B-Rep Distance Functions (BR-DF)
How to Represent a B-Rep Model by Volumetric Distance Functions?

Fuyang Zhang[1]   Pradeep Kumar Jayaraman[2]   Xiang Xu[2]   Yasutaka Furukawa[1,3]

[1]Simon Fraser University   [2]Autodesk   [3]Wayve

{fuyangz, furukawa}@sfu.ca   {xiang.xu, pradeep.kumar.jayaraman}@autodesk.com
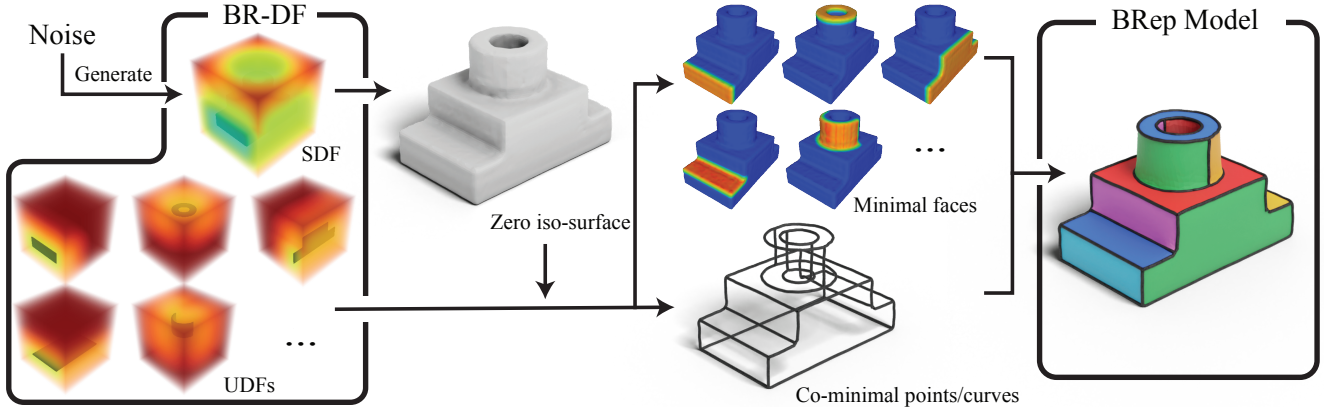
Figure 1. BR-DF is a geometric representation for Boundary Representation (B-Rep) models. An SDF encodes surface geometry. UDFs encode vertices, edges, faces, and their connectivity. An extension of the Marching Cubes converts BR-DF to a faceted B-Rep model.

Abstract

This paper presents a novel geometric representation for CAD Boundary Representation (B-Rep) based on volumetric distance functions, dubbed B-Rep Distance Functions (BR-DF). BR-DF encodes the surface mesh geometry of a CAD model as signed distance function (SDF). B-Rep vertices, edges, faces and their topology information are encoded as per-face unsigned distance functions (UDFs). An extension of the Marching Cubes algorithm converts BR-DF directly into watertight CAD B-Rep model (strictly speaking a faceted B-Rep model). A surprising characteristic of BR-DF is that this conversion process never fails. Leveraging the volumetric nature of BR-DF, we propose a multi-branch latent diffusion with 3D U-Net backbone for jointly generating the SDF and per-face UDFs of a BR-DF model. Our approach achieves comparable CAD generation performance against SOTA methods while reaching the unprecedented 100% success rate in producing (faceted) B-Rep models.

1. Introduction

Volumetric distance functions are fundamental in geometry processing. The introduction of the Marching Cubes al-

gorithm [16] and signed distance functions (SDFs) marked a significant breakthrough in surface reconstruction for the field of computer graphics [1, 12]. These concepts were later adopted by the computer vision community through level-set and geometric active contour methods [3], allowing geometry optimization without explicitly managing the surface topology changes. In the era of deep learning and generative artificial intelligence (GenAI), signed distance functions remain central to geometry reconstruction and generation [2, 21, 33], enabling high-quality 3D asserts creation through large-scale training [7, 13, 37].

Solid modeling in computer-aided design (CAD) focuses on designing and editing shapes intended for real-world manufacturing. Ensuring watertightness is therefore a critical requirement and a key challenge for GenAI models designed to synthesize CAD data structures, such as Boundary Representations (B-Reps) [30]. Signed distance functions, by their very nature, guarantee watertight representations, making them a compelling choice for representing CAD B-Rep model in solid modeling. This paper advances the use of such volumetric distance functions by introducing an innovative representation that is both equivalent to and mutually convertible with a faceted B-Rep model [24], i.e., a segmented triangle mesh where each segment corresponds

1

to a face in B-Rep, and the boundary curves between segments correspond to B-Rep edges.

Our proposed representation, B-Rep Distance Functions (BR-DF), combines a signed distance function (SDF) with per-face unsigned distance functions (UDFs) to fully capture the geometry and topology of a watertight CAD model. Surprisingly, we find that a simple algorithm can reliably convert BR-DF into faceted B-Rep—a process that *never fails*. Specifically, we introduce a novel extension of the Marching Cubes algorithm that reconstructs the CAD surface boundary from SDF, and then conditioned on the surface mesh and UDFs, recovers the complete B-Rep vertices, edges, and faces together with their topology. Concretely, for a CAD model with F faces, any point on the zero isosurface of SDF has an associated F-dimensional face UDF scalar fields. A B-Rep face is determined by the minimal UDF value in the face UDF fields. A B-Rep edge is defined when at least two faces share the same minimum UDF value, and similarly a B-Rep vertex occurs where at least three faces share the minimum UDF value.

Since our representation is a set of scalar functions on volumetric grids, we present a simple latent diffusion with 3D U-Net backbone for BR-DF generation. Qualitative and quantitative evaluations on DeepCAD [32] and ABC [11] datasets demonstrate that BR-DF based generative model matches the performance of recent SOTA methods [32, 36], while achieving the unprecedented 100% success rate of producing faceted B-Rep models. While faceted B-Reps still needs to be converted into B-Reps using standard mesh-to-BRep conversion tools available in CAD software, the topology generated by our method greatly aids in this process. We believe this representation will play a key role in developing generative models for CAD that guarantee watertightness. To summarize, our paper makes three key contributions: 1) a novel volumetric distance function representation of B-Rep models, named BR-DF; 2) a Marching Cubes and Triangles algorithm to convert BR-DF to faceted B-rep models with 100% success rate; 3) A simple B-Rep generative model based on BR-DF with performance competitive to the state-of-the-art approaches. This paper potentially marks a key innovation in CAD geometry representation—analogous to the impact that Marching Cubes had on raster geometry modeling 25 years ago.

2. Related Work

This section reviews related work on distance functions and CAD model generation.

2.1. Signed Distance Functions

Signed distance functions (SDFs) have been instrumental in geometry processing. The introduction of the Marching Cubes algorithm [15] enabled the creation of 3D mesh models from SDFs, setting the foundation for many surface re-

construction techniques. The Digital Michelangelo Project [1] is a notable example, which utilized SDFs to capture complex 3D geometry. In the Computer Vision domain, level-set and active contour methods utilized SDFs, allowing implicit surface representation that facilitated topology adaptation [3]. Over time, distance functions became less prominent in Computer Vision, as depth maps, point clouds, and mesh representations gained popularity; however, SDFs combined with Marching Cubes remain fundamental for surface extraction in geometry processing.

The integration of deep learning has renewed interest in distance functions. Neural implicit representations, such as multilayer perceptron (MLP), predict SDF values for single-image reconstruction or scan completion [2, 21, 33]. Volumetric grids are a popular alternative for the SDF representation and can be scaled to high resolutions [6, 20]. NeuralRecon reconstructs 3D scenes in real-time from a monocular video by predicting local TSDF volumes [25]. Similarly, Atlas directly regresses a TSDF volume from a set of posed RGB images, utilizing a 2D convolutional neural network (CNN) to extract features from each image [19].

