GEOMCAD: A BENCHMARK FOR PRECISION RECONSTRUCTION OF GENERATIVE CAD OPERATIONS FROM POINT CLOUDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Point-cloud-to-CAD models are judged by whether their programs execute and resemble the target shape, yet a CAD program is only useful if its parameters are right. We show that these criteria diverge sharply: models whose programs execute on over 90% of inputs recover the exact parameter set for at most 2.7% of them. To measure this gap, we introduce **GeomCAD**, a benchmark of 4 million executable CadQuery programs, four times larger than any prior program-paired corpus and balanced across *extrude*, *revolve*, *sweep*, and *loft*, with every parameter an integer on a $1{,}000$-level grid so that exact parameter match is well defined; 36% of its faces are neither planar nor cylindrical, versus at most 12% in prior program-paired corpora. Its protocol **GeomCAD-Eval** scores exact parameter recovery alongside shape and validity. Our reference model **CAD-Exact**, trained on GeomCAD, outperforms all zero-shot and fine-tuned baselines on every operation, recovering 27–49% of ground-truth parameter values (best baseline: 13–40%) and matching the target volume within 5% on 49–95% of inputs (best baseline: 36–84%). The corpus, evaluation code, and all checkpoints are released.[1]

1 INTRODUCTION

Computer-aided design (CAD) programs preserve dimensions, tolerances, and constructive intent (Shapiro, 2002). Unlike meshes or voxel grids, they retain the sketches and modeling operations that produced a shape, so designs can be re-executed and modified through parameter changes. Reconstructing executable CAD programs from 3D scans is therefore valuable for reverse engineering, design reuse, and digital-twin construction (Willis et al., 2021; Wu et al., 2021; Várady et al., 1997).

Recent CAD reconstruction methods predict extrusion parameters (Uy et al., 2022; Li et al., 2023; Ren et al., 2022; Sharma et al., 2020; Wang et al., 2020; Li et al., 2024; Lambourne et al., 2022), autoregressive sketch sequences (Wu et al., 2021; Khan et al., 2024a; Yang et al., 2025; Xu et al., 2022; 2023; Dupont et al., 2024; Ma et al., 2024), or executable Python via point-conditioned LLMs (Rukhovich et al., 2025; Kolodiazhnyi et al., 2026), but evaluation has converged on shape-similarity metrics (Chamfer Distance, volumetric IoU) that are insufficient for program-level reconstruction: visually similar outputs may correspond to substantially different CAD parameters (Figure 1). We show that the gap is large: models whose programs execute on over 90% of inputs reproduce the full parameter set exactly for at most 2.7% of them, even while recovering up to 55% of individual values (Table 3).

This limitation is compounded by existing benchmarks. Most program-paired datasets remain extrude-centric (Willis et al., 2021; Wu et al., 2021; Rukhovich et al., 2025). At the surface level, the corpus behind the strongest current systems contains exactly two surface families, planes and cylinders (§3), and parameters are typically floating-point or coarsely discretized, preventing exact parameter-level comparison and biasing evaluation toward aggregate geometric similarity.

To address these limitations, we introduce **GeomCAD**, a procedural benchmark of 4 million (M)[2] executable CadQuery programs, four times larger than any prior program-paired corpus and balanced across four generative operations (extrude, revolve, sweep, loft) at $\sim$1 M each. Every numerical literal

[1] Code: `https://anonymous.4open.science/r/CAD-Exact-5707/README.md`.

[2] Uppercase "M" denotes million; lowercase "m" denotes the meter unit.

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.

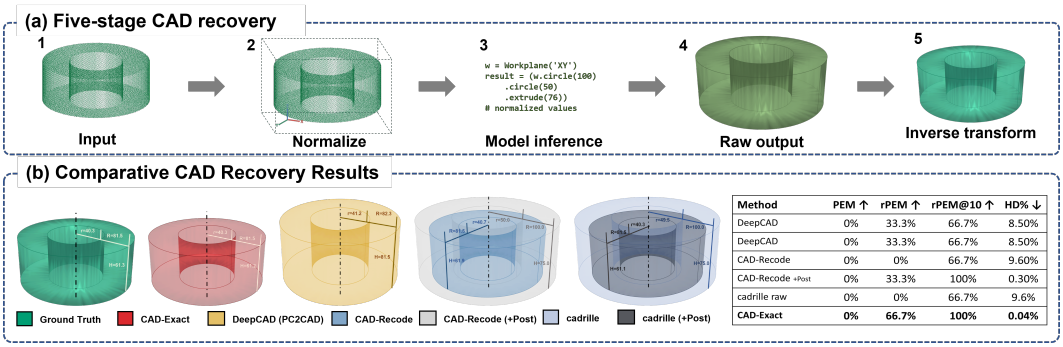


Figure 1: **No method achieves parameter-exact reconstruction** even on a simple hollow cylinder. Raw baselines break in two ways, uniform frame-scale error (all of H, R, r off together) or thickness-only error (only H off), both above 20%. Per-prediction bounding-box canonicalization (+*post*, §4) closes the shape gap but leaves 0.2–0.6% relative residuals on H, R, r (about two to six steps on the 1000-level integer grid). CAD-Exact (solid), our reference model, still differs from GT (dashed) by $\sim 0.1\%$ on H, i.e., about one grid step (three significant digits). This precision gap is invisible to shape-similarity metrics and is exactly what GeomCAD-Eval measures via PEM, rPEM, and rPEM@10 (§3.5).

is an integer on a 1,000-level grid (§3.2), so exact parameter match against the ground-truth program is well defined. The corpus is procedural by design, as is CAD-Recode's (Rukhovich et al., 2025): human-authored collections are small (DeepCAD 0.18 M, Fusion 360 8.6 K) and almost entirely extrude-based, and none offers a uniform integer parameterization. Like all point-cloud-to-CAD benchmarks, GeomCAD samples its point clouds from ideal CAD surfaces. We further present **GeomCAD-Eval**, a three-tier protocol covering parameter precision, shape similarity, and program validity; because different programs can build the same solid, exact match is always reported beside graded and tolerance-based recovery (§3.5). Finally, we train a reference model, **CAD-Exact**, on GeomCAD (§4).

Our contributions are summarized as follows:

- **Exact parameter recovery made measurable:** an integer-canonical program representation on which Parameter Exact Match (PEM) is strictly defined, and GeomCAD-Eval, which pairs PEM with graded recovery (rPEM, rPEM@10), shape similarity, and validity with a failure-mode decomposition.
- **GeomCAD:** 4 M programs over four generative operations balanced at $\sim$1 M each, with a far richer surface mix than prior program-paired corpora: 36% of its faces are neither planar nor cylindrical, versus 0% in CAD-Recode (Rukhovich et al., 2025) and 12% in Zero-to-CAD (Ataei et al., 2026) (§3.4).
- **CAD-Exact:** a reference model that outperforms all zero-shot and fine-tuned baselines on every operation, recovering 27–49% of ground-truth parameter values (best baseline: 13–40%) and matching the target volume within 5% on 49–95% of inputs (best baseline: 36–84%; §5).

2 Related Work

Our work draws on program-paired CAD datasets, point-cloud-to-CAD reconstruction methods, and precision-oriented evaluation for 3D reconstruction.

2.1 Program-Paired CAD Datasets

Program-paired CAD datasets trade off scale, operation coverage, and parameter format (Table 1). Shape-only ABC (Koch et al., 2019) and ShapeNet (Chang et al., 2015) carry no programs. Fusion 360 Gallery (Willis et al., 2021), DeepCAD (Wu et al., 2021), and CAD-Recode (Rukhovich et al., 2025) provide executable programs but only for sketch-and-extrude. CC3D-Ops (Dupont et al., 2022)

Table 1: **Program-paired CAD datasets.** Scale, available data, generative operations, parameter format, and program representation across 8 corpora. Surface-composition details in Table 2.

Dataset	#Samples	Available data	Generative ops	Param. type	Program
ABC (Koch et al., 2019)	1 M	pc / mesh / B-Rep	B-Rep only	n/a	–
Fusion 360 (Willis et al., 2021)	8.6 K	B-Rep / param. seq.	sketch+extrude	float	Fusion 360 JSON
DeepCAD (Wu et al., 2021)	0.18 M	param. seq.	sketch+extrude	256-level int (8-bit)	custom op seq
CC3D-Ops (Dupont et al., 2022)	37 K+	B-Rep + op labels	multi-op (labels only)	n/a	–
CAD-Recode (Rukhovich et al., 2025)	1 M	pc / mesh / B-Rep / prog	extrude-only	int $[-100, 100]$	CadQuery
VideoCAD (Man et al., 2025)	41 K	video / UI / image	extrude (UI)	float	UI actions
Zero-to-CAD (Ataei et al., 2026)	1 M	image / mesh / prog	multi-op (fillet/chamfer/shell)	float symbolic	CadQuery
GeomCAD (ours)	**4 M**	**pc / mesh / B-Rep / prog**	**4 ops** $\times$ **$\sim$1 M each**	**int** $[1, 1000]$	**CadQuery**

annotates per-face operations without executable programs. Multi-modal extensions add text, image, and video conditioning (Kolodiazhnyi et al., 2026; Khan et al., 2024b; Chen et al., 2025; Man et al., 2025); sketch-level datasets cover 2D (Seff et al., 2020; 2022; Para et al., 2021). The concurrent Zero-to-CAD (Ataei et al., 2026) adds fillet, chamfer, and shell operations but uses symbolic float parameters, so exact-match evaluation is not defined on it. Across these corpora, no existing program-paired dataset jointly provides integer literals with PEM-definable evaluation, multi-op coverage at $\sim$1 M samples each, and audited train/val/test splits with release-gated distribution matching. GeomCAD jointly satisfies these three axes (Table 1), with all four splits passing distribution matching on the release-gate axes (Appendix C.5); surface-composition details are in Table 2.

2.2 Point Cloud to CAD Reconstruction

CAD reconstruction methods differ in what they output. Some predict extrusion parameters (Uy et al., 2022; Li et al., 2023; Ren et al., 2022); SfmCAD (Li et al., 2024) extends this to four operations. Others fit primitives, boundaries (Sharma et al., 2020; Wang et al., 2020; Schnabel et al., 2007; Guo et al., 2022; Xu et al., 2024; Jayaraman et al., 2024; 2021; Lambourne et al., 2021; 2022; Dupont et al., 2022) or CSG trees (Sharma et al., 2018; Yu et al., 2022; 2023), producing geometry or topology without runnable code. Construction-sequence generators (Wu et al., 2021; Khan et al., 2024a; Yang et al., 2025; Dupont et al., 2024; Ma et al., 2024; Xu et al., 2022; 2023; Nash et al., 2020; Jones et al., 2020; Li et al., 2022) output executable CAD but train mostly on extrude data and evaluate with shape similarity. The newest group uses point-conditioned LLMs to emit CadQuery (Contributors, 2026): CAD-Recode (Rukhovich et al., 2025) and cadrille (Kolodiazhnyi et al., 2026) (with online RL) still evaluate on extrude-heavy data. We contribute a benchmark and evaluation code, not a new architecture, and measure how well models handle multiple operations and recover exact parameters.

