SAR4CAD: STRUCTURE-AWARE REINFORCEMENT LEARNING FOR PARAMETRIC CAD RECONSTRUCTION FROM MULTI-VIEW DRAWINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Computer-Aided Design (CAD) models can be reconstructed from engineering drawings; however, existing methods are limited by the scarcity of well-annotated drawing datasets and an excessive reliance on a single sequence representation. Specifically, most sequence generators optimize only one serialization scheme, although neither sketch ordering nor the decomposition of certain features is necessarily unique. SAR4CAD is a two-stage framework that trains a multimodal large language model (MLLM) using multi-view renderings, scalable vector graphics (SVG) projection tokens, and concise textual descriptions. The SVG tokens encode visible and hidden edges, primitive types, and quantized coordinates extracted from the CAD model, thereby providing reliable geometric information for reconstruction. In the first stage, supervised fine-tuning teaches the model the required output syntax and establishes a stable generation policy. In the second stage, structure-aware group relative policy optimization (GRPO) evaluates sampled outputs after they have been parsed and executed. The associated reference-guided structural reward integrates CAD-kernel validity, structural-count similarity, sketch overlap computed via Hungarian matching, ordered extrusion-parameter similarity, and 3D Chamfer similarity. Experiments on a constructed parametric CAD dataset demonstrate that SAR4CAD consistently outperforms the evaluated baselines across all evaluation metrics. Code is available at https://anonymous.4open.science/r/SAR4CAD-F64F.

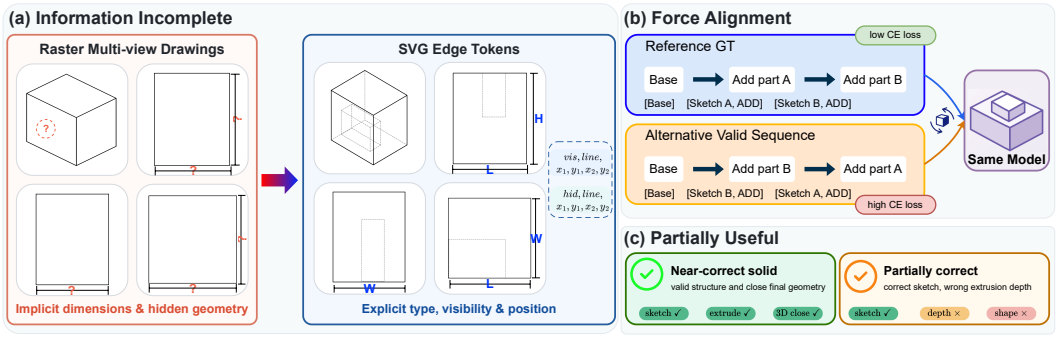

Figure 1: Illustrative examples of existing problems. (a) Raster multi-view drawings leave dimensions and hidden geometry ambiguous, whereas visibility-aware SVG edge tokens provide explicit primitive types, visibility, and quantized positions. (b) Token-level supervision penalizes alternative construction sequences even when they produce the same solid as the reference. (c) Non-exact candidates still provide useful supervision.

1 INTRODUCTION

Computer-aided design has long been perceived as a tedious and complicated process with a steep learning curve. Constructing a parametric CAD model requires users to master geometric primitives, dimensional constraints, feature dependencies, and modeling operations, which makes CAD authoring difficult for non-experts and time-consuming even for experienced designers (Willis et al.,

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.

2021). Recent advances in large language models and sequence-based parametric representations have made automatic CAD generation increasingly feasible: sketches, extrusions, and Boolean operations can be serialized as discrete tokens and generated autoregressively (Xu et al., 2022; 2023). This progress has also enabled image-conditioned CAD reconstruction, where a model takes one or multiple views of an object as input, infers its parametric construction sequence, and executes the sequence to recover an editable 3D CAD model. Representative multimodal approaches include CAD-MLLM, which supports CAD generation conditioned on text, images, and point clouds (Xu et al., 2024), and Drawing2CAD, which reconstructs native CAD sequences from multi-view vector drawings (Qin et al., 2025). Despite their progress, current reconstruction methods still face two limitations that restrict geometric fidelity and learning efficiency.

First, image-conditioned reconstruction often lacks explicit and sufficiently detailed numerical constraints. Raster views convey appearance, projected geometry, and coarse spatial relationships, but exact dimensions and relative scales remain implicit (Wang et al., 2025), as shown in Fig. 1 (a). A generated model may therefore resemble the target while containing incorrect lengths, radii, offsets, or overall proportions. Additional geometric modalities, such as point clouds, can reduce this ambiguity by providing direct 3D measurements, but they are often difficult to obtain in practical scenarios because they require specialized sensors, calibrated capture settings, or additional reconstruction pipelines. A practical CAD reconstruction system should instead recover accurate geometry from readily available drawings while receiving numerical requirements in an accessible form.

Second, conventional sequence supervision assumes that each CAD model corresponds to a single correct parametric sequence. This assumption conflicts with the many-to-one nature of parametric modeling: different sketch orders, primitive decompositions, and feature constructions can produce the same final solid (Wu et al., 2021), as shown in Fig. 1 (b). Consequently, a predicted sequence may be syntactically valid and geometrically correct yet still be treated as a failure because it differs from the recorded reference. Sequences that satisfy only part of the target structure receive even less useful feedback, although their correct sketches, primitives, or extrusion parameters reveal which decisions the model has already learned, as shown in Fig. 1 (c). Discarding these "failed" sequences wastes informative training signals and limits further optimization under token-level maximum-likelihood objectives.

To address these limitations, we introduce SAR4CAD, a multimodal framework for parametric CAD reconstruction with explicit numerical conditioning and structure-aware reinforcement learning. Each training example combines multi-view images, a concise image description, and detailed numerical requirements represented by typed, visibility-aware, quantized SVG edge tokens. We first adapt Qwen2.5-VL-7B-Instruct (Bai et al., 2025) through supervised fine-tuning to learn the CAD sequence grammar and establish a stable generation policy. We then apply structure-aware Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which parses and executes sampled sequences and assigns graded rewards according to validity, structural counts, Hungarian-matched sketch geometry, ordered extrusion parameters, and final 3D shape similarity. This design provides useful feedback to both geometrically equivalent and partially correct sequences instead of evaluating every prediction solely against one token-level reference.

The main contributions of this work are summarized as follows:

- A multimodal CAD reconstruction dataset is constructed by pairing multi-view images and textual descriptions with detailed numerical requirements, thereby providing both semantic context and explicit geometric constraints for parametric reconstruction.
- A structure-aware GRPO strategy is developed to evaluate generated CAD programs at multiple levels and extract informative optimization signals from functionally equivalent and partially correct sequences.
- Extensive experiments on the constructed dataset demonstrate that SAR4CAD consistently outperforms existing CAD reconstruction methods across all reported evaluation metrics.

