TOPOGEN: TOPOLOGY-GUIDED MULTIMODAL GENERATION FOR EXECUTABLE CADQUERY SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal models have shown promise in programmatic CAD generation; however, generating executable and structurally faithful programs from visual observations remains challenging. A model may reproduce the overall shape while failing to capture the structural relations and modeling operations required for engineering validity. Unlike many generative tasks, CAD provides an executable geometric representation that a geometric kernel can deterministically verify, making structural errors explicitly measurable. We present **TopoGen**, a topology-guided multimodal framework for CadQuery synthesis from point clouds and multi-view images. Our central insight is that CAD topology provides a shared structural space bridging heterogeneous visual observations and executable programs. TopoGen encodes topology as a constraint graph and derives geometry-grounded structural constraints from BReps for supervision during training. For execution-grounded reinforcement learning, we introduce Topology Relative Policy Optimization (ToRPO), which uses essential-operation matching to prevent reward exploitation and provide structural feedback during program generation. This design preserves both the geometric and topological relations of generated CAD models. We further construct **TopoCAD-1M**, a topology-augmented corpus that enriches large-scale mechanical engineering data with structural constraints and alleviates the long-tailed distribution of CAD operations.

1 INTRODUCTION

Computer-Aided Design (CAD) is central to modern engineering and manufacturing because it provides precise, editable representations of physical objects (Briere-Cote et al., 2012). CAD reconstruction seeks to automate this process by recovering structured models from visual observations such as point clouds and images (Mallis et al., 2023; Chen et al., 2025). However, what constitutes a correct reconstruction? A geometrically similar result may appear satisfactory, yet visual similarity alone does not guarantee editability or consistency with the intended function. A CAD model is defined not only by its visible surface but also by the primitives, constraints, modeling operations, and parametric relations that determine its construction and modification.

Early methods represent CAD models as ordered sequences of sketches and modeling operations (Willis et al., 2021; Wu et al., 2021; Xu et al., 2022). More recent work generates executable programs, particularly Python or CadQuery scripts, which can be compiled into parametric BReps (Rukhovich et al., 2025; Authors, 2024). This programmatic representation preserves modeling history and editability while providing a natural interface for foundation models. Recent systems have extended this paradigm to images, text, and multimodal inputs (Chen et al., 2024; Khan et al., 2024; Xu et al., 2024a; Kolodiazhnyi et al., 2025). Nevertheless, most methods primarily supervise program tokens or final geometry, leaving the structural relations underlying a CAD model implicit.

This limitation matters because visually similar parts may serve different functions. For example, an eccentric wheel and a flywheel can exhibit similar cylindrical appearances in sparse point clouds or rendered images. An eccentric wheel requires an offset between its rotation axis and outer cylinder axis, whereas a flywheel requires these axes to be coaxial for rotational energy storage. Their geometry may appear similar, but their constraints encode different functions and design intentions. Such examples show that geometric features alone provide insufficient evidence about relations such as

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.

incidence, parallelism, orthogonality, and coaxiality (Mills et al., 2001; Langbein et al., 2004; Li et al., 2011; Cheng et al., 2025).

We argue that CAD topology provides a structural bridge between multimodal observations and executable programs. BReps organize faces, edges, vertices, and their incidence relations into an explicit structure (Ansaldi et al., 1985; Lambourne et al., 2021). Based on this representation, we construct a CAD-grounded constraint graph that encodes directional relations, such as parallelism and orthogonality; parametric relations, such as equal lengths and angles; and incidence relations among geometric entities. This graph supplies explicit structural supervision beyond surface appearance and program tokens. Then we use these constraints to guide full-parameter fine-tuning of Qwen3-VL-4B (Yang et al., 2025), enabling the model to associate multimodal observations with the structural organization of executable CAD programs.

Large-scale procedural CAD corpora provide abundant executable programs but exhibit highly long-tailed operation distributions (Rukhovich et al., 2025; Kolodiazhnyi et al., 2025). Common primitives and simple extrusions dominate, whereas complex compositions and structurally important operations are relatively rare. We therefore construct **TopoCAD-1M**, a corpus of one million CAD models with topological constraint graphs and improved coverage of long-tailed operations.

CAD programs can be executed and evaluated automatically, making CAD generation well suited to reinforcement learning with programmatically verifiable rewards (Chen et al., 2025; Wang et al., 2025; Guan et al., 2025; Kolodiazhnyi et al., 2025). However, rewards dominated by surface similarity can be exploited. A program may achieve a high IoU while omitting a hole, replacing a helix with a simple primitive, or using an incorrect operation that produces a similar silhouette. Effective optimization must therefore assess both geometric fidelity and the operations and structural relations that define the intended design.

To address these challenges, we present TopoGen, a topology-guided multimodal framework that synthesizes executable CadQuery programs from point clouds and multi-view images. During supervised fine-tuning, constraint graph supervision aligns multimodal observations with explicit structural relations. During reinforcement learning, our Topology Relative Policy Optimization (ToRPO) jointly evaluates geometric fidelity, topological consistency, and essential modeling operations. Experiments show that TopoGen improves program generation and structural fidelity, with clear gains on complex CAD designs.

Our contributions are threefold:

- We introduce a structural formulation for multimodal CAD synthesis. A CAD-grounded constraint graph explicitly encodes primitive attributes and directional, parametric, and incidence relations, providing a modality-agnostic interface between visual observations and executable programs.
- We introduce **ToRPO**, an execution-grounded objective that augments geometric rewards with graph consistency and essential-operation matching. ToRPO shifts optimization from surface resemblance toward intended design recovery while suppressing rewards for geometrically plausible but structurally incorrect programs.
- We propose constraint-aware multimodal learning and construct **TopoCAD-1M**, a dataset of one million CAD models annotated with constraint graphs. The proposed supervision directly incorporates CAD structure into full-parameter fine-tuning and improves the coverage of rare operations in long-tailed distributions.

2 Related Work

CAD Generation and Programmatic CAD Modeling Early learning-based methods represented CAD models as ordered sketch-and-extrude sequences. DeepCAD introduced a tokenized representation preserving construction history and editability (Wu et al., 2021), later extended to richer inputs and operations (Ren et al., 2022; Xu et al., 2022; 2023; Zhang et al., 2025; Tan et al., 2026). However, these sequence-based methods rely on predefined vocabularies and struggle with long sequences, complex Booleans, and unsupported operations.

Another line of work directly generates boundary representations (BReps), which describe solids through faces, edges, vertices, and their connectivity (Ansaldi et al., 1985; Lambourne et al., 2021).

Recent approaches primarily adopt autoregressive (Van Den Oord et al., 2016; Radford et al., 2019; Esser et al., 2021) or diffusion models (Ho et al., 2020; Song et al., 2020b;a) to generate BRep geometry and topology. Although recent methods improve generation quality and structural validity (Xu et al., 2024b; Li et al., 2025), jointly modeling discrete topology and continuous geometry remains difficult, often leading to inconsistent structures and limited editability.

Programmatic CAD generation formulates reconstruction as executable program synthesis. CAD-Recode, CADFit, and CADFS generate or optimize construction programs using Python, geometry-driven optimization, or FeatureScript (Rukhovich et al., 2025; Nehme et al., 2026; Pyatov et al., 2026). Recent studies further apply foundation models to CadQuery generation in text-based and multimodal settings (Xie & Ju, 2025; Doris et al., 2025; Kolodiazhnyi et al., 2025). Although these methods produce executable outputs, they leave construction logic and design-defining relations implicit. We address this gap by representing constraints as a graph and integrating essential-operation matching, moving from surface similarity toward design-structure recovery.

Constraint-Aware CAD Representation and Reconstruction Unlike general reconstruction, which minimizes deviation from observations, constraint-aware reconstruction simultaneously recovers design intent. Constraints describe topology, relations such as parallelism, orthogonality, coaxiality, coplanarity, and symmetry, and parameters such as equal dimensions or ratios.

Model beautification in traditional reverse engineering laid the foundation for constraint-aware reconstruction. Mills et al. (2001) and Langbein (2003) surveyed 600 mechanical components and found regularities in 97% of models, showing that canonical angles, alignments and symmetry are prevalent in engineered objects. Langbein et al. (2001) and Mitra et al. (2007) detected these regularities in imperfect BReps via feature-space clustering and approximate-symmetry analysis. Benko et al. (2002) jointly optimized entities under prioritized relations while resolving conflicts. Langbein et al. (2004) applied constraint priorities and solvability tests before optimization, and Li et al. (2011) extended this paradigm to noisy point clouds by coupling local fitting with global relations. These traditional methods treat constraints as post-hoc regularities to detect or enforce, without jointly reasoning about geometry, topology, and design intent.

Recent methods incorporate constraints into learned representations and BRep reconstruction. Cst-Net (Cheng et al., 2025) encoded primitive types, principal directions, and adjacency from BReps or point clouds, improving recognition and rotation robustness. Mesh2Brep (Shen et al., 2025) recovered BReps from meshes via robust primitive fitting and relation inference, using similarity and intersection-aware constraints to preserve regularity, tangency, and topology. However, existing methods either learn constraints for shape understanding or enforce them during optimization, leaving unified reasoning about geometry, topology, and design intent underexplored.