2.3 Precision-Oriented Evaluation for 3D Reconstruction

3D reconstruction commonly reports F-score under relative distance thresholds (Tatarchenko et al., 2019; Knapitsch et al., 2017), which compares geometry rather than program parameters. The closest program-level metrics are the command and parameter accuracies of DeepCAD (Wu et al., 2021), but they address a different task. DeepCAD evaluates an autoencoder: the input is the ground-truth command sequence itself, so the reconstruction stays aligned with it, and parameters are compared position by position for commands whose type is recovered, within a tolerance of 2 out of 256 levels. In that setting, parameter accuracy exceeds 97% for every variant reported. Reconstruction from a point cloud has no such alignment: the model sees only geometry, and its program may differ from the ground truth in length, order, and operations. GeomCAD-Eval (§3.5) is designed for this case. It reports exact match (PEM) together with order-free graded recovery (rPEM, rPEM@10), whose one-to-one matching and ground-truth denominator remain meaningful when the predicted structure differs from the ground truth.

3 The GeomCAD Benchmark

GeomCAD is a benchmark for point-cloud-to-CAD reconstruction. The corpus contains $4{,}006{,}182$ executable CadQuery programs, balanced at $\sim$1 M samples per operation across four generative operations (*extrude*, *revolve*, *sweep*, *loft*), with every numerical literal stored as an integer so that Parameter Exact Match (PEM) against a ground-truth program is well defined (Figure 2). Its evaluation protocol **GeomCAD-Eval** scores predictions along precision-aware, shape-similarity, and program-validity tiers (§3.5).

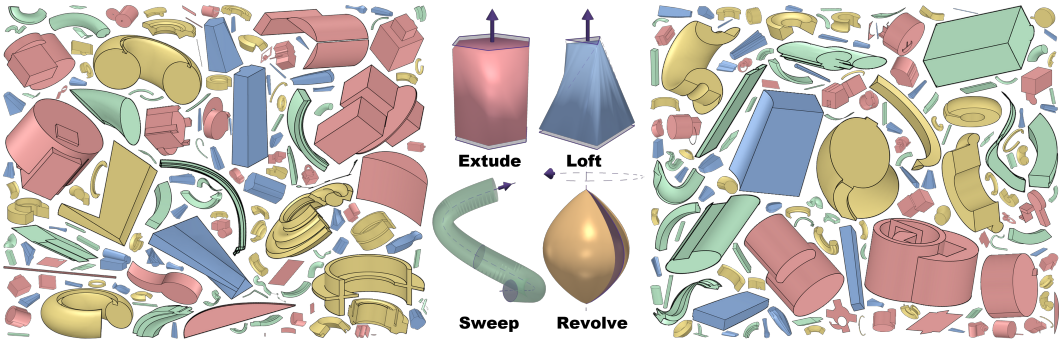


Figure 2: **GeomCAD overview**: 280 parts sampled from the four operations (extrude, loft, sweep, revolve); the four parametric icons at the centre define each operation's construction primitive.

3.1 Procedural Program Generation

Extrude programs are drawn from two existing corpora, CAD-Recode v1.5 ($\sim$1 M sketch-and-extrude programs) and SketchGraphs (Seff et al., 2020) ($\sim$15 M constrained 2D sketches), which supply 73.7% and 26.3% of the extrude split; SketchGraphs sketches are lifted to 3D with a fixed workplane-plus-depth template. The rewrite keeps each construction graph and moves every literal onto GeomCAD's integer grid (§3.2), which is $5\times$ finer than CAD-Recode's $[-100, 100]$ grid (Table 1).

Revolve, sweep, and loft have no comparable program-paired corpus. Each uses a procedural template that samples the 3D parameters the sketch lacks (rotation axis and angle, sweep path, ordered sections) and applies them to a sketch profile drawn from CAD-Recode v1.5, the source of most extrude programs. Because the four operations largely share this sketch source, the surface-composition differences in §3.4 mainly reflect the operation itself.

All four pipelines pass through a shared validity gate (OCCT execution, watertight solid, minimum face count; per-operation pass rates in Appendix B).

3.2 Integer-Parameter Representation

The generation pipeline maps every numerical literal (dimensions, coordinates, radii, angles, section counts) to an integer on the 1,000-level grid $[1, 1000]$ (Appendix B). The only exception is the value 0, kept verbatim where a construction requires it: revolve start angles, sweep translations that coincide with the origin, and loft sections anchored at the workplane origin ($\sim$0.5% of literals). Extrude never uses 0, since zero thickness or radius yields a degenerate solid. An audit of all 4,006,182 programs confirms that every literal is an integer.

We use 1,000 levels for two reasons. A grid step is a fixed fraction of the part's extent: CAD-Recode's $[-100, 100]$ grid gives 0.5% steps, i.e. 0.5 mm on a typical 100 mm part, while our 1,000-level grid gives 0.1%, i.e. 0.1 mm (Appendix A). Under the Qwen2 BPE tokenizer, which splits numbers into single digits, every grid value takes at most four sub-tokens (three below 1,000), whereas a 10^4-level grid would need up to five per literal with no precision gain above the noise floor of our metrics (Appendix A).

Coordinate frames differ between the model's encoder and the evaluator. The *canonical cube* $[0, 1]^3 \subset \mathbb{R}^3$ is the model's encoder input frame: an invertible affine transform (§4) maps each input point cloud into it, and the inverse transform returns predictions to the raw mesh's own coordinates. Program literals live on a separate $[1, 1000]$ integer grid (§3.2); executing a program yields a mesh in grid units, which a factor of 10^{-3} maps into the canonical cube, so one grid step is 10^{-3} of the part's largest bounding-box extent, i.e. three significant digits regardless of absolute size. The *evaluation frame* is used only for scoring (§3.5): meshes are centered and scaled so that the GT bounding-box diagonal equals 1,000. By N-level we mean N distinct integer values.

Table 2: **Surface-type composition (%), full corpora under one OCCT kernel**. ABC (Koch et al., 2019) included as a not-program-paired reference for kernel-level calibration (B-rep/STEP source).

Surface type (%)	GeomCAD	CAD-Recode	Zero-to-CAD	ABC
plane	55.08	84.00	68.94	54.61
cylinder	8.89	16.00	18.85	24.14
B-spline	24.85	0.00	4.33	8.23
revolution	10.67	0.00	0.12	0.76
beyond plane + cylinder	36.03	0.00	12.21	21.25
B-spline + revolution	35.52	0.00	4.45	8.99
scale (solids / faces / median faces per solid)	4.01 M / 42.4 M / 9	0.98 M / 16.8 M / 16	0.98 M / 43.5 M / 30	1.00 M / 187.0 M / 187

3.3 Splits and Release

Each operation's $\sim$1 M pool is split into train and held-out validation from disjoint seeds. A separate 1,500-sample test split per operation is generated independently of the pool by the same pipeline (counts in Table 5). Deduplication by exact program text removes 0.037% of training rows, including all overlaps with held-out splits (Appendix B). Each release split matches its pool in construct frequency, triangle count, and numeric scale (Appendix C.5).

The corpus is released as a 77 GB bundle with programs and meshes, and as a 271 MB programs-only bundle from which the meshes regenerate deterministically (Appendix C.6). We also release the generator scripts, split-audit tools, per-sample quality flags, and the weights of every evaluated checkpoint.

3.4 Full-Corpus Geometry Statistics

Surface types that are absent from a training corpus give a model no in-distribution signal for them. We therefore classify all 102.7 M faces of GeomCAD, CAD-Recode, and Zero-to-CAD with a single OCCT kernel (Table 2). 36.03% of GeomCAD faces are neither planar nor cylindrical, compared with 0% for CAD-Recode and 12.21% for Zero-to-CAD.

Per-operation surface breakdowns and per-sample quality flags are in Appendix C.

3.5 Evaluation Protocol (GeomCAD-Eval)

GeomCAD-Eval scores predictions in three tiers: Tier-1 precision-aware fidelity, Tier-2 shape similarity, and Tier-3 program validity. It also splits invalid predictions into five failure modes. All twelve main-table metrics (six Tier-1, five Tier-2, one Tier-3) are reported as percentages.

Each sample i has a predicted program $\hat{y}^i$ and a GT program y^i. Executing each through CadQuery's OCCT backend gives a solid with volume $\hat{V}^i, V^i$ and a surface point cloud $\hat{S}^i, S^i$ in the evaluation frame. Both clouds use the GT's centering and scale, so scale and offset errors in the prediction are preserved (baselines are the exception; see §4). $\hat{\boldsymbol{\theta}}^i$ and $\boldsymbol{\theta}^i$ are the integer literals of each program in textual order, clamped to $\{0\} \cup [1, 1000]$. $\mathbb{1}[\cdot]$ is the indicator function, and $\mathcal{V}$ is the set of valid predictions (Tier-3). Full definitions are in Appendix E.

Tier-1 (precision-aware fidelity) measures whether the predicted program recovers the ground-truth values. We report six metrics on $\mathcal{V}$. ***TVF***$@\delta$ is the fraction with relative volume error $\leq \delta$, for $\delta \in \{1\%, 5\%\}$. ***PEM*** $= \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbb{1}[\hat{\boldsymbol{\theta}}^i = \boldsymbol{\theta}^i]$ is the fraction whose literal sequence matches GT exactly, with the same length and the same value at every position; no reordering is applied. Two graded variants complement it: ***rPEM*** is the multiset recall of GT literals, and ***rPEM@10*** is the same recall with a tolerance of 10 grid levels ($\approx 1\%$ of the grid span). Both use one-to-one matching over the full sequences, so missing literals count against the prediction. ***MFM***$_{\text{op}}$ requires the counts of the four generative operations to match GT.

PEM assumes that the GT program is identifiable from its shape. This holds for extrude, revolve, and sweep, but not for loft, where different section parametrizations can produce the same solid (§5). The gap $\text{rPEM} - \text{PEM}$ measures predictions that recover some values but not the whole program, including correct values placed at the wrong positions.

