UNICAD: UNIFIED CAD GENERATION WITH CONTINUOUS CONDITIONAL GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating editable CAD models with precise geometry from single-view images is crucial for downstream design and manufacturing applications. However, existing methods are either limited to unconditional generation or rely on restrictive sketch-and-extrude pipelines, limiting user control and generative diversity. We propose UniCAD, a unified generative model that bridges 2D images and structured 3D CAD representations. It models topology with a connectivity-guided autoregressive Transformer decoder and geometry with a Transformer-based diffusion model. In addition, UniCAD supports controllable CAD generation from a single image, enabling smooth interpolation and precise control over conditioning strength. Extensive experiments show that our model supports conditional and unconditional generation within a unified framework, delivering high-quality results.

1 INTRODUCTION

In recent years, the field of 3D model generation has experienced significant growth, with generative works emerging across various representations such as mesh (Gao et al., 2022; Wu et al., 2024; Siddiqui et al., 2024; Chen et al., 2025b), NeRF (Schwarz et al., 2020; Poole et al., 2022; Chen et al., 2023; Deng et al., 2023), and point cloud (Luo & Hu, 2021; Huang et al., 2024; Lee & Benes, 2025). These representations have played a crucial role in downstream applications, including animation, gaming, and virtual reality. However, despite their versatility, these 3D models often lack clear topological structures and precise geometric parameterizations. Consequently, they struggle to satisfy the rigorous tolerances required for high-precision tasks like industrial design and manufacturing.

This limitation has led researchers to explore boundary representation (BRep) (Weiler, 1986), the standard CAD format that defines objects via exact boundaries. However, automatically generating BRep models remains challenging, as it requires modeling the complex joint distribution of discrete topology and continuous geometry. Recent works such as BrepGen (Xu et al., 2024b) and DTG-BrepGen (Li et al., 2025a) address this by transforming BRep into neural-friendly representations. For example, BrepGen models shapes as hierarchical latent structures, while DTGBrepGen adopts a topology-geometry decoupled formulation. Despite these advances, existing methods are largely limited to unconditional generation and lack mechanisms for incorporating visual guidance, restricting their applicability in user-directed design scenarios.

Conversely, works on image-conditioned CAD generation, such as GenCAD (Alam & Ahmed, 2024), rely on intermediate representations based on sketch-and-extrude pipelines (Wu et al., 2021). Such representations restrict the generative modeling space to a limited class of CAD geometries constructed exclusively from basic extrusion operations. Furthermore, these methods are often trained on relatively restricted datasets, fundamentally limiting their ability to generalize to the diverse and unstructured geometries encountered in real-world scenarios.

To bridge this modality gap, we propose UniCAD, a generative framework for direct CAD generation with single-view image conditioning. By treating conditional CAD generation as a joint distribution modeling problem over discrete graphs and continuous parametric spaces, we factorize this intractable distribution into a two-stage process: vision-guided discrete topology modeling followed by topology-conditioned continuous geometry diffusion.

First, we formulate topology generation as an autoregressive process. Instead of treating topology as a flat sequence, we introduce a connectivity-guided prior that prioritizes high-degree core faces,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

enabling more effective modeling of global face–face relationships with an autoregressive Transformer decoder. Local edge–vertex connectivity is then modeled using a pointer network (Vinyals et al., 2015).

Second, to generate continuous geometric attributes, we deploy a structure-aware Transformer-based diffusion model (Peebles & Xie, 2023; Ho et al., 2020; Vaswani et al., 2017). This model operates in a progressive coarse-to-fine cascade, predicting spatial bounding boxes before refining precise curve and surface parameters, strictly conditioned on visual features and the previously inferred topology.

Throughout both stages, visual semantics extracted by a Vision Transformer (Dosovitskiy et al., 2020) are consistently injected via cross-attention. Crucially, by integrating condition dropout during training and classifier-free guidance (CFG) (Ho & Salimans, 2022) at inference, we establish a unified probabilistic space. This formulation not only provides precise control over the conditioning strength but also enables smooth traversal across the entire design intent spectrum, from strict image-based reconstruction to diverse unconditional exploration.

In summary, the contributions of this paper are as follows:

- We propose UniCAD, a generative framework designed to address the modality gap between unstructured 2D visual inputs and highly structured 3D CAD manifolds. By directly grounding discrete topology and continuous geometric parameters in visual semantics, our method enables parametric CAD generation without relying on restricted proxy representations.
- We formulate a unified probabilistic generative space that integrates both conditional and unconditional distributions within a single model. Our formulation enables smooth interpolation and precise traversal across the generative spectrum, ranging from high-fidelity reconstruction to diverse unconditional exploration.
- We model the complex joint CAD distribution with a two-stage decoupled architecture: a connectivity-guided topological prior for autoregressive discrete connectivity modeling, followed by topology-conditioned diffusion for progressively modeling continuous geometric attributes with structural awareness.
- Extensive experiments demonstrate that our method achieves competitive performance in both conditional and unconditional settings, improving CD for conditional generation and JSD/MMD for unconditional generation.

2 Related work

Generative Models: Autoregressive and diffusion models represent two dominant generative paradigms. Autoregressive models (Van Den Oord et al., 2016; Radford et al., 2019; Esser et al., 2021) capture dependencies through explicit causal structures by sequentially predicting discretized tokens. Conversely, diffusion models (Ho et al., 2020; Song et al., 2020b;a) approximate complex distributions via a reverse denoising process, offering stable optimization and broad mode coverage. Integrating Transformer backbones (Vaswani et al., 2017; Peebles & Xie, 2023) into diffusion models has proven highly effective for modeling continuous data with global structural consistency. For conditional generation, CFG (Ho & Salimans, 2022) jointly models conditional and unconditional distributions via condition dropout during training. This formulation enables continuous control over the fidelity-diversity trade-off during inference, making it widely applicable across multimodal tasks.

Unconditional CAD Generation: Existing CAD generative methods generally follow two paradigms. The first models CAD objects autoregressively as sequences of construction operations, such as sketch-and-extrude commands (Wu et al., 2021; Xu et al., 2022; 2023; Guo et al., 2025). While effective at capturing sequential dependencies, these methods require costly sequence annotations and struggle to represent complex geometries beyond predefined operation types.

The second paradigm directly generates BRep models. BrepGen (Xu et al., 2024b) employs diffusion to generate hierarchical latent geometry, implicitly recovering topology via node duplication and merging. DTGBrepGen (Li et al., 2025a) improves structural validity by explicitly decoupling topology and geometry generation. BrepDiff (Lee et al., 2025) introduces a single-stage diffusion model for BRep generation, while BrepGPT (Li et al., 2025b) and AutoBrep (Xu et al., 2025) model unified BRep tokens autoregressively. Despite these advances, a fundamental representation

bottleneck remains: it is inherently difficult to design encodings that simultaneously capture discrete topological relationships and continuous geometric attributes.

Image-Conditioned CAD Generation: Generating CAD models from a single-view image holds significant practical value for design automation. Recent works map visual inputs to CAD models through direct BRep generation, including diffusion-based methods such as HoLa (Liu et al., 2025) and masked autoregressive methods such as CMT (Wu et al., 2025), intermediate structured visual geometry (You et al., 2025), or image-conditioned CAD program generation (Alam & Ahmed, 2024; Xu et al., 2024a; Chen et al., 2025a; Wang et al., 2025).