Multimodal Reverse Engineering CAD reverse engineering recovers structured and editable models from geometric or visual observations. Traditional analytical methods rely on primitive segmentation and surface fitting but often require manual intervention and provide no watertightness guarantees (Roth & Levine, 1993; Xia et al., 2020). Learning-based methods automate this process from geometric inputs: TransCAD predicts construction sequences from point clouds (Dupont et al., 2024), AutoBRep reconstructs BRep topology through surface adjacency learning (Xu et al., 2025), and CAD-Recode translates point clouds into executable Python programs (Rukhovich et al., 2025).

Recent multimodal methods extend reverse engineering to visual inputs. Img2CADSeq connects images with construction sequences via sequence-based diffusion and intermediate geometry (Tan et al., 2026), while Cadrille directly generates executable programs from multiple modalities (Kolodiazhnyi et al., 2025). CADFit combines image-based 3D reconstruction with program fitting (Nehme et al., 2026). However, geometric and visual inputs mainly capture appearance, whereas CAD programs also encode construction logic and structural relations. TopoGen bridges this gap by using explicit topological constraints to connect point clouds and multi-view images with executable CadQuery programs.

3 METHOD

3.1 OVERVIEW

We formulate multimodal CAD synthesis as a structured inverse modeling problem. Given a visual observation, multiple executable programs may produce geometrically similar solids, while only a subset preserves the structural relations of the underlying design. This creates an inherent gap between geometric alignment and structural equivalence.

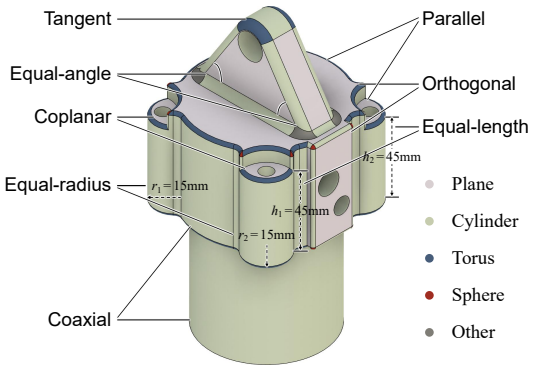


Figure 1: BRep-grounded constraint annotations. Surface colors denote primitive types, while labels indicate geometric and parametric relations between entities.

TopoGen addresses this gap by introducing a structural space between visual observations and executable programs. Given a ground-truth BRep, we use OpenCASCADE Technology (OCCT) along with a custom engine to extract deterministic feature constraints in a white-box manner (as shown in Figure 1). This graph serves as a shared structural target for both point clouds and multi-view images, allowing heterogeneous observations to be mapped toward the same underlying CAD organization.

The framework contains two complementary stages. In Stage I, **Topology-Guided Supervised Fine-Tuning** learns the correspondence between visual observations, CAD structure, and executable CadQuery programs. In Stage II, **Topology Relative Policy Optimization (ToRPO)** closes the loop and evaluates generated programs via CAD execution and jointly rewards to prevent exploit shortcuts from mismatched modeling operations

Importantly, point clouds and multi-view images are the only input modalities required at inference time. Structural annotations are used exclusively as training and verification signals (as shown in Figure 2).

3.2 Problem Formulation

Formally, given an observation $x \in \mathcal{P} \cup \mathcal{I}$, where $\mathcal{P}$ denotes a point cloud and $\mathcal{I} = \{(I_1, \ldots, I_K)\}$ denotes K multi-view images, TopoGen learns

$$x \longrightarrow \mathbf{z} \longrightarrow \tau \xrightarrow{\mathcal{E}} \hat{S} \xrightarrow{\Phi} \mathcal{G}, \tag{1}$$

where $\mathbf{z}$ denotes the learned multimodal representation, $\tau = (\tau_1, \ldots, \tau_T)$ is an executable CadQuery program, $\mathcal{E}$ is the CAD execution kernel, $\hat{S}$ is the generated solid and Φ extracts its structural graph.

Given an observation x, the model samples a CadQuery program $\tau \sim \pi_\theta(\cdot \mid x)$, which is executed by the CAD kernel. A conventional program synthesis objective focuses primarily on geometric agreement:

$$\theta^* = \arg\max_\theta \mathbb{E}_{\tau \sim \pi_\theta(\cdot|x)} \left[R_{\text{geo}}(\hat{S}, S^*) \right], \tag{2}$$

where $\mathcal{S}^*$ denote the ground-truth solid. However, CAD program synthesis is inherently underdetermined by final geometry. Let d_{geo} measure the discrepancy between executed solids, while d_{struct} measure the discrepancy between their BRep structural graphs. In particular, there may exist two programs $\tau_a \neq \tau_b$ such that $d_{\text{geo}}\left(\mathcal{E}(\tau_a), \mathcal{E}(\tau_b)\right) \approx 0$, but $d_{\text{struct}}\left(\tau_a, \tau_b\right) \gg 0$. Thus, geometric alignment does not imply equivalence of construction logic or structural organization.

3.3 Topology-Guided Supervised Fine-Tuning

Stage I aligns visual observations with CAD structure and executable programs. For each CAD model, we construct an edge-aware point cloud and deterministically extract a constraint graph from its ground-truth BRep. The graph is serialized as a structural target and jointly learned with the corresponding CadQuery program.

Edge-Aware Point Sampling Uniform surface sampling over-represents large planar regions and may miss sharp boundaries that indicate feature transitions, holes, chamfers, and fillets (Xie et al., 2025; Cheng et al., 2025). We therefore reserve part of the point budget for structurally informative

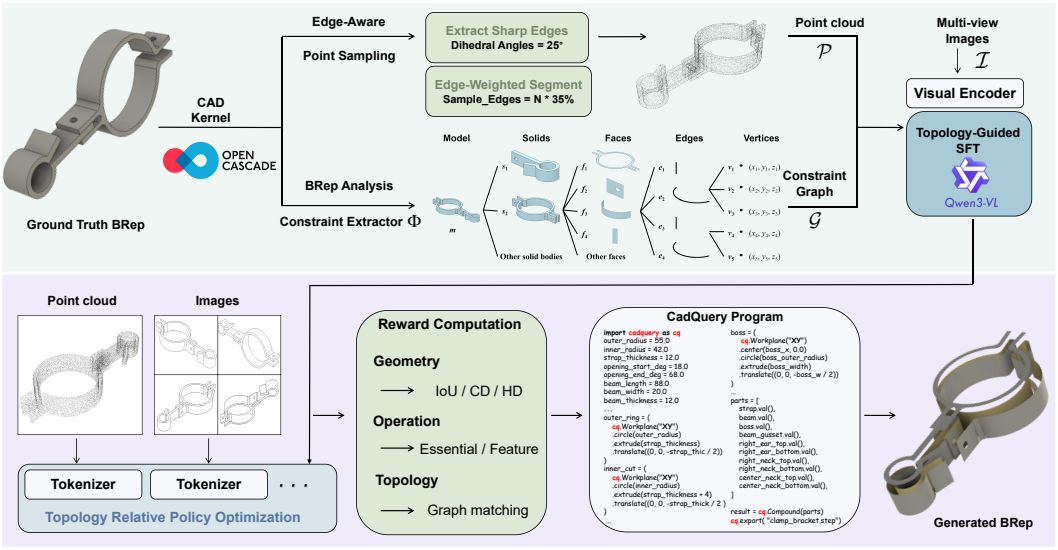


Figure 2: Overview of the TopoGen pipeline. In Stage I, a deterministic extractor converts BReps into constraint-grounded graphs with primitive and structural relations, providing shared supervision for CadQuery synthesis from point clouds and multi-view images. In Stage II, ToRPO jointly optimizes geometric essential-operation and topological fidelity to prevent structural shortcuts.

edges. Let S_{sharp} be the set of solid edges whose dihedral angles exceed $25°$. Given a budget of M points, the sampled point cloud is

$$\mathcal{P} = \mathcal{P}_{\text{surface}} \cup \mathcal{P}_{\text{boundary}}, \qquad |\mathcal{P}_{\text{boundary}}| = \rho M, \tag{3}$$

where $\rho = 0.35$ in our implementation. The remaining $(1-\rho)M$ points are sampled over BRep faces, while $\mathcal{P}_{\text{boundary}}$ is sampled near S_{sharp}. This allocation preserves global surface coverage while increasing the visibility of local geometric discontinuities.

Deterministic Constraint Graph A CAD BRep explicitly organizes a solid through faces, edges, and vertices. We use OCCT (Banović et al., 2018) and a custom geometry engine to convert each ground-truth solid $S^\star$ into a typed constraint graph

$$\mathcal{G}_{\mathcal{R}} = \Phi(S^\star) = (\mathcal{V}, \mathcal{E}_{rel}), \tag{4}$$

where $\mathcal{V}$ denotes geometric entities, $\mathcal{E}_{rel}$ denotes pairwise relations, and $\mathcal{R}$ is the relation vocabulary.