Tier-2 (shape similarity) reports five standard shape-similarity metrics: **IoU**; bidirectional **Chamfer Distance** (CD, % of GT bbox diagonal); **F-score** at $\tau = 2\%$ of the diagonal (Tatarchenko et al., 2019;

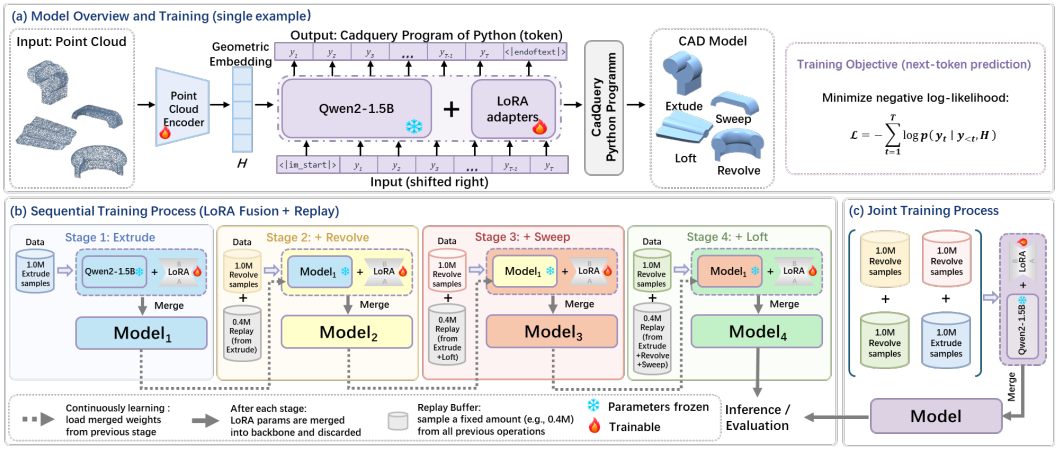


Figure 3: **CAD-Exact (a)** Point-cloud encoder feeds embeddings to `Qwen2-1.5B` with LoRA (base LM frozen). **(b)** Two budget-matched regimes: joint over the pooled corpus vs. a sequential chain with replay, each stage's LoRA merged before the next.

Knapitsch et al., 2017); bidirectional **Hausdorff Distance** (HD, % of diagonal); and **Volumetric-Gated F-score** VGF@5%, i.e. F@2% set to zero when the relative volume error exceeds 5% or the prediction is invalid. IoU, CD, F@2%, and HD are computed on valid predictions only, whereas VGF covers the full split, so the gap between VGF and F@2% shows how much failure the conditional metrics leave out. Prior protocols (Rukhovich et al., 2025; Wu et al., 2021; Kolodiazhnyi et al., 2026) report the same kinds of shape metrics plus a validity rate; differences in normalization and aggregation are documented in Appendix E. We include Tier-2 for comparison with this prior work and use Tier-1 for the main comparison.

Tier-3 (program validity) checks whether each prediction executes. The harness executes each predicted program under a 30-second CadQuery timeout, within a 180-second per-input limit that also covers generation (Appendix E.3), and writes an STL; a prediction is *valid* if the STL loads as a non-empty triangle mesh. **Validity** is the valid fraction of the full split. Invalid predictions are assigned to one of five failure modes: syntax error, API misuse, bad argument, kernel rejection, or empty output. All reported numbers use unconstrained decoding (no syntax or API filter on the token distribution), as in prior work.

Aggregation uses the mean for metrics on the valid set, except CD% and HD%, which are reported as medians because their per-sample distributions are heavy-tailed. VGF@5% and Validity are averaged over the full split.

Reproducibility is supported by fixing the OCCT kernel, surface sampling (65,536 points per mesh, seeded per sample), literal extraction and matching, tolerance thresholds, and the containerized harness in the release. We chose 65,536 points because the Hausdorff distance between two independent samplings of the same mesh drops from 2.2% of the diagonal at 8,192 points to $\approx 0.9\%$ at 65,536. HD% differences below this level are sampling noise, while the 2% F-score threshold lies above it. This scoring density is separate from the 8,192-point input cloud of the model encoder (§4).

4 CAD-Exact

CAD-Exact keeps the CAD-Recode v1.5 architecture (Rukhovich et al., 2025) unchanged (Figure 3). A point-cloud encoder farthest-point-samples n=256 points from an 8,192-point surface cloud, projects them into geometric embeddings that enter `Qwen2-1.5B` (Yang et al., 2024) as soft prompts, and the language model, adapted with LoRA (Hu et al., 2022) at rank r=64, autoregressively decodes a CadQuery program. It differs from CAD-Recode in training corpus (GeomCAD's four operations vs. extrude only) and parameter encoding (integer literals on GeomCAD's 1,000-level grid, 5× finer than CAD-Recode's $[-100, 100]$).

Two training arms start from scratch with identical per-step hyperparameters and the same total step budget. The *joint* arm pools all four operations for $187{,}500$ steps and serves as our main reference. The *staged* arm chains four $46{,}875$-step stages in the order extrude $\rightarrow$ revolve $\rightarrow$ sweep $\rightarrow$ loft, replaying 0.4 M samples from earlier operations at each stage and merging each stage's LoRA into the backbone before the next. Section 5 compares them; the chain's main advantage is extension efficiency (a fifth operation costs one further stage vs. a full retrain). We treat the chain as a fine-tuning schedule and do not compare with continual-learning methods (§6). Full hyperparameters in Appendix D.

Symmetric inference uses an invertible affine normalization: the input point cloud is translated and scaled into the canonical cube $[0, 1]^3$ (§3.2), and the inverse transform maps predictions back to the input frame. Because program literals live on a $[1, 1000]$ integer grid whose executed mesh is rescaled by 10^{-3} into the canonical cube, one grid step corresponds to one thousandth of the part's largest bounding-box extent, so accuracy scales with part size rather than being fixed in absolute units (Appendix D).

Baselines in Table 3 are CAD-Recode v1.5 (Rukhovich et al., 2025) and cadrille (GRPO/RL) (Kolodiazhnyi et al., 2026), both zero-shot on our splits, plus a cadrille fine-tuned on GeomCAD (LoRA r=64, 20k steps; a 3k budget in Appendix F.3). DeepCAD (Wu et al., 2021) is separately reported on its own test split (§5.4) since its sequence vocabulary makes on-grid PEM/rPEM undefined. Each method emits *one* program per input (*best-of-1*); cadrille's published protocol selects among multiple candidates, so best-of-1 is stricter than what it reports for itself.

Baseline scoring. For shape metrics, each zero-shot baseline prediction is canonicalized by its own bounding box (centered, diagonal rescaled to $1{,}000$)—rows marked *+post*. This approximates CAD-Recode's "+post" variant, whose exact inverse the baselines' APIs do not expose, and removes any global scale or translation mismatch, whereas CAD-Exact is scored in the GT-normalized frame where such errors count against it; the asymmetry favors the baselines. For PEM, rPEM, and rPEM@10, off-grid literals (CAD-Recode's $[-100, 100]$, cadrille's continuous values) are min-max rescaled per script into $[1, 1000]$ and rounded; this remapping can turn correct values into wrong integers and vice versa, so baseline PEM/rPEM are for reference only. The fine-tuned cadrille is scored natively (no *+post*, no remapping): it is already trained on our grid and frame.

5 Results

5.1 Evaluation Setup

Nine systems are scored on 1,500-sample per-operation splits (loft: 1,744) under one pipeline (180 s timeout, best-of-1; $65{,}536$-point scoring sample per mesh, seeded). Baselines predict in their own normalized frames and shape-similarity metrics are reported after per-prediction bounding-box canonicalization (*+post*, §4); CAD-Exact emits the corpus's integers directly and is scored in the GT-normalized evaluation frame. *+post* lifts only the baselines, e.g., moving CAD-Recode's Extrude TVF@1% from near zero (raw frame) to 52.1 (Table 3), while structure metrics (PEM/rPEM/MFM_{op}) are unaffected. CAD-Exact numbers come from the two-arm run of §4. Raw CD/HD and the failure-mode breakdown are in Appendix F.

5.2 Multi-Operation Comparison

Table 3 shows the 9-method $\times$ 4-op grid. In the staged chain each stage adds its operation (revolve at stage 2: rPEM 53.7, IoU 91.9; sweep at stage 3: rPEM 47.6, IoU 80.3; loft at stage 4: MFM_{op} $0.0 \rightarrow 99.8$, TVF@5% $0.1 \rightarrow 58.4$); loft on stages 1–3 is a hard zero since those stages never see it. Earlier operations are largely retained, but extrude validity drops from $\geq 91\%$ (stages 1–3) to 72.5% at stage 4. Neither regime dominates: the joint model matches the best staged checkpoint on extrude/revolve/sweep and leads by a wide margin on loft (TVF@5% 94.8 vs. 58.4, IoU 96.5 vs. 84.3, Val 96.6 vs. 61.9), while the best staged checkpoint edges revolve (stage 3, rPEM 54.7) and sweep (stage 4, TVF@5% 59.0).

Baselines expose a metric interaction the paper depends on. CAD-Recode v1.5 executes on 65–95% of inputs across all four operations because its output is well-formed CadQuery, but on revolve and sweep the emitted program is a stacked extrude (MFM op-count 0.0%; Figure 4) whose shape metrics collapse (TVF@5% ≤ 5.0, CD% ≥ 6.7). cadrille zero-shot behaves similarly (MFM op-count 0.0%

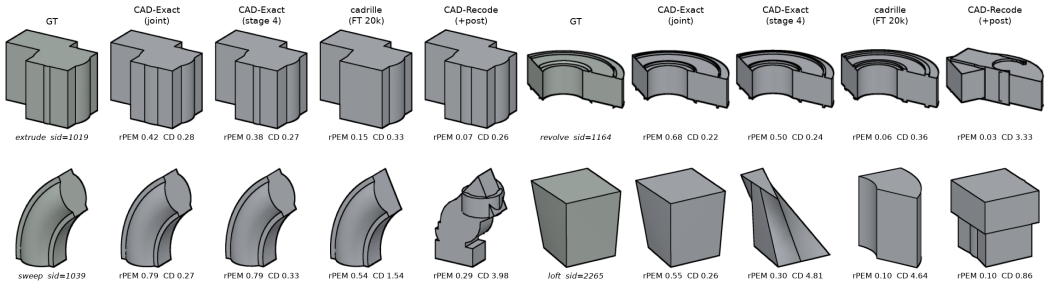


Figure 4: Qualitative comparison of five methods against the ground truth, one sample per operation. Per-panel rPEM and CD% for reference (§3.5); CAD-Recode is drawn in its +*post* frame.

on all non-extrude operations). DeepCAD (via the PC2CAD point-cloud head) generates meshes from its own quantized DSL rather than CadQuery, so PEM/rPEM and MFM_{op} are undefined for it; at the shape level it exhibits the same failure (TVF@$5\% = 0$ across all operations, IoU 19–46, validity 52–78%), confirming that pre-LLM CAD sequence generators do not clear the parameter-level bar even when their meshes execute. Validity therefore fails to signal what is broken; Tier-1 (PEM, TVF, MFM op-count) does.

Fine-tuning cadrille on GeomCAD (LoRA r=64) raises its MFM op-count on revolve and sweep from 0.0% to 99.8% and 100.0%, matching CAD-Exact on operation structure. Scored natively on the GeomCAD grid (no +*post*, no literal renormalization, since fine-tuning has already aligned it to the target frame), cadrille FT reaches PEM $0.0/0.0/1.3/0.0$ and rPEM $20.6/22.6/40.0/13.2$ across extrude/revolve/sweep/loft, still below the joint model's rPEM $35.1/49.2/43.9/27.3$. Loft is where the gap is largest: joint TVF@5% is 94.8 against cadrille FT's 8.4, and validity 96.6 vs. 76.9 (Appendix F).