However, existing methods face two fundamental challenges. First, mitigating single-view information loss by introducing intermediate modalities, such as multi-view images or point clouds, can accumulate errors across transformations and often violates the high precision required for downstream BRep reconstruction. Second, current approaches primarily focus on faithful reconstruction from input conditions, lacking mechanisms for flexible, user-controllable generation.

To address this, our framework directly infers BRep from a single image. By incorporating CFG during training and inference, we enable continuous control over conditioning strength. This unified formulation seamlessly supports both image-conditioned and unconditional modeling, facilitating flexible CAD generation beyond strict reconstruction.

3 PRELIMINARIES

Following DTGBrepGen (Li et al., 2025a), we represent each BRep model by separating its topological structure and geometric attributes. Formally, a CAD model is represented as

$$\mathcal{B}_i = (\mathcal{T}_i, \mathcal{G}_i), \tag{1}$$

where $\mathcal{T}_i$ denotes the topology and $\mathcal{G}_i$ denotes the geometry.

Topology Representation: Global topology is represented by a face adjacency matrix: $\mathbf{FeF}_i \in \mathbb{Z}_{\geq 0}^{N_f \times N_f}$, where $\mathbf{FeF}_i[p, q]$ denotes the number of shared edges between faces p and q. Since $\mathbf{FeF}_i$ is symmetric, its strictly upper-triangular part is flattened into a sequence:

$$\mathbf{FeF}_i^{\text{seq}} = (\mathbf{FeF}_i[1,2], \mathbf{FeF}_i[1,3], \ldots, \mathbf{FeF}_i[N_f - 1, N_f]) \,. \tag{2}$$

However, $\mathbf{FeF}_i$ alone does not fully specify the topology, lacking information about edge ordering and intra-face loop structures. To capture this local connectivity, an edge connectivity sequence is further introduced:

$$\mathbf{EV}_i^{\text{seq}} := \left(e_{j_1}^i, e_{j_2}^i, \ldots, e_{j_{K_i}}^i\right), \tag{3}$$

where K_i denotes the sequence length for sample $\mathcal{B}_i$, and j_k is the k-th selected edge index. Each $e_{j_k}^i$ represents either an edge in $\mathcal{B}_i$ or a special token: $e_{j_k}^i \in \{e_1, \ldots, e_{N_e},\ e_{\text{loop}},\ e_{\text{face}}\}$. Here, e_{loop} and e_{face} indicate the end of a loop and a face, respectively. The sequence is constructed by traversing faces and serializing the ordered edge connectivity within each face. Together, $\mathbf{FeF}_i$ and $\mathbf{EV}_i^{\text{seq}}$ capture global face adjacency and local edge connectivity, forming the topology representation used in this work.

Geometry Representation: For geometry representation, we follow the staged parameterization of DTGBrepGen (Li et al., 2025a), decomposing geometric attributes into four components: face-level bounding boxes, vertex geometry, edge geometry, and face geometry. This organization aligns with the coarse-to-fine generation process, enabling different levels of geometric information to be modeled by dedicated modules.

We first represent coarse spatial layouts using face-level bounding boxes $\mathbf{B}_i^f \in \mathbb{R}^{N_f \times 6}$, where the bounding box of the p-th face is defined as $\mathbf{b}_{i,p}^f = (x_p^{\min}, y_p^{\min}, z_p^{\min}, x_p^{\max}, y_p^{\max}, z_p^{\max})$. Vertex geometry is represented as $\mathbf{V}_i \in \mathbb{R}^{N_v \times 3}$, where the coordinate of the k-th vertex is given by $\mathbf{v}_{i,k} = (x_k, y_k, z_k)$. Each edge is modeled as a cubic B-spline curve with four control points, resulting in $\mathbf{E}_i \in \mathbb{R}^{N_e \times 4 \times 3}$: $\mathbf{e}_{i,m} = [\mathbf{p}_{i,m,1}, \mathbf{p}_{i,m,2}, \mathbf{p}_{i,m,3}, \mathbf{p}_{i,m,4}] \in \mathbb{R}^{4 \times 3}$, where $\mathbf{p}_{i,m,r} \in \mathbb{R}^3$ denotes the

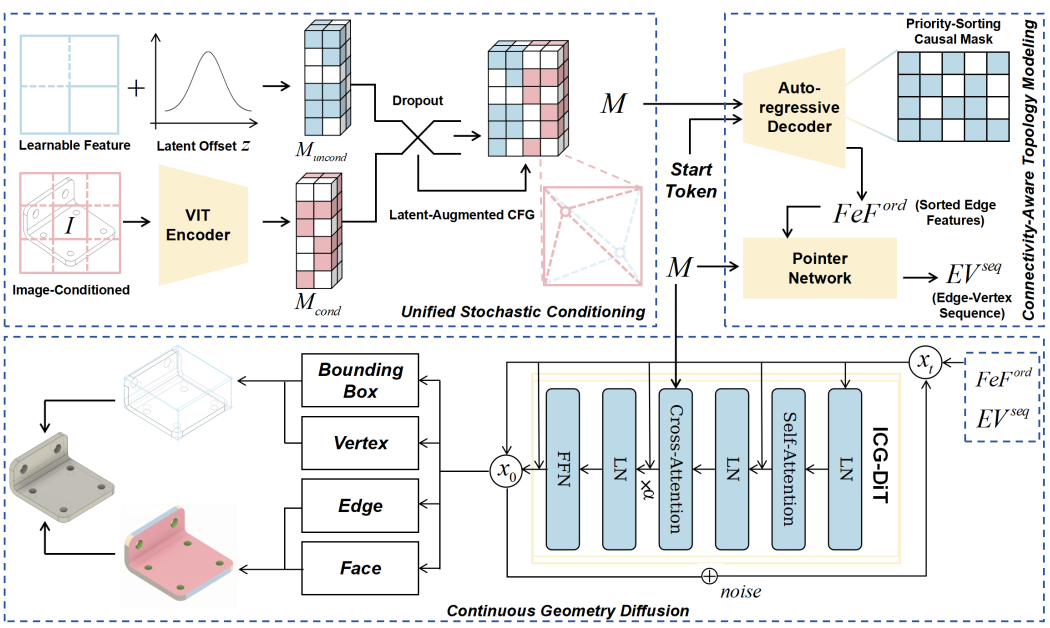

Figure 1: Overview of the proposed UniCAD framework. Given an input image, the model first generates topology and then progressively generates geometry in a coarse-to-fine manner. The stochastic latent offset $\mathbf{z}$ is applied only in the **FeF** generation stage.

r-th control point of the m-th edge. Similarly, each face is modeled as a bicubic B-spline surface with a 4×4 control grid, resulting in $\mathbf{F}_i \in \mathbb{R}^{N_f \times 16 \times 3}$: $\mathbf{f}_{i,p} = [\mathbf{q}_{i,p,1,1}, \mathbf{q}_{i,p,1,2}, \ldots, \mathbf{q}_{i,p,4,4}] \in \mathbb{R}^{16\times 3}$, where $\mathbf{q}_{i,p,u,v} \in \mathbb{R}^3$ denotes the control point at grid position (u, v) of the p-th face.