We first decompose the graph nodes according to BRep entity type: $\mathcal{V} = \mathcal{V}_F \cup \mathcal{V}_E \cup \mathcal{V}_V$, corresponding to faces, edges, and vertices. For each node $v \in \mathcal{V}$, we define an attribute vector $[c_v, t_v, \mathbf{o}_v, \mathbf{g}_v]$, where c_v is the BRep entity type, t_v denotes the primitive type when applicable, $\mathbf{o}_v$ describes direction information, and $\mathbf{g}_v$ contains geometric parameters.

Each structural relation is represented by a typed edge $e_{ij} = (v_i, r, v_j), \quad r \in \mathcal{R}$. We organize the relation space into three families: $\mathcal{R} = \mathcal{R}_{\text{ori}} \cup \mathcal{R}_{\text{par}} \cup \mathcal{R}_{\text{inc}}$, where $\mathcal{R}_{\text{ori}} = \{\text{parallel}, \text{orthogonal}, \text{coplanar}, \text{coaxial}\}$, $\mathcal{R}_{\text{par}} = \{\text{equal length}, \text{equal radius}\}$, $\mathcal{R}_{\text{inc}} = \{\text{vertex}-\text{edge}, \text{vertex}-\text{face}, \text{edge}-\text{face}\}$. We encode orientation, parameter, and incidence constraints using geometric invariants, as detailed in Table 1.

The resulting graph can therefore be viewed as a compact structural description, which is extracted deterministically from the module rather than predicted from noisy observations. This provides a modality-agnostic structural target shared by point clouds and multi-view images.

Structural Space for Program Learning For supervised fine-tuning, we construct $\mathfrak{D}_{\text{SFT}} = \{(o_n, \mathcal{G}_n, \tau_n^\star)\}_{n=1}^{L}$, where o_n is an edge-aware point cloud or a set of multi-view images, $\mathcal{G}_n$ is

the extracted constraint graph (Eq. 4), and $\tau_n^\star$ is the ground-truth CadQuery program. We serialize the graph using a deterministic template $\mathrm{Ser}(\cdot)$ and form the target sequence $\mathbf{y}_n^\star = [\mathrm{Ser}(\mathcal{G}_n); \tau_n^\star]$.

Qwen3-VL-4B is then optimized end to end to predict this sequence from the visual observation. Let $\mathcal{J}_{\mathrm{con}}$ and $\mathcal{J}_{\mathrm{code}}$ denote the token positions belonging to the constraint description and CadQuery program, respectively. The objective is

$$\mathcal{L}_{\mathrm{SFT}} = -\lambda_{\mathrm{con}} \sum_{t \in \mathcal{J}_{\mathrm{con}}} \log p_\theta(y_t^\star \mid \mathbf{y}_{<t}^\star, o) - \lambda_{\mathrm{code}} \sum_{t \in \mathcal{J}_{\mathrm{code}}} \log p_\theta(y_t^\star \mid \mathbf{y}_{<t}^\star, o). \tag{5}$$

The constraint tokens require the model to recover design-relevant relations before completing the executable program. Consequently, point clouds and multi-view images are mapped to a common structural description $\mathcal{G} = \Phi(S^\star)$ without providing ground-truth constraints at inference time.

3.4 Topology Relative Policy Optimization

Although topology-guided SFT aligns visual observations with structural constraints and CadQuery programs, token-level imitation cannot verify whether a generated program executes correctly or preserves the intended construction. DPO (Rafailov et al., 2023) relies on predefined preference pairs and therefore cannot directly exploit deterministic feedback from a CAD kernel. Existing execution-based CAD optimization, including Cadrille (Kolodiazhnyi et al., 2025), provides geometric feedback but remains vulnerable to shortcuts when the reward is dominated by surface similarity. A program may match the target silhouette while replacing a design-defining operation with a simpler primitive. ToRPO addresses this limitation by executing each sampled program and ranking candidates using both geometric fidelity and construction-aware checks, without requiring human preference labels or a learned reward model.

Table 1: Structural constraints used for topology-guided CAD synthesis.

Constraint Family	Relations	Formulation
Orientation	Parallel	$\mathbf{d}_i \times \mathbf{d}_j = \mathbf{0}$
	Orthogonal	$\mathbf{d}_i \cdot \mathbf{d}_j = 0$
	Coplanar	$\mathbf{n} \cdot (\mathbf{p}_i - \mathbf{p}_j) = 0$
	Coaxial	$\mathbf{d}_i \times \mathbf{d}_j = \mathbf{0}$, $\mathbf{d}_i \times (\mathbf{p}_i - \mathbf{p}_j) = \mathbf{0}$
Parameter	Equal length	$\|l_i - l_j\| = 0$
	Equal radius	$\|r_i - r_j\| = 0$
Incidence	Vertex–Edge	$\mathbf{p}_v \in \mathbf{e}_{ij}$
	Face–Vertex	$\mathbf{p}_v \in \mathbf{f}_{ijk}$
	Edge–Face	$\mathbf{e}_{ij} \subset \mathbf{f}_{klm}$

For each observation x, we sample a group of J programs $\{\tau_j\}_{j=1}^J$ from the current policy and execute them with the CAD kernel. An invalid program receives zero reward. For a valid program, the score is

$$R(\tau) = 0.70R_{\mathrm{IoU}} + 0.10R_{\mathrm{ess}} + 0.10R_{\mathrm{feat}} + 0.05R_{\mathrm{CD}} + 0.05R_{\mathrm{HD}}, \tag{6}$$

where R_{IoU} is the voxel IoU computed at a fixed orientation on a 64^3 grid. We do not optimize rotation-invariant IoU because recovering the correct orientation is part of the task; IoU over 24 rotations is reported only as a diagnostic metric. R_{ess} is a family-specific binary test for design-defining operations. For example, a torsion spring must contain both sweep and helix operations, preventing a cylindrical surrogate from receiving full credit despite a similar silhouette. R_{feat} is the F1 score over the presence of chamfers, fillets, and holes. The normalized Chamfer and Hausdorff terms receive smaller weights because they are strongly correlated with IoU and mainly differentiate candidates with similarly high overlap.

The rewards within each sampled group are standardized to obtain relative advantages, $A_j = \frac{R(\tau_j) - \mu_R}{\sigma_R + \varepsilon}$, which increase the likelihood of candidates that outperform alternatives generated for the same observation. We optimize the clipped objective

$$\mathcal{L}_{\mathrm{ToRPO}} = -\mathbb{E}_{j,t}\left[\min\left(q_{j,t}A_j,\ \mathrm{clip}(q_{j,t}, 1-\epsilon, 1+\epsilon)A_j\right)\right] + \beta D_{\mathrm{KL}}\left(\pi_\theta \| \pi_{\mathrm{ref}}\right), \tag{7}$$

where $q_{j,t} = \pi_\theta(\tau_{j,t} \mid \tau_{j,<t}, x) / \pi_{\mathrm{old}}(\tau_{j,t} \mid \tau_{j,<t}, x)$. The anti-shortcut terms R_{ess} and R_{feat} contribute 0.20 of the total score, sufficiently penalizing programs that reproduce the visible surface but omit defining operations. Meanwhile, retaining most of the reward for geometry provides a stable ranking signal while current models still struggle with basic shape recovery.

4 EXPERIMENTS

Datasets. Before discussing the model architecture, we first address the quality and composition of the training data. Existing procedural CAD datasets are strongly biased toward simple shapes with few faces, while complex parts with multiple operations and design-relevant structures are relatively rare. Moreover, some generated samples contain multiple disconnected solids and lack a clear single-part design intent. Such samples are less suitable for CAD construction programs learning. To address these issues, we construct **TopoCAD-1M**, a topology-augmented corpus containing approximately one million single-solid CAD models. We exclude samples containing more than two separate solids and augment each sample with an extra BRep-grounded constraint graph. To establish balanced complexity tiers, we organize the corpus according to the number of BRep faces: 200K models (1-15 faces), 600K models (16-50 faces), 100K models (51-100 faces), and 100K models (100+ faces). We use a standard 80/10/10 train/val/test split in line with existing baselines to ensure a fair comparison. The corpus curates models from the ABC dataset (Koch et al., 2019), the DeepCAD (Wu et al., 2021) and CC3D (Mallis et al., 2023) subset used by CAD-Recode and Zero-to-CAD (Ataei et al., 2026). This design reduces the dominance of simple shapes while increasing exposure to complex construction patterns and structural constraints.

Experimental Setup. We evaluate TopoGen in both point-cloud and multi-view image settings. Unless otherwise specified, the model uses four rendered views and a point cloud containing 1,024 points. The backbone is Qwen3-VL-4B-Instruct with full-parameter supervised fine-tuning in bfloat16 precision. Training is performed on eight NVIDIA A100 GPUs with 80 GB memory using distributed data parallelism. The per-device batch size is 1, the gradient accumulation step is 2, and the resulting global batch size is 16. We train for one epoch with a learning rate of 1×10^{-5}, a weight decay of 0.01, and a warmup ratio of 0.03. For execution-grounded optimization, we initialize ToRPO from the supervised checkpoint. Each input produces a group of four candidate programs, and each rollout performs two policy updates. We use a learning rate of 1×10^{-6}, a clipping range of 0.2, a KL coefficient of 0.01, a generation temperature of 0.9, and a top-p value of 0.95. All generated programs are compiled with the CAD kernel before reward computation. Invalid programs receive no geometric or structural credit.