5.3 Where Invalidity Comes From

Invalidity is decomposed into syntax, API, argument, kernel, and empty-output modes in Appendix F.1. On the staged chain, extrude breaks only once loft is added: its syntax-error rate stays at 1–4% through stages 1–3, which add single-sketch operations, and rises to 12.7% after the loft stage. The loft stage rarely injects loft syntax (explicit `.loft(` calls appear in 1.1% of extrude predictions); instead it shifts the extrude output distribution: `union` appears in 77.4% of predictions against 61.6% in the ground truth, and operation counts mismatch on about twice as many valid predictions as at stages 1–3 (Appendix **??**). Joint training shows none of these shifts. Syntax errors are removable in principle by grammar-constrained decoding; kernel rejection is the residue. Among valid predictions, shape similarity does not reveal how many parameters are right: on the extrude sample of Figure 4, all four methods reproduce the shape closely (CD 0.26–0.33%, with CAD-Recode lowest), yet their rPEM ranges from 0.07 to 0.42. For the joint model, strict PEM peaks at 1.5% per operation (revolve; 2.7% for the best staged checkpoint), and loft PEM stays at 0.0 because free-form loft sections admit multiple valid parametrizations of the same shape. Under multiset recall (rPEM), the same joint model recovers 27–49% of GT literals per operation, and 63–86% under a ± 10 tolerance (rPEM@10); the PEM/rPEM gap is what makes ordered exact match a useful stress test while still crediting the recovered subset. Full failure-mode and graded-recovery tables are in Appendix F.

5.4 External Evaluation

On the DeepCAD test split (real Onshape designs, zero-shot for all our checkpoints), we report two protocols. *Under cadrille's published protocol* (best-of-10 selection over 10 seeded samplings), joint ranks 5th of 12 on median Chamfer distance (2.46), 1st of 12 on invalidity (0.25% = the fraction of inputs on which no seed produced a valid mesh), and 10th of 12 on IoU (59.2). Mean CD is heavy-tailed; CAD-Recode and cadrille achieve lower mean CD (Appendix F). Under *best-of-1* (a single seed, the stricter setting that matches Table 3), executability is 75.2% / 65.0% / 56.5% for joint / stage 1 / stage 4, with stage 4 losing another $\sim$25 points to syntactic invalidity (28.2% vs. joint's 1.9%). Baseline rankings there are taken from each method's own paper on the same DeepCAD split;

Table 3: **Main results on GeomCAD** (all values in %). Zero-shot baselines are scored with *+post* and literal renormalization (§4); the fine-tuned cadrille and CAD-Exact are scored natively in the GT frame. CD% and HD% are medians relative to the GT bounding-box diagonal (↓: lower is better). Red, yellow, and cyan mark the best, 2nd, and 3rd value per column within each operation; full ties and zeros are left unmarked. ⋆DeepCAD is evaluated through the PC2CAD point-cloud head trained on DeepCAD's original dataset; it emits a quantized DSL that is converted to a mesh by its own pipeline (not CadQuery), so PEM/rPEM and MFM_{op} are *n/a*. Shape metrics and Validity are computed on the successful subset only.

		Tier 1						Tier 2					Tier 3
Op	**Method**	TVF1↑	TVF5↑	PEM↑	rPEM↑	rPEM10↑	MFMop↑	IoU↑	CD%↓	F@2%↑	HD%↓	VGF5↑	Val↑
Extrude	DeepCAD (PC2CAD⋆)	0.0	0.0	n/a	n/a	n/a	n/a	36.3	6.25	18.1	21.26	0.0	78.5
	CAD-Recode +post	52.1	83.7	0.0	7.3	29.0	71.2	93.8	0.28	**96.2**	**2.17**	77.9	94.5
	cadrille (0-shot, +post)	1.2	5.3	0.0	7.8	27.2	22.7	0.0	11.54	17.8	34.67	1.4	37.6
	cadrille (FT 20k)	29.8	63.4	0.0	20.6	58.6	81.7	83.4	0.54	86.1	6.04	49.6	83.5
	CAD-Exact (stage 1)	62.7	**85.3**	0.6	35.1	64.3	85.2	95.3	**0.27**	95.6	2.88	76.9	91.6
	CAD-Exact (stage 2)	**63.5**	84.7	0.5	35.6	64.3	85.6	**96.0**	**0.27**	95.9	2.81	78.7	94.4
	CAD-Exact (stage 3)	63.4	84.8	0.5	**35.9**	**64.6**	84.8	95.7	**0.27**	95.9	2.59	77.7	92.9
	CAD-Exact (stage 4)	39.8	58.2	0.3	29.6	59.5	66.5	76.2	0.47	77.8	7.96	40.3	72.5
	CAD-Exact (joint)	63.4	84.3	**0.7**	35.1	62.8	**89.4**	95.5	**0.27**	95.9	2.85	**79.5**	**95.9**
Revolve	DeepCAD (PC2CAD⋆)	0.0	0.0	n/a	n/a	n/a	n/a	22.4	6.63	19.0	26.20	0.0	52.6
	CAD-Recode +post	0.8	5.0	0.0	10.8	39.2	0.0	17.0	6.72	22.1	26.75	0.9	69.7
	cadrille (0-shot, +post)	0.6	2.0	0.0	11.2	26.8	0.0	8.5	10.89	10.6	35.71	0.1	36.3
	cadrille (FT 20k)	14.5	57.5	0.0	22.6	70.8	99.8	81.5	0.58	91.8	2.40	53.1	95.3
	CAD-Exact (stage 1)	1.1	5.0	0.0	7.7	32.3	0.0	23.9	5.71	35.9	30.34	1.8	67.5
	CAD-Exact (stage 2)	39.3	82.0	2.3	53.7	86.5	**100.0**	91.9	**0.28**	**96.9**	1.25	78.4	96.1
	CAD-Exact (stage 3)	**42.5**	**82.9**	**2.7**	**54.7**	**87.4**	99.9	**92.1**	**0.28**	96.7	**1.22**	**80.0**	**97.1**
	CAD-Exact (stage 4)	34.3	78.2	1.3	47.6	84.4	**100.0**	89.7	0.32	95.7	1.43	71.6	92.7
	CAD-Exact (joint)	34.7	77.7	1.5	49.2	83.6	99.9	89.7	0.31	95.0	1.43	71.2	92.9
Sweep	DeepCAD (PC2CAD⋆)	0.0	0.0	n/a	n/a	n/a	n/a	18.7	8.34	15.3	31.49	0.0	72.1
	CAD-Recode +post	0.3	1.8	0.0	15.2	35.1	0.0	10.8	9.66	15.0	30.68	0.3	76.0
	cadrille (0-shot, +post)	0.5	2.1	0.0	15.3	35.8	0.0	8.7	10.02	13.4	30.82	0.1	25.1
	cadrille (FT 20k)	21.8	36.2	1.3	40.0	73.6	**100.0**	61.2	1.59	62.9	8.29	19.9	66.4
	CAD-Exact (stage 1)	2.0	8.6	0.0	18.0	46.2	0.0	39.6	3.43	47.9	17.15	2.7	46.6
	CAD-Exact (stage 2)	1.9	7.3	0.0	19.9	43.8	0.0	32.6	5.65	39.8	25.42	1.3	21.1
	CAD-Exact (stage 3)	29.6	**59.2**	2.0	**47.6**	**81.7**	**100.0**	80.3	0.42	86.9	1.95	51.4	89.2
	CAD-Exact (stage 4)	**30.0**	59.0	**2.2**	46.7	81.4	99.9	**80.9**	**0.40**	**87.6**	**1.84**	**51.6**	**90.5**
	CAD-Exact (joint)	23.7	49.0	1.1	43.9	76.6	99.9	71.6	0.68	78.8	3.37	41.9	89.6
Loft	DeepCAD (PC2CAD⋆)	0.0	0.0	n/a	n/a	n/a	n/a	46.4	4.83	24.6	17.59	0.0	63.3
	CAD-Recode +post	8.2	40.1	0.0	11.5	45.1	0.0	83.0	1.15	78.5	7.35	22.5	65.3
	cadrille (0-shot, +post)	0.4	2.6	0.0	12.0	35.1	0.0	15.0	10.69	12.0	34.71	0.5	77.4
	cadrille (FT 20k)	1.3	8.4	0.0	13.2	42.0	58.3	48.7	5.61	38.3	24.41	3.1	76.9
	CAD-Exact (stage 1)	0.0	0.2	0.0	5.4	29.4	0.0	2.3	20.33	3.6	64.69	0.0	45.9
	CAD-Exact (stage 2)	0.0	0.0	0.0	5.1	29.2	0.0	2.3	21.22	3.4	66.03	0.0	39.3
	CAD-Exact (stage 3)	0.0	0.1	0.0	5.5	29.3	0.0	2.3	20.64	3.6	64.98	0.0	42.8
	CAD-Exact (stage 4)	42.3	58.4	0.0	**28.1**	**86.5**	99.8	84.3	0.37	85.6	1.71	35.9	61.9
	CAD-Exact (joint)	**63.8**	**94.8**	0.0	27.3	86.0	**100.0**	**96.5**	**0.31**	**98.9**	**1.25**	**91.2**	**96.6**

protocol differences (best-of-N, decoding constraints) mean these comparisons should be read as coarse positioning rather than head-to-head.

6 Conclusion and Limitations

We introduce GeomCAD, a 4M-program benchmark with integer parameters balanced over four generative operations, and GeomCAD-Eval, which pairs shape metrics with exact parameter match. Trained on GeomCAD, CAD-Exact recovers more parameter values and more accurate volumes than every baseline on every operation. Exact recovery nonetheless remains open: models that execute on more than 90% of inputs reproduce the full parameter set for at most 2.7% of them, a gap that shape metrics hide.

Limitations include operation coverage: fillet, chamfer, and shell need a topological selector that a benchmark would first have to score. Samples are single-operation solids with clean point clouds; evidence on real designs is limited to zero-shot transfer to DeepCAD (§5.4), and noisy scans, multi-operation parts, and continual-learning baselines (EWC (**?**), LwF (**?**)) are left to future work. The DeepCAD numbers in Table 3 use the PC2CAD point-cloud extension trained on DeepCAD's own dataset, not an official DeepCAD paper release; an evaluation with a first-party PC-conditioned DeepCAD checkpoint, when available, would sharpen the comparison.