2 Related Work

Parametric CAD sequence reconstruction. Unlike voxels, point clouds, and meshes, parametric CAD programs preserve editable sketches, dimensions, and feature operations, motivating reconstruction methods that predict construction sequences rather than only final geometry. DeepCAD

established the dominant sketch–extrude formulation by serializing CAD commands and geometric parameters for Transformer-based autoencoding and generation (Wu et al., 2021). Subsequent methods improved the expressiveness and controllability of this representation: SkexGen disentangles sketch topology, sketch geometry, and extrusion through separate codebooks (Xu et al., 2022), while HNC-CAD organizes curves, loops, profiles, and solids into hierarchical neural codes (Xu et al., 2023). FlexCAD instead converts the full construction hierarchy into structured text and fine-tunes an LLM to infill masked fields at different hierarchy levels, enabling a single model to support coarse-to-fine controllable generation (Zhang et al., 2025). Conditional reconstruction further introduces visual evidence (Sharma et al., 2018; Ren et al., 2021a). Drawing2CAD directly encodes multi-view vector drawings with coordinate-bearing primitives and uses separate command and argument decoders to recover native CAD sequences (Qin et al., 2025), whereas CAD-MLLM conditions a multimodal language model on text, images, or point clouds (Xu et al., 2024).

Reinforcement and preference post-training. Modern language-model alignment commonly begins with supervised fine-tuning and then incorporates preferences or task rewards. The canonical RLHF pipeline learns a reward model from ranked responses and updates the policy with Proximal Policy Optimization (PPO) (Ouyang et al., 2022; Schulman et al., 2017); PPO stabilizes policy-gradient updates through a clipped surrogate objective, but typically requires both a reward model and a learned value function. Direct Preference Optimization (DPO) derives a direct classification objective from preference pairs, avoiding online rollouts and a separately fitted reward model during policy optimization (Rafailov et al., 2023). Generalized Knowledge Distillation (GKD) addresses a different but complementary problem: it trains a student on its own sampled sequences using token-level feedback from a teacher, reducing train–inference distribution mismatch and allowing on-policy distillation to be combined with reinforcement fine-tuning (Agarwal et al., 2024). Group Relative Policy Optimization (GRPO) removes PPO's learned critic by sampling multiple responses to the same prompt and estimating advantages from their relative rewards (Shao et al., 2024).

Multimodal large language models. Multimodal large language models connect pretrained visual encoders with language models so that images and text can be processed within a generative interface. BLIP jointly supports vision–language understanding and generation through a multimodal encoder–decoder and improves noisy web supervision with caption bootstrapping and filtering (Li et al., 2022); BLIP-2 subsequently bridges frozen image encoders and frozen language models with a lightweight Querying Transformer, substantially reducing the number of trainable parameters (Li et al., 2023). LLaVA connects a pretrained vision encoder to an LLM and uses GPT-generated visual instruction data to develop general-purpose image-grounded dialogue and reasoning (Liu et al., 2023a). The Qwen-VL family extends this paradigm with multilingual image–text understanding, visual grounding, and text reading (Bai et al., 2023); Qwen2.5-VL supports native dynamic-resolution processing and strengthens spatial understanding and structured output (Bai et al., 2025).

3 METHODOLOGY

3.1 PROBLEM DEFINITION

Let $\mathbf{x} = (\mathcal{I}, \mathcal{V}, d)$ denote the multimodal observation of a CAD object, where $\mathcal{I} = \{I_v\}_{v=1}^{4}$ contains one isometric and three orthographic raster views, $\mathcal{V} = \{V_v\}_{v=1}^{4}$ contains the corresponding structured SVG edge records, and d is a concise textual description. Let $\mathbf{y}^* = (y_1^*, \ldots, y_T^*)$ be its reference parametric construction sequence, where T denotes the total number of tokens in the serialized sequence. The sequence organizes curves into loops, loops into faces, and faces into sketches, with each sketch followed by an extrusion block specifying a Boolean operation and 17 numerical parameters. A deterministic CAD execution function $R(\cdot)$ maps every valid sequence $\mathbf{y}$ to a solid model $M = R(\mathbf{y})$. Parametric CAD reconstruction aims to learn a conditional policy $\pi(\cdot)$ that generates a predicted sequence $\hat{\mathbf{y}}$ from the multimodal observation. A successful prediction must be syntactically valid, executable by the CAD kernel, and geometrically consistent with the target object $M^* = R(\mathbf{y}^*)$.

3.2 OUR ARCHITECTURE

Fig. 2 illustrates the complete SAR4CAD pipeline, from the construction of aligned multimodal examples to the generation of an executable parametric model. Training comprises two stages. The

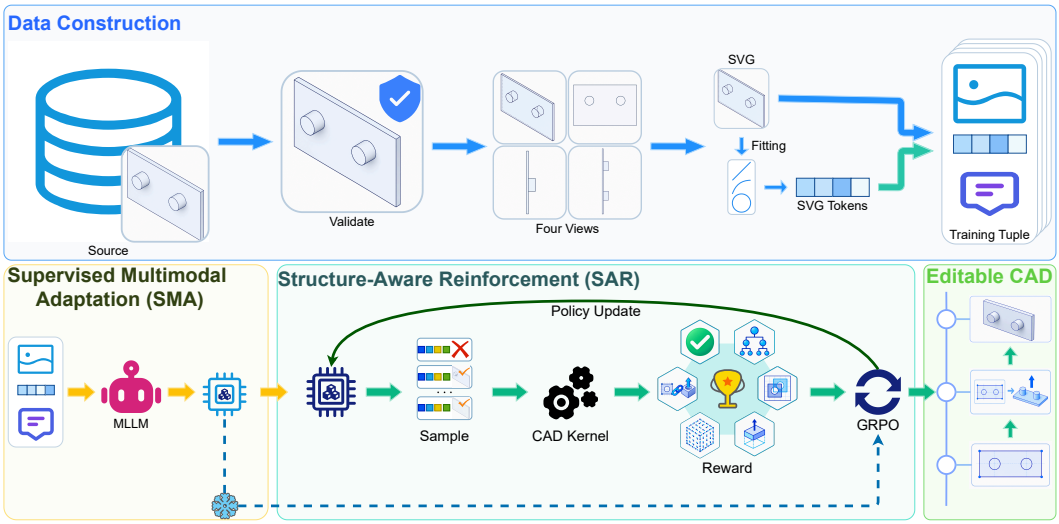

Figure 2: Overview of SAR4CAD with its data construction pipeline. Source sketch–extrude sequences and descriptions are filtered by executability, projected into four raster and vector views, and converted into visibility-aware SVG tokens. The resulting multimodal tuples condition an MLLM generator. SMA learns the native CAD grammar through supervised multimodal adaptation, while SAR samples and executes candidate programs and optimizes them with structure-aware GRPO. The final output is an editable sketch–extrude sequence and its executable solid.

first stage applies Supervised Multimodal Adaptation (SMA). SMA aligns the raster, vector, and textual inputs with the target CAD program and learns a policy that reliably generates programs conforming to the sketch–extrude syntax. The second stage applies Structure-Aware Reinforcement (SAR). For each multimodal observation, the current policy samples a set of candidate CAD programs. A strict parser and OpenCASCADE first filter out candidates that cannot be parsed or executed. Each valid program then receives a composite reward that evaluates six complementary aspects of reconstruction. This two-stage design separates the learning of valid construction syntax from the subsequent non-differentiable optimization of executable geometry.