Metrics. Following Cadrille, we normalize each ground-truth CAD model to fit within $[-0.5, 0.5]^3$ and evaluate reconstruction quality using Chamfer Distance (CD), voxel Intersection over Union (IoU), and Invalidity Ratio (IR). CD is computed between 8,192 sampled surface points and is reported after multiplication by 10^3. IoU is computed from voxelized reconstructed and ground-truth solids. IR denotes the percentage of generated programs that fail to produce a valid CAD model. Because invalid outputs can strongly bias mean-based estimates, we report both the mean and median CD over valid predictions. We also report mean and median IoU, together with IR, to distinguish average performance from the typical reconstruction quality.

Results on Image-to-BRep. Table 2 compares TopoGen with existing methods for multi-view image-conditioned CAD reconstruction across four datasets, while Fig. 3 provides qualitative comparisons. CADDreamer captures coarse primitive geometry but often produces incomplete or over-simplified structures, while CADCrafter recovers finer details but may misplace or omit local features such as holes and protrusions.

Cadrille generally achieves stronger geometric fidelity and performs particularly well in terms of CD on some datasets. However, its reconstructions can suffer from thin features that break connectivity, misplaced holes, and self-intersections, leading to structurally incorrect or invalid BReps even when the overall shape is well recovered.

In contrast, our method demonstrates more faithful reconstruction that aligns closely with the ground-truth CAD models, preserving both geometric details and structural relations. By introducing CAD-grounded structural supervision, TopoGen explicitly guides the model to recover feature relations, while ToRPO further encourages executable programs to preserve the underlying CAD construction.

Results on Point-to-BRep. Table 2 reports quantitative results of cross-dataset for point-cloud-conditioned CAD reconstruction , and Fig. 4 provides qualitative comparisons. Compared with

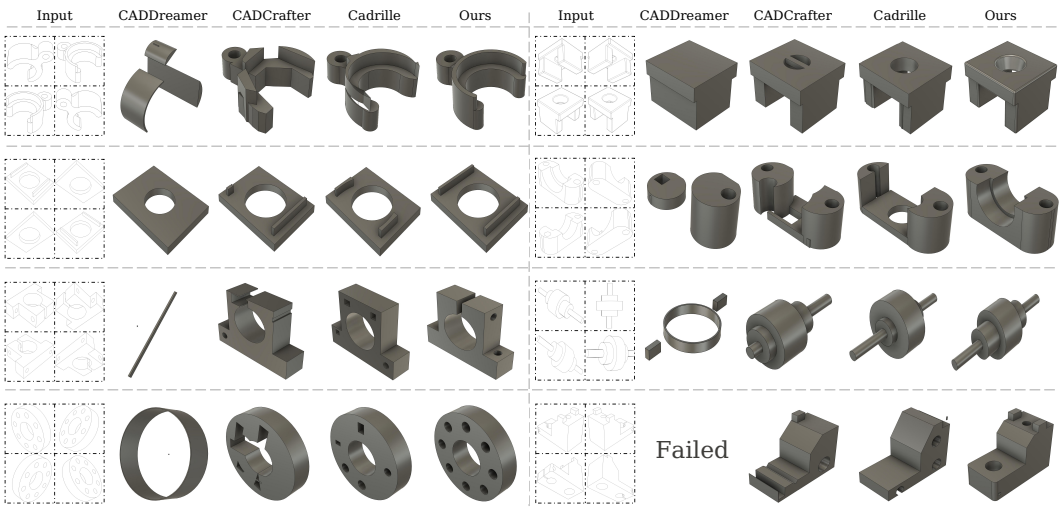


Figure 3: Qualitative comparison of multi-view image-conditioned CAD reconstruction. TopoGen produces geometrically faithful BRep models while better preserving structural relations and design-relevant features, such as holes, intersections, and repeated geometric patterns.

(a) Multi-view Image Input

Method	DeepCAD			Fusion 360			CC3D			TopoCAD		
	CD↓	IoU↑	IR↓	CD↓	IoU↑	IR↓	CD↓	IoU↑	IR↓	CD↓	IoU↑	IR↓
CADDreamer	0.964	72.43	25.87	1.134	67.58	29.84	3.478	60.72	31.73	2.761	61.73	30.58
CADCrafter	0.367	78.04	12.74	0.416	79.47	14.63	0.439	59.24	20.83	0.554	51.36	17.55
Cadrille	0.263	87.14	2.28	**0.209**	82.76	5.47	0.475	67.93	6.01	0.596	70.17	4.89
TopoGen	**0.173**	**90.08**	**1.21**	0.281	**83.49**	**5.17**	**0.412**	**73.38**	**5.79**	**0.471**	**80.14**	**4.28**

(b) Point Cloud Input

Method	DeepCAD			Fusion 360			CC3D			TopoCAD		
	CD↓	IoU↑	IR↓	CD↓	IoU↑	IR↓	CD↓	IoU↑	IR↓	CD↓	IoU↑	IR↓
CAD-SIGNet	0.374	75.27	6.13	0.736	60.94	10.17	3.718	49.14	15.08	4.693	49.58	14.72
CAD-Recode	0.214	79.86	4.83	0.217	74.43	9.14	0.674	67.18	11.26	0.657	68.14	10.93
Cadrille	0.203	88.34	3.38	0.268	80.76	5.24	0.437	**77.86**	5.47	0.413	74.66	4.93
TopoGen	**0.153**	**91.76**	**2.74**	**0.237**	**82.57**	**3.16**	**0.247**	72.47	**3.71**	**0.314**	**78.63**	**3.19**

Table 2: Cross-dataset evaluation of CAD program synthesis from multi-view images and point clouds. We report CD, IoU and IR. Lower CD and IR and higher IoU indicate better performance. Each panel includes methods to the corresponding input modality.

image inputs, point clouds provide more explicit geometric evidence, yet sparse sampling can still make it difficult to recover small features and maintain consistent scales across primitives. CAD-SIGNet and CAD-Recode generally recover the main primitive structures, but their predictions can exhibit scale drift and locally plausible yet globally incoherent primitive compositions, particularly for parts with multiple interacting features.

Cadrille produces more geometrically faithful reconstructions and achieves particularly strong IoU on several datasets. However, it can still suffer from local structural inconsistencies when combining multiple primitives, resulting in mismatched feature relations or inaccurate global configurations.

In contrast, TopoGen more consistently preserves both local geometric details and global structural coherence. Its CAD-grounded constraint graph explicitly models relations between primitives, helping mitigate scale drift and structurally inconsistent compositions, while ToRPO further encourages the generated programs to remain executable and faithful to the target CAD structure.

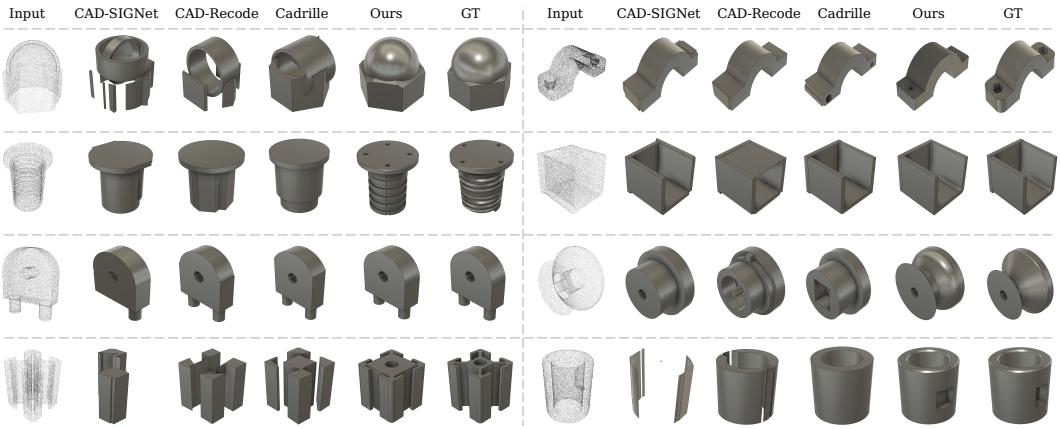


Figure 4: We evaluate our method against state-of-the-art approaches on inputs with point clouds. While baseline methods often produce distorted shapes or fail to recover topology under heavy noise, our approach can still synthesize multiple plausible CAD candidates.

Ablation Studies. We perform ablation studies on the TopoCAD dataset to evaluate the contribution of each core component (Table 3). To comprehensively assess modality-agnostic robustness, all models are evaluated on a mixed-modality validation split (`pc_img`), which uniformly samples both point clouds and multi-view images.

Table 3: Ablation on core components. Best in bold.