AI USE STATEMENT

In this work, we used generative AI tools (Anthropic Claude Code and comparable LLM-based assistants) in four ways: (i) language editing and copy-editing of the manuscript, including the abstract and results-section prose; (ii) retrieval and discovery within internal artifacts (evaluation CSVs, checkpoint provenance, related-work references); (iii) drafting and debugging of analysis, evaluation, and packaging scripts (the metric pipeline in Appendix C.6, the appendix-table generator, and the dataset-release packing script); and (iv) drafting the text and tables of Appendix F (the full 5-method $\times$ 4-op grid, the Validity Ladder, and their notation). We have not used generative AI tools for generating experimental results or for synthesizing the dataset: the four operation splits are produced by deterministic CadQuery generators, not by an LLM. All reported numbers were produced by the evaluation pipeline provided in the anonymized repository. All AI-assisted text and code were reviewed by the authors; AI-assisted code was verified by the authors through testing and independent recomputation of its outputs, and AI-assisted text was checked against the underlying measurements. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

GeomCAD is procedurally generated and contains no personal data or human subjects. The extrude split derives from the publicly released CAD-Recode corpus (CC-BY-NC-4.0), and the remaining three operations are newly generated from CAD-Recode sketch profiles; GeomCAD therefore inherits the non-commercial term and is released under CC-BY-NC-4.0. As CAD reconstruction systems become increasingly capable of recovering executable designs with accurate dimensions, parameter-level evaluation also becomes relevant for understanding potential misuse, including unauthorized replication of proprietary industrial components; we release a synthetic benchmark rather than reconstructions of real proprietary parts, and we believe measurement of this capability supports informed assessment of that risk.

REPRODUCIBILITY STATEMENT

The inference and evaluation code, including the exact metric implementation that produced every reported table, and the Singularity definitions pinning the CAD kernel are provided through an anonymized repository for double-blind review at `https://anonymous.4open.science/r/CAD-Exact-5707`. The full corpus (4,006,182 samples; 232 parquet shards with inline meshes, and the same programs as four gzipped JSONL files), the generation code, the full-corpus audit scripts, the training code, and the weights of our trained checkpoints (baselines use their publicly released weights) will be publicly released upon acceptance. With that release, three central claims are recomputable by a reader rather than asserted: that every literal is an integer (full-corpus audit script), that the meshes follow deterministically from the programs under the pinned container (mesh round-trip check; Appendix C.6), and that each test split is representative of its pool (split-versus-pool audit, which is also the release gate; §3.3). Splits, checkpoints, inference, and scoring are frozen and versioned end to end; the release will document the hash of every artifact, the environment, seeds, decoding configuration, commands, and expected outputs. Dataset generation details are in Appendix B, metric definitions in Appendix E, and training details in Appendix D.

REFERENCES

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. URL `https://arxiv.org/abs/2604.24479`.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. URL `https://arxiv.org/abs/1512.03012`.

Tianrun Chen, Chunan Yu, Yuanqi Hu, Jing Li, Tao Xu, Runlong Cao, Lanyun Zhu, Ying Zang, Yong Zhang, Zejian Li, and Lingyun Sun. Img2CAD: Conditioned 3D CAD model generation from

single image with structured visual geometry. *IEEE Transactions on Industrial Informatics*, 21 (11):8539–8549, 2025. doi: 10.1109/TII.2025.3584476. URL https://doi.org/10.1109/TII.2025.3584476.

CadQuery Contributors. Cadquery, 2026. URL https://doi.org/10.5281/zenodo.18633916.

Elona Dupont, Kseniya Cherenkova, Anis Kacem, Sk Aziz Ali, Ilya Arzhannikov, Gleb Gusev, and Djamila Aouada. CADOps-Net: Jointly learning CAD operation types and steps from boundary-representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022. URL https://arxiv.org/abs/2208.10555.

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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 19–36, 2024. URL https://arxiv.org/abs/2407.12702.

Haoxiang Guo, Shilin Liu, Hao Pan, Yang Liu, Xin Tong, and Baining Guo. ComplexGen: CAD reconstruction by B-rep chain complex generation. *ACM Transactions on Graphics*, 41(4): 129:1–129:18, 2022. doi: 10.1145/3528223.3530078. URL https://doi.org/10.1145/3528223.3530078.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.09685.

Pradeep Kumar Jayaraman, Aditya Sanghi, Joseph G. Lambourne, Karl D. D. Willis, Thomas Davies, Hooman Shayani, and Nigel Morris. UV-Net: Learning from boundary representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11703–11712, 2021. URL https://arxiv.org/abs/2006.10211.

Pradeep Kumar Jayaraman, Joseph G. Lambourne, Nishkrit Desai, Karl D. D. Willis, Aditya Sanghi, and Nigel J. W. Morris. SolidGen: An autoregressive model for direct B-rep synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://iclr.cc/virtual/2024/poster/21768.

R. Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J. Mitra, and Daniel Ritchie. ShapeAssembly: Learning to generate programs for 3D shape structure synthesis. *ACM Transactions on Graphics*, 39(6):234:1–234:20, 2020. URL https://arxiv.org/abs/2009.08026.

Mohammad Sadil Khan, Elona Dupont, Sk Aziz Ali, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. CAD-SIGNet: CAD language inference from point clouds using layer-wise sketch instance guided attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4713–4722, 2024a. URL https://arxiv.org/abs/2402.17678.

Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin Sheikh, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. Text2CAD: Generating sequential CAD models from beginner-to-expert level text prompts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b. URL https://arxiv.org/abs/2409.17106.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):78:1–78:13, 2017. doi: 10.1145/3072959.3073599. URL https://doi.org/10.1145/3072959.3073599.

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 (CVPR)*, pp. 9601–9611, 2019. URL https://arxiv.org/abs/1812.06216.

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 reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2505.22914.

Joseph G. Lambourne, Karl D. D. 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 (CVPR)*, pp. 12773–12782, 2021. URL https://arxiv.org/abs/2104.00706.

Joseph G. Lambourne, Karl D. D. Willis, Pradeep Kumar Jayaraman, Longfei Zhang, Aditya Sanghi, and Kamal Rahimi Malekshan. Reconstructing editable prismatic CAD from rounded voxel models. In *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–9, 2022. doi: 10.1145/3550469.3555424. URL https://doi.org/10.1145/3550469.3555424.

Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J. Mitra. Free2CAD: Parsing freehand drawings into CAD commands. *ACM Transactions on Graphics (SIGGRAPH)*, 41(4), 2022. doi: 10.1145/3528223.3530133. URL https://doi.org/10.1145/3528223.3530133.

Pu Li, Jianwei Guo, Xiaopeng Zhang, and Dong-Ming Yan. SECAD-Net: Self-supervised CAD reconstruction by learning sketch-extrude operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16816–16826, 2023. URL https://arxiv.org/abs/2303.10613.

Pu Li, Jianwei Guo, Huibin Li, Bedrich Benes, and Dong-Ming Yan. SfmCAD: Unsupervised CAD reconstruction by learning sketch-based feature modeling operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4671–4680, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Li_SfmCAD_Unsupervised_CAD_Reconstruction_by_Learning_Sketch-based_Feature_Modeling_Operations_CVPR_2024_paper.html.

Weijian Ma, Shuaiqi Chen, Yunzhong Lou, Xueyang Li, and Xiangdong Zhou. Draw step by step: Reconstructing CAD construction sequences from point clouds via multimodal diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27154–27163, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Ma_Draw_Step_by_Step_Reconstructing_CAD_Construction_Sequences_from_Point_CVPR_2024_paper.html.

Brandon Man, Ghadi Nehme, Md Ferdous Alam, and Faez Ahmed. VideoCAD: A dataset and model for learning long-horizon 3D CAD UI interactions from video. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.24838.

Charlie Nash, Yaroslav Ganin, S. M. Ali Eslami, and Peter Battaglia. PolyGen: An autoregressive generative model of 3D meshes. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. URL https://arxiv.org/abs/2002.10880.

Wamiq Para, Shariq Bhat, Paul Guerrero, Tom Kelly, Niloy J. Mitra, Leonidas J. Guibas, and Peter Wonka. SketchGen: Generating constrained CAD sketches. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.02711.

Daxuan Ren, Jianmin Zheng, Jianfei Cai, Jiatong Li, and Junzhe Zhang. ExtrudeNet: Unsupervised inverse sketch-and-extrude for shape parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 482–498, 2022. URL https://arxiv.org/abs/2209.15632.

Danila Rukhovich, Elona Dupont, Dimitrios Mallis, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. CAD-Recode: Reverse engineering CAD code from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL https://arxiv.org/abs/2412.14042.

Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2007. doi: 10.1111/j.1467-8659.2007.01016.x. URL https://doi.org/10.1111/j.1467-8659.2007.01016.x.

Ari Seff, Yaniv Ovadia, Wenda Zhou, and Ryan P. Adams. SketchGraphs: A large-scale dataset for modeling relational geometry in computer-aided design. In *ICML 2020 Workshop on Object-Oriented Learning*, 2020. URL https://arxiv.org/abs/2007.08506.

Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P. Adams. Vitruvion: A generative model of parametric CAD sketches. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2109.14124.

Vadim Shapiro. Solid modeling. In Gerald Farin, Josef Hoschek, and Myung-Soo Kim (eds.), *Handbook of Computer Aided Geometric Design*, chapter 20, pp. 473–518. Elsevier, 2002. doi: 10.1016/B978-0-444-51104-1.X5000-X. URL https://doi.org/10.1016/B978-0-444-51104-1.X5000-X.

Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. CSGNet: Neural shape parser for constructive solid geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5515–5523, 2018. URL https://arxiv.org/abs/1712.08290.

Gopal Sharma, Difan Liu, Subhransu Maji, Evangelos Kalogerakis, Siddhartha Chaudhuri, and Radomír Měch. ParSeNet: A parametric surface fitting network for 3D point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 261–276, 2020. URL https://arxiv.org/abs/2003.12181.

Maxim Tatarchenko, Stephan R. Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3D reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3405–3414, 2019. URL https://arxiv.org/abs/1905.03678.

Mikaela Angelina Uy, Yen-Yu Chang, Minhyuk Sung, Purvi Goel, Joseph G. Lambourne, Tolga Birdal, and Leonidas Guibas. Point2Cyl: Reverse engineering 3d objects from point clouds to extrusion cylinders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11850–11860, 2022. URL https://arxiv.org/abs/2112.09329.

Tamás Várady, Ralph R. Martin, and Jordan Cox. Reverse engineering of geometric models – an introduction. *Computer-Aided Design*, 29(4):255–268, 1997. doi: 10.1016/S0010-4485(96)00054-1. URL https://doi.org/10.1016/S0010-4485(96)00054-1.

Xiaogang Wang, Yuelang Xu, Kai Xu, Andrea Tagliasacchi, Bin Zhou, Ali Mahdavi-Amiri, and Hao Zhang. PIE-Net: Parametric inference of point cloud edges. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2007.04883.

Karl D. D. 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*, 40(4):1–24, August 2021. ISSN 0730-0301, 1557-7368. doi: 10.1145/3450626.3459818. URL https://doi.org/10.1145/3450626.3459818.