4 METHOD

Based on this topology–geometry decomposition, UniCAD generates a BRep model in two stages: topology generation and geometry generation, with the image condition incorporated into both stages. Given an input image $\mathbf{I}_i$, the model first predicts the topology $\mathcal{T}_i$, and then progressively generates the geometry $\mathcal{G}_i$ conditioned on both $\mathcal{T}_i$ and $\mathbf{I}_i$, yielding the final BRep model $\mathcal{B}_i = (\mathcal{T}_i, \mathcal{G}_i)$. The overall pipeline is illustrated in Fig. 1.

From a probabilistic perspective, the conditional generation process is formulated as

$$p(\mathcal{B}_i \mid \mathbf{I}_i) = p(\mathbf{FeF}_i \mid \mathbf{I}_i)\, p(\mathbf{EV}_i^{\text{seq}} \mid \mathbf{FeF}_i, \mathbf{I}_i) \cdot p(\mathbf{B}_i^f \mid \mathcal{T}_i, \mathbf{I}_i)\, p(\mathbf{V}_i \mid \mathbf{B}_i^f, \mathcal{T}_i, \mathbf{I}_i)\, p(\mathbf{E}_i \mid \mathbf{V}_i, \mathbf{B}_i^f, \mathcal{T}_i, \mathbf{I}_i)\, p(\mathbf{F}_i \mid \mathbf{E}_i, \mathbf{V}_i, \mathbf{B}_i^f, \mathcal{T}_i, \mathbf{I}_i). \tag{4}$$

This decomposition corresponds to a coarse-to-fine generation process: the model first predicts global face adjacency, then recovers local edge connectivity, and finally generates geometry progressively from bounding boxes to vertices, edges, and faces.

Our method consists of three key components. First, we design a unified stochastic scheme with conditional encoding and dropout (Srivastava et al., 2014), allowing image features to be shared across both topology and geometry stages. Second, in the topology stage, we model $\mathbf{FeF}_i$ using a connectivity-guided autoregressive Transformer decoder with importance-aware reordering to better capture critical adjacency patterns. Finally, in the geometry stage, we propose an Image-Conditioned Geometric DiT (ICG-DiT) module to progressively generate continuous geometry while jointly attending to image features, topology constraints, and previously generated geometric context.

4.1 Unified Stochastic Conditioning

To consistently incorporate image conditions into both topology and geometry generation, we introduce a unified stochastic conditioning scheme. Given an input image $\mathbf{I}_i$, a pretrained ViT (Dosovitskiy et al., 2020) extracts semantic features, which are projected into the model embedding space to form the condition representation $\mathbf{M}_{\text{cond}}$.

To unify conditional and unconditional modeling, we introduce learnable memory tokens for the unconditional branch. Rather than using a fixed unconditional embedding, we augment these tokens with a stochastic offset generated from random variables $\mathbf{z}$ in the $\mathbf{FeF}$ generation stage, enabling richer variation during generation.

During training, we adopt condition dropout to jointly learn conditional and unconditional generation within a single model:

$$\mathbf{M} = \begin{cases} \mathbf{M}_{\text{uncond}}, & \text{with probability } p_{\text{drop}}, \\ \mathbf{M}_{\text{cond}}, & \text{otherwise.} \end{cases} \tag{5}$$

This strategy enables the model to approximate conditional and unconditional distributions.

At inference time, we employ CFG (Ho & Salimans, 2022) based on the latent augmentation strategy to control the conditioning strength. Specifically, the final prediction is obtained by combining conditional and unconditional outputs:

$$\Phi(\mathbf{x}) = (1 - s)\Phi(\mathbf{x}; \mathbf{M}_{\text{uncond}}) + s\Phi(\mathbf{x}; \mathbf{M}_{\text{cond}}), \tag{6}$$

where s is the guidance scale. It controls the trade-off between image fidelity and generative diversity, with $s = 0$ corresponding to unconditional generation and larger values enforcing stronger alignment with the input image.

The resulting conditioning representation $\mathbf{M}$ is shared across all generation stages. For simplicity, we omit the explicit dependence on $\mathbf{I}$ in the following formulations.

4.2 Connectivity-Guided Topology Modeling

Given the conditioning representation $\mathbf{M}$, the topology stage generates $\mathcal{T} = \{\mathbf{FeF}, \mathbf{EV}^{\text{seq}}\}$ in two steps. It first predicts the face adjacency matrix $\mathbf{FeF}$, and then generates the edge connectivity sequence $\mathbf{EV}^{\text{seq}}$ conditioned on $\mathbf{FeF}$.

The strictly upper-triangular entries of $\mathbf{FeF}$ naturally form a discrete sequence. We therefore model its conditional distribution autoregressively, with each element predicted from preceding elements. To obtain a stable and structure-aware serialization order, we introduce a priority-based sorting mechanism. Faces are first ranked by topological importance, measured by their adjacency degree, with faces connected to more neighbors being generated earlier. For faces with identical adjacency degrees, a secondary ordering based on geometric scale is applied. This produces a reordered sequence $\mathbf{FeF}^{\text{ord}}$, prioritizing structurally central faces and improving autoregressive modeling stability. At inference, the model samples directly in this priority-ordered sequence space.

Under this formulation, we model the conditional distribution using an autoregressive Transformer-based decoder:

$$p_\theta(\mathbf{FeF}^{\text{ord}} \mid \mathbf{M}) = \prod_{l=1}^{L} p_\theta(y_l \mid y_{<l}, \mathbf{M}), \tag{7}$$

where y_l denotes the l-th element in the reordered sequence and $\mathbf{M}$ is the conditioning representation defined in Sec. 4.1. In implementation, sequence generation follows standard autoregressive decoding with causal masking, ensuring each position depends solely on previous tokens. A start token is used to initialize decoding. The model outputs a categorical distribution at each step, trained by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{FeF}} = -\sum_{l=1}^{L} \log p_\theta(y_l \mid y_{<l}, \mathbf{M}). \tag{8}$$

Given $\mathbf{FeF}$, we further generate the edge connectivity sequence $\mathbf{EV}^{\text{seq}}$. This stage follows the serialization strategy and pointer network introduced in DTGBrepGen (Li et al., 2025a). We inject

the conditioning representation $\mathbf{M}$ as memory into the pointer network, replacing the original key–value features. This allows the model to incorporate conditioning information when predicting edge connections. Formally, the process is defined as:

$$p_\theta(\mathbf{EV}^{\text{seq}} \mid \mathbf{FeF}, \mathbf{M}) = \prod_t p_\theta(e_t \mid e_{<t}, \mathbf{FeF}, \mathbf{M}), \tag{9}$$

where e_t denotes the generated topology token at step t. The model is trained using autoregressive maximum likelihood, with the loss function

$$\mathcal{L}_{\text{EV}} = -\sum_t \log p_\theta(e_t \mid e_{<t}, \mathbf{FeF}, \mathbf{M}). \tag{10}$$

4.3 Continuous Geometry Diffusion

Given the topology $\mathcal{T} = \{\mathbf{FeF}, \mathbf{EV}^{\text{seq}}\}$, we generate the geometric attributes $\mathcal{G} = \{\mathbf{B}^f, \mathbf{V}, \mathbf{E}, \mathbf{F}\}$ of the BRep model. Geometry is generated in a staged coarse-to-fine manner, progressively predicting face bounding boxes, vertices, edges, and faces. At each stage, our ICG-DiT denoiser models the corresponding geometric variable. This design enables the conditioning representation $\mathbf{M}$ to be consistently incorporated into the generation process, allowing each stage to selectively attend to different image features via cross-attention. As a result, the model aligns image conditions with geometric structures across different levels.