2.2. Generative Models for B-Reps

Recent advancements in boundary representation (B-Rep) generation have focused on learning the construction sequences of sketch-and-extrude operations. This approach allows the generated sequences to be parsed by solid modeling kernels, enabling the rebuilding of B-Reps. Deep-CAD [32] pioneered this direction by releasing a dataset of sketch-and-extrude sequences, laying the groundwork for further research. Subsequent works introduced significant improvements: SkexGen [34] leveraged autoregressive Transformer models [28] with disentangled learning of geometry and topology, leading to a substantial enhancement in generation quality. Later, HNC-CAD [35] introduced hierarchical latent codes that capture natural hierarchies in the data structure, better preserving design intent and improving editability. A key advantage of the sketch-and-extrude paradigm is its inherent guarantee of watertightness, as extruded 2D sketches are constrained to form closed loops. However, these methods rely on the availability of explicit construction sequences alongside B-Reps, limiting their applicability to simpler, primitive-based shapes.

Direct B-Rep generation creates 3D geometry while maintaining topological consistency among faces, edges, and vertices. SolidGen [9] employs Transformer models and pointer networks [29] to generate B-Rep entities in a bottom-up manner. While this method benefits from larger datasets compared to sketch-extrude techniques, it remains constrained to basic geometric primitives and does not consume freeform data. BrepGen [36] introduced a structured latent representation, learning face and edge embeddings via uniformly sampled point grids in the parameter

space [8]. By incorporating diffusion models, it progressively generates faces and edges in a top-down manner, facilitating the creation of complex freeform surfaces while ensuring topological consistency. However, both of these methods adopt a surface-modeling approach to generating solid models and struggle with maintaining watertightness. A particularly relevant method to our research is Split-and-Fit [14], which reconstructs B-Reps from point clouds using learned Voronoi partitions. By integrating geometric algorithms and heuristics, it significantly improves watertightness in post-processing compared to prior approaches. In contrast, our BR-DF representation offers a simpler and more robust alternative based on signed and unsigned distance functions, enabling conversion to faceted B-Reps using a straightforward algorithm with guaranteed success.

3. B-Rep Distance Functions (BR-DF)

BR-DF is the core of our technical innovation. This section provides a detailed explanation of its representation. Next section explains the novel extension of Marching Cubes algorithm that converts BR-DFs to faceted B-Rep models.

3.1. Signed Distance Function (SDF)

BR-DF consists of two volumetric distance functions (see Figure 1). The first function is the widely used signed distance function (SDF) that returns a signed distance to the nearest point on the surface, with negative values indicating inside and positive values for outside.

SDF zero iso-surface. Following existing literature, the zero iso-surface of an SDF yields the surface geometry [1, 12] which we refer to as an "SDF-surface." Note that our representation assumes a solid B-Rep model, which is watertight and has a well-defined SDF.

3.2. Per-face Unsigned Distance Function (UDF)

The second function is an unsigned distance function (UDF) that returns an unsigned distance to the closest point on the face of a CAD B-Rep model. When UDFs are sliced along the SDF zero iso-surface, they encode B-Rep vertices, edges, faces, and their connectivity as follows.

UDF minimum face. A straightforward interpretation of UDF would be to consider a set of points with the zero UDF value as a face on the SDF surface. However, UDFs are unsigned and never zero. One might ask why not a per-face SDF, with the zero iso-curve defining the face boundary. However, this leads to inconsistencies of the zero iso-curves between adjacent faces. Instead, we consider a "UDF minimum face" as the set of points where the UDF of a face is the minimum among all the faces. This definition effectively partitions the SDF-surface without overlaps or gaps.

UDF co-minimum curve. The UDF minimum face definition deduces "UDF co-minimal curves". B-Rep edges or a face boundary is a set of points where two face UDFs are the minimum simultaneously.

UDF co-minimum point. The UDF co-minimum curve definition deduces "UDF co-minimum points". A B-Rep vertex is where three UDFs are the minimum simultaneously. B-Rep points at X-junctions or higher-order degrees are degenerate configurations in the BR-DF representation, represented by two or more nearby co-minimum points.

4. Marching Cubes and Triangles (MCT)

Marching Cubes and Triangles (MCT) takes a discrete BR-DF of distance functions sampled at 3D grid points as input and outputs a faceted B-Rep model, i.e., a B-Rep model except for parametric curve/surface fitting. The remarkable aspect of the BR-DF representation and MCT algorithm is their robustness—this process never fails and consistently guarantees watertightness and valid topology. MCT always converts any BR-DF into a valid faceted B-Rep model.

Marching Cubes extracts a triangulated mesh from a surface SDF. Triangle vertices are on the edges of the uniform grid, and per-face UDF values are linearly interpolated at the mesh vertices (see Figure 2 Step 1 for details).

Marching Triangles is a novel extension that extracts CAD B-Rep vertices, edges, faces, and their topology from the triangulated mesh by identifying UDF minimal faces and tracing the co-minimal points and curves. Marching Cubes applies a 256-way rule to each cube independently based on the signs of the SDF at its 8 grid points, achieving global consistency and a watertight mesh. Similarly, Marching Triangles applies a 3-way rule to each triangle independently depending on the face UDFs at its 3 vertices, achieving global consistency and a valid faceted B-Rep model.

Concretely, we determine the minimal face at each mesh vertex by selecting the face with the smallest UDF, applying perturbations in case of a tie (see Step 2 in Figure 2). For each triangle, we extract a B-Rep vertex and/or edges via the following 3-way rule as illustrated in Figure 2 Step 3: 1) if the minimal faces at all the vertices are the same, no vertex or edge are created; 2) if two vertices share the same minimal face, a B-Rep edge is created to separate the two vertices from the third by cutting edges where the two UDFs are equal via linear interpolation; and 3) if all three minimal faces are different, a B-Rep vertex and three incident B-Rep edges are created based on the UDF values at these three vertices (detailed equations are in the supplementary). Note that this operation is performed locally for each triangle but the algorithm does achieve global consistency.

5. BR-DF Generative Model

BR-DF is a set of 3D scalar fields with an arbitrary number of UDFs (i.e., faces). To determine the number of UDFs, we follow BrepGen [36], which first generates axis-aligned
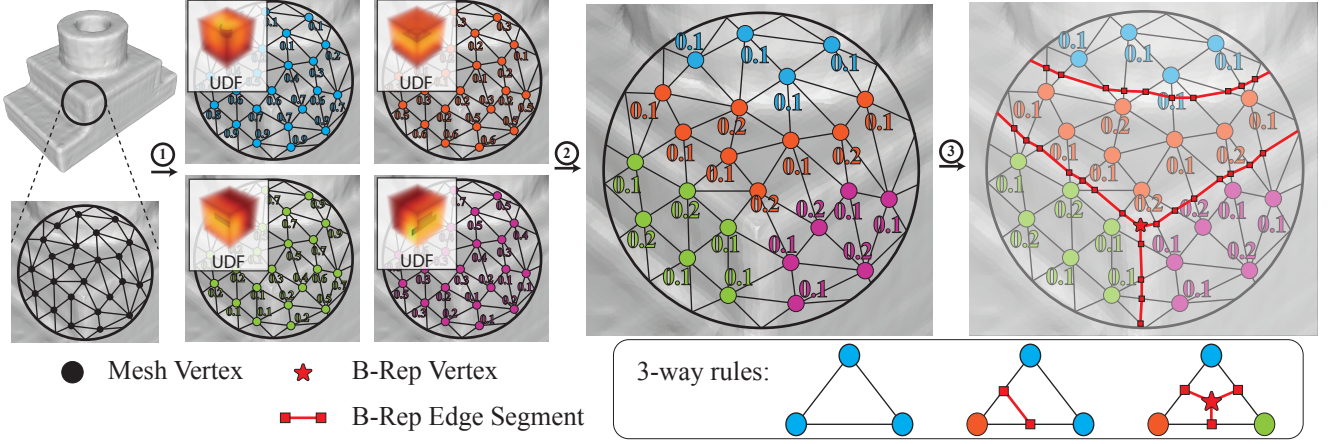
Figure 2. MCT algorithm: 1) Given the volumetric UDFs for each face, interpolate a UDF value at each mesh vertex; 2) Select the face with the smallest UDF at each mesh vertex; and 3) Extract B-Rep vertices and edges by applying the 3-way rules.

bounding boxes of the faces. Given the number of faces, a latent diffusion model with a 3D U-Net backbone generates the SDF and UDFs, using 3D VQ-VAEs as the latent encoder/decoder. This section explains the bounding box generation, 3D VQ-VAEs, and the latent diffusion model.