3.3 Dataset Construction

The experiments use the FlexCAD-formatted training split deduplicated by extrusion sequence. These records originate from the public DeepCAD corpus as preprocessed by SkexGen (Wu et al., 2021; Xu et al., 2022; Zhang et al., 2025). SkexGen provides two training variants, deduplicated at the sketch and extrusion-sequence levels, respectively; the latter is selected to reduce repeated construction patterns during supervised training. Each sequence is linked to the corresponding abstract description released with Text2CAD using its DeepCAD model identifier (Khan et al., 2024).

To assess executability, each source sequence is processed using the CAD conversion pipeline and assigned `valid=true` only if the conversion successfully produces a CAD solid. Only valid records with nonempty Text2CAD abstract descriptions are retained. For each retained sequence, the corresponding STEP solid is imported into CadQuery and projected along the directions $(1, 1, 1)$, $(0, 0, 1)$, $(1, 0, 0)$, and $(0, 1, 0)$ to obtain isometric, top, front, and right views, respectively. CadQuery exports each projection as a 1200×1200 SVG with hidden-line classification enabled. Visible and hidden paths are retained in separate groups, and their stroke widths are normalized using the scale factor encoded in the SVG. CairoSVG subsequently rasterizes each projection as a 2400×2400 PNG image. Any sample missing one or more of the four projections is discarded.

The FlexCAD/SkexGen sequence representation quantizes sketch coordinates as 6-bit integers in the range $0, 63$. Each vector projection is independently normalized to the same 6-bit grid. Specifically, the polylines in each exported SVG are parsed, and every path is fitted to one of three geometric primitives: a line, a circular arc, or a circle. For a fitted edge, let u denote either its x- or y-coordinate, and let $u_{\min}$ and $u_{\max}$ denote the corresponding coordinate-wise bounds computed over all fitted edges in that projection view. All coordinates in the view are quantized using these view-

specific bounds as follows:

$$q(u) = \begin{cases} \mathrm{clip}_{[0,63]}\left(\mathrm{round}\left(63\dfrac{u-u_{\min}}{u_{\max}-u_{\min}}\right)\right), & u_{\max}-u_{\min} > 10^{-10}, \\ 32, & u_{\max}-u_{\min} \le 10^{-10}, \end{cases} \tag{1}$$

where $\mathrm{round}(\cdot)$ maps a real-valued coordinate to its nearest integer, while $\mathrm{clip}_{[0,63]}(\cdot)$ truncates values below 0 to 0 and values above 63 to 63.

Each edge retains its visible or hidden label. Edges are then deduplicated according to their quantized geometry, with visible edges taking precedence when identical geometry is also labeled as hidden. Each view is serialized as follows:

$$V_v = \texttt{[svg_}v\texttt{]}\ e_{v,1}\ldots e_{v,n_v}\ \texttt{[svg_end]}, \tag{2}$$

where $v \in \{\mathrm{iso}, \mathrm{top}, \mathrm{front}, \mathrm{right}\}$ indexes the projection view, n_v is the number of retained edges in view v, and $e_{v,i}$ denotes its i-th serialized edge; hence, e_{v,n_v} is the final edge record of that view. The closing marker `[svg_end]` indicates that no further edge records belong to the current view.

For example, `vis_line,6,11,56,11 <edge_end>` denotes a visible line (i.e., $e_{v,i}$) with quantized endpoints $(6, 11)$ and $(56, 11)$. Hidden lines use the `hid_line` prefix, while arcs and circles follow analogous serialization formats. This construction places every SVG view on the same 6-bit grid as the CAD sequence while preserving within-view relative geometry. Because the views are normalized independently, their tokens do not retain a shared physical scale or coordinate frame across views. Each example therefore comprises four raster views, four SVG edge records, one textual description, and one target sketch–extrude sequence from the same model.

3.4 Supervised Multimodal Adaptation

The SMA stage teaches the multimodal language model the native sketch–extrude grammar before reinforcement learning. Let $\mathcal{D}_{\mathrm{SMA}} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{*(n)})\}_{n=1}^{N}$ denote the supervised corpus constructed in Sec. 3.3, where N is the number of training examples and n indexes an individual example. For each example, the model receives four raster views, their visibility-aware SVG records, and the textual description. Its input is assembled as

$$P_{\mathrm{SMA}}(\mathbf{x}) = \mathrm{Concat}\left(P_{\mathrm{task}}, \mathbf{x}, P_{\mathrm{format}}\right), \tag{3}$$

where P_{task} states the reconstruction task, and P_{format} specifies the required CAD sequence format. The operator $\mathrm{Concat}(\cdot)$ denotes ordered prompt concatenation: it serializes the task instruction, the multimodal observation $\mathbf{x}$, and the format instruction into a single model input while preserving their listed order. Interleaving each image with the SVG record carrying the same view label preserves their correspondence while exposing the model to curve syntax, hierarchy delimiters, quantized parameters, and Boolean operations.

Only the assistant response is supervised; prompt and padding tokens are excluded from the loss. For a reference sequence $\mathbf{y}^* = (y_1^*, \ldots, y_T^*)$, the SMA objective is

$$\mathcal{L}_{\mathrm{SMA}}(\theta) = -\mathbb{E}_{(\mathbf{x},\mathbf{y}^*)\sim\mathcal{D}_{\mathrm{SMA}}}\left[\sum_{t=1}^{T} \log \pi_\theta(y_t^* \mid P_{\mathrm{SMA}}(\mathbf{x}), y_{<t}^*)\right], \tag{4}$$

where θ denotes the trainable model parameters and T is the number of tokens in the reference CAD sequence. Although token-level supervision cannot account for equivalent construction sequences, it supplies the syntactic and structural initialization required by the next stage: without a grammar-aware policy, most sampled programs fail before geometric feedback can be computed.