Module	Optimization	CD↓	IoU↑	IR↓
M1	CADRILLE on Qwen3-VL-4B	0.403	75.5	4.20
M2	TopoGen with basic SFT	0.381	79.2	3.98
M3	Topology-Guided SFT	0.362	80.9	3.01
M4	**Full TopoGen with ToRPO**	**0.283**	**85.7**	**2.33**

We compare the full pipeline (M4) against three progressive variants. The transition from M1 to M2 demonstrates that our basic SFT formulation already outperforms the Cadrille baseline on mixed observations. More importantly, removing deterministic constraint supervision (M2) causes the Invalid Ratio (IR) to increase from 3.01 to 3.98 compared to Topology-Guided SFT (M3). This indicates that without explicit structural targets, the model struggles to deduce valid topological relations purely from heterogeneous visual data, leading to more compilation failures. Furthermore, removing ToRPO (M3) limits the overall IoU to 80.9 and degrades the Chamfer Distance (CD). This drop confirms that token-level SFT alone is vulnerable to structural shortcuts; execution-grounded optimization is essential to verify design-defining operations and prevent the model from generating geometric surrogates. Ultimately, the full model (M4) validates the complementary necessity of topology-guided SFT and ToRPO for robust cross-modal CAD synthesis.

5 Conclusions and Future Work

We presented **TopoGen**, a multimodal framework for synthesizing executable CAD programs from point clouds and multi-view images. TopoGen combines deterministic BRep-grounded constraint extraction, complexity-balanced data construction, and topology-aware execution optimization. TopoCAD-1M supplies structural annotations for diverse CAD operations, while constraint-guided SFT and ToRPO jointly improve geometric fidelity, topological consistency, and essential-operation recovery while reducing surface-similarity reward exploitation.

Experiments demonstrate consistent improvements in geometric and structural fidelity, particularly for complex parts and rare operations. Future work will extend the constraint vocabulary toward higher-level design intent, improve robustness to real-world observations with continuously interactive editing, and develop agentic workflows that iteratively generate, execute, inspect, and revise CAD programs through feedback from CAD kernels and engineering tools.

AI USE STATEMENT

During manuscript preparation, the authors used LLM-based tools exclusively for copy-editing and language polishing. AI was not involved in generating ideas, experiments, or scientific contributions. All AI-assisted text was reviewed and revised by the authors, who bear full responsibility for the content.

ETHICS STATEMENT

This work develops methods for automated CAD modeling and programmatic 3D geometry synthesis. It does not involve human subjects, animals, or personal data. All datasets are synthetic or from public 3D repositories, posing no significant societal bias or ethical risk. The authors identify no negative societal impact or dual-use concerns.

REFERENCES

Silvia Ansaldi, Leila De Floriani, and Bianca Falcidieno. Geometric modeling of solid objects by using a face adjacency graph representation. *ACM SIGGRAPH Computer Graphics*, 19(3):131–139, 1985.

Mohammadmehdi Ataei, Farzaneh Askari, Kamal Rahimi Malekshan, and Pradeep Kumar Jayaraman. Zero-to-cad: Agentic synthesis of interpretable cad programs at million-scale without real data. *arXiv preprint arXiv:2604.24479*, 2026.

CadQuery Authors. Cadquery/cadquery: Cadquery 2.4.0, January 2024. URL `https://doi.org/10.5281/zenodo.10513848`.

Mladen Banović, Orest Mykhaskiv, Salvatore Auriemma, Andrea Walther, Herve Legrand, and Jens-Dominik Müller. Algorithmic differentiation of the open cascade technology cad kernel and its coupling with an adjoint cfd solver. *Optimization Methods and Software*, 33(4-6):813–828, 2018.

Pal Benko, Géza Kós, Tamás Várady, László Andor, and Ralph Martin. Constrained fitting in reverse engineering. *Computer Aided Geometric Design*, 19(3):173–205, 2002.

Antoine Briere-Cote, Louis Rivest, and Roland Maranzana. Comparing 3d cad models: uses, methods, tools and perspectives. *Computer-Aided Design and Applications*, 9(6):771–794, 2012.

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*, 2025.

Tianrun Chen, Chunan Yu, Yuanqi Hu, Jing Li, Tao Xu, Runlong Cao, Lanyun Zhu, Ying Zang, Yong Zhang, Zejian Li, et al. Img2cad: Conditioned 3d cad model generation from single image with structured visual geometry. *arXiv preprint arXiv:2410.03417*, 2024.

Xi Cheng, Ruiqi Lei, Di Huang, Zhichao Liao, Fengyuan Piao, Yan Chen, Pingfa Feng, and Long Zeng. Constraint-aware feature learning for parametric point cloud. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28114–28124. IEEE, 2025.

Anna C Doris, Md Ferdous Alam, Amin Heyrani Nobari, and Faez Ahmed. Cad-coder: An open-source vision-language model for computer-aided design code generation. *arXiv preprint arXiv:2505.14646*, 2025.

Elona Dupont, Kseniya Cherenkova, Dimitrios Mallis, Gleb Gusev, Anis Kacem, and Djamila Aouada. Transcad: A hierarchical transformer for cad sequence inference from point clouds. In *European Conference on Computer Vision*, pp. 19–36. Springer, 2024.

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.

Yandong Guan, Xilin Wang, Xingxi Ming, Jing Zhang, Dong Xu, and Qian Yu. Cad-coder: Text-to-cad generation with chain-of-thought and geometric reward. *arXiv preprint arXiv:2505.19713*, 2025.

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

Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. Text2cad: Generating sequential cad designs from beginner-to-expert level text prompts. *Advances in Neural Information Processing Systems*, 37:7552–7579, 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Maksim Kolodiazhnyi, Denis Tarasov, Dmitrii Zhemchuzhnikov, Alexander Nikulin, Ilya Zisman, Anna Vorontsova, Anton Konushin, Vladislav Kurenkov, and Danila Rukhovich. cadrille: Multi-modal cad reconstruction with online reinforcement learning. *arXiv e-prints*, pp. arXiv–2505, 2025.

Joseph G Lambourne, Karl DD Willis, Pradeep Kumar Jayaraman, Aditya Sanghi, Peter Meltzer, and Hooman Shayani. Brepnet: A topological message passing system for solid models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12773–12782, 2021.

Frank C Langbein, Bruce I Mills, A David Marshall, and Ralph R Martin. Recognizing geometric patterns for beautification of reconstructed solid models. In *Proceedings International Conference on Shape Modeling and Applications*, pp. 10–19. IEEE, 2001.

Frank C Langbein, A David Marshall, and Ralph R Martin. Choosing consistent constraints for beautification of reverse engineered geometric models. *Computer-Aided Design*, 36(3):261–278, 2004.

Frank Curd Langbein. *Beautification of reverse engineered geometric models*. PhD thesis, Cardiff University, 2003.

Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. Globfit: Consistently fitting primitives by discovering global relations. In *Acm siggraph 2011 papers*, pp. 1–12. 2011.

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 IEEE/CVF International Conference on Computer Vision*, 2025.

Dimitrios Mallis, Ali Sk Aziz, Elona Dupont, Kseniya Cherenkova, Ahmet Serdar Karadeniz, Mohammad Sadil Khan, Anis Kacem, Gleb Gusev, and Djamila Aouada. Sharp challenge 2023: Solving cad history and parameters recovery from point clouds and 3d scans. overview, datasets, metrics, and baselines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1786–1795, 2023.

BI Mills, FC Langbein, AD Marshall, and RR Martin. Estimate of frequencies of geometric regularities for use in reverse engineering of simple mechanical components. *Internat. J. Shape Modeling. Submitted*, 2001.

Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Symmetrization. *ACM Transactions on Graphics (TOG)*, 26(3):63–es, 2007.

Ghadi Nehme, Eamon Whalen, and Faez Ahmed. Cadfit: Precise mesh-to-cad program generation with hybrid optimization. *arXiv preprint arXiv:2605.01171*, 2026.

Vladislav Pyatov, Gleb Bobrovskikh, Saveliy Galochkin, Nikita Boldyrev, Oleg Voynov, Alexander Filippov, Gonzalo Ferrer, Peter Wonka, and Evgeny Burnaev. Cadfs: A big cad program dataset and framework for computer-aided design with large language models. *arXiv preprint arXiv:2605.01925*, 2026.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Daxuan Ren, Jianmin Zheng, Jianfei Cai, Jiatong Li, and Junzhe Zhang. Extrudenet: Unsupervised inverse sketch-and-extrude for shape parsing. In *European Conference on Computer Vision*, pp. 482–498. Springer, 2022.

Gerhard Roth and Martin D Levine. Extracting geometric primitives. *CVGIP: image understanding*, 58(1):1–22, 1993.

Danila Rukhovich, Elona Dupont, Dimitrios Mallis, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. Cad-recode: Reverse engineering cad code from point clouds. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9801–9811. IEEE, 2025.

Zeyu Shen, Mingyang Zhao, Dong-Ming Yan, and Wencheng Wang. Mesh2brep: B-rep reconstruction via robust primitive fitting and intersection-aware constraints. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):6661–6676, 2025.

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.

Shiyu Tan, Zixuan Zhao, Hao Gao, Zhiheng Chen, Xiaolong Yin, and Enya Shen. Img2cadseq: Image-to-cad generation via sequence-based diffusion. *arXiv preprint arXiv:2605.13293*, 2026.

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

Ruiyu Wang, Yu Yuan, Shizhao Sun, and Jiang Bian. Text-to-cad generation through infusing visual feedback in large language models. *arXiv preprint arXiv:2501.19054*, 2025.