5.1. Preliminaries

We represent an SDF or a UDF as 64^3 values sampled at volumetric grid points within a cube. Given a B-Rep model, we determine its tightest axis-aligned bounding box and expand it to a cube while maintaining the center and adding a 20% margin on all sides. We normalize the 3D coordinates so that the cube center is the origin, and its dimensions are 2.0. We employ simple heuristics to compute SDF and UDF values by using *compute_signed_distance* and *compute_distance* from Open3D [38], and truncate results to $[-0.1, 0.1]$ range because important values are near zero.

5.2. Bounding Box Generation

We borrow the *face position denoiser module* from Brep-Gen [36], which predicts axis-aligned bounding boxes as the 3D coordinates of their bottom-left and top-right corners. It generates a predetermined maximum of 40 bounding boxes (60 during inference to minimize false negatives), while applying a non-maximum suppression (i.e., two bounding boxes become duplicates if the coordinates are within 0.04 distance for the two corners).

5.3. 3D VQ-VAEs

We use a standard VQ-VAEs [27] with 3D ResNet backbone [4] to encode the distance function of 64^3 spatial resolution into a latent volume of 4^3 resolution. The VQ-VAE architecture consists of four downsampling and four upsampling blocks. Each block contains three layers of 3D-ResNet blocks. The feature dimensions for downsampling

are 64-128-128-256, and in reverse order for upsampling. We train two VQ-VAEs (one for an SDF and one for a UDF) using a combination of MSE reconstruction loss and code-book loss. The codebook size is set to 8192.

5.4. BR-DF Latent Diffusion Module

BR-DF generation follows a multi-branch diffusion architecture [26]. It comprises of a surface branch for the surface SDF and the face branches for the face UDFs (see Figure 3). The number of face branches is set to the number of generated bounding boxes at inference and that of the ground-truth at training.

The BR-DF Latent Diffusion Model Backbone is a 3D U-Net [22] with three encoding and three decoding blocks. The first two encoding blocks are followed by a downsampling operation, *nn.Conv3d(stride=2)*, and the last two decoding blocks have an upsampling operation, *nn.Upsample(scale_factor=2, model="nearest")*. Each block has two layers of 3D-ResNet blocks. A two-layer MLP injects 6-DoF bounding box parameters into the face branch by adding it to the face feature. Similarly, the average of all bounding boxes is added to the surface branch.

Three inter-branch layers are added following each U-Net block: face-to-face (f2f) layer, face-to-surface (f2s) layer, and surface-to-face (s2f) layer.
• face-to-face (f2f) layer and face-to-surface (f2s) layer have the same structure. The source feature volume (a face volume for f2f layer and the surface volume for f2s layer) cross attention to the target feature volumes (remaining face features for f2f layer and all face features for f2s layer) at the same location. That means each token of the source feature attends only to the corresponding location in the target feature maps. A 3D ResNet block then refines the feature.
• surface-to-face (s2f) layer is a 3D ResNet block that takes an input as the concatenation of the surface volume and the
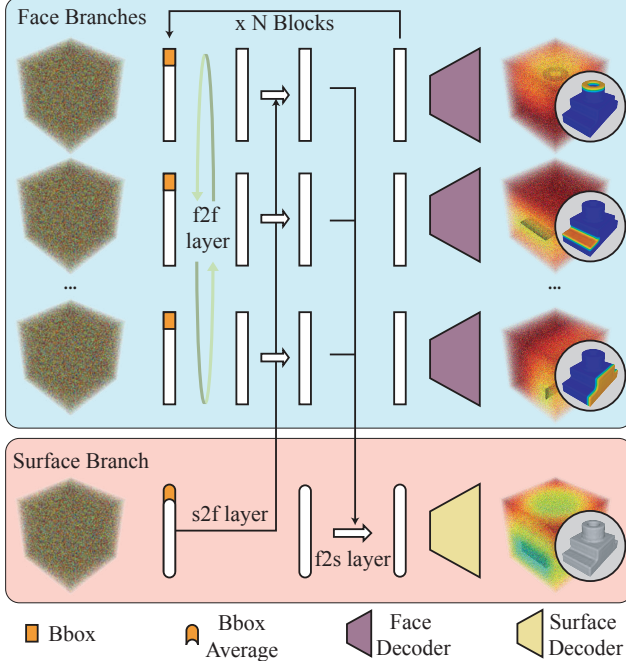
4

Figure 3. The multi-branch diffusion architecture. Top and bottom represent the face and surface branches, respectively. The number of faces corresponds to the generated bounding boxes during inference and the ground truth bounding boxes during training. Three inter-branch cross-attention modules (f2f, f2s, s2f) are added.

face volume over the channel dimension, and outputs the updated face latent.

Loss Function. The model is trained to predict the L2-norm regression loss of the added sampled noise.

$$L = \mathbb{E}_{t, \{Z_f^i\}_{i=1}^N, Z_s, \{\epsilon^i \sim \mathcal{N}(0,I)\}_{i=1}^N, \epsilon \sim \mathcal{N}(0,I)}$$
$$\left[\sum_{i=1}^N \|\epsilon^i - \epsilon_f^i(\overline{Z_f^i}, t)\|^2 + \|\epsilon - \epsilon_s(\overline{Z_s}, t)\|^2 \right], \quad (1)$$

where ϵ, ϵ^i are gaussian noises; ϵ_f, ϵ_s denotes the model outputs; Z_f^i and Z_s are face latent and surface latent; $\overline{Z}$ is the latent with added noise.

6. Experiments

This section presents the generation results of our approach. Extensive analysis demonstrates the advantages of B-Rep Distance Functions (BR-DF). Unlike traditional Boundary Representations which involves complex geometric structures, BR-DF employs volumetric-based distance functions that bypasses these complexities, making learning much easier. The Marching Cube and Triangles (MCT) algorithm ensures robust conversion to faceted B-Rep with 100% watertightness and valid topology. The results show that our approach matches the state-of-the-art performance while having a clear advantage in terms of robustness.

6.1. Experiment Setup

Dataset. We evaluate generation quality on two datasets: 1) DeepCAD dataset [32] comprises of mechanical parts made from sketch-and-extrude operations, 2) ABC dataset [11] includes a wide variety of CAD parts in B-Rep formats from industrial design. Following [31], we remove duplicated models in the datasets. Periodic surfaces (cylinders, etc.) are split on the seams following SolidGen and BrepGen [10, 36]. B-Reps with more than 40 faces are also filtered out to manage memory usage. After filtering, a total of 55,897 DeepCAD data and 120,868 ABC data are used for training.

Implementation Details. We implement our model in Py-Torch, using AdamW [17] optimizer with a learning rate of with $5e^{-4}$ for all modules. We use continuous latent for subsequent latent diffusion and the quantization layer is absorbed by the decoder. Following BrepGen [36], the bounding box generation module is a standard Transformer diffusion model with pre-layer normalization, 12 self-attention layers, and 12 heads. Hidden dimension is 1024, and feature dimension is 768. Both bounding box generation module and BR-DF latent diffusion module use 1,000 diffusion steps and a linear beta schedule from $1e^{-4}$ to 0.02. The bounding box generation module is trained for 500 epochs with a batch size 256 on a single NVIDIA A100 GPU, while the BR-DF latent diffusion module is trained for 1000 epochs with a batch size of 64 on two A100 GPUs.