First, we factorize conditional geometry generation as

$$p(\mathcal{G} \mid \mathcal{T}, \mathbf{M}) = p(\mathbf{B}^f \mid \mathcal{T}, \mathbf{M})\, p(\mathbf{V} \mid \mathbf{B}^f, \mathcal{T}, \mathbf{M})\, p(\mathbf{E} \mid \mathbf{V}, \mathbf{B}^f, \mathcal{T}, \mathbf{M})\, p(\mathbf{F} \mid \mathbf{E}, \mathbf{V}, \mathbf{B}^f, \mathcal{T}, \mathbf{M}). \tag{11}$$

This factorization implies that each stage depends on the image condition, the generated topology, and preceding geometric variables, resulting in a progressive refinement process from coarse spatial layouts to detailed surface geometry. For clarity, we describe the design of the conditional geometric denoiser using face geometry $\mathbf{F}$ as an example; the same formulation applies to $\mathbf{B}^f$, $\mathbf{V}$, and $\mathbf{E}$, differing only in input variables and output dimensionality.

Second, we construct geometry-aware tokens for the ICG-DiT denoiser by aggregating local context from $\mathbf{FeF}$ and $\mathbf{EV}^{\text{seq}}$. Let $\mathbf{x}_0$ denote the ground-truth geometric variable and $\mathbf{x}_t$ its noisy version. For face geometry, $\mathbf{x}_t$ is the perturbed control point representation. For each face p, the initial token is

$$\mathbf{h}_p^{(0)} = E_x(\mathbf{x}_{t,p}) + E_b(\mathbf{b}_p^f) + \text{Pool}_v\big(E_v\big(\mathbf{V}_{\mathcal{N}_V(p)}\big)\big) + \text{Pool}_e\big(E_e\big(\mathbf{E}_{\mathcal{N}_E(p)}\big)\big) + E_t(t), \tag{12}$$

where $\mathcal{N}_V(p)$ and $\mathcal{N}_E(p)$ denote the vertices and edges associated with face p, respectively, and Pool_v and Pool_e denote the corresponding mask-aware aggregation operations.

The denoiser adopts a Transformer backbone that sequentially applies self-attention, cross-attention, and feed-forward layers. Self-attention captures dependencies among geometry tokens, while cross-attention injects the conditioning representation $\mathbf{M}$ as memory. We scale the cross-attention branch by a constant factor α to balance geometric context and image-conditioned information:

$$\mathbf{H}^{(l+1)} = \text{FFN}\Big(\text{SelfAttn}(\mathbf{H}^{(l)}) + \alpha\ \text{CrossAttn}\Big(\text{SelfAttn}(\mathbf{H}^{(l)}), \mathbf{M}\Big)\Big), \tag{13}$$

where $\mathbf{H}^{(l)}$ denotes the geometry tokens at layer l. Residual connections and layer normalization are omitted for clarity.

Third, we train each geometric stage with the same diffusion objective. At each stage, let $\mathbf{x} \in \mathbf{B}^f, \mathbf{V}, \mathbf{E}, \mathbf{F}$ denote the geometric variable generated at the current stage. Accordingly, $\mathbf{x}_0$ denotes the ground-truth geometric variable, and $\mathbf{x}_t$ denotes its noisy version at diffusion timestep t. The denoiser predicts the noise conditioned on the topology $\mathcal{T}$, the conditioning representation $\mathbf{M}$, and previously generated geometric variables $\mathcal{G}_<$.The training objective is

$$\mathcal{L}_{\text{geom}} = \mathbb{E}_{\mathbf{x}_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathcal{T}, \mathbf{M}, \mathcal{G}_<)\|_2^2\right], \tag{14}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Table 1: Quantitative comparison on the DeepCAD and ABC datasets. Img2CAD is evaluated under its original dataset and protocol due to its specific input requirements.

Dataset	Model	Mean CD ↓	Validity ↑
DeepCAD	GenCAD	0.195	**0.940**
DeepCAD	CADDreamer	0.166	0.741
DeepCAD	Ours	**0.072**	0.877
ABC	Ours	0.076	0.508
Original	Img2CAD	0.116	0.953

5 EXPERIMENTS

5.1 EXPERIMENT SETUP

Datasets. We evaluate our method on DeepCAD (Wu et al., 2021) and ABC (Koch et al., 2019), following the preprocessing and split protocol of BrepGen (Xu et al., 2024b). For DeepCAD, we retain samples with fewer than 30 faces and at most 20 edges per face, resulting in 67,909/5,489/5,152 train/val/test samples. For ABC, we use samples with fewer than 50 faces and at most 30 edges per face, yielding 195,604/16,012/15,915 train/val/test splits.

For image-conditioned generation, each CAD model is rendered into four views, and one view is randomly sampled at each training epoch. We follow each baseline's rendering protocol when available.

Metric. For image-conditioned generation, we evaluate on the full test set using Chamfer Distance (CD) and Validity. CD measures geometric consistency via point cloud distance, while Validity measures whether the generated BRep can be successfully constructed and passes structural and watertightness checks using OCCT through pythonOCC. We follow Img2CAD (You et al., 2025) for CD and DTGBrepGen (Li et al., 2025a) for Validity.

For unconditional generation, we report MMD, COV, JSD, and Validity to evaluate generation fidelity, diversity, and distribution similarity. Validity follows DTGBrepGen, while the other metrics follow BrepGen. We follow the BrepGen protocol, generating 3,000 samples per run and averaging over 10 trials against 1,000 randomly sampled test shapes.

Implementation Details. During training, we use a learning rate of 5×10^{-4} and a batch size of 1024. We apply condition dropout with probability $p_{\text{drop}} = 0.25$ at each generation stage, including both topology and geometry generation. We use $\alpha = 0.5$ for cross-attention scaling in all geometry denoisers. We employ early stopping based on validation loss and terminate training when the validation performance no longer improves.

We adopt confidence-aware sampling for topology generation. When the maximum probability exceeds 0.7, we apply greedy decoding; otherwise, we use top-p sampling with $p = 0.9$ for $\mathbf{FeF}$ generation and top-k sampling with $k = 3$ for $\mathbf{EV}^{\text{seq}}$ generation.

5.2 RESULTS

Image-Conditioned Generation. We evaluate our method on the test splits of DeepCAD and ABC defined in preprocessing. For quantitative comparison, we evaluate GenCAD (Alam & Ahmed, 2024) and CADDreamer (Li et al., 2025c) on the same DeepCAD test set. For Img2CAD, due to its specific input format requirements, we follow its original dataset and evaluation settings for reproduction. In addition, we provide qualitative visual comparisons with HoLa (Liu et al., 2025).

Fig. 2 presents qualitative results under single-view image conditioning with conditioning strength set to 1, demonstrating strong alignment with the input image. Table 1 reports the quantitative results. Our method achieves significantly lower CD while maintaining competitive validity, indicating accurate geometric alignment and high-quality BRep generation.