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 (ICCV)*, pp. 6772–6782, 2021. URL https://arxiv.org/abs/2105.09492.

Xiang Xu, Karl D. D. Willis, Joseph G. Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. SkexGen: Autoregressive generation of CAD construction sequences with disentangled codebooks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2207.04632.

Xiang Xu, Joseph G. Lambourne, Pradeep Kumar Jayaraman, Zhengqing Wang, Karl D. D. Willis, and Yasutaka Furukawa. Hierarchical neural coding for controllable CAD model generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2307.00149.

Xiang Xu, Pradeep Kumar Jayaraman, Joseph G. Lambourne, Karl D. D. Willis, and Yasutaka Furukawa. BrepGen: A B-rep generative diffusion model with structured latent geometry. *ACM Transactions on Graphics (SIGGRAPH)*, 43(4), 2024. URL https://arxiv.org/abs/2401.15563.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. URL https://arxiv.org/abs/2407.10671.

Bingchen Yang, Haiyong Jiang, Hao Pan, Guosheng Lin, Jun Xiao, and Peter Wonka. PS-CAD: Local geometry guidance via prompting and selection for CAD reconstruction. *ACM Transactions on Graphics*, 44(3):28:1–28:21, 2025. doi: 10.1145/3733595. URL https://doi.org/10.1145/3733595.

Fenggen Yu, Zhiqin Chen, Manyi Li, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. CAPRI-Net: Learning compact CAD shapes with adaptive primitive assembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11768–11778, 2022. URL https://arxiv.org/abs/2104.05652.

Fenggen Yu, Qimin Chen, Maham Tanveer, Ali Mahdavi-Amiri, and Hao Zhang. D^{2}CSG: Unsupervised learning of compact CSG trees with dual complements and dropouts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2301.11497.

A Integer Grid

CAD-Recode v1.5 stores literals as integers on $[-100, 100]$ (Rukhovich et al., 2025, Appendix B), a step of 0.5% of the part's extent. Floating-point literals would leave PEM undefined, since `0.5` and `0.4999` have no canonical equivalence and any comparison would need a tolerance. GeomCAD therefore stores every literal on the integer grid $[1, 1000]$ (§3.2). Table 4 compares the three grid sizes we considered.

Table 4: **Grid size, resolution, and token cost.** A step is a fixed fraction of the part's largest extent; the absolute step is given for a $100\,\mathrm{mm}$ part. The Qwen2 tokenizer splits numbers into single digits.

Grid	Literal range	Step (% of extent)	Step @ $100\,\mathrm{mm}$	Digits per literal
CAD-Recode v1.5	$[-100, 100]$	0.5%	$0.5\,\mathrm{mm}$	≤ 3 (plus sign)
GeomCAD	$[\mathbf{1}, \mathbf{1000}]$	$\mathbf{0.1}\%$	$\mathbf{0.1}\,\mathrm{mm}$	$\leq \mathbf{4}$ (≤ 3 below 1000)
10^4 levels	$[1, 10^4]$	0.01%	$0.01\,\mathrm{mm}$	≤ 5

The 10^4-level grid costs one more digit per literal and gives no precision gain above the noise floor of our metrics (§3.5). A power-of-two bound such as 1024 costs the same four digits as 1000 but loses the decimal step of exactly 0.1%.

B Data Generation

The CAD-Recode fraction of extrude (73.7%) is rewritten in place (Algorithm 1). Revolve, sweep, loft, and the SketchGraphs fraction of extrude (26.3%) are composed from a sketch profile and a template (Algorithm 2). Both pipelines call the same literal-rewriting routine (Algorithm 3).

Sketch profiles for revolve, sweep, and loft are extracted from CAD-Recode v1.5 programs and re-emitted as standalone closed sketches; profiles built from primitives (`circle`, `rect`) or with more than one closed loop are dropped. Each profile is centered, rescaled, and clamped to an operation-specific axis range. Each non-extrude generator also has `scratch_*` modes that draw profiles from simple shape grammars (polygons, lobes, slot-and-key). SketchGraphs sketches are solved to 2D polylines, filtered to a single closed loop, centered, and rescaled in the same way.

Algorithm 1 *Extrude split generation by in-place rewriting (CAD-Recode fraction, 73.7%)*

Require: CAD-Recode v1.5 Python program c.
Ensure: (c', STL) pair, or REJECT.
$c' \leftarrow \textsc{TransformScript}(c, [1, 1000])$ ▷ Alg. 3, full integer range
if $c' = \varnothing$ **return** REJECT ▷ degenerate parameter range
Execute c' in CadQuery; **return** REJECT on exception or $\mathrm{Volume}(r) \leq 0$
Let $\mathbf{p}_{\min}, \mathbf{p}_{\max} \in \mathbb{R}^3$ denote the min and max corners of $\mathrm{BBox}(r)$.
if $\min_{i \in \{x,y,z\}} p_{\min,i} < -0.01$ **then** ▷ solid extends below positive octant; attempt auto-translate fix-up
    Compute per-axis translation $t_i \leftarrow \lceil 1 - p_{\min,i} \rceil$ for $i \in \{x, y, z\}$.
    if any $t_i > 1000$ **return** REJECT ▷ required translation outside parameter range
    Append `.translate`$((t_x, t_y, t_z))$ to c' and re-execute.
    if the re-executed solid fails any of the above checks, **return** REJECT.
end if
return REJECT if the solid is not watertight or falls below the minimum face count; otherwise export STL via `trimesh`.
return (c', STL)

Integer discretization applies one affine map per program (Algorithm 3). The extrude rewrite uses the fixed range $[1, 1000]$; the composed programs use a random sub-range of $[0, 1000]$ per program, so that operation parameters such as sweep length or loft spacing vary across samples. Zeros in the workplane-normal component of a translation are kept verbatim so that sections in loft and sweep stay aligned.

The validity gate is shared by all four pipelines: a program is rejected if it fails to execute in CadQuery's OCCT backend, does not produce a watertight solid, or falls below the minimum face count. The executed solid is not required to lie inside the $[1, 1000]^3$ grid box (revolving a profile carries it to the far side of the axis); scoring is unaffected because metrics are computed in the evaluation frame (§3.5).

Algorithm 2 *Bottom-up composition for loft/revolve/sweep and the SketchGraphs fraction of extrude*
Loft/revolve/sweep draw profiles from the CAD-Recode-derived pool $\mathcal{P}$; the SketchGraphs fraction of extrude draws from SketchGraphs and uses a fixed extrusion template ($o = \text{EXTRUDE}$).

Require: Operation $o \in \{\text{LOFT}, \text{REVOLVE}, \text{SWEEP}, \text{EXTRUDE}\}$; sketch profile pool $\mathcal{P}$ (CAD-Recode for loft/revolve/sweep; SketchGraphs for the extrude SketchGraphs fraction, §B).
Ensure: (c', STL) pair, or REJECT.
for attempt $= 1$ **to** 10 **do** ▷ retry-bounded sampling loop
  method $\sim \{\text{PROFILE}_o, \text{PROFILESIMPLE}, \text{SCRATCH}_o, \text{SCRATCHSKETCH}\}$ ▷ 4 sketch-source modes
  $c \leftarrow \text{GENERATERAWSCRIPT}(o, \text{method}, \mathcal{P})$ ▷ compose op-template with sketch in canonical 2D region
  if $c = \varnothing$ **continue**
  Sample randomized target sub-range $[t_{\min}, t_{\max}] \subset [0, 1000]$ with non-trivial span and $t_{\min}$ near zero ▷ exact distribution in code release
  $c' \leftarrow \text{TRANSFORMSCRIPT}(c, [t_{\min}, t_{\max}])$ ▷ Alg. 3
  if $c' = \varnothing$ **continue**
  Execute c'; **continue** on exception or $\text{Volume}(r) \leq 0$.
  continue if the solid is not watertight or falls below the minimum face count; export STL.
  return (c', STL)
end for
return REJECT ▷ all 10 attempts failed

Algorithm 3 TRANSFORMSCRIPT: *source-level numerical rewriting*

Require: Python program c; target range $[t_{\min}, t_{\max}] \subseteq [0, 1000]$ ($[1, 1000]$ when invoked from the extrude pipeline).
Ensure: Rewritten program c', or $\varnothing$.
Replace prefix zeros in `Vector(0,0,N)` / `Location((0,0,N))` with sentinel Z. ▷ coord-frame protection: workplane normal axis
Resolve inline divisions, e.g., `offset=X/N`, to floats. ▷ normalize before regex match
$L \leftarrow$ standalone numerical literals in c via regex. ▷ skips numerals embedded in identifier names
$v_{\min}, v_{\max} \leftarrow \min(L), \max(L)$.
return $\varnothing$ if $L = \varnothing$ or $v_{\max} - v_{\min} < 10^{-3}$. ▷ degenerate range
for each literal $v \in L$ in reverse position order **do** ▷ back-to-front replacement preserves earlier match positions
  $q \leftarrow \text{clamp}\big(\text{round}\big(\frac{v - v_{\min}}{v_{\max} - v_{\min}}(t_{\max} - t_{\min}) + t_{\min}\big),\ 1, 1000\big)$. ▷ affine map + edge-case clamp
  Replace v with q in c.
end for
Restore sentinel Z to literal 0. ▷ workplane axis preserved verbatim
return the resulting program as c'.

Deduplication runs twice. At generation time the composed programs are deduplicated by canonical-form hash; the extrude rewrite is not. Before release the whole corpus is deduplicated by exact program text across splits. This removes 15 training rows whose text also appears in a test split (revolve 1, sweep 14), one that appears in validation, one empty-program row found by the integer audit, and 1,462 within-training duplicates (all extrude), 0.037% of training rows in total. The removed IDs ship with the corpus.

Splits are made at the program level. Validation is drawn from each operation's pool; the test split is generated separately from disjoint seeds through the same pipeline and validity gate. Counts are in Table 5.

C CORPUS STATISTICS

C.1 COMPOSITION

Each row stores the CadQuery program and the STL mesh of the solid it produces (Figure 5). Input point clouds are sampled from the mesh on demand ($8{,}192$ surface points, farthest-point-sampled to 256). Because every program executes in OCCT, the solid can also be exported as STEP, BREP, or IGES. The corpus is released under CC-BY-NC-4.0, the license of the CAD-Recode corpus it partly derives from.

Table 5: **GeomCAD composition as released** (4,006,182 samples; post-audit counts and removed IDs ship with the corpus). Every row carries the executable program and its paired mesh; the programs alone are 271 MB as four gzipped JSONL files.