6.2. Inference

At inference, we use a 1,000 step DDPM [5] to denoise the face bounding boxes. Following BrepGen, we increase the number of bounding box tokens to 60, which exceeds the maximum threshold used during training. This prevents noisy bounding boxes from incorrectly merging in the denoising process. Non-maximum suppression eliminates duplicated bounding boxes. Next, the face UDFs and surface SDF in BR-DF are jointly denoised conditioned on the previously generated bounding boxes. We use a DDIM [23] scheduler with 200 steps for fast sampling. Finally, MCT extracts the faceted B-Rep from the generated BR-DF.

6.3. Post-Processing

Marching Cubes and Triangles (MCT) algorithm extracts faceted B-Rep models from BR-DF with 100% success rate. To improve visual quality and support further B-Rep conversion, we apply a conventional post-processing procedure to smooth both the mesh surface and its boundaries. We first compute parametric curves for each boundary while preserving T-junctions. We then perform multiple iterations of Laplacian smoothing and quadric-based edge-collapse remeshing to simplify and refine each face. During this process, boundary vertices at the same location from different faces are averaged to maintain geometric consistency. Please refer to the supplementary for further details.
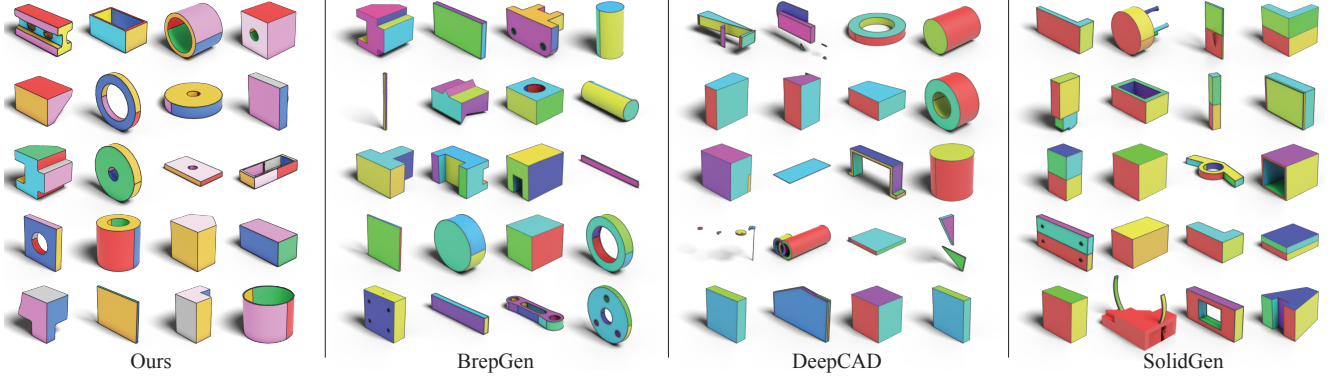
5

Figure 4. Unconditional generation results on DeepCAD dataset [32]. Our method achieves comparable performance to the state-of-the-art methods, while maintaining 100% success rate.

Method	COV % ↑	MMD ↓	JSD ↓	Novel % ↑	Unique % ↑	Valid % ↑
DeepCAD	65.5	1.29	1.67	87.4	89.3	46.1
SolidGen	71.0	1.08	1.31	99.1	96.2	60.3
BrepGen	73.9	1.04	1.28	99.8	99.7	62.9
Ours	73.7	1.09	1.40	99.9	99.1	100.*
BrepGen$_{ABC}$	57.9	1.35	3.69	99.7	99.4	48.2
Ours$_{ABC}$	56.8	1.35	3.80	99.9	99.5	100.*

Table 1. DeepCAD and ABC unconditional generation evaluation based on *Coverage* (COV), *Minimum Matching Distance* (MMD), *Jensen-Shannon Divergence* (JSD) and *Unique*, *Novel*, *Valid* ratio.

6.4. Evaluations

Following BrepGen [36], we randomly sampled 3,000 results from the generated data and 1,000 from the reference test set to evaluate our model with two sets of metrics: distribution metrics and CAD metrics. The distribution metrics are based on 2,000 points sampled from the solid surface:

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

CAD metrics compute the following scores:

- *Novel* percentage is the ratio of the generated data that does not appear in the training set.
- *Unique* percentage is the ratio of the data that appears only once in the generated results.
- *Valid* percentage is the ratio of the generated B-Rep data that are watertight solids.

6.5. Unconditional B-Rep Generation

We compare against DeepCAD [32], SolidGen [10] and BrepGen [36] for unconditional generation. For DeepCAD we further rebuild the generated sketch-and-extrude parameters into B-Rep solids.

Qualitative Evaluation. Figure 4 presents qualitative results on DeepCAD dataset. Our results demonstrate comparable performance to state-of-the-art methods. Additional results are provided in the supplementary materials. Figure 5 further showcases generated results trained on the ABC dataset. The key strength of our method is its unprecedented 100% success rate in producing faceted B-Rep models. To the best of our knowledge, no existing method comes close to achieving this level of reliability.

Quantitative Evaluation. Table 1 presents the quantitative results of our method against three other existing approaches. Our model is able to achieve comparable scores in COV, MMD, and JSD metrics. It is only slightly lower than the more complicated BrepGen method which has four different diffusion modules [36]. Another reason is that our approach represents B-Rep through volumetric distance functions. This introduces a disadvantage for distribution-based metrics, as these metrics rely on sampling points from the mesh surface, making our performance sensitive to resolution. However, our key strength lies in achieving a 100% validity rate. Validity is defined as generating watertight models with no topological errors—a criterion our method never fails. It is important to note that our success rate is calculated based on faceted B-Rep. Nonetheless, we argue that given a watertight and topologically valid structure, converting it into a standard B-Rep format is a straightforward and manageable process. Two B-Reps are considered identical if they have the same topological connections and if the 4-bit quantized positions of B-Rep vertices and edges (represented by the concatenation of their two endpoint positions) are also identical.
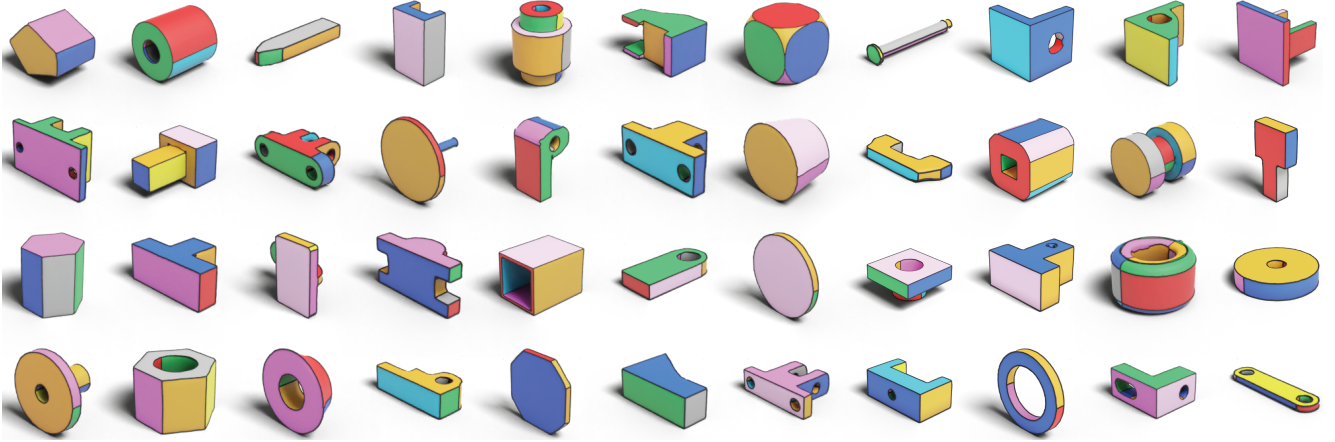
Figure 5. Unconditional generation results on the ABC dataset [11].