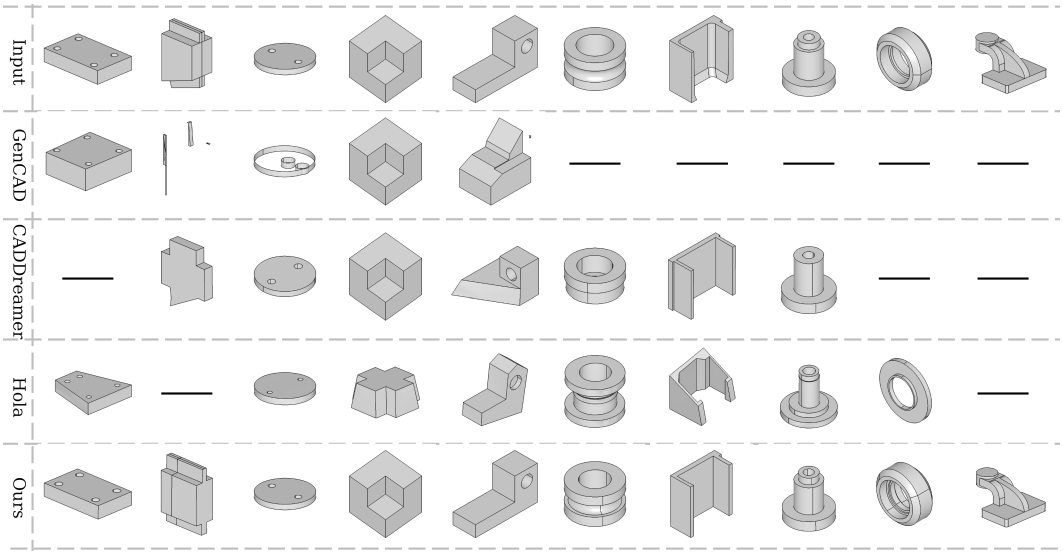

Figure 2: Qualitative comparison of single-view image-conditioned BRep generation with conditioning strength set to 1. The left five columns show results on DeepCAD, and the right five columns show results on ABC. Each row presents the input image and the generated BRep models from different methods; "-" denotes generation failure.

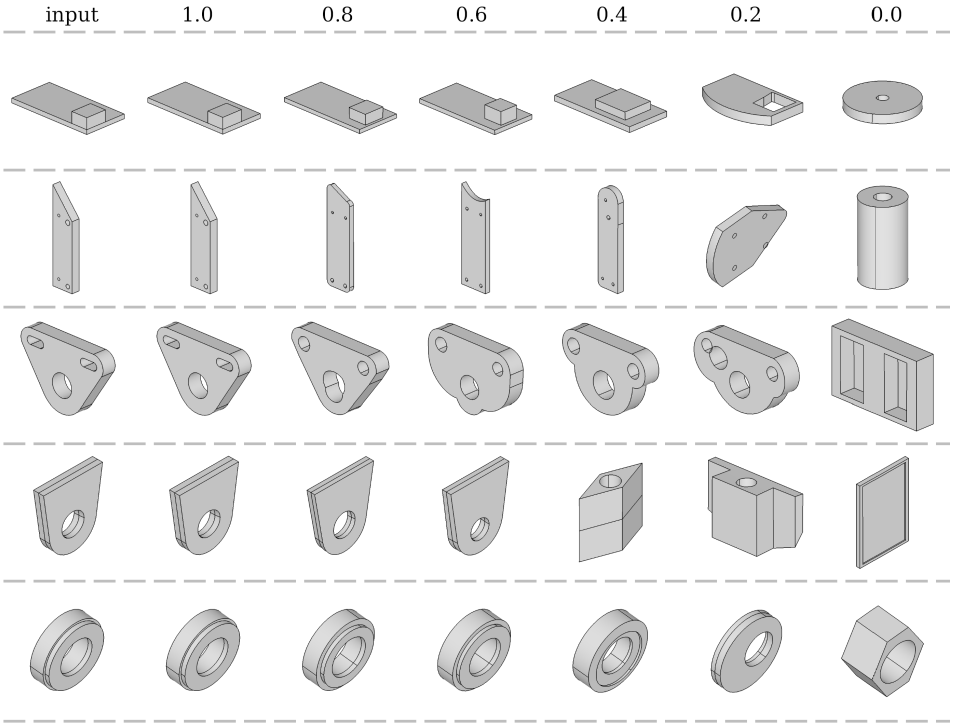

Figure 3: Effect of conditioning strength. Larger strengths improve input alignment, while strength 0 corresponds to unconditional generation.

Fig. 3 further illustrates generation results under varying conditioning strengths. When the conditioning strength is set to 1, the model exhibits reconstruction-like behavior with high fidelity to the input. When the conditioning strength is 0, the model performs unconditional generation. For intermediate values, the results smoothly interpolate between conditional and unconditional modes.

Table 2: Unconditional generation results on DeepCAD and ABC. MMD and JSD values are multiplied by 10^2.

	DeepCAD				ABC			
Model	MMD↓	COV↑	JSD↓	Validity ↑	MMD↓	COV↑	JSD↓	Validity ↑
GenCAD	1.425	**0.772**	3.947	**0.936**	-	-	-	-
BrepGen	1.124	0.711	1.376	0.698	1.484	0.589	2.866	0.448
DTGBrepGen	1.081	0.740	1.327	0.808	1.443	**0.614**	2.551	0.493
BrepDiff	1.219	0.697	1.297	0.718	1.637	0.566	2.757	0.392
Ours	**0.988**	0.731	**1.205**	0.912	**1.390**	0.606	**2.475**	**0.514**

Unconditional Generation. For unconditional generation, we compare our method with representative BRep generation approaches, including GenCAD, BrepGen, DTGBrepGen, and BrepDiff (Lee et al., 2025), under the same evaluation setting. To enable unconditional generation, we directly use our image-conditioned model by setting the conditioning strength to zero. As shown in Table 2, our method achieves the best MMD and JSD on both DeepCAD and ABC, while maintaining competitive coverage and validity. These results show that our unified framework generalizes effectively across both conditional and unconditional BRep generation, achieving strong performance in each setting.

Table 3: Ablation study on DeepCAD under single-view image-conditioned and unconditional generation settings. MMD and JSD values are multiplied by 10^2.

	Conditional		Unconditional			
Method	CD↓	Validity ↑	MMD↓	COV↑	JSD↓	Validity ↑
w/o importance ordering	0.098	0.837	1.149	0.692	1.298	0.824
w/o AR topology decoder (cVAE)	0.132	0.804	0.906	0.742	1.172	0.926
Ours	0.072	0.877	0.988	0.731	1.205	0.912

Ablation Study. Table 3 reports ablations on DeepCAD for both single-view image-conditioned and unconditional generation. Removing importance-aware ordering in the **FeF** stage degrades performance, indicating that prioritizing topologically important faces guides subsequent generation. Replacing the autoregressive topology decoder with a conditional VAE modestly improves unconditional generation but causes a larger drop in conditional generation. We therefore retain the autoregressive decoder for a better balance between the two settings.

