hao Zhang. 2020. PIE-NET: Parametric Inference of Point Cloud Edges. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 20167–20178.
- K.J. Weiler. 1986. *Topological structures for geometric modeling*. University Microfilms.
- Karl DD Willis, Pradeep Kumar Jayaraman, Hang Chu, Yunsheng Tian, Yifei Li, Daniele Grandi, Aditya Sanghi, Linh Tran, Joseph G Lambourne, Armando Solar-Lezama, and Wojciech Matusik. 2022. JoinABLE: Learning Bottom-up Assembly of Parametric CAD Joints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Karl DD Willis, Pradeep Kumar Jayaraman, Joseph G Lambourne, Hang Chu, and Yewen Pu. 2021a. Engineering sketch generation for computer-aided design. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshop)*. 2105–2114.
- Karl D. D. Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. 2021b. Fusion 360 Gallery: A Dataset and Environment for Programmatic CAD Construction from Human Design Sequences. *ACM Transactions on Graphics (TOG)* 40, 4 (2021).
- Rundi Wu, Chang Xiao, and Changxi Zheng. 2021. DeepCAD: A Deep Generative Network for Computer-Aided Design Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Xiang Xu, Pradeep Kumar Jayaraman, Joseph G. Lambourne, Karl D.D. Willis, and Yasutaka Furukawa. 2023. Hierarchical Neural Coding for Controllable CAD Model Generation. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) (ICML ’23). JMLR.org, Article 1601, 19 pages.
- Xianghao Xu, Wenzhe Peng, Chin-Yi Cheng, Karl D.D. Willis, and Daniel Ritchie. 2021. Inferring CAD Modeling Sequences Using Zone Graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6062–6070.
- Xiang Xu, Karl DD Willis, Joseph G Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. 2022. SkexGen: Autoregressive Generation of CAD Construction Sequences with Disentangled Codebooks. In *International Conference on Machine Learning*. PMLR, 24698–24724.
- Fenggen Yu, Qimin Chen, Maham Tanveer, Ali Mahdavi Amiri, and Hao Zhang. 2023. DualCSG: Learning Dual CSG Trees for General and Compact CAD Modeling. *arXiv preprint arXiv:2301.11497* (2023).
- Fenggen Yu, Zhiqin Chen, Manyi Li, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. 2022. CAPRI-Net: Learning compact CAD shapes with adaptive primitive assembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11768–11778.
- Shengdi Zhou, Tianyi Tang, and Bin Zhou. 2023. CADParser: A Learning Approach of Sequence Modeling for B-Rep CAD. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*. ijcai.org, 1804–1812. <https://doi.org/10.24963/IJCAI.2023/200>

Fig. 16. Unconditionally generated mechanical part B-reps from *BrepGen*.

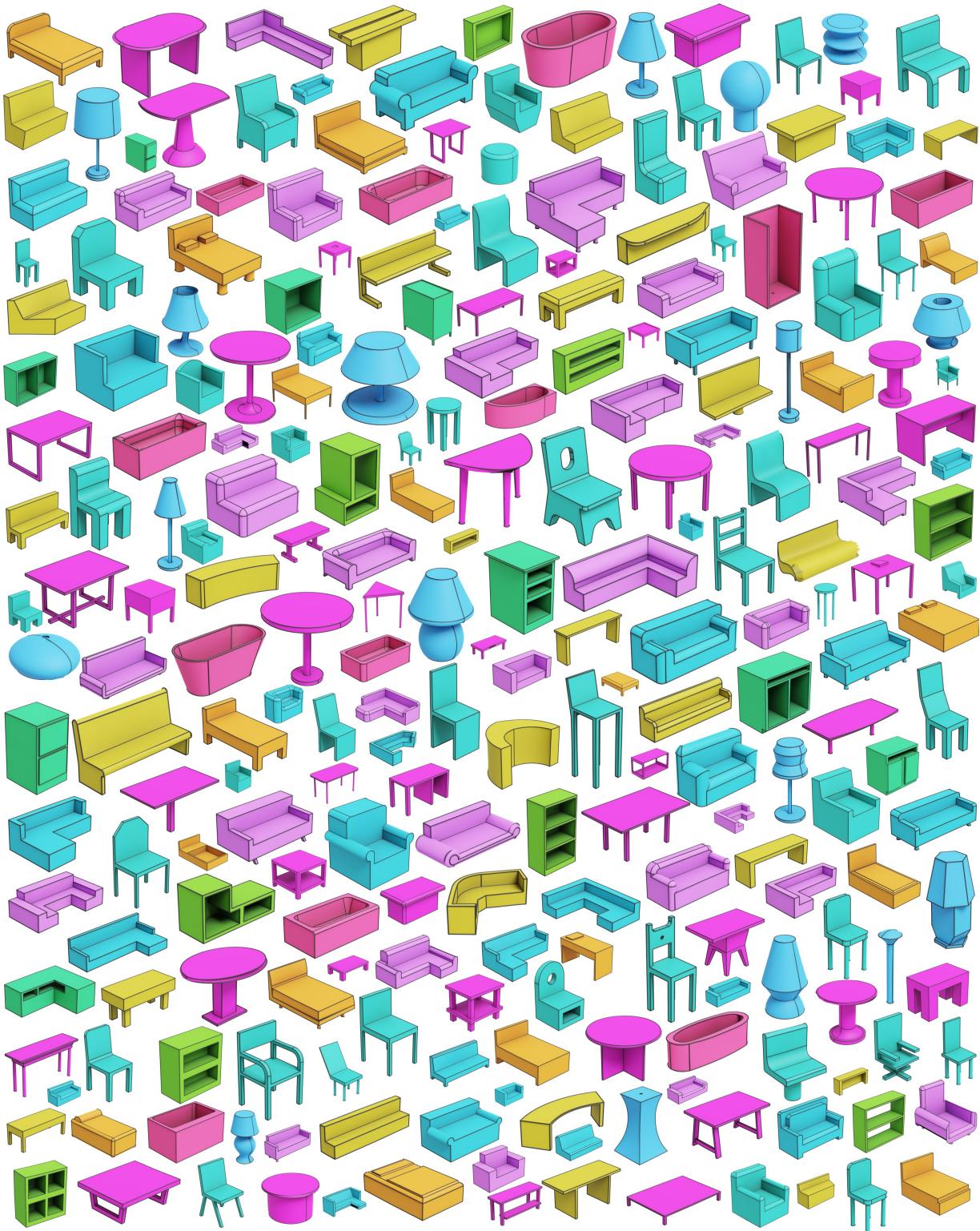


Fig. 17. Conditionally generated furniture B-reps from *BrepGen*, colored by category.