6.6. Reconstruction Evaluation

BR-DF representation is equivalent to a facated B-Rep model. We evaluate the numerical accuracy of this equivalence through the following steps: 1) Convert a faceted B-Rrep model into BR-DF via distance computation; 2) Reconstruct a faceted B-Rep model using MCT; and 3) Evaluate the reconstruction accuracy by comparing the reconstructed model against the original model. We use three metrics. The first one is Chamfer Distance (CD) based on 5,000 points uniformly sampled on the surfaces of both models. We compute CD metrics before and after the postprocessing step. The second one is Invalid Rate (IR) defined in DeepCAD [32], measuring the proportion of cases where B-Rep conversion failed due to invalid topology. The third one is the Same Topology Rate (STR), which measures the ratio of samples where the CAD topology (i.e., the numbers of points, curves, and faces together with their connectivity) match exactly. Note that STR is more "strict" than IR, because the reconstruction must lead to a valid B-Rep model and the topology must match exactly.

In Table 2, Ours$_{\text{raw}}$ shows the number for our reconstruction procedure described above. Ours$_{\text{latent}}$ is a version where we further add the 3D VQ-VAE encoder and decoder (subsection 5.3) to the BRDF representation. Lastly, we also evaluate DeepCAD system, where a transformer encodes a BRep-model into a 256 dimensional embedding and another transformer decodes an embedding back to a BRep-model. The table shows that Ours$_{\text{raw}}$ achieves impressive 0% IR and 100% STR. CD is at the scale of $6.6e$-4, which is negligible given that the model coordinates are normalized to a scale of 2.0 (subsection 5.1).

6.7. Conditional B-Rep Generation

Given partial bounding boxes, our model can generate a diverse set of CAD models with complete geometry and topology. Inspired by RePaint [18], we replace a subset of

tokens with the provided bounding boxes, which are diffused to the corresponding time step during the bounding box generation stage. To enhance result quality, we adopt the same resampling technique as RePaint. After that, the generation of UDFs and SDF follows the same process as unconditional generation. Figure 7 illustrates the diversity of results produced from different Gaussian noise inputs.

6.8. Constructive BR-DF Geometry

BR-DF is a set of 3D scalar fields. Constructive solid geometry operations—such as intersection and union—become applicable to B-rep models through their BR-DF representations. Figure 6 illustrates such examples. Animations of more examples are provided in the supplementary video.

Suppose we are to take intersection/union of multiple B-Rep models, each of which is a surface SDF and an arbitrary number of face UDFs. The surface SDF of a combined model follows the standard formula (i.e., max for intersection and min for union). Computing face UDFs of the combined model is not trivial. Since we only care the order of UDF values (i.e., minimum face or co-minimum curve/point), we use each UDF without modification.

6.9. Ablation Studies

To evaluate the effectiveness of the two-stage generation process, we conduct an ablation study by removing the bounding box generation module and using only the BR-DF Latent Diffusion Module, as described in Sec. 5.4. Since the number of faces forming a solid varies and is unknown during inference, we randomly duplicate faces during training until the maximum face count is reached. At inference, we apply non-maximum suppression (NMS) with a predefined threshold to eliminate duplicate face latents. Table 3 presents the results trained on the DeepCAD dataset. While the CAD metrics (Novel, Unique, Valid) remain similar, the distribution metrics (COV, MMD, JSD) show a significant
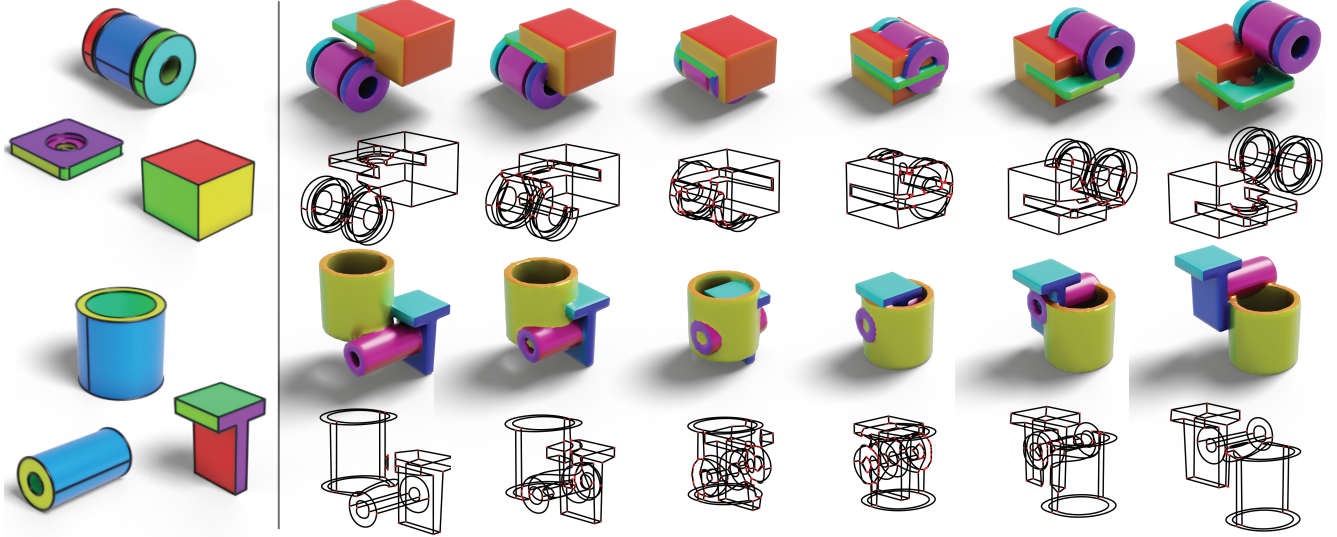
Figure 6. Constructive BR-DF Geometry. Multiple B-Rep models (left) are converted to their BR-DF representations and combined. The resulting BR-DF representation is then converted back to a valid B-Rep model using the MCT algorithm.
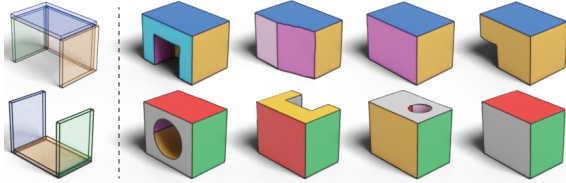

Figure 7. Given partial bounding boxes (left), our model generates a diverse set of CAD models with complete geometry/topology.
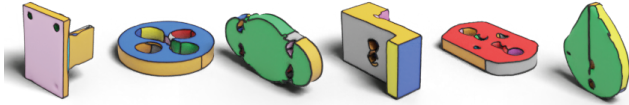

Figure 8. Failure examples occur at thin geometries and where irregular meshes are generated by the marching cubes.

Method	CD↓	IR↓	STR↑
DeepCAD	7.6e-4	3.21%	95.21%
Ours$_{raw}$	6.6e-4/6.9e-4	0.00%	100%
Ours$_{latent}$	7.5e-4/8.0e-4	0.00%	93.33%

Table 2. Reconstruction evaluation. For Ours$_{raw}$ and Ours$_{latent}$, we report results with and without post-processing for CD.

Bbox	COV % ↑	MMD ↓	JSD ↓	Novel % ↑	Unique % ↑	Valid % ↑
Pred	73.7	1.09	1.40	100.	99.1.	100.*
w/o	66.1	1.29	1.52	99.9	99.3.	100.*
GT	75.8	1.06	1.22	99.8	98.7	100.*

Table 3. Effectiveness of the two-stage generation process. Ablation using predicted bounding box (pred), no bounding box (w/o), and ground-truth bounding box (GT).

gap compared to the full two-stage model. This suggests that joint denoising for a variable number of latents is considerably more challenging for high-dimensional features (face latents) than for low-dimensional features (bounding box coordinates). Additionally, we report results using ground-truth bounding boxes.

6.10. Failure Cases

Figure 8 presents failure cases and show the limitations of our method. Note that these failures are not caused by the Marching Cubes and Triangles (MCT) algorithm but rather by defects in the BR-DF generative model. MCT is guaranteed to convert BR-DF with a 100% success rate. Improving the generative model will significantly mitigate these issues.