5.3 Limitations and Future Work

Our method currently supports image-conditioned generation as the primary input modality, leaving broader multimodal conditioning underexplored. Extending the framework to additional modalities, such as point clouds, within a unified model is a direction for future work. In addition, the improvement on the ABC dataset is less pronounced than on DeepCAD, highlighting the greater challenge of handling complex geometries within the multi-stage generation pipeline. Developing more robust generation strategies for such complex geometries remains an important direction for future work.

6 Conclusion

We presented UniCAD, a unified generative model for single-view image-conditioned and unconditional BRep generation. UniCAD combines connectivity-aware topology modeling and topology-conditioned geometry diffusion for BRep generation, while classifier-free guidance enables continuous control over conditioning strength. Experiments on DeepCAD and ABC demonstrate strong performance in both conditional and unconditional settings, suggesting a promising direction for controllable automatic BRep generation.

AI USE STATEMENT

In this work, we used generative AI tools only for English-language polishing, LaTeX formatting and modification, and debugging author-written code. These uses were limited to improving readability and presentation and to identifying or correcting programming errors; generative AI was not used to generate the research ideas, propose or refine hypotheses, design or provide feedback on the research methodology or experiments, implement the proposed method, or interpret the experimental results. We also did not use generative AI to generate synthetic datasets, develop theoretical or conceptual frameworks, formulate mathematical claims or proofs, translate the manuscript, clean or reformat datasets, or conduct qualitative or thematic data analysis. All AI-assisted edits and debugging suggestions were reviewed by the authors, and any resulting code changes were independently checked and tested before use. We take responsibility for the final content of this work, including all text, claims, code, results, and artifacts produced with the aid of generative AI.

REFERENCES

Md Ferdous Alam and Faez Ahmed. Gencad: Image-conditioned computer-aided design generation with transformer-based contrastive representation and diffusion priors. *arXiv preprint arXiv:2409.16294*, 2024.

Cheng Chen, Jiacheng Wei, Tianrun Chen, Chi Zhang, Xiaofeng Yang, Shangzhan Zhang, Bingchen Yang, Chuan-Sheng Foo, Guosheng Lin, Qixing Huang, et al. Cadcrafter: Generating computer-aided design models from unconstrained images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11073–11082, 2025a.

Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2416–2425, 2023.

Zilong Chen, Yikai Wang, Wenqiang Sun, Feng Wang, Yiwen Chen, and Huaping Liu. Meshgen: Generating pbr textured mesh with render-enhanced auto-encoder and generative data augmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5835–5848, 2025b.

Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20637–20647, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances in neural information processing systems*, 35:31841–31854, 2022.

Xufei Guo, Xiao Dong, Juan Cao, and Zhonggui Chen. Cadtrans: A code tree-guided cad generative transformer model with regularized discrete codebooks. *Graphical Models*, 139:101262, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Zixuan Huang, Justin Johnson, Shoubhik Debnath, James M Rehg, and Chao-Yuan Wu. Pointinfinity: Resolution-invariant point diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10050–10060, 2024.

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*, pp. 9601–9611, 2019.

Jae Joong Lee and Bedrich Benes. Rgb2point: 3d point cloud generation from single rgb images. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2952–2962. IEEE, 2025.

Mingi Lee, Dongsu Zhang, Clément Jambon, and Young Min Kim. Brepdiff: Single-stage b-rep diffusion model. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025.

Jing Li, Yihang Fu, and Falai Chen. Dtgbrepgen: A novel b-rep generative model through decoupling topology and geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21438–21447, 2025a.

Pu Li, Wenhao Zhang, Weize Quan, Biao Zhang, Peter Wonka, and Dongming Yan. Brepgpt: Autoregressive b-rep generation with voronoi half-patch. *ACM Transactions on Graphics (TOG)*, 44(6):1–18, 2025b.

Yuan Li, Cheng Lin, Yuan Liu, Xiaoxiao Long, Chenxu Zhang, Ningna Wang, Xin Li, Wenping Wang, and Xiaohu Guo. Caddreamer: Cad object generation from single-view images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21448–21457, 2025c.

Yilin Liu, Duoteng Xu, Xingyao Yu, Xiang Xu, Daniel Cohen-Or, Hao Zhang, and Hui Huang. Hola: B-rep generation using a holistic latent representation. *ACM Transactions on Graphics (TOG)*, 44 (4):1–25, 2025.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2837–2845, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

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

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in neural information processing systems*, 33: 20154–20166, 2020.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19615–19625, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.

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.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *Advances in neural information processing systems*, 28, 2015.

Siyu Wang, Cailian Chen, Xinyi Le, Qimin Xu, Lei Xu, Yanzhou Zhang, and Jie Yang. Cad-gpt: Synthesising cad construction sequence with spatial reasoning-enhanced multimodal llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7880–7888, 2025.

Kevin J Weiler. *Topological structures for geometric modeling (Boundary representation, manifold, radial edge structure)*. Rensselaer Polytechnic Institute, 1986.

Jianyu Wu, Yizhou Wang, Xiangyu Yue, Xinzhu Ma, Jinyang Guo, Dongzhan Zhou, Wanli Ouyang, and Shixiang Tang. Cmt: A cascade mar with topology predictor for multimodal conditional cad generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7014–7024. IEEE, 2025.

Kailu Wu, Fangfu Liu, Zhihan Cai, Runjie Yan, Hanyang Wang, Yating Hu, Yueqi Duan, and Kaisheng Ma. Unique3d: High-quality and efficient 3d mesh generation from a single image. *Advances in Neural Information Processing Systems*, 37:125116–125141, 2024.

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*, pp. 6772–6782, 2021.

Jingwei Xu, Chenyu Wang, Zibo Zhao, Wen Liu, Yi Ma, and Shenghua Gao. Cad-mllm: Unifying multimodality-conditioned cad generation with mllm. *arXiv preprint arXiv:2411.04954*, 2024a.

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*, pp. 24698–24724. PMLR, 2022.

Xiang Xu, Pradeep Kumar Jayaraman, Joseph G Lambourne, Karl DD Willis, and Yasutaka Furukawa. Hierarchical neural coding for controllable cad model generation. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 38443–38461, 2023.

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, 2024b.

Xiang Xu, Pradeep Jayaraman, Joseph Lambourne, Yilin Liu, Durvesh Malpure, and Pete Meltzer. Autobrep: Autoregressive b-rep generation with unified topology and geometry. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

Yang You, Mikaela Angelina Uy, Jiaqi Han, Rahul Thomas, Haotong Zhang, Yi Du, Hansheng Chen, Francis Engelmann, Suya You, and Leonidas Guibas. Img2cad: Reverse engineering 3d cad models from images through vlm-assisted conditional factorization. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

A Additional Analysis of Conditioning Strength

We provide additional quantitative analysis of the conditioning strength used during inference. The conditioning strength controls the trade-off between input image alignment and generative diversity. Specifically, we evaluate each conditioning strength on the full DeepCAD test set and report the corresponding CD.

Table 4: Quantitative analysis under different conditioning strengths on DeepCAD.