Karl DD Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. *ACM Transactions on Graphics (TOG)*, 40(4): 1–24, 2021.

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.

Shaobo Xia, Dong Chen, Ruisheng Wang, Jonathan Li, and Xinchang Zhang. Geometric primitives in lidar point clouds: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:685–707, 2020.

Haoyang Xie and Feng Ju. Text-to-cadquery: A new paradigm for cad generation with scalable large model capabilities. *arXiv preprint arXiv:2505.06507*, 2025.

Jingpeng Xie, Shiyu Tan, Yuanlei Wang, Tianle Du, Yifei Xue, and Yizhen Lao. Iovs4nerf: Incremental optimal view selection for large-scale nerfs. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

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 *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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhanwei Zhang, Shizhao Sun, Wenxiao Wang, Deng Cai, and Jiang Bian. Flexcad: Unified and versatile controllable cad generation with fine-tuned large language models. *International Conference on Learning Representations*, 2025.

A Appendix

A.1 Evaluation Metrics

We report metrics to assess both geometric fidelity and program validity. Following standard evaluation protocols, instances where the geometric kernel fails to compile a valid STEP file are excluded from the geometric metric computation. Meanwhile, we generate 3,000 valid models, randomly sample 1,000 instances from the test set 10 separate times, and report the averaged results. Specifically, we employ the following three metrics:

Invalid Ratio (IR) IR measures the syntactical and structural validity of the generated CadQuery programs. It is defined as the percentage of generated programs that crash during execution or fail to produce a valid CAD solid via the CAD kernel:

$$\mathrm{IR} = \frac{N_{\mathrm{invalid}}}{N_{\mathrm{total}}} \times 100\%, \tag{8}$$

where N_{invalid} is the number of execution failures and N_{total} is the total number of generated programs. A lower IR indicates better structural and grammatical validity.

Intersection over Union (IoU) To assess the global volumetric agreement between the generated and ground-truth solids, we convert both CAD models into voxel grids (e.g., 64^3 resolution). The volumetric IoU is calculated as:

$$\mathrm{IoU} = \frac{|V_{\mathrm{pred}} \cap V_{\mathrm{gt}}|}{|V_{\mathrm{pred}} \cup V_{\mathrm{gt}}|}, \tag{9}$$

where V_{pred} and V_{gt} represent the binary voxel sets of the predicted and ground-truth solids, respectively.

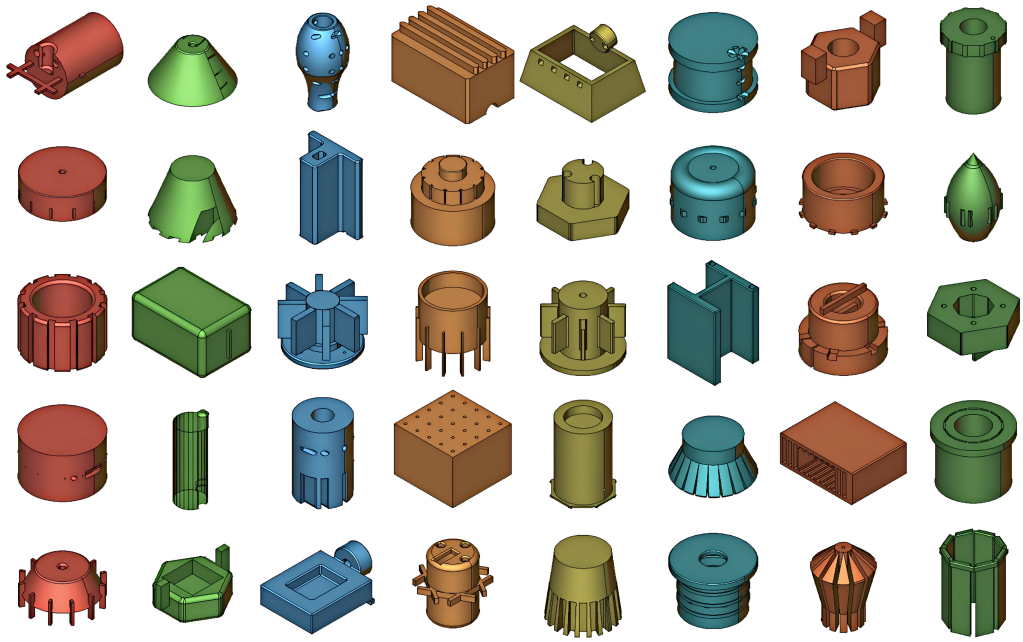


Figure 5: Representative CAD models from TopoCAD-1M across different BRep face-count tiers. The examples illustrate the geometric and structural diversity of the corpus, including holes, fillets, shells, patterns, and rotational features.

Chamfer Distance (CD) To evaluate fine-grained surface accuracy, we uniformly sample point clouds from the surfaces of the generated solid (P) and the ground-truth solid (Q). The symmetric Chamfer Distance is defined as:

$$d_{\mathrm{CD}}(P,Q) = \frac{1}{|P|}\sum_{p\in P}\min_{q\in Q}\|p-q\|_2^2 + \frac{1}{|Q|}\sum_{q\in Q}\min_{p\in P}\|q-p\|_2^2. \tag{10}$$

A lower CD score reflects tighter boundary alignment and better preservation of local geometric details.

A.2 TopoCAD-1M

Figure 5 presents representative models from **TopoCAD-1M**, covering prismatic, rotational, thin-walled, patterned, and compound-feature parts across different complexity tiers. Each model is paired with its executable CadQuery program, BRep geometry, multimodal observations, and extracted constraint graph. This structural diversity supports learning beyond simple extrusions and improves the coverage of design-relevant operations and relations.

A.3 Limitations

While TopoGen effectively bridges the gap between visual observations and CAD structures, certain areas remain open for future refinement. The current framework relies on a predefined constraint vocabulary and rule-based essential-operation checks. Although robust for standard CAD families, this approach may not explicitly capture every high-level functional intent or seamlessly generalize to highly unconventional part categories. Furthermore, incorporating CAD kernel execution directly into the training loop introduces computational overhead that inherently bounds candidate sampling during ToRPO, occasionally yielding sparse reward signals when ungrammatical programs fail to compile. Finally, extending this structural reasoning pipeline to handle the inherent complexities of real-world scans and photographs—such as sensor noise, severe occlusions, and illumination shifts—presents a natural and exciting direction for future research.

A.4 Robust Extraction of Structural CAD Constraints

Setup and notation. Let the B-rep be $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$. The primitive types include vertices v with point $\mathbf{p}_v$; edges e_{ij} with endpoints $\mathbf{p}_i, \mathbf{p}_j$, unit direction $\mathbf{d}_{ij} = (\mathbf{p}_j - \mathbf{p}_i)/\|\mathbf{p}_j - \mathbf{p}_i\|_2$, and length $l_{ij} = \|\mathbf{p}_j - \mathbf{p}_i\|_2$; planar faces $\mathbf{f}_{ijk}$ with unit normal $\mathbf{n}_{ijk}$ and point $\mathbf{p}_{ijk}$; cylindrical faces $\mathbf{c}$ with axis point $\mathbf{p}_\mathbf{c}$, unit axis direction $\mathbf{d}_\mathbf{c}$, and radius $r_\mathbf{c}$; conical faces $\mathbf{k}$ with apex point $\mathbf{p}_\mathbf{k}$, unit axis direction $\mathbf{d}_\mathbf{k}$, and half-angle $\alpha_\mathbf{k}$; toroidal faces $\mathbf{t}$ with center point $\mathbf{p}_\mathbf{t}$, unit axis direction $\mathbf{d}_\mathbf{t}$, major radius $R_\mathbf{t}$, and minor radius $r_\mathbf{t}$; and spherical faces $\mathbf{s}$ with center point $\mathbf{p}_\mathbf{s}$ and radius $r_\mathbf{s}$. Here $\mathbf{s}$ is a face symbol and should not be confused with the scalar s. Axes $a_\mathbf{c}$, $a_\mathbf{k}$, and $a_\mathbf{t}$ are associated with $\mathbf{c}$, $\mathbf{k}$, and $\mathbf{t}$, respectively, and inherit their point, direction, and shape parameters; spherical faces have no axis.

For a primitive i, let $\mathbf{d}_i$ denote its direction if defined, $\mathbf{n}_i$ its normal if it is planar, $\mathbf{p}_i$ a representative point, l_i its length if defined, r_i its radius if defined, α_i its half-angle if conical, and R_i its major radius if toroidal. Let $\mathcal{F}_p \subseteq \mathcal{F}$ be the set of planar faces, and let $\mathcal{L}$ and $\mathcal{R}$ denote the sets of primitives with defined length and radius, respectively. This notation follows Table 1. Parameters are extracted directly from the B-rep model.

The edge notation e is shorthand for a generic e_{ij}; thus $\mathbf{p}_e^0 = \mathbf{p}_i$, $\mathbf{p}_e^1 = \mathbf{p}_j$, $\mathbf{d}_e = \mathbf{d}_{ij}$, and $l_e = l_{ij}$ when $e = e_{ij}$. Similarly, $\mathbf{f}$ denotes a generic face, including a planar face $\mathbf{f}_{ijk}$ when applicable, with $\mathbf{n}_\mathbf{f} = \mathbf{n}_{ijk}$ and $\mathbf{p}_\mathbf{f} = \mathbf{p}_{ijk}$ when $\mathbf{f} = \mathbf{f}_{ijk}$.