7. Conclusions

We introduced B-Rep Distance Functions, a novel geometric representation for CAD B-Rep models using volumetric distance functions. BR-DF encodes surface geometry as an SDF and represents B-Rep elements (vertices, curves, faces) and their connectivity as per-face UDFs. Our Marching Cubes and Triangles algorithm always converts any BR-DF into a watertight faceted B-Rep model. Compared to other B-Rep modeling methods, BR-DF is 3D scalar fields, making its generation immediately applicable to popular approaches such as diffusion models. Future work is to enhance generation quality and handle more complex models.

8. Acknowledgements

References

[1] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 1, 2, 3

[2] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018. 1, 2

[3] Olivier Faugeras and Renaud Keriven. *Variational principles, surface evolution, PDE's, level set methods and the stereo problem*. IEEE, 2002. 1, 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 5

[6] Jingyu Hu, Ka-Hei Hui, Zhengzhe Liu, Ruihui Li, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation, inversion, and manipulation. *ACM Trans. Graph.*, 43 (2), 2024. 2

[7] Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. Make-a-shape: a ten-million-scale 3d shape model. In *Forty-first International Conference on Machine Learning*, 2024. 1

[8] Pradeep Kumar Jayaraman, Aditya Sanghi, Joseph G. Lambourne, Karl D.D. Willis, Thomas Davies, Hooman Shayani, and Nigel Morris. Uv-net: Learning from boundary representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[9] Pradeep Kumar Jayaraman, Joseph G Lambourne, Nishkrit Desai, Karl DD Willis, Aditya Sanghi, and Nigel JW Morris. Solidgen: An autoregressive model for direct b-rep synthesis. *arXiv preprint arXiv:2203.13944*, 2022. 2

[10] Pradeep Kumar Jayaraman, Joseph G Lambourne, Nishkrit Desai, Karl DD Willis, Aditya Sanghi, and Nigel JW Morris. Solidgen: An autoregressive model for direct b-rep synthesis. *arXiv preprint arXiv:2203.13944*, 2022. 5, 6

[11] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019. 2, 5, 7

[12] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, et al. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144, 2000. 1, 3

[13] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025. 1

[14] Yilin Liu, Jiale Chen, Shanshan Pan, Daniel Cohen-Or, Hao Zhang, and Hui Huang. Split-and-Fit: Learning B-Reps via structure-aware Voronoi partitioning. *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, 43(4):108:1–108:13, 2024. 3

[15] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 2

[16] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 1

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 5

[18] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 7

[19] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 414–431. Springer, 2020. 2

[20] Ken Museth. Nanovdb: A gpu-friendly and portable vdb data structure for real-time rendering and simulation. In *ACM SIGGRAPH 2021 Talks*, New York, NY, USA, 2021. Association for Computing Machinery. 2

[21] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1, 2

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 4

[23] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[24] Ian Stroud. *Boundary representation modelling techniques*. Springer Science & Business Media, 2006. 1

[25] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15598–15607, 2021. 2

[26] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *Advances in neural information processing systems*, 2023. 4

[27] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 4

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[29] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015. 2

[30] Kevin J. Weiler. Topological structures for geometric modeling. Rensselaer Polytechnic Institute, 1986. 1

[31] Karl DD Willis, Pradeep Kumar Jayaraman, Joseph G Lambourne, Hang Chu, and Yewen Pu. Engineering sketch generation for computer-aided design. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2105–2114, 2021. 5

[32] Rundi Wu, Chang Xiao, and Changxi Zheng. Deepcad: A deep generative network for computer-aided design models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6772–6782, 2021. 2, 5, 6, 7

[33] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in neural information processing systems*, 32, 2019. 1, 2

[34] Xiang Xu, Karl DD Willis, Joseph G Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. Skexgen: Autoregressive generation of cad construction sequences with disentangled codebooks. In *International Conference on Machine Learning*, pages 24698–24724. PMLR, 2022. 2

[35] Xiang Xu, Pradeep Kumar Jayaraman, Joseph G Lambourne, Karl DD Willis, and Yasutaka Furukawa. Hierarchical neural coding for controllable cad model generation. In *International Conference on Machine Learning*, pages 38443–38461, 2023. 2

[36] Xiang Xu, Joseph Lambourne, Pradeep Jayaraman, Zhengqing Wang, Karl Willis, and Yasutaka Furukawa. Brepgen: A b-rep generative diffusion model with structured latent geometry. *ACM Transactions on Graphics (TOG)*, 43 (4):1–14, 2024. 2, 3, 4, 5, 6

[37] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024. 1

[38] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 4

B-Rep Distance Functions (BR-DF)
How to Represent a B-Rep Model by Volumetric Distance Functions?

Supplementary Material

8.1. Extracting boundary segments

The Marching Cubes and Triangles (MCT) algorithm traces all the mesh triangles and extracts the boundaries following 3-way rules (1) If the minimal faces of all the three vertices are the same, no point or curves are created; (2) If two vertices share the same minimal face, a B-Rep curve is created to cut the triangle via linear interpolation; (3) If all three minimal faces are different, a B-Rep vertex and three incident B-Rep curves are created. The equations for computing the B-Rep points on the mesh edges or inside the triangles are shown in Figure 9.

$$t \times d_1 + (1-t) \times d_2 = t \times d_1 + (1-t) \times d_2 \quad (1)$$

$$Loc_a = t \times Loc_{v_1} + (1-t) \times Loc_{v_2} \quad (2)$$

$$w_1, w_2, w_3 = 1/d_1, 1/d_2, 1/d_3 \quad (3)$$

$$Loc_b = \frac{w_1 Loc_{v_1} + w_2 Loc_{v_2} + w_3 Loc_{v_3}}{w_1 + w_2 + w_3} \quad (4)$$
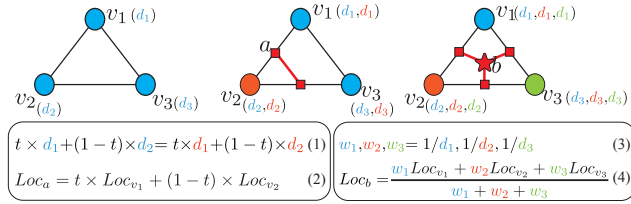
Figure 9. Brep Vertex/Curve Interpolation.

The boundary between every two neighbor faces is defined by a set of intersection points (marked as red cubes or stars in Figure 9) and their connections. The boundary is guaranteed to begin and end with 3-degree junctions (red stars).

8.2. Boundary Smoothing

For the boundaries of every two neighbor faces, we fit them to either a straight line (1-degree polynomial function) or a curve line (3-degree polynomial function). The start and end points are preserved by assigning them a high fitting weight. After fitting, the adjusted boundary curves are projected back onto the mesh to update the positions of the intersection points.

8.3. Establishing Faces and Boundaries

To properly define faces and boundaries, new triangles must be introduced within the mesh so that boundary curves become actual mesh edges. Figure 10 illustrates this process, where dark lines represent newly added edges, and different colors indicate separated faces.

8.4. Mesh Smoothing with Topology Preservation

The initial mesh generated by Marching Cubes often has rough surfaces and rounded boundary regions. To improve
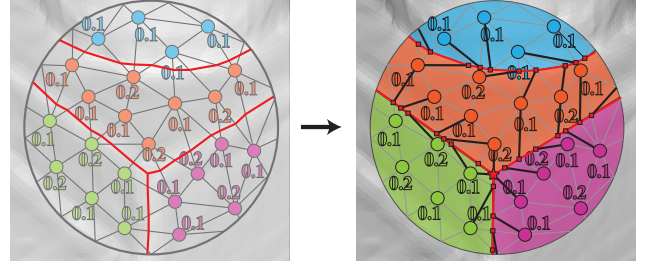

Figure 10. Illustration of face and boundary establishment.

quality, we apply multiple iterations of Laplacian smoothing and Quadric-based edge-collapse remeshing (with the PyMeshLab package). During this process, boundary vertices are averaged to maintain geometric consistency while preserving the mesh's overall structure.