Conditioning Strength	Mean CD ↓
1.0	0.072
0.8	0.078
0.6	0.089
0.4	0.131
0.2	0.312
0.0	0.493

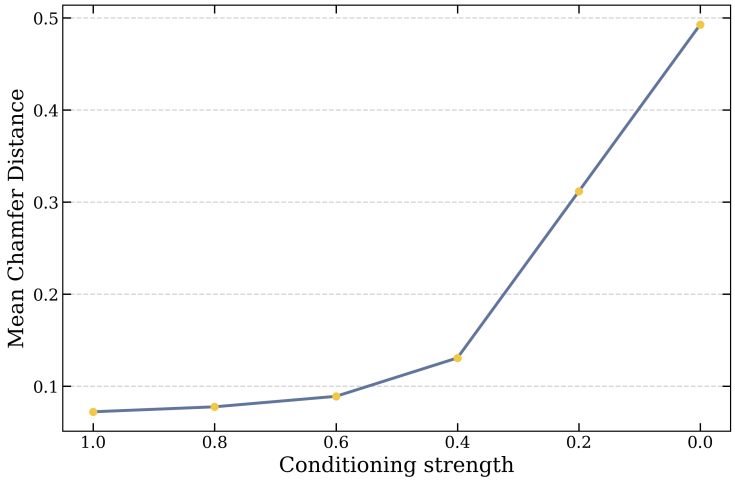

Figure 4: Quantitative trends under different conditioning strengths. Larger conditioning strengths improve alignment with the input image, while smaller values encourage more diverse generation.

As shown in Table 4 and Fig. 4, the Mean CD increases as the conditioning strength decreases, indicating weaker alignment with the input image. The increase is relatively mild from strength 1.0 to 0.4, where the generated results mainly exhibit changes in geometric parameters, bounding box scale, and minor topology variations. When the strength is further reduced from 0.4 to 0.0, the CD increases more sharply, suggesting that the model gradually departs from image-conditioned generation and exhibits larger topological changes. This trend quantitatively demonstrates that the conditioning strength provides continuous control during inference, ranging from image-faithful generation to unconditional exploration.

B Additional Unconditional Generation Results

We provide additional visual results for unconditional BRep generation in Fig. 5. These examples are generated by setting the conditioning strength to zero. This complements the quantitative evaluation in the main paper and illustrates the diversity of shapes generated under the unconditional setting.

C Failure Analysis

We provide representative failure cases in Fig. 6. The failures mainly fall into two categories. The first category is caused by topology generation errors, where incorrect face adjacency or edge connectivity

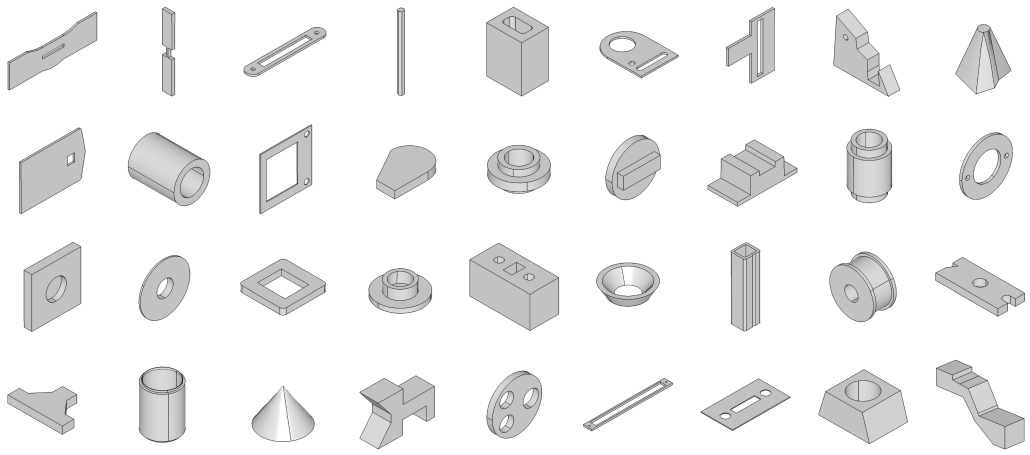

Figure 5: Additional unconditional BRep generation results on DeepCAD and ABC. The samples are generated with conditioning strength set to 0.

leads to severe structural artifacts. Since geometry generation is conditioned on the generated topology, such errors can propagate to subsequent stages and result in large deviations from plausible CAD shapes.

The second category is caused by minor geometric inaccuracies. Although the overall topology is largely reasonable, small errors in curve or surface parameters may lead to abnormal tangencies, slight gaps between adjacent faces, or imperfect boundary closure. These local geometric defects can break watertightness and make the generated BRep invalid. This suggests that improving topology robustness and enforcing stronger geometric consistency are important directions for future work.

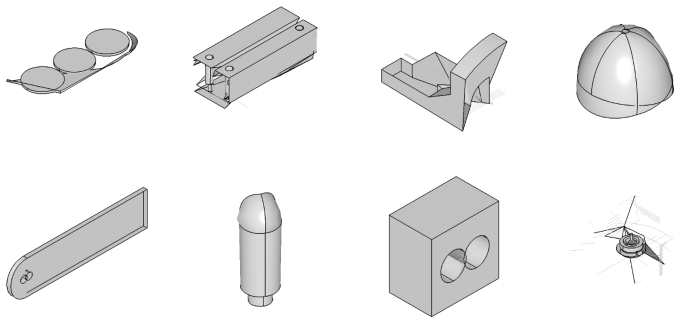

Figure 6: Representative failure cases. Failures mainly arise from topology generation errors that cause severe structural artifacts, or from local geometric inaccuracies that produce abnormal tangencies, small gaps, and non-watertight BReps.

D BREP VALIDITY EVALUATION

D.1 VALIDITY CRITERION.

Following the evaluation protocol of DTGBrepGen (Li et al., 2025a), we evaluate BRep validity using Open CASCADE Technology (OCCT) through pythonOCC. A generated sample is regarded as invalid if either its predicted geometry cannot be successfully constructed into a BRep model by OCCT, or the constructed model fails subsequent structural or watertightness checks. The latter includes invalid wire ordering, wire self-intersections, bad shell edges, open or free edges, and non-

watertight boundaries. Therefore, topological errors and edge–surface inconsistencies that prevent successful construction or closure are also reflected by the Validity metric.

D.2 NUMERICAL SETTINGS.

All geometric operations are performed in normalized model coordinates. Table 5 summarizes the numerical tolerances used during BRep construction and healing. The maximum wire and face tolerances are upper bounds used by the healing procedures rather than geometric fitting errors.

Table 5: Numerical settings used for BRep construction and validity checking. All values are reported in normalized model coordinates.

Stage	Value	Meaning
Analytic surface selection	0.01	Primitive-fitting error threshold
B-spline surface fitting	0.05	Surface approximation tolerance
B-spline curve fitting	0.005	Preferred curve-fitting tolerance
Curve fallback 1	0.008	First relaxed curve tolerance
Curve fallback 2	0.05	Final relaxed curve tolerance
Wire healing	0.01	Wire-healing precision
Wire minimum tolerance	0.0001	Minimum checking tolerance
Wire maximum tolerance	1.0	Maximum healing tolerance
Face healing	0.01	Face-healing precision
Face maximum tolerance	0.1	Maximum face tolerance
P-curve construction	0.001	Edge-on-surface tolerance

D.3 FAILURE BREAKDOWN.

To further analyze the causes of invalid generations, we divide the invalid samples into two groups. The first consists of samples whose predicted geometry cannot be successfully constructed as a BRep model by OCCT. The second consists of samples that can be constructed as BRep models but fail subsequent structural or watertightness checks. These failures may result from invalid wire ordering, wire self-intersections, bad shell edges, open or free edges, or non-watertight boundaries.