3.5 Structure-Aware Reinforcement

The SAR stage improves executability and geometric accuracy through the non-differentiable CAD kernel using GRPO (Shao et al., 2024). Let $\mathcal{D}_{\mathrm{SAR}}$ denote the reinforcement-training set. For each pair $(\mathbf{x}, \mathbf{y}^*)$, the behavior policy $\pi_{\theta_{\mathrm{old}}}$ samples G candidate programs $\{\hat{\mathbf{y}}_i\}_{i=1}^{G}$ conditioned on $P_{\mathrm{SMA}}(\mathbf{x})$. Each candidate is parsed, executed with OpenCASCADE, and compared with the reference sequence at complementary levels.

Reward composition. The scalar reward combines conversion validity, structural topology, 2D sketch geometry, ordered extrusion similarity, 3D shape similarity, and joint geometry–extrusion consistency:

$$r = r_{\text{conv}}(\hat{\mathbf{y}})\big(w_{\text{conv}} + w_{\text{topo}} r_{\text{topo}} + w_{\text{geo}} r_{\text{geo}} + w_{\text{ext}} r_{\text{ext}} + w_{\text{3D}} r_{\text{3D}} + w_{\text{joint}} r_{\text{joint}}\big), \tag{5}$$

All six terms lie in $[0, 1]$. A candidate receives $r_{\text{conv}}(\hat{\mathbf{y}}) = 1$ only if it passes the strict parser and OpenCASCADE converts it into a valid solid. Otherwise, its total reward is zero, and the remaining terms are not evaluated.

Structural topology (r_{topo}). This term compares the numbers of sketches, faces, loops, and curves, as well as the distributions of `add`, `cut`, and `intersect` operations. The similarity between two structure counts is their ratio of the smaller value to the larger value; it is 1 if both counts are zero and 0 if only one is zero. Operation similarity is $1 - \frac{1}{2}\|\mathbf{p} - \mathbf{g}\|_1$, where $\mathbf{p}$ and $\mathbf{g}$ are the normalized predicted and reference operation-count vectors, with the same convention for empty vectors. Averaging these five similarities yields r_{topo}.

2D sketch geometry (r_{geo}). This term measures profile accuracy without requiring the predicted and reference sketches to have the same order. Each sketch is rasterized on a 64×64 grid, with outer loops filled and inner loops removed as holes. Let $P_{1:m}$ and $Q_{1:n}$ denote the predicted and reference masks, where m and n are the corresponding sketch counts. Hungarian matching (Kuhn, 1955) pairs the masks by IoU. The score r_{match} is the mean matched IoU after assigning zero to each unmatched sketch. The final score combines this sketch-level agreement with the overlap between the two mask unions:

$$r_{\text{geo}} = \left[w_{\text{union}}\ \text{IoU}\left(\bigcup_{i=1}^{m} P_i, \bigcup_{j=1}^{n} Q_j \right) + w_{\text{match}}\, r_{\text{match}} \right]^2, \tag{6}$$

where w_{union} and w_{match} weight the union-overlap and sketch-matching components, respectively, with $w_{\text{union}} + w_{\text{match}} = 1$. Squaring the weighted sum suppresses inaccurate near-matches.

Extrusion similarity (r_{ext}). This term evaluates construction information absent from 2D masks. Predicted and reference extrusions are paired by sequence position because their order affects the final solid. Each block contains a Boolean operation and a 17-dimensional parameter vector partitioned into extent, translation, rotation, scale, and planar-offset groups of dimensions 2, 3, 9, 1, and 2. For the k-th group in pair j, $s_{j,k}$ is one minus the mean absolute error normalized by the parameter range and then clipped to $[0, 1]$. The range is 63 for 6-bit parameters and 2 for rotation parameters. With group weights $\boldsymbol{\alpha} = (0.25, 0.20, 0.20, 0.15, 0.20)$, the score is

$$r_{\text{ext}} = \frac{1}{\max(N_p, N_g)} \sum_{j=1}^{\min(N_p, N_g)} \mathbb{I}[a_j^{\text{pred}} = a_j^*] \left(\sum_{k=1}^{5} \alpha_k s_{j,k} \right)^2, \tag{7}$$

where N_p and N_g are the predicted and reference extrusion counts, a_j^{pred} and a_j^* are their Boolean operations, and $\mathbb{I}[\cdot]$ is the indicator function. Operation mismatches and unmatched blocks contribute zero.

3D shape similarity (r_{3D}). This term compares the executed solid with the target shape. Up to 2,000 surface points are sampled from the generated STL and compared with the normalized reference point cloud using the bidirectional squared Chamfer distance CD. The score $r_{\text{3D}} = e^{-5\,\text{CD}}$ assigns larger rewards to closer shapes. It is set to zero if the 3D module or reference point cloud is unavailable.

Joint geometry–extrusion consistency (r_{joint}). The final term, $r_{\text{joint}} = r_{\text{geo}} r_{\text{ext}}$, acts as a soft logical AND. It assigns substantial credit only when both the 2D profiles and the ordered extrusion parameters are accurate, preventing either term from compensating for a poor value of the other.

SAR maximizes the clipped GRPO objective while penalizing deviation from the frozen SMA stage reference policy π_{ref}:

$$\mathcal{J}_{\text{SAR}}(\theta) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}^*) \sim \mathcal{D}_{\text{SAR}},\, \{\hat{\mathbf{y}}_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}}$$
$$\left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|\hat{\mathbf{y}}_i|} \sum_{t=1}^{|\hat{\mathbf{y}}_i|} \min\big(\rho_{i,t}(\theta) \hat{A}_{i,t},\ \text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\, \hat{A}_{i,t}\big) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right], \tag{8}$$

where $\rho_{i,t}(\theta) = \pi_\theta(\hat{y}_{i,t} \mid \cdot)/\pi_{\theta_{\text{old}}}(\hat{y}_{i,t} \mid \cdot)$ is the importance-sampling ratio; $\hat{A}_{i,t}$ is the group-normalized advantage estimate; ϵ is the clipping threshold; and β controls the KL divergence penalty. Group-relative advantages eliminate the learned value model, while executable rewards provide graded feedback through the CAD kernel.

4 EXPERIMENTS

4.1 DATASET AND METRICS

Dataset. The data are derived from the publicly available DeepCAD corpus (Wu et al., 2021). We use the extrusion-sequence-deduplicated split released with SkexGen (Xu et al., 2022) and serialize its sketch–extrude programs in the FlexCAD format (Zhang et al., 2025). Of the 103,281 source training records, 67,150 pass the executability filter, 67,144 also contain nonempty descriptions, and 67,143 additionally provide all four rendered views; these 67,143 examples form the Supervised Multimodal Adaptation (SMA) set. A subset of 1,000 examples is used for Structure-Aware Reinforcement (SAR). Applying the same filters to the source test split, excluding overly simple models, and sampling the remainder yields a held-out test set of 2,000 examples.