9. Additional Results

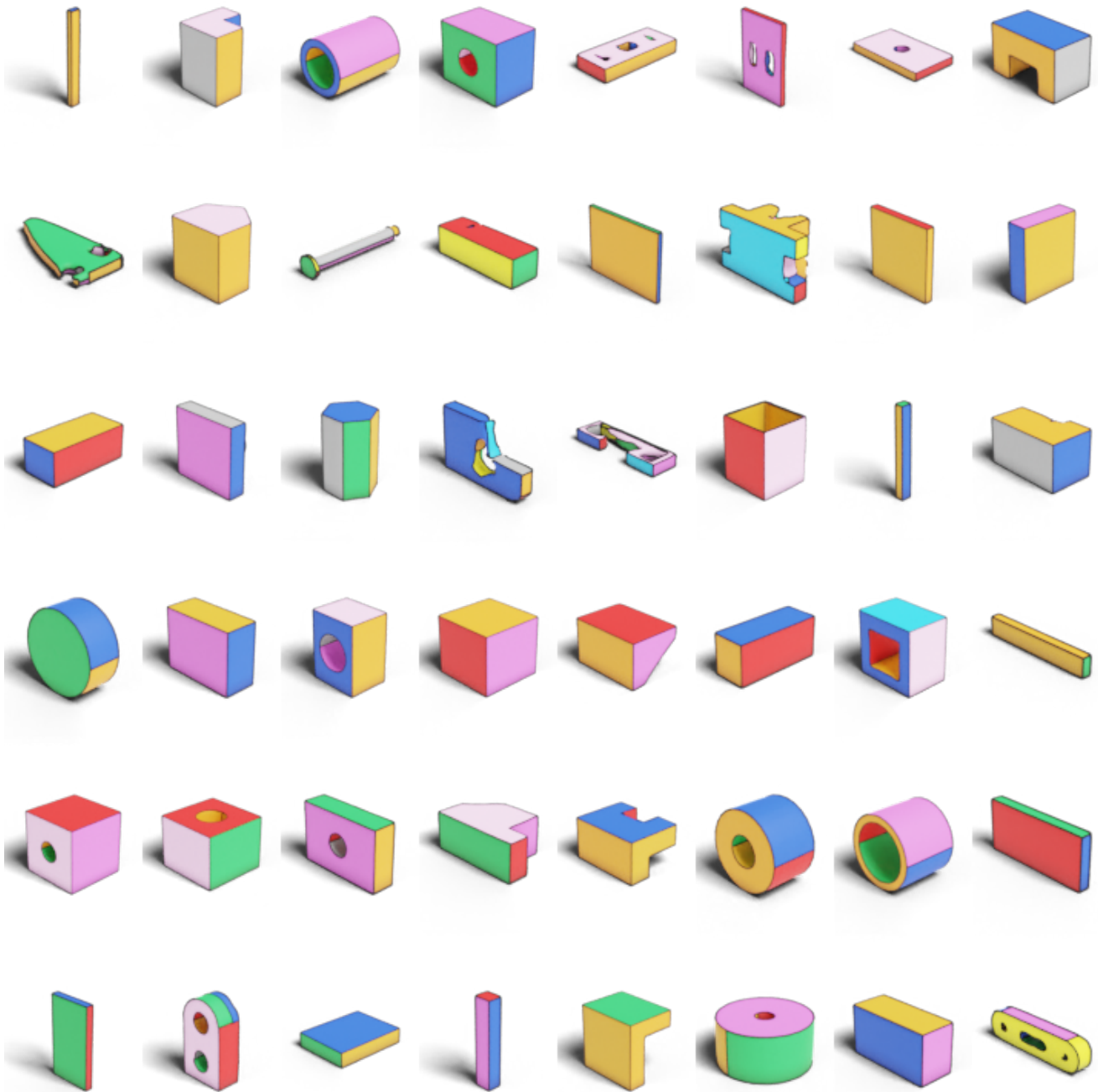Figures 11,12,13,14 shows additional results on DeepCAD dataset.
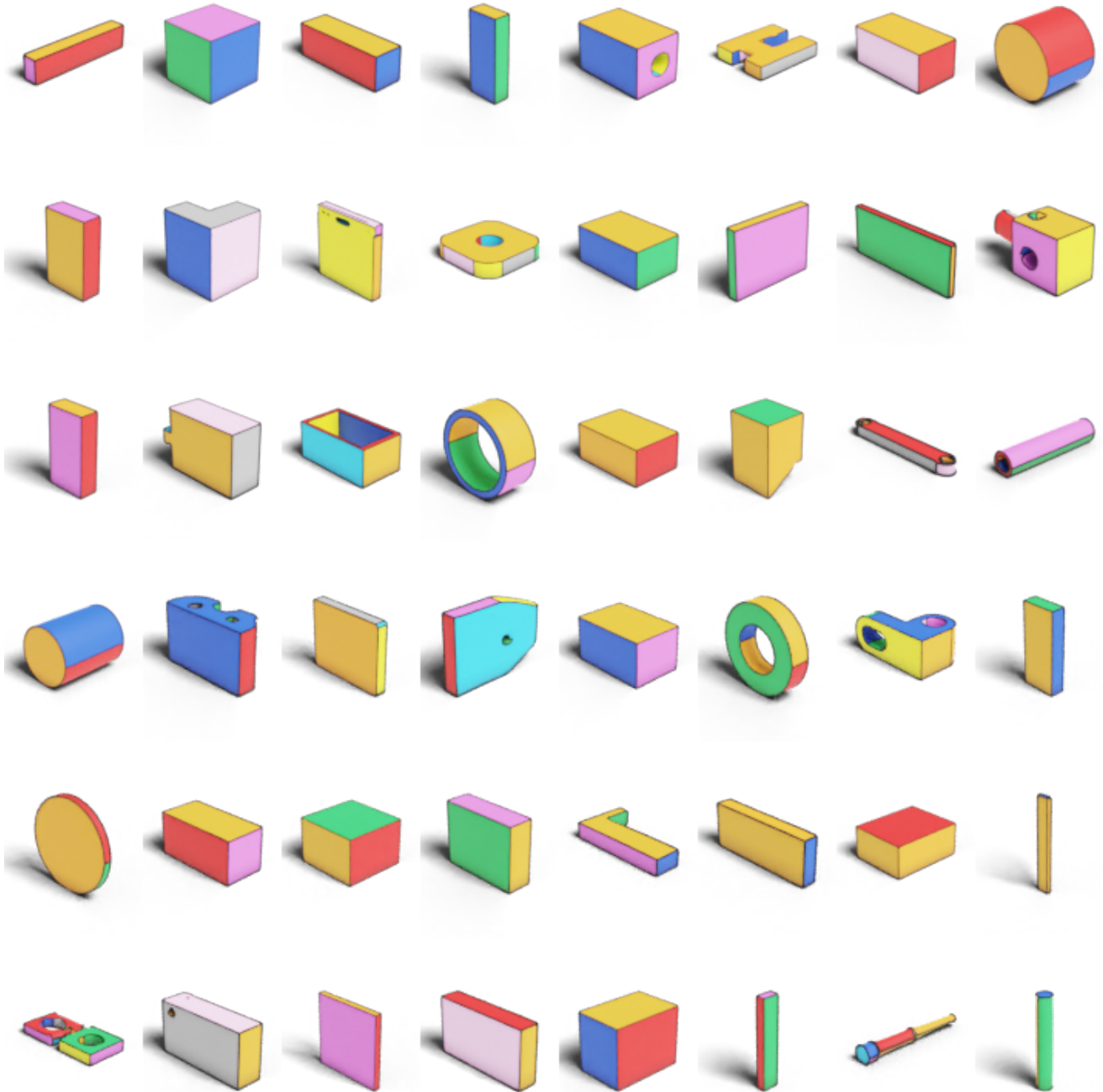