For an edge e, let $\mathbf{p}_e^0, \mathbf{p}_e^1$ be its endpoints, $\mathbf{u}_e = \mathbf{p}_e^1 - \mathbf{p}_e^0$ its edge vector, $l_e = \|\mathbf{u}_e\|_2$ its length, and $\mathbf{d}_e = \mathbf{u}_e/l_e$ its unit direction. We parameterize the edge as $\mathbf{p}_e(t) = \mathbf{p}_e^0 + t\mathbf{u}_e$, $t \in [0,1]$. For sampled parameters $\{t_k\} \subset [0,1]$, $\mathbf{p}_e(t_k)$ are the corresponding points on the edge. For a planar face $\mathbf{f}$, we write $\mathbf{n}_\mathbf{f}$ and $\mathbf{p}_\mathbf{f}$ for its unit normal and a representative point, respectively. For a general face $\mathbf{f}$, $\mathrm{Proj}_\mathbf{f}(\mathbf{p})$ denotes the closest-point projection of $\mathbf{p}$ onto $\mathbf{f}$, $\mathbf{q}_\mathbf{f}(\mathbf{p}) = \mathrm{Proj}_\mathbf{f}(\mathbf{p})$, and $\mathrm{dist}(\mathbf{p}, \mathbf{f}) = \|\mathbf{p} - \mathrm{Proj}_\mathbf{f}(\mathbf{p})\|_2$. The predicate $\mathrm{in}(\mathbf{f}, v)$ means that the projection of $\mathbf{p}_v$ lies in the bounded region of $\mathbf{f}$.

All coordinates are normalized by the bounding-box diagonal s. We use

$$\epsilon_a = \max(\epsilon_a^0, \kappa_a \sigma_a), \quad \epsilon_d = \max(\epsilon_d^0, \kappa_d \sigma_d), \quad \epsilon_p = \max(\epsilon_p^0, \kappa_p \sigma_p), \tag{11}$$

and, for an edge e of length l_e,

$$\epsilon_{t,e} = \epsilon_d / \max(l_e, \epsilon_d). \tag{12}$$

Here $\epsilon_a^0, \epsilon_d^0, \epsilon_p^0$ are base tolerances for angle, distance, and relative length/radius, respectively; $\kappa_a, \kappa_d, \kappa_p$ are noise scaling factors; and $\sigma_a, \sigma_d, \sigma_p$ are estimated standard deviations of angular, distance, and positional noise. The resulting $\epsilon_a, \epsilon_d, \epsilon_p$ are the angle, distance, and relative length/radius tolerances. For exact CAD B-rep, $\sigma_a = \sigma_d = \sigma_p = 0$.

Residuals and weights. For compatible primitive pairs, use

$$r_\parallel = \|\mathbf{d}_i \times \mathbf{d}_j\|_2, \tag{13}$$

$$r_\perp = |\mathbf{d}_i^\top \mathbf{d}_j|, \tag{14}$$

$$r_{\text{coplanar}} = \max\big(\|\mathbf{n}_i \times \mathbf{n}_j\|_2, |\mathbf{n}_i^\top (\mathbf{p}_i - \mathbf{p}_j)|\big), \tag{15}$$

$$r_{\text{coaxial}} = \max\big(\|\mathbf{d}_i \times \mathbf{d}_j\|_2, \|\mathbf{d}_i \times (\mathbf{p}_i - \mathbf{p}_j)\|_2\big), \tag{16}$$

$$r_{\text{length}} = \frac{|l_i - l_j|}{\max(l_i, l_j, \epsilon_p)}, \tag{17}$$

$$r_{\text{radius}} = \frac{|r_i - r_j|}{\max(r_i, r_j, \epsilon_p)}. \tag{18}$$

A constraint with residual r and tolerance ϵ receives confidence

$$w(r, \epsilon) = \exp\left(-\frac{r^2}{2\epsilon^2}\right) \mathbf{1}[r \leqslant \epsilon]. \tag{19}$$

Here $\mathbf{1}[\cdot]$ is the indicator function, equal to 1 if the condition holds and 0 otherwise. A constraint is represented as a tuple (τ, i, j, w, r), where τ is the constraint type label, i, j are primitive identifiers (written as $v, e, \mathbf{f}$ when referring to vertices, edges, and faces explicitly), w is the confidence weight, and r is the residual. This yields soft constraints rather than hard binary decisions.

Stage 1: pairwise structural constraints. We first extract pairwise relations among compatible primitives. Candidate pairs are generated by spatial hashing, kd-tree, and adjacency queries. For each candidate pair, we test parallelism, orthogonality, coplanarity, coaxiality, equal length, and equal radius, and store each accepted relation as a soft constraint. The procedure is summarized in Algorithm 1.

Algorithm 1 Pairwise structural CAD constraint extraction.

Require: $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, s, $\epsilon_a, \epsilon_d, \epsilon_p$, $w(r, \epsilon) = e^{-r^2/(2\epsilon^2)}\mathbf{1}[r \leqslant \epsilon]$
Ensure: $\mathcal{C}_{\text{pair}} = \{(\tau, i, j, w, r)\}$

$\mathcal{C}_{\text{pair}} \leftarrow \emptyset$
$\{\mathbf{d}_i, \mathbf{n}_i, \mathbf{p}_i, l_i, r_i\} \leftarrow \text{Prim}(\mathcal{M}, s)$
$\mathcal{P} \leftarrow \text{Cand}(\mathcal{M}; \text{SH}, \text{KD}, \text{Adj})$
for all $(i, j) \in \mathcal{P}$ **do**
  $r_{\parallel} \leftarrow \|\mathbf{d}_i \times \mathbf{d}_j\|_2$
  if $\text{Dir}(i) \wedge \text{Dir}(j) \wedge r_{\parallel} \leqslant \epsilon_a$ **then**
    $\mathcal{C}_{\text{pair}} \leftarrow \mathcal{C}_{\text{pair}} \cup \{(\text{Parallel}, i, j, w(r_{\parallel}, \epsilon_a), r_{\parallel})\}$
  end if
  $r_{\perp} \leftarrow |\mathbf{d}_i^{\top} \mathbf{d}_j|$
  if $\text{Dir}(i) \wedge \text{Dir}(j) \wedge r_{\perp} \leqslant \epsilon_a$ **then**
    $\mathcal{C}_{\text{pair}} \leftarrow \mathcal{C}_{\text{pair}} \cup \{(\text{Orthogonal}, i, j, w(r_{\perp}, \epsilon_a), r_{\perp})\}$
  end if
  $r_{\text{coplanar}} \leftarrow \max(\|\mathbf{n}_i \times \mathbf{n}_j\|_2, |\mathbf{n}_i^{\top}(\mathbf{p}_i - \mathbf{p}_j)|)$
  if $\text{Planar}(i) \wedge \text{Planar}(j) \wedge r_{\text{coplanar}} \leqslant \max(\epsilon_a, \epsilon_d)$ **then**
    $\mathcal{C}_{\text{pair}} \leftarrow \mathcal{C}_{\text{pair}} \cup \{(\text{Coplanar}, i, j, w(r_{\text{coplanar}}, \max(\epsilon_a, \epsilon_d)), r_{\text{coplanar}})\}$
  end if
  $r_{\text{coaxial}} \leftarrow \max(\|\mathbf{d}_i \times \mathbf{d}_j\|_2, \|\mathbf{d}_i \times (\mathbf{p}_i - \mathbf{p}_j)\|_2)$
  if $\text{Axis}(i) \wedge \text{Axis}(j) \wedge r_{\text{coaxial}} \leqslant \max(\epsilon_a, \epsilon_d)$ **then**
    $\mathcal{C}_{\text{pair}} \leftarrow \mathcal{C}_{\text{pair}} \cup \{(\text{Coaxial}, i, j, w(r_{\text{coaxial}}, \max(\epsilon_a, \epsilon_d)), r_{\text{coaxial}})\}$
  end if
  $r_{\text{length}} \leftarrow |l_i - l_j| / \max(l_i, l_j, \epsilon_p)$
  if $i, j \in \mathcal{L} \wedge r_{\text{length}} \leqslant \epsilon_p$ **then**
    $\mathcal{C}_{\text{pair}} \leftarrow \mathcal{C}_{\text{pair}} \cup \{(\text{EqualLength}, i, j, w(r_{\text{length}}, \epsilon_p), r_{\text{length}})\}$
  end if
  $r_{\text{radius}} \leftarrow |r_i - r_j| / \max(r_i, r_j, \epsilon_p)$
  if $i, j \in \mathcal{R} \wedge r_{\text{radius}} \leqslant \epsilon_p$ **then**
    $\mathcal{C}_{\text{pair}} \leftarrow \mathcal{C}_{\text{pair}} \cup \{(\text{EqualRadius}, i, j, w(r_{\text{radius}}, \epsilon_p), r_{\text{radius}})\}$
  end if
end for
return $\mathcal{C}_{\text{pair}}$

Stage 2: incidence and projection constraints. We then extract incidence and projection relations between lower-dimensional and higher-dimensional primitives. This includes vertex–edge incidence, face–vertex incidence or projection, and edge–face incidence or projection. For each edge, the parameter tolerance is scaled by its length. The final constraint set is post-processed by weight thresholding with threshold $w_{\min} \in [0, 1]$, retaining only constraints with $w \geqslant w_{\min}$; this operation is denoted by $\mathrm{Post}_{w_{\min}}$. The sampled parameters $\{t_k\} \subset [0, 1]$ are used to discretely test whether an edge lies on a face. The procedure is summarized in Algorithm 2.