Metrics. Reconstruction quality is evaluated using average Chamfer distance (Avg-CD), median Chamfer distance (Median-CD), average Jensen–Shannon divergence (Avg-JSD), and validity rate (VR). Avg-CD, defined as the mean CD across individual models, quantifies the overall geometric error (Fan et al., 2017). By contrast, Median-CD is less sensitive to a small number of challenging or degenerate cases. Both metrics are computed over prediction–reference pairs whose predicted sequences can be converted into valid solids. Lower values indicate higher reconstruction accuracy. Avg-JSD quantifies the distributional discrepancy between the generated and reference shape sets. To compute Avg-JSD, the normalized point clouds from all valid prediction–reference pairs are discretized on a 28^3 occupancy grid. The resulting occupancies are then aggregated into separate empirical distributions for the generated and reference sets (Achlioptas et al., 2018), from which a single dataset-level Jensen–Shannon divergence is computed (Lin, 1991). VR measures executability and is defined as the proportion of test predictions that can be successfully parsed and converted by OpenCASCADE into valid 3D solids. Because CD and JSD are computed only for successfully constructed shapes, VR is reported alongside these metrics to capture failures that geometric discrepancies alone cannot reveal.

4.2 IMPLEMENTATION DETAILS

All models are trained using ms-swift (Zhao et al., 2025) on two NVIDIA RTX A6000 GPUs. Qwen2.5-VL-7B-Instruct serves as the base model. During SMA, LoRA (Hu et al., 2022) is applied to all linear layers of the language model, with a rank of 16 and a scaling factor of 64. The vision encoder and aligner remain frozen. SMA is performed for three epochs with a per-device batch size of 4 and four gradient-accumulation steps, yielding an effective global batch size of 32. The learning rate is set to 1×10^{-4}, with a warmup ratio of 0.05, and a maximum sequence length of 10,240 tokens. For SAR, both the trainable policy and the frozen reference policy are initialized from the SMA checkpoint. The SAR policy is adapted using LoRA with a rank of 8 and a scaling factor of 32. SAR training is conducted for at most one epoch on the 1,000-example subset, using a per-device prompt batch size of 3 and $G = 6$ sampled completions for each prompt. The sampling temperature and repetition penalty are set to 0.7 and 1.15, respectively. The learning rate is 4×10^{-6}, with a weight decay of 0.01, a warmup ratio of 0.1, and a KL coefficient of 0.1. The total context length is set to 8,192 tokens, with each sampled completion limited to 2,048 tokens. The weights of the composite reward terms in Eq. 5 are set to 0.02, 0.08, 0.25, 0.35, 0.10, and 0.20, respectively. In Eq. 6, w_{union} and w_{match} are set to 0.4 and 0.6, respectively.

4.3 COMPARISON WITH BASELINES

SAR4CAD is compared with Qwen2.5-VL-7B-Instruct and Drawing2CAD on the test set constructed as described in Sec. 3.3. Samples overlapping with the training data used by Drawing2CAD were excluded from the comparison to prevent data leakage. Test samples without corresponding ground-truth (GT) annotations in the Drawing2CAD dataset were also excluded. After these exclusions, 1,672 of the original 2,000 test samples remained for comparative evaluation. For a fair

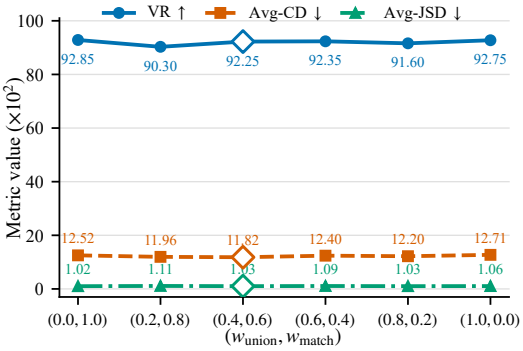

Figure 3: Effect of the 2D sketch geometry reward weights in Eq. 6. The two reward sub-terms enable the model to capture both global and local features simultaneously.

Table 1: Comparison of different methods on the filtered test set. * denotes results obtained on the complete test set. The results for Qwen2.5-VL-7B-Instruct* on the complete test set are omitted because they are identical, thereby saving space.

Benchmark	VR ↑	Avg-JSD ↓	Avg-CD ↓	Median-CD ↓
Qwen2.5-VL-7B-Instruct (4x)	0.00	–	–	–
Drawing2CAD (4x)	69.26	5.05	22.55	11.51
SAR4CAD (4x)	**90.97**	**1.11**	**13.53**	**0.84**
Qwen2.5-VL-7B-Instruct (3x)	0.00	–	–	–
Drawing2CAD (3x)	68.90	4.87	23.73	11.10
SAR4CAD (3x)	**94.73**	**1.12**	**12.39**	**1.13**
Qwen2.5-VL-7B-Instruct (1x)	0.00	–	–	–
Drawing2CAD (1x)	67.88	5.19	24.97	11.78
SAR4CAD (1x)	**94.56**	**1.04**	**12.95**	**1.10**
SAR4CAD* (4x)	**92.25**	**1.03**	**11.82**	**1.13**
SAR4CAD* (3x)	**93.20**	**0.92**	**13.29**	**1.58**
SAR4CAD* (1x)	**93.45**	**1.13**	**12.61**	**1.47**

Table 2: An ablation study was conducted to evaluate the effects of six reward functions.

Variants	r_{conv}	r_{topo}	r_{geo}	r_{ext}	r_{3D}	r_{joint}	VR ↑	Avg-JSD ↓	Avg-CD ↓	Median-CD ↓
①	✔						90.50	1.05	13.57	1.26
②	✔	✔					92.00	1.19	13.80	1.33
③	✔	✔	✔				92.00	1.11	13.85	1.48
④	✔	✔	✔	✔			91.85	1.17	11.97	**1.08**
⑤	✔	✔	✔	✔	✔		91.60	1.13	12.24	1.40
⑥	✔	✔	✔	✔	✔	✔	**92.25**	**1.03**	**11.82**	1.13

comparison, Drawing2CAD was retrained using the officially released reproduction code and evaluated on the resulting filtered test subset. The results of this comparison are reported in Tab. 1. SAR4CAD's performance on the full test set is reported in the final three rows of Tab. 1. For readability, all metric values in the table are multiplied by 100. The best and second-best results are indicated by boldface and underlining, respectively. "3x" denotes the three orthographic views (front, right, and top), "1x" denotes the isometric (Iso) view, and "4x" denotes the complete set comprising all four views. Across all applicable comparisons and input-view configurations, SAR4CAD demonstrates the strongest overall performance. On the filtered test split, Drawing2CAD ranks second, whereas Qwen2.5-VL-7B-Instruct exhibits the worst performance. These results suggest that explicitly modeling the structural and geometric properties of CAD programs facilitates the reconstruction of executable solid models from multi-view drawings.