Table 6 summarizes the failure statistics for unconditional and conditional generation on the DeepCAD and ABC datasets. Across all settings, most invalid samples fail during BRep construction, while only a relatively small proportion can be constructed successfully but fail the subsequent structural or watertightness checks.

Table 6: Breakdown of invalid BRep generations. Construction failure refers to samples that cannot be successfully constructed as BRep models by OCCT, while structural failure refers to constructed models that fail subsequent structural or watertightness checks.

Dataset	setting	Total invalid (%)	Construction failure (%)	Structural failure (%)
DeepCAD	unconditional	8.77	8.15	0.62
ABC	unconditional	48.63	45.91	2.72
DeepCAD	conditional	12.40	11.08	1.32
ABC	conditional	49.20	45.34	3.86

D.4 DATASET COMPLEXITY ANALYSIS.

We further compare the structural and geometric characteristics of DeepCAD and ABC to provide additional context for evaluating generation performance across the two datasets. Table 7 summarizes the distribution of the number of faces per model. ABC contains substantially more faces than DeepCAD, with an average of 16.95 faces per model compared with 10.16 on DeepCAD. The difference becomes more pronounced for more complex shapes: the 75th percentile increases from 12 faces on DeepCAD to 23 faces on ABC, while the maximum number of faces increases from 30 to 50.

Table 7: Statistics of the number of BRep faces per model in DeepCAD and ABC.

Dataset	Average	Minimum	25th percentile	Median	75th percentile	Maximum
DeepCAD	10.1565	5	6	8	12	30
ABC	16.9471	2	9	14	23	50

The two datasets also differ substantially in surface composition. As shown in Table 8, DeepCAD consists entirely of planar and cylindrical surfaces in our evaluation split, with planar surfaces accounting for approximately 75.92

Table 8: Distribution of surface types in DeepCAD and ABC.

Surface type	DeepCAD average	DeepCAD proportion	ABC average	ABC proportion
Plane	7.7107	75.92%	9.2543	54.61%
Cylinder	2.4458	24.08%	6.0719	35.83%
Cone	0.0000	0.00%	0.7397	4.36%
Torus	0.0000	0.00%	0.5251	3.10%
Sphere	0.0000	0.00%	0.1677	0.99%
B-spline	0.0000	0.00%	0.1105	0.65%
Extrusion	0.0000	0.00%	0.0455	0.27%
Revolution	0.0000	0.00%	0.0259	0.15%
Offset	0.0000	0.00%	0.0064	0.04%

Overall, ABC contains models with more faces and substantially greater surface-type diversity than DeepCAD. These statistics characterize the increased structural and geometric complexity of ABC and provide additional context for the performance differences observed between the two datasets.

E Rendering View Setup

For image-conditioned generation, each CAD model is rendered from four predefined oblique views. The views are specified by azimuth and elevation angles, as summarized in Table 9. During training, one of the four rendered views is randomly sampled at each epoch, as described in the main paper. Fig. 7 illustrates the camera directions used for rendering.

Table 9: Camera angles used for rendering the four input views.

View	Azimuth	Elevation
View 1	45°	35°
View 2	135°	25°
View 3	225°	45°
View 4	315°	55°

F Network and Implementation Details

We provide dataset-specific preprocessing settings and network configurations used in our experiments. The same overall architecture is used for DeepCAD and ABC, with dataset-specific limits and several hidden dimensions adjusted according to dataset complexity.

F.1 Dataset and Preprocessing Settings

Table 10 summarizes the dataset-specific preprocessing settings and sequence limits used for DeepCAD and ABC.

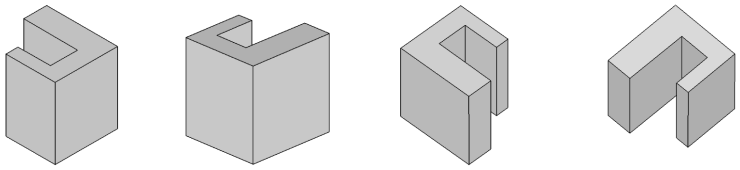

Figure 7: Illustration of the four predefined camera directions used for rendering input images.

Table 10: Dataset-specific preprocessing and sequence limits.

Parameter	DeepCAD	ABC
Maximum number of faces	30	50
Maximum edges per face	20	30
Edge classes	5	5
Threshold	0.05	0.05
Bounding box scale	3	3

F.2 Topology Generation Networks

The topology generation stage consists of a face–face adjacency model and an edge–vertex connectivity model. The face–face model generates $\mathbf{FeF}$, while the edge–vertex model generates $\mathbf{EV}^{\mathrm{seq}}$ conditioned on the predicted face adjacency. The detailed configurations are listed in Table 11.

Table 11: Network configurations for topology generation.

Module	Parameter	DeepCAD	ABC
$\mathbf{FeF}$ model	Hidden dimension d_{model}	128	128
$\mathbf{FeF}$ model	Attention heads	4	4
$\mathbf{FeF}$ model	Layers	4	4
$\mathbf{EV}$ model	Hidden dimension d_{model}	256	256
$\mathbf{EV}$ model	Layers	4	4

F.3 Geometry Generation Networks

The geometry generation stage follows the coarse-to-fine order described in the main paper: face bounding boxes, vertices, edges, and faces. Edge and face geometries are generated from B-spline control point representations, with edge geometry dimension 12 and face geometry dimension 48. Table 12 provides the network configurations for all geometry generation stages.

F.4 Conditioning and Dropout Settings

For stochastic conditioning, we use the same condition dropout probability across all stages. During training, the conditioning representation is dropped with probability $p_{\mathrm{drop}} = 0.25$. For the stochastic unconditional branch, we use $z_{\mathrm{scale}} = 1.0$ and apply latent dropout with probability 0.1.

G Compute Resources

All models are trained on 4 NVIDIA A100 GPUs with 40GB memory. Inference is performed on a single NVIDIA A100 GPU with 40GB memory. The total training time is approximately 45 hours for DeepCAD and 178 hours for ABC, including all staged topology and geometry generation modules.

Table 12: Network configurations for geometry generation.

Module	Parameter	DeepCAD	ABC
Face bbox model	MLP hidden dimension for encoding $\mathbf{x}_t$	256	256
Face bbox model	Token hidden dim d_x	512	768
Face bbox model	Condition hidden dim d_e	256	256
Face bbox model	Attention heads	8	12
Face bbox model	FFN hidden dim	512	512
Face bbox model	Layers	8	8
Vertex geometry model	MLP hidden dimension for encoding $\mathbf{x}_t$	256	256
Vertex geometry model	Token hidden dim d_x	512	512
Vertex geometry model	Condition hidden dim d_e	256	256
Vertex geometry model	Attention heads	8	8
Vertex geometry model	FFN hidden dim	512	512
Vertex geometry model	Layers	8	8
Edge geometry model	Geometry dimension	12	12
Edge geometry model	Hidden dimension d_{model}	768	768
Edge geometry model	Attention heads	12	12
Edge geometry model	Layers	8	8
Face geometry model	Geometry dimension	48	48
Face geometry model	Hidden dimension d_{model}	768	768
Face geometry model	Attention heads	12	12
Face geometry model	Layers	8	8