Algorithm 2 Incidence and projection CAD constraint extraction.

Require: $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, s, ϵ_a, ϵ_d, $w(r, \epsilon)$, $w_{\min}$, $\mathcal{C}_{\text{pair}}$
Ensure: $\mathcal{C} = \{(\tau, i, j, w, r)\}$

$\mathcal{C} \leftarrow \mathcal{C}_{\text{pair}}$
for all $(v, e) \in \mathcal{V} \times \mathcal{E}$ **do**
  $l_e \leftarrow \|\mathbf{u}_e\|_2$
  $\epsilon_{t,e} \leftarrow \epsilon_d / \max(l_e, \epsilon_d)$
  $t \leftarrow ((\mathbf{p}_v - \mathbf{p}_e^0)^\top \mathbf{u}_e)/(\mathbf{u}_e^\top \mathbf{u}_e)$
  $r_v \leftarrow \|\mathbf{p}_v - (\mathbf{p}_e^0 + t\mathbf{u}_e)\|_2$
  if $-\epsilon_{t,e} \leqslant t \leqslant 1 + \epsilon_{t,e} \wedge r_v \leqslant \epsilon_d$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathrm{VertexEdge}, v, e, w(r_v, \epsilon_d), r_v)\}$
  end if
end for
for all $(\mathbf{f}, v) \in \mathcal{F} \times \mathcal{V}$ **do**
  if $\mathbf{f} \in \mathcal{F}_p$ **then**
    $r_p \leftarrow |\mathbf{n}_\mathbf{f}^\top (\mathbf{p}_v - \mathbf{p}_\mathbf{f})|$
    if $r_p \leqslant \epsilon_d \wedge \mathrm{in}(\mathbf{f}, v)$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathrm{FaceVertex}, \mathbf{f}, v, w(r_p, \epsilon_d), r_p)\}$
    end if
  end if
  $\mathbf{q}_\mathbf{f}(\mathbf{p}_v) \leftarrow \mathrm{Proj}_\mathbf{f}(\mathbf{p}_v)$
  if $\mathbf{f} \notin \mathcal{F}_p \wedge \|\mathbf{q}_\mathbf{f}(\mathbf{p}_v) - \mathbf{p}_v\|_2 \leqslant \epsilon_d$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathrm{FaceVertex}, \mathbf{f}, v, w(\|\mathbf{q}_\mathbf{f}(\mathbf{p}_v) - \mathbf{p}_v\|_2, \epsilon_d), \|\mathbf{q}_\mathbf{f}(\mathbf{p}_v) - \mathbf{p}_v\|_2)\}$
  end if
end for
for all $(e, \mathbf{f}) \in \mathcal{E} \times \mathcal{F}$ **do**
  $\{t_k\} \subset [0, 1]$
  if $\mathbf{f} \in \mathcal{F}_p$ **then**
    $r_a \leftarrow |\mathbf{n}_\mathbf{f}^\top \mathbf{d}_e|$
    $r_d \leftarrow \max_k \mathrm{dist}(\mathbf{p}_e(t_k), \mathbf{f})$
    if $r_a \leqslant \epsilon_a \wedge \{\mathbf{p}_e(t_k)\} \subset \mathbf{f} \wedge r_d \leqslant \epsilon_d$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathrm{EdgeFace}, e, \mathbf{f}, w(r_d, \epsilon_d), r_d)\}$
    end if
  end if
  $r_s \leftarrow \max_k \|\mathbf{q}_\mathbf{f}(\mathbf{p}_e(t_k)) - \mathbf{p}_e(t_k)\|_2$
  if $\mathbf{f} \notin \mathcal{F}_p \wedge \{\mathbf{p}_e(t_k)\} \subset \mathbf{f} \wedge r_s \leqslant \epsilon_d$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathrm{EdgeFace}, e, \mathbf{f}, w(r_s, \epsilon_d), r_s)\}$
  end if
end for
$\mathcal{C} \leftarrow \mathrm{Post}_{w_{\min}}(\mathcal{C})$
return $\mathcal{C}$

A.5 More Experimental Results

Figures 6 and 7 present additional qualitative results for image- and point-cloud-conditioned CAD reconstruction, respectively. Across diverse part geometries and structural complexities, TopoGen generates executable CadQuery programs whose reconstructed BReps preserve both the overall geometry and fine-grained design features of the ground-truth models.

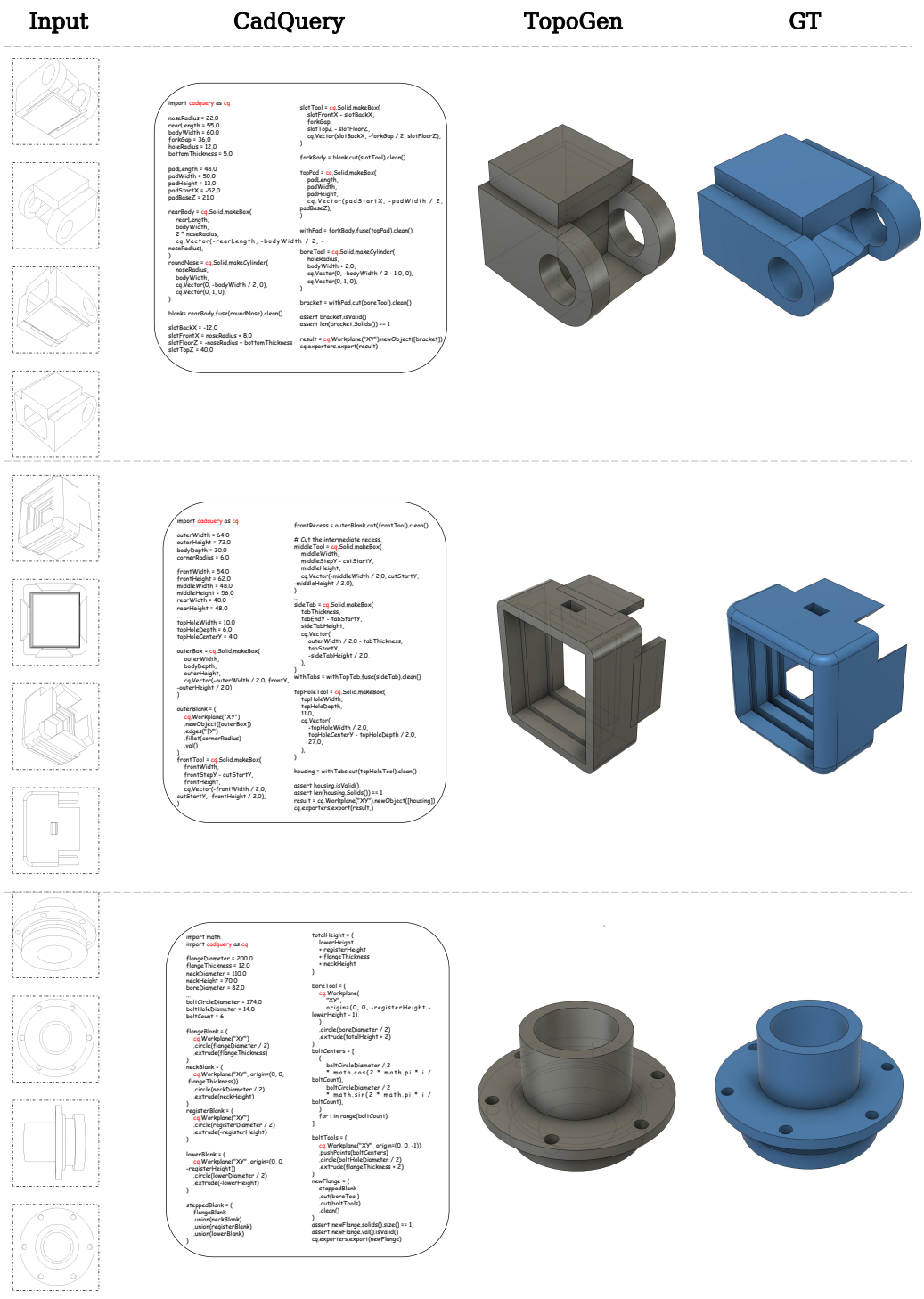


Figure 6: Additional qualitative results for multi-view image-conditioned CAD reconstruction. From left to right, each example shows the input images, generated CadQuery program, reconstructed BRep, and ground-truth CAD model.



Figure 7: Additional qualitative results for point-cloud-conditioned CAD reconstruction. From left to right, each example shows the input point cloud, generated CadQuery program, reconstructed BRep, and ground-truth CAD model.