4.4 Ablation Study

A progressive ablation study assesses the contribution of each reward component to structure-aware reinforcement learning. As shown in Tab. 2, the conversion-validity reward establishes a reliable optimization foundation, while structural supervision improves program executability. Geometry- and extrusion-aware rewards enhance shape reconstruction, although gains are not monotonic across all metrics, reflecting trade-offs among structural correctness, local geometric fidelity, and global shape consistency. The complete formulation achieves the most balanced performance, particularly in distributional similarity and typical-case reconstruction quality, supporting a unified reward that jointly evaluates program validity, construction structure, and geometric consistency.

Tab. 3 examines the contributions of different input modalities. Textual descriptions provide essential semantic guidance and substantially improve the model's ability to generate valid CAD programs. Incorporating SVG tokens provides explicit geometric and structural priors, yielding more consistent shape distributions and higher reconstruction fidelity in typical cases while preserving a high validity rate. These results demonstrate that semantic descriptions and vectorized geometric cues complement visual inputs, thereby reducing reconstruction ambiguity and enhancing structure-aware CAD generation.

Tab. 4 examines the effects of the two-stage training strategy. The original Qwen2.5-VL-7B-Instruct model lacks the task-specific knowledge needed to interpret the specialized serialized-text represen-

Table 3: Ablation of input modalities without SAR.

Benchmark	VR ↑	Avg-JSD ↓	Avg-CD ↓	Median-CD ↓
Images	46.25	2.84	19.41	5.96
Images + description	**92.50**	1.88	19.16	6.68
Images + description + SVG Tokens	89.30	**1.08**	**13.58**	**1.08**

Table 4: Ablation of the two training stages.

Variants	VR ↑	Avg-JSD ↓	Avg-CD ↓	Median-CD ↓
Qwen2.5-VL-7B-Instruct	0.00	–	–	–
+ SMA	89.30	1.08	13.58	**1.08**
+ SAR	**92.25**	**1.03**	**11.82**	1.13

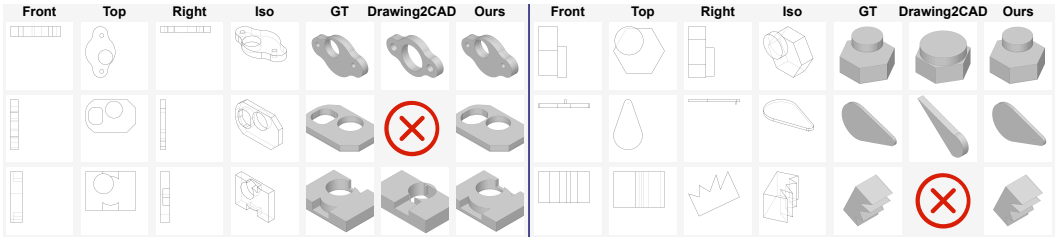

Figure 4: Qualitative results showing the input multi-view drawings, ground-truth CAD models, and reconstruction outputs generated by different methods.

tation and therefore fails to produce valid outputs. SMA enables the model to learn the underlying reconstruction patterns, substantially improving its ability to generate valid outputs. Subsequent SAR with tailored reward functions provides more effective structural and geometric guidance, resulting in further improvements in overall performance.

4.5 Parameter Analysis

The sensitivity analysis presented in Fig. 3 highlights the complementary contributions of the two components defined in Eq. 6. The union-overlap term is more effective in reducing Avg-CD, whereas the Hungarian-matching term yields better VR and Avg-JSD performance. Combining the two terms with nonzero weights achieves a competitive balance across all three metrics. These results suggest that the two reward components provide complementary supervision for distinct aspects of geometric quality.

4.6 Qualitative Results

Fig. 4 presents qualitative results on examples with increasing geometric complexity. In the first row, Drawing2CAD shifts circular holes toward the center of each part, whereas the proposed method recovers their specified positions while preserving the overall geometry. In the second row, which requires precise reasoning about dimensions and relative lengths, Drawing2CAD either fails to generate a valid model or distorts lengths and proportions, while the proposed method closely matches the ground truth. For the complex examples in the third row, the proposed method reliably reconstructs stepped and repeated structures where Drawing2CAD produces invalid or geometrically inaccurate models. Overall, these results demonstrate greater reconstruction accuracy and robustness across positional, dimensional, and structural variations.

5 Conclusion

This work presents SAR4CAD, a structure-aware framework for reconstructing editable parametric CAD models from multi-view drawings. SAR4CAD augments raster views and textual descriptions with visibility-aware SVG edge tokens, providing explicit geometric constraints that reduce the ambiguity of image-only reconstruction. The two-stage training first employs SMA to learn the sketch–extrude grammar. It then applies SAR with CAD-kernel feedback to optimize validity, structural consistency, sketch geometry, extrusion parameters, and 3D shape similarity. Experiments demonstrate that SAR4CAD consistently outperforms the evaluated baselines across input-view configurations. Ablation studies confirm the complementary contributions of structured geometric inputs and multi-level rewards. The current formulation is limited to sketch–extrude programs and independently normalized projection views. These constraints restrict support for broader CAD operations and prevent the SVG tokens from preserving a globally consistent physical scale across views. Future work will expand the reconstruction space to broader parametric operations and explore cross-view geometric representations based on globally consistent coordinate systems.

AI USE STATEMENT

Generative AI tools were used solely to improve the clarity, grammar, and readability of the manuscript. They were not used to formulate the research ideas, design the methodology or experiments, implement the methods, generate or analyze data, interpret the results, or produce scientific claims. The authors reviewed all AI-assisted edits and take full responsibility for the final content of the paper.

ETHICS STATEMENT

This work uses only publicly available CAD datasets and does not involve human subjects, personally identifiable information, sensitive data, or unlawfully obtained data. To the best of our knowledge, the research does not introduce ethical concerns beyond those ordinarily associated with automated CAD reconstruction.

REPRODUCIBILITY STATEMENT

Implementation code and supporting materials are provided in an anonymized repository. The manuscript reports the dataset construction procedure, evaluation metrics, training configuration, and reward formulation needed to understand and reproduce the experiments.

REFERENCES

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 40–49. PMLR, 10–15 Jul 2018.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pp. 628–644. Springer, 2016.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359. Curran Associates, Inc., 2022. doi: 10.52202/068431-1189.