Operation	Train	Validation	Test	Total	Integer range
extrude	998,815	1,500	1,500	1,001,815	[1, 1000]
revolve	998,965	1,035	1,500	1,001,500	[0, 1000]
sweep	998,796	1,071	1,500	1,001,367	[0, 1000]
loft	998,945	1,055	1,500	1,001,500	[0, 1000]
total	3,995,521	4,661	6,000	4,006,182	[0, 1000]

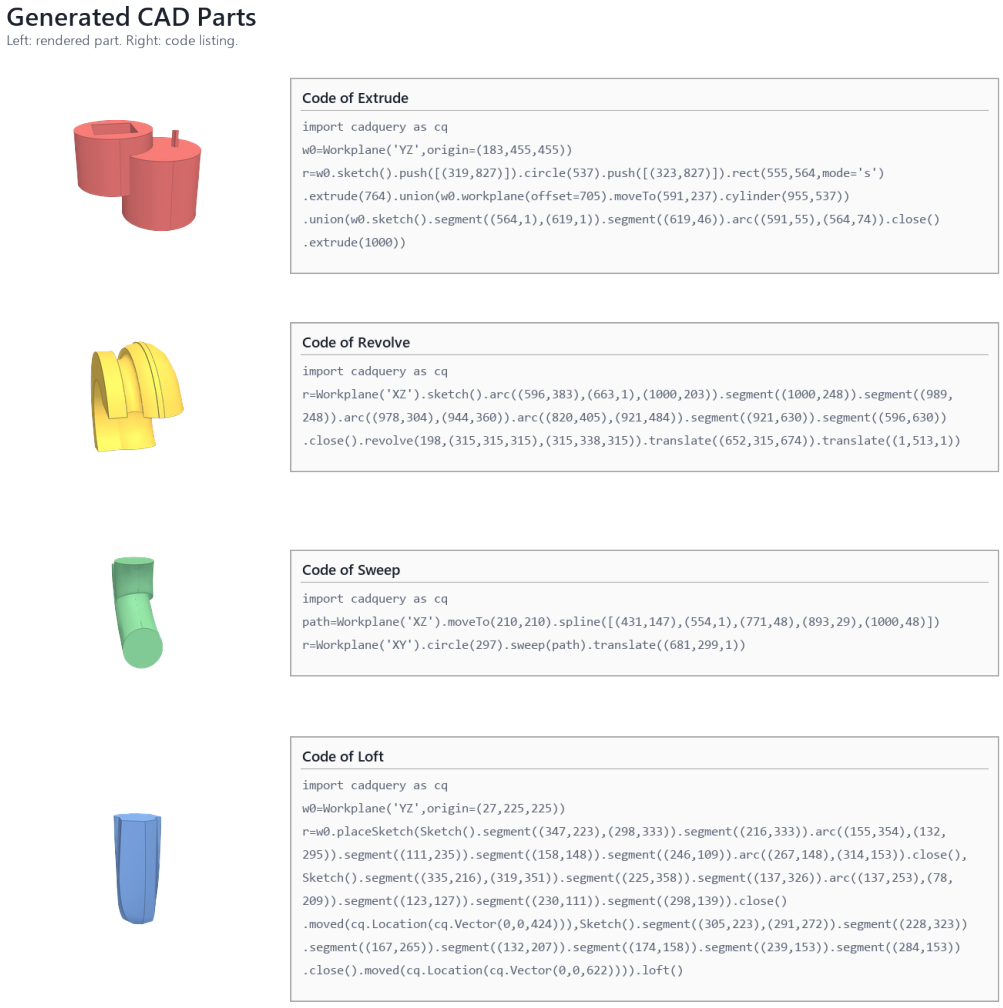


Figure 5: A GeomCAD sample: CadQuery program with integer literals and the solid it produces.

C.2 Parameters and Program Length

Figure 6 shows the literal values per operation (0 excluded). Sweep literals concentrate below 200, revolve literals peak near both ends of the grid, and loft literals concentrate mid-range; the two-sample Kolmogorov–Smirnov distance exceeds 0.15 for every pair of operations. Loft programs are the longest (median 36 literals, maximum 302; Table 7, Figure 7).

Table 6: **Construction structure per operation** (156,500-program sample).

Operation	Curve primitives (med)	Ordered sections
Extrude	7	1
Revolve	3	1
Sweep	4	1 + explicit path
Loft	0	median 2, max 5

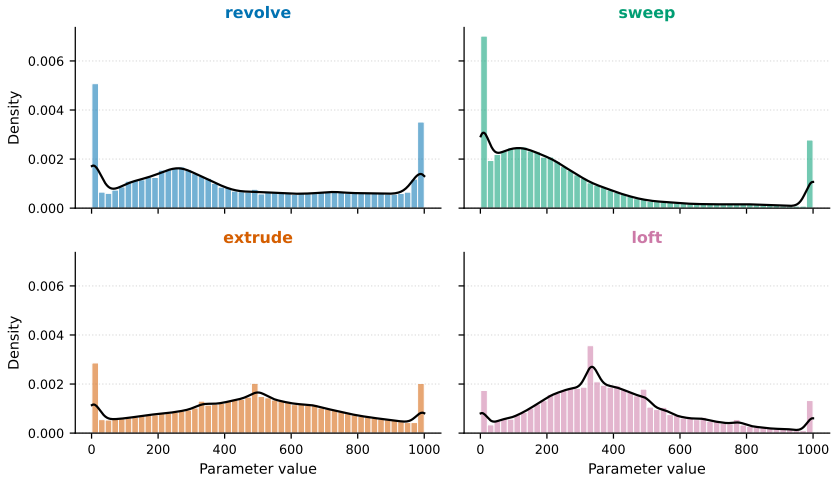


Figure 6: **Literal values per operation**, histogram and kernel density over $[1, 1000]$.

Table 7: **Program length per operation** over the full pool.

	Program size (bytes)			API calls			Parameters		
Operation	Median	Mean	Max	Median	Mean	Max	Median	Mean	Max
Extrude	384	406.0	940	20	21.4	55	31	33.7	97
Loft	377	438.6	2551	21	24.5	153	36	42.5	302
Revolve	241	254.8	882	11	12.0	43	24	25.1	101
Sweep	258	265.6	874	13	13.7	47	18	20.4	95

C.3 Surface Types per Operation

Table 8 splits Table 2 by operation. B-spline faces come from loft and sweep, surfaces of revolution from revolve and sweep. The four operations produce cone, torus, and sphere faces only as negligible residue.

C.4 Solid-Level Composition

A solid is *multi-body* when its program yields more than one disjoint solid and *thin* when its smallest bounding-box extent is under 2% of the largest; rates are over programs that execute (Table 9). Zero-to-CAD has fewer thin and low-face-count solids than GeomCAD. GeomCAD's multi-body solids are almost all extrude (27.75% of extrude, down from 57.45% in CAD-Recode; sweep 0.07%, revolve 0.003%, loft 0.00%), since extruding a multi-loop sketch can yield several solids while revolve, sweep, and loft yield one. Thin solids are 1.73% of extrude and 0.91% of sweep. Per-sample flags for all three properties ship with the release.

C.5 Split-versus-Pool Audit

All four release splits match their pools on construct frequency (within about one point, e.g. extrude `union` 61.6 vs. 61.2%, sweep `radiusArc` 27.1 vs. 27.3%) and on triangle count at the median and 90th percentile. Validation, drawn from the pool, serves as the control.

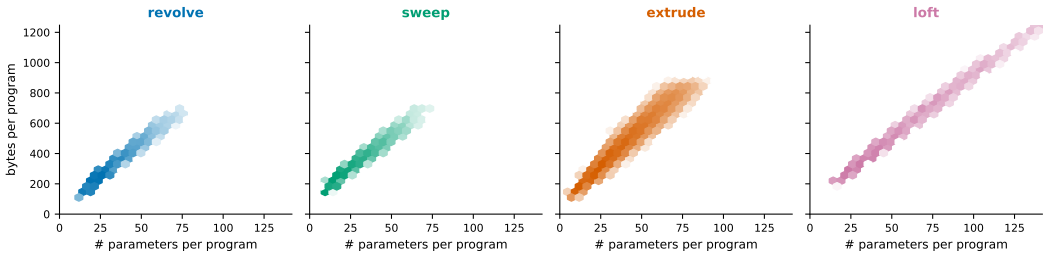


Figure 7: **Program size (bytes) against literal count**, per operation.

Table 8: Per-operation surface composition (%, full corpus) and median B-Rep faces per solid.

Operation	plane	B-spline	cylinder	revolution	extrusion	med. faces
extrude	86.48	0.00	13.52	0.00	0.00	13
revolve	39.43	0.00	16.44	44.13	0.00	8
sweep	67.62	15.17	5.27	9.27	2.67	7
loft	20.44	79.28	0.28	0.00	0.00	10

C.6 Deterministic Mesh Regeneration

We re-executed 60 test programs per operation under the pinned container (Table 10). All 240 execute. Where face counts agree, the triangle data is bit-identical; the only byte differences are the 80-byte STL header and one-ulp noise in per-facet normals. The 18 face-count differences come from the library version: two fresh runs of the same program agree exactly with each other and differ from the released mesh by 0.2–0.6% of facets, always on curved surfaces, which is why extrude is unaffected. On 200 samples per operation, the released meshes are watertight in 91.0% of extrude, 99.0% of revolve, 85.0% of sweep, and 100.0% of loft; the rest are open at the mesh level, not at the solid level, and are flagged per sample.

D CAD-Exact Implementation Details

The point encoder applies Fourier positional encoding to the 256 farthest-point-sampled points and projects them linearly to d=1536, the hidden size of `Qwen2-1.5B`.

LoRA adapters of rank r=64 are attached to all attention and feed-forward projections. The backbone is frozen; the adapters and the point encoder are trained. An adapter parameterizes the update

$$W' = W + BA, \tag{1}$$

with frozen $W \in \mathbb{R}^{d\times k}$ and trainable $A \in \mathbb{R}^{r\times k}$, $B \in \mathbb{R}^{d\times r}$.

The staged arm starts stage 1 with a fresh adapter on the pretrained `Qwen2-1.5B` and trains on extrude alone. Each later stage merges the previous adapter into the backbone ($W \leftarrow W + BA$), starts a fresh adapter, and keeps training the same point encoder. Before each merge, an adapter/base fingerprint check and a post-merge rank test confirm that the adapter is merged onto the right base. The sample mixture of each stage is given in Table 11.

Shared settings are identical in both arms: initial weights, encoder, backbone, LoRA configuration, optimizer, batch size, preprocessing, checkpoint selection, corpus version, and splits. Early stopping is disabled, so both arms run their full step budget.

Optimization uses AdamW with learning rate 10^{-4}, cosine decay with 6% warmup, and weight decay 0.01. The effective batch size is 64 (per-device batch 8, gradient accumulation 4, over 2 GPUs; equivalently 32 per device with 2-step accumulation on one H200), i.e. 3.0 M samples per stage and 12.0 M samples (3 epochs over ∼4 M) for the joint arm. Maximum sequence length is 1024 tokens, with gradient checkpointing and `bfloat16` throughout; all runs use seed 42. Each arm takes about 10 GPU-days on 4× NVIDIA A100 80 GB (∼1.10 s per step).