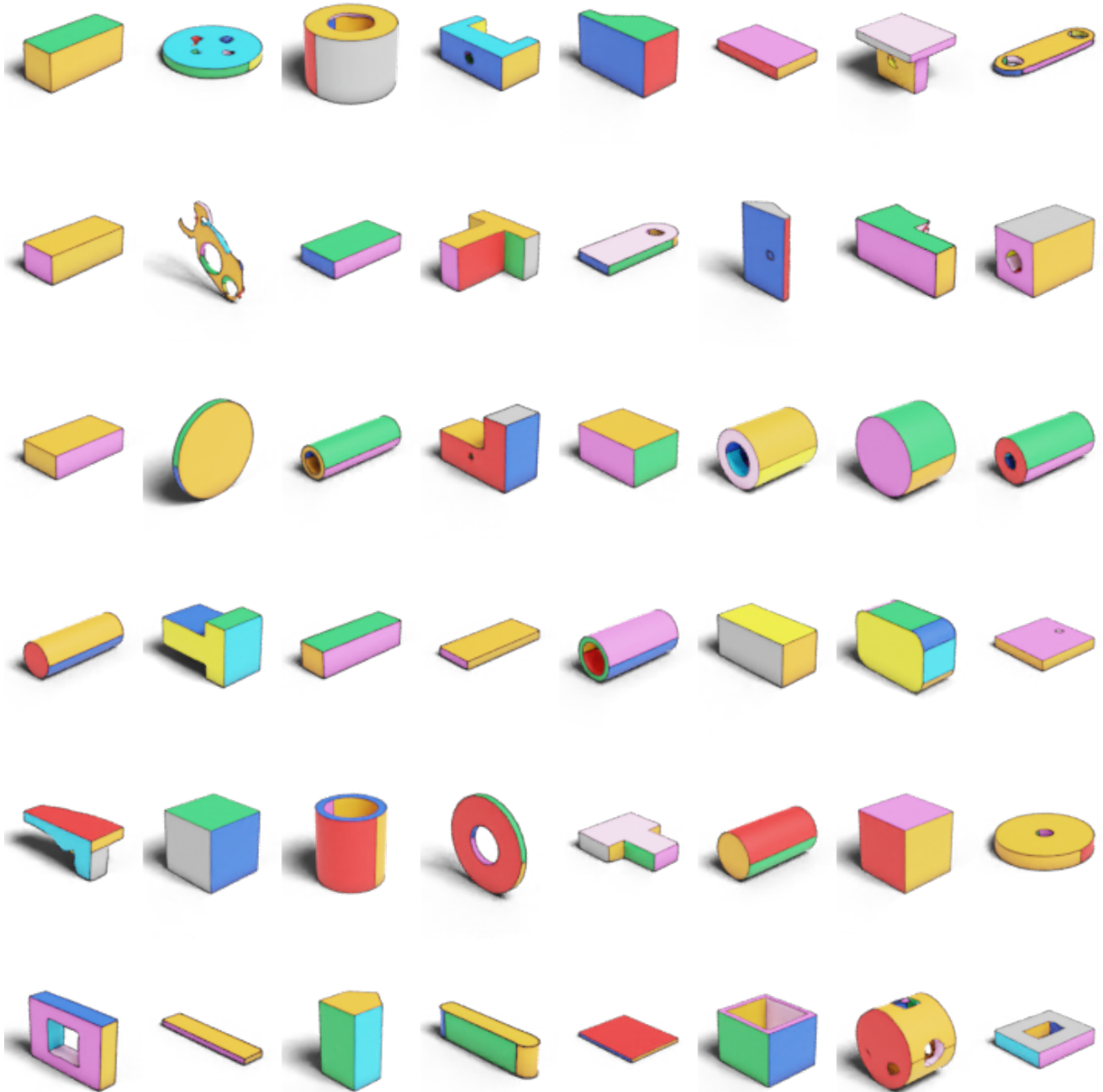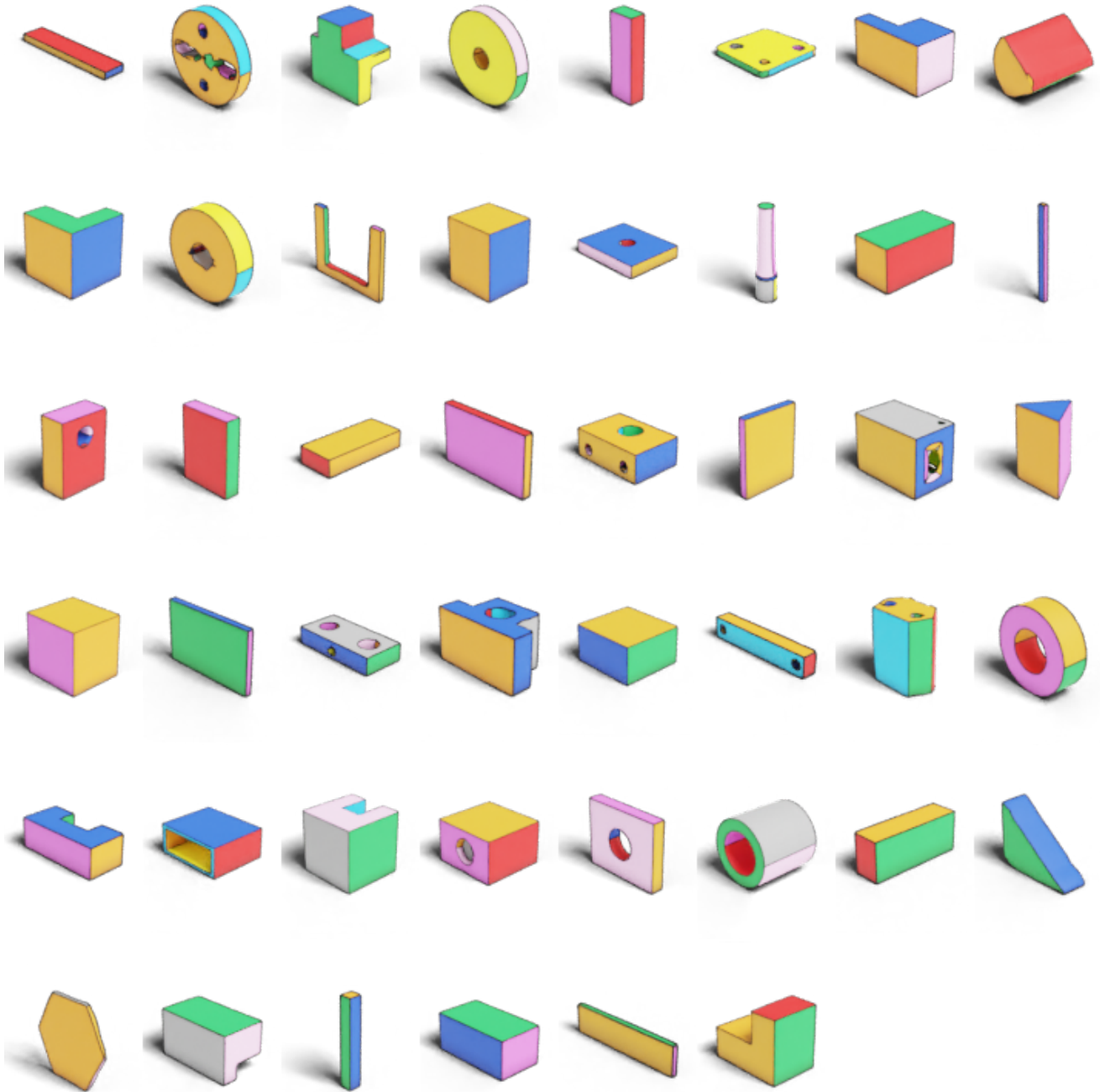
Figure 11. Additional Results.

Figure 12. Additional Results.

Figure 13. Additional Results.

Figure 14. Additional Results.