Kevin Ellis, Maxwell Nye, Yewen Pu, Felix Sosa, Josh Tenenbaum, and Armando Solar-Lezama. Write, execute, assess: Program synthesis with a repl. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Kavi Gupta, Peter Ebert Christensen, Xinyun Chen, and Dawn Song. Synthesize, execute and debug: Learning to repair for neural program synthesis. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17685–17695. Curran Associates, Inc., 2020.

Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, Yanning Chen, and Zhipeng Wang. Liger-kernel: Efficient triton kernels for LLM training. In *Championing Open-source DEvelopment in ML Workshop @ ICML25*, 2025.

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 *International Conference on Learning Representations*, 2022.

Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin Sheikh, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. Text2cad: Generating sequential cad designs from beginner-to-expert level text prompts. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 7552–7579. Curran Associates, Inc., 2024.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023a.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9), January 2023b. ISSN 0360-0300.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Feiwei Qin, Shichao Lu, Junhao Hou, Changmiao Wang, Meie Fang, and Ligang Liu. Drawing2cad: Sequence-to-sequence learning for cad generation from vector drawings. In *Proceedings of the 33rd ACM international conference on multimedia*, pp. 10573–10582, 2025.

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.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, 2020.

Daxuan Ren, Jianmin Zheng, Jianfei Cai, Jiatong Li, Haiyong Jiang, Zhongang Cai, Junzhe Zhang, Liang Pan, Mingyuan Zhang, Haiyu Zhao, and Shuai Yi. Csg-stump: A learning friendly csg-like representation for interpretable shape parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12478–12487, October 2021a.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. ZeRO-Offload: Democratizing Billion-Scale model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pp. 551–564. USENIX Association, July 2021b. ISBN 978-1-939133-23-6.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. Csgnet: Neural shape parser for constructive solid geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Maxim Tatarchenko, Stephan R. Richter, Rene 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)*, June 2019.

Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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 (ICCV)*, pp. 6772–6782, October 2021.

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

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 Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24698–24724. PMLR, 17–23 Jul 2022.

Xiang Xu, Pradeep Kumar Jayaraman, Joseph George Lambourne, Karl D.D. Willis, and Yasutaka Furukawa. Hierarchical neural coding for controllable CAD model generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38443–38461. PMLR, 23–29 Jul 2023.

Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, et al. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales. *arXiv preprint arXiv:2308.01320*, 2023.

Zhanwei Zhang, Shizhao Sun, Wenxiao Wang, Deng Cai, and Jiang Bian. Flexcad: Unified and versatile controllable cad generation with fine-tuned large language models. In *International Conference on Learning Representations*, volume 2025, pp. 3204–3227, 2025.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, et al. Swift: a scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 29733–29735, 2025.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023.

OVERVIEW OF SUPPLEMENTARY MATERIAL

This supplementary material provides additional details supporting the main paper, *SAR4CAD: Structure-Aware Reinforcement Learning for Parametric CAD Reconstruction from Multi-View Drawings*. The contents are organized as follows:

- Section A provides additional details on training and evaluation.
- Section B provides the detailed formulation and implementation of the six structure-aware reward components used by SAR.
- Section C catalogs the complete set of prompts utilized throughout the two-stage pipeline.
- Section D presents supplementary experiments.
- Section E provides more qualitative cases.

A ADDITIONAL IMPLEMENTATION DETAILS

Experiments were conducted on a server equipped with two NVIDIA RTX A6000 GPUs, each with 48 GB of VRAM. Both training stages used bfloat16 precision and FlashAttention (Dao et al., 2022). The SMA and SAR stages used DeepSpeed ZeRO-2 and ZeRO-3 optimization (Rajbhandari et al., 2020), respectively. Training was implemented using the ms-swift framework. The Liger Kernel (Hsu et al., 2025) was adopted to provide fused, memory-efficient operators for large language model training under hardware resource constraints. During the SAR stage, rollout generation imposed additional memory pressure (Yao et al., 2023). Therefore, the optimizer states and model parameters were offloaded to CPU memory to prevent out-of-memory (OOM) errors during concurrent training and sampling (Ren et al., 2021b). The SMA and SAR stages required 106 and 28 hours of training, respectively.

Choice of Evaluation Metrics. The command-accuracy and parameter-accuracy metrics adopted by Drawing2CAD (Qin et al., 2025) were not used in this study. These metrics quantify the average position-wise agreement with a single reference sequence. However, a high score does not guarantee that the complete CAD program is executable. A single error in a structurally critical command, delimiter, or parameter field can disrupt the construction hierarchy and prevent successful parsing or solid generation. More generally, token-level agreement does not guarantee behavioral equivalence or even successful execution, motivating the use of execution-aware evaluation in program synthesis (Ellis et al., 2019; Gupta et al., 2020). Conversely, a sequence with different parameter values or a different command order may remain executable and produce a geometrically similar solid. Therefore, the selected metrics directly evaluate the two properties essential to this task: whether the program generates a valid solid and how closely the generated geometry matches the target geometry.

B DETAILED FORMULATION OF STRUCTURE-AWARE REWARDS

This section provides the implementation-level definitions of the reward components summarized in Sec. 3.5. The overall composition and execution gate are already defined in Eq. 5; hence, we focus below on how each component is computed after a candidate has passed that gate.

Conversion gate. The strict parser verifies the complete sketch–extrude hierarchy, including the terminal `<extrude_end>` marker, primitive fields, numerical parameters, and the pairing between sketches and extrusion blocks. The parsed program is then executed in construction order, and OpenCASCADE checks the boundary representation after each `add`, `cut`, or `intersect` operation. These implementation checks operationalize $r_{\text{conv}}(\hat{\mathbf{y}})$: any parsing, conversion, timeout, or solid-validity failure terminates reward evaluation immediately.

Structural topology (r_{topo}). Let c_h^p and c_h^* be the predicted and reference counts at hierarchy level $h \in \mathcal{H} = \{\text{sketch}, \text{face}, \text{loop}, \text{curve}\}$. Their count similarity is

$$s_{\text{cnt}}(a,b) = \begin{cases} 1, & a = b = 0, \\ 0, & \min(a,b) = 0 < \max(a,b), \\ \dfrac{\min(a,b)}{\max(a,b)}, & \text{otherwise}, \end{cases} \tag{9}$$

For the Boolean operations, let $\mathbf{n}^p, \mathbf{n}^* \in \mathbb{N}^3$ contain the numbers of `add`, `cut`, and `intersect` operations, and let $\mathbf{p} = \mathbf{n}^p/\|\mathbf{n}^p\|_1$ and $\mathbf{g} = \mathbf{n}^*/\|\mathbf{n}^*\|_1$. The operation-distribution similarity is the complement of their total variation distance,

$$s_{\text{op}} = 1 - \frac{1}{2}\|\mathbf{p} - \mathbf{g}\|_1, \tag{10}$$

If both operation vectors are empty, s_{op} is defined as 1; if only one is empty, it is 0. The topology reward averages the four hierarchy scores and the operation score:

$$r_{\text{topo}} = \frac{1}{5}\left(\sum_{h\in\mathcal{H}} s_{\text{cnt}}(c_h^p, c_h^*) + s_{\text{op}}\right), \tag{11}$$

This term measures whether the prediction has the correct construction complexity and operation composition, while leaving geometric placement and operation order to the subsequent rewards.