Table 9: Solid-level composition, full corpora under one kernel.

Corpus	multi-body	single-body	zero solids	thin	<7 faces
CAD-Recode	57.45%	42.55%	0.00%	0.30%	11.50%
GeomCAD	6.89%	93.11%	0.00% (1)	0.65%	27.24%
Zero-to-CAD	0.12%	99.11%	0.77% (7,570)	0.09%	0.01%

Table 10: Mesh round-trip check under the pinned container (60 programs per operation).

Operation	rebuilt	exact face count	bbox within 10^{-6}	volume within 0.01%
extrude	60/60	60/60	60/60	60/60
revolve	60/60	55/60	60/60	56/60
sweep	60/60	57/60	59/60	60/60
loft	60/60	50/60	60/60	59/60

Symmetric inference maps an input point $\mathbf{x}$ with bounding-box minimum $\mathbf{b}_0$ and largest extent $B_{\max}$ to $(\mathbf{x}-\mathbf{b}_0)/B_{\max} \in [0,1]^3$. A predicted mesh vertex $\mathbf{m}$, in grid units, is returned to the input frame as $10^{-3}B_{\max}\,\mathbf{m}+\mathbf{b}_0$.

E Metric Definitions

Notation follows §3.5. Literals are the integers of a program in textual order, clamped to $\{0\} \cup [1, 1000]$; frames and baseline normalization are described in §3.5 and §4. Tier-1 metrics and IoU, CD%, F@2%, HD% are averaged over the valid set $\mathcal{V}$; VGF@5% and Validity over all N samples.

E.1 Tier 1

Volume error and TVF:

$$\mathrm{VolErr}^i \;=\; \frac{\left|\hat{V}^i - V^i\right|}{V^i}. \tag{2}$$

$$\mathrm{TVF@}\delta \;=\; \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathbb{1}\left[\,\mathrm{VolErr}^i \le \delta\,\right], \quad \delta \in \{1\%, 5\%\}. \tag{3}$$

PEM compares the GT literals $\boldsymbol{\theta}^i = (\theta^i_1, \ldots, \theta^i_{n_i})$ with the predicted literals $\hat{\boldsymbol{\theta}}^i = (\hat{\theta}^i_1, \ldots, \hat{\theta}^i_{\hat{n}_i})$ position by position, without tolerance:

$$\mathrm{PEM} \;=\; \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathbb{1}\Big[\,\hat{n}_i = n_i \;\wedge\; \hat{\theta}^i_j = \theta^i_j \;\forall j\,\Big]. \tag{4}$$

rPEM is the multiset recall of GT literals, where $\cap_m$ counts each value $\min(c_{\hat{\boldsymbol{\theta}}}, c_{\boldsymbol{\theta}})$ times, so matching is one-to-one:

$$\mathrm{rPEM} \;=\; \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \frac{|\hat{\boldsymbol{\theta}}^i \,\cap_m\, \boldsymbol{\theta}^i|}{|\boldsymbol{\theta}^i|}, \tag{5}$$

rPEM@k replaces the intersection with a maximum-cardinality matching M_i^* in the bipartite graph that links a GT literal θ^i_a and a predicted literal $\hat{\theta}^i_b$ whenever $|\theta^i_a - \hat{\theta}^i_b| \le k$:

$$\mathrm{rPEM@}k \;=\; \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \frac{|M_i^*|}{|\boldsymbol{\theta}^i|}. \tag{6}$$

We compute M_i^* by nearest-first greedy assignment and report $k = 10$. By construction $\mathrm{rPEM@}0 = \mathrm{rPEM}$.

MFM$_{\mathrm{op}}$ checks the counts n_k of the four generative operations $\mathcal{F}_{\mathrm{op}} = \{\texttt{extrude}, \texttt{revolve}, \texttt{sweep}, \texttt{loft}\}$:

$$\mathrm{MFM}_{\mathrm{op}} \;=\; \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathbb{1}\left[\,\forall k \in \mathcal{F}_{\mathrm{op}} : n_k(\hat{y}^i) = n_k(y^i)\,\right]. \tag{7}$$

Table 11: **Sample mixture per stage** (%). Staged rows: the new operation plus replay of earlier ones.

Stage	Extrude	Revolve	Sweep	Loft
1 (extrude)	100.0	–	–	–
2 (+revolve)	28.6	71.4	–	–
3 (+sweep)	8.2	20.4	71.4	–
4 (+loft)	2.3	5.8	20.4	71.4
Joint (reference)	25.0	25.0	25.0	25.0

E.2 Tier 2

With $\hat{Y}^i, Y^i$ the predicted and GT solids and $\hat{S}^i, S^i$ their surface samples:

$$\mathrm{IoU}^i = \frac{|\hat{Y}^i \cap Y^i|}{|\hat{Y}^i \cup Y^i|}, \tag{8}$$

$$\mathrm{CD}^i = \frac{1}{|\hat{S}^i|}\sum_{a\in\hat{S}^i}\min_{b\in S^i}\|a-b\|_2^2 \;+\; \frac{1}{|S^i|}\sum_{b\in S^i}\min_{a\in\hat{S}^i}\|a-b\|_2^2, \tag{9}$$

$$\mathrm{F}^i@\tau_{\text{rel}} = \frac{2\,P_\tau\,R_\tau}{P_\tau + R_\tau}, \quad P_\tau = \tfrac{1}{|\hat{S}^i|}\sum_{a\in\hat{S}^i}\mathbb{1}\!\left[\min_{b\in S^i}\|a-b\|_2 \le \tau\right],\; R_\tau \text{ symmetric}, \tag{10}$$

where $\tau = 2\%$ of the GT bounding-box diagonal (Tatarchenko et al., 2019), and

$$d_H^i = \max\!\left(\max_{a\in\hat{S}^i}\min_{b\in S^i}\|a-b\|_2,\; \max_{b\in S^i}\min_{a\in\hat{S}^i}\|a-b\|_2\right). \tag{11}$$

CD and HD are reported as percentages of the GT bounding-box diagonal (1,000 in the evaluation frame):

$$\mathrm{CD}\%^i \;=\; \frac{\sqrt{\mathrm{CD}^i}}{\mathrm{diag}(\mathrm{bbox}^i_{\mathrm{GT}})}\times 100\%, \qquad \mathrm{HD}\%^i \;=\; \frac{d_H^i}{\mathrm{diag}(\mathrm{bbox}^i_{\mathrm{GT}})}\times 100\%. \tag{12}$$

VGF sets F@2% to zero for invalid predictions ($v_i = 0$) and for volume errors above δ:

$$\mathrm{VGF}@\delta \;=\; \frac{1}{N}\sum_{i=1}^{N} v_i \cdot \mathrm{F}^i@\tau_{\text{rel}} \cdot \mathbb{1}[\mathrm{VolErr}^i \le \delta], \quad \delta = 5\%. \tag{13}$$

E.3 Tier 3

A program is invalid if CadQuery execution exceeds 30 s, if generation plus execution exceeds 180 s per input, or if the written STL does not load as a non-empty triangle mesh. Each invalid prediction is assigned to one failure mode: *syntax* (the text does not parse as Python), *API misuse* (a CadQuery method is called incorrectly), *bad argument* (numerically invalid parameters, e.g. a self-intersecting sketch), *kernel* (OCCT rejects the construction), or *empty* (execution produces no non-empty mesh).

F Additional Results

F.1 Failure Modes on Extrude

Table 12: **Decomposition of invalid predictions on extrude** (% of the full split; rows sum to 100 up to rounding). *Syntax* = text does not compile; *API* = wrong CadQuery call; *arg* = impossible arguments; *kernel* = OCCT rejects the construction; *empty* = executes with no volume.

Model / split	valid	syntax	API	arg	kernel	empty
stage 1, extrude	91.6	3.9	2.0	0.2	2.3	0.0
joint, extrude	95.9	0.4	1.0	0.3	1.3	0.4

F.2 ZERO-SHOT DEEPCAD

Table 13 scores the joint model on the 816 DeepCAD test models with released normalized GT meshes, using cadrille's published protocol: ten candidates per input, each from a different sampling of the input cloud, the one with the lowest Chamfer distance to that cloud kept, and the released evaluation code for normalization. The other rows are quoted from cadrille's table; our harness reproduces its CAD-Recode entry (0.184). Our mean CD is 6.27 against a median of 2.46, and invalidity is 0.25% at ten candidates but 23.5% at one. Nine rows train on DeepCAD and two on the CAD-Recode corpus; ours is trained on neither. DeepCAD is extrusion-only and shares construction-graph provenance with our extrude subset, so it is only partly external.

Table 13: Zero-shot on the DeepCAD benchmark (816 models with released GT meshes; comparison rows quoted from cadrille's published table).

Method	Trained on	med CD ↓	IoU ↑	IR ↓
PointNet+DeepCAD	DeepCAD	9.64	46.7	7.1%
Point-BERT+HNC-CAD	DeepCAD	8.64	65.3	5.6%
MultiCAD	DeepCAD	8.09	–	11.5%
TransCAD	DeepCAD	4.51	65.5	1.1%
PrismCAD	DeepCAD	4.28	72.1	16.2%
Point2Cyl	DeepCAD	4.27	73.8	3.9%
CAD-Diffuser	DeepCAD	3.02	74.3	1.5%
CAD-Exact joint (zero-shot)	**GeomCAD**	**2.46**	**59.2**	**0.25%**
CAD-SIGNet	DeepCAD	0.29	77.3	5.0%
cadrille	DeepCAD + img + text	0.25	79.4	0.4%
CAD-Recode	CAD-Recode 1M	0.18	87.1	3.1%
cadrille	CAD-Recode 1M + img	0.18	87.1	2.1%

Table 14 uses a separate draw of 1,498 shapes, taken uniformly from all 100 directories of the official 8,052-model test split, at best-of-1. F@2% is computed against the input cloud; on GeomCAD extrude, mesh- and cloud-referenced F@2% differ by 0.0001.

Table 14: Three GeomCAD-trained checkpoints, zero-shot on 1,498 DeepCAD test shapes, best-of-1; invalid predictions scored 0.

Model	Executability	F@2% (all)	F@2% median (valid)	Syntax-invalid
CAD-Exact stage 1	65.0%	0.419	0.695	3.5%
CAD-Exact stage 4	56.5%	0.369	0.692	28.2%
Jointly trained	75.2%	0.496	0.698	1.9%

F.3 CADRILLE FINE-TUNED ON GEOMCAD

We fine-tune the released `cadrille-rl` checkpoint with LoRA (r=64, α=128) and the point encoder unfrozen, on 200,000 GeomCAD programs (50,000 per operation), for 3,000 and for 20,000 steps, and decode greedily from 256 points. At effective batch 32, 20,000 steps cover 640,000 samples, about three passes over the subset. The 20k-step model is the *cadrille (FT 20k)* row of Table 3, scored natively without +*post* or literal remapping.