2D sketch geometry (r_{geo}). Each loop is rasterized on the same 64×64 grid used by the quantized sketch coordinates. Within a face, the first loop is filled as the outer boundary and every subsequent loop is subtracted as a hole. Multiple faces in one sketch are merged by union. This produces predicted sketch masks $P_{1:m}$ and reference masks $Q_{1:n}$, together with the composite masks $P_\cup = \bigcup_{i=1}^m P_i$ and $Q_\cup = \bigcup_{j=1}^n Q_j$. For two binary masks A and B, we use

$$\text{IoU}(A,B) = \frac{|A \cap B|}{|A \cup B|}, \tag{12}$$

with a zero score when the union is empty. To remove irrelevant sketch-order differences, we form the matrix $H_{ij} = \text{IoU}(P_i, Q_j)$ and compute a maximum-weight one-to-one matching $\mathcal{M}^*$ using the Hungarian algorithm. Unmatched sketches are assigned zero through the denominator

$$r_{\text{match}} = \frac{1}{\max(m,n)} \max_{\mathcal{M}} \sum_{(i,j)\in\mathcal{M}} H_{ij}, \tag{13}$$

When both sets are empty, r_{match} is defined as 1, whereas it is 0 when exactly one set is empty. The final geometry term is

$$r_{\text{geo}} = \left[0.4\ \text{IoU}(P_\cup, Q_\cup) + 0.6\, r_{\text{match}}\right]^2, \tag{14}$$

The composite IoU supplies a stable global-profile signal, while the matched term preserves the decomposition into individual sketches. The square retains continuous feedback but suppresses predictions that only approximately match the target profiles.

Ordered extrusion similarity (r_{ext}). The 17 numerical parameters of an extrusion are divided into five groups: bidirectional extent (2 values), translation (3), rotation matrix (9), scale (1), and planar offset (2). Let $\mathcal{G}_k$ be the index set of group k, with dimension $d_k = |\mathcal{G}_k|$. For the j-th predicted–reference extrusion pair, its group similarity is

$$s_{j,k} = \text{clip}_{[0,1]}\left(1 - \frac{1}{d_k \Delta_k} \sum_{\ell\in\mathcal{G}_k} \left| z_{j,\ell}^p - z_{j,\ell}^* \right|\right), \tag{15}$$

where $\Delta_k = 63$ for the 6-bit extent, translation, scale, and offset parameters, and $\Delta_k = 2$ for rotation-matrix entries in $\{-1, 0, 1\}$. With group weights $\boldsymbol{\alpha} = (0.25, 0.20, 0.20, 0.15, 0.20)$, the score of the j-th pair is

$$e_j = \mathbb{I}[a_j^p = a_j^*] \left(\sum_{k=1}^5 \alpha_k s_{j,k}\right)^2, \tag{16}$$

Predicted and reference blocks are paired strictly by construction order because permuting Boolean operations can change the resulting solid. If their respective counts are N_p and N_g, unmatched blocks receive zero and

$$r_{\text{ext}} = \frac{1}{\max(N_p, N_g)} \sum_{j=1}^{\min(N_p,N_g)} e_j, \tag{17}$$

The operation indicator provides a hard semantic gate, whereas the squared numerical score distinguishes approximate parameters from accurate construction histories.

3D shape similarity (r_{3D}). The valid predicted solid is tessellated, and up to $2{,}000$ points are sampled from its surface. The reference point cloud is precomputed in the same manner. To match the evaluation protocol, each point cloud X is scaled without centroid subtraction:

$$\widetilde{X} = \left\{ \frac{\mathbf{x}}{s(X)} : \mathbf{x} \in X \right\}, \qquad s(X) = \max_{\mathbf{x}\in X} \|\mathbf{x}\|_\infty, \tag{18}$$

For the normalized predicted and reference point clouds $\widetilde{X}$ and $\widetilde{Y}$, the bidirectional squared Chamfer distance is

$$\text{CD}(\widetilde{X}, \widetilde{Y}) = \frac{1}{|\widetilde{X}|} \sum_{\mathbf{x}\in\widetilde{X}} \min_{\mathbf{y}\in\widetilde{Y}} \|\mathbf{x}-\mathbf{y}\|_2^2 + \frac{1}{|\widetilde{Y}|} \sum_{\mathbf{y}\in\widetilde{Y}} \min_{\mathbf{x}\in\widetilde{X}} \|\mathbf{y}-\mathbf{x}\|_2^2, \tag{19}$$

The distance is mapped to a bounded reward by

$$r_{\text{3D}} = \exp\Big[-5\ \text{CD}(\widetilde{X}, \widetilde{Y})\Big], \tag{20}$$

If the reference point cloud or the 3D comparison module is unavailable, this term is set to zero rather than approximated from the token sequence.

Joint geometry–extrusion consistency (r_{joint}). The geometry and extrusion terms supervise complementary parts of the program: r_{geo} evaluates the unordered set of 2D profiles, whereas r_{ext} evaluates their ordered 3D construction parameters. Their consistency reward is

$$r_{\text{joint}} = r_{\text{geo}}\, r_{\text{ext}}, \tag{21}$$

This product acts as a soft logical AND. A candidate therefore receives substantial joint credit only when both its sketch geometry and its extrusion history agree with the reference, preventing a strong score in one component from masking a failure in the other.

C PROMPT CATALOGS

Following prior studies, prompt engineering is employed to improve the model's interpretation of task instructions and adherence to the required output format (Liu et al., 2023b; Wei et al., 2022; Zhou et al., 2023). Multitask instruction tuning further shows that explicit natural-language task descriptions improve instruction following and generalization to previously unseen tasks (Sanh et al., 2022; Chung et al., 2024). This section outlines the prompts employed in the training phase, specifically P_{task} and P_{format}, as defined in Eq. 3.

Input

P_{task}**:**

You are a senior CAD engineer. There is now a new structure for representing real-world 3D CAD models. The following is a detailed description of this structure.

1. Basic Unit: Curve

*Each curve type is represented by its geometric points. All coordinates are discretized into integers within a **0–63 range** (i.e., on a 64×64 grid) and directly written as decimal numbers in the text.*

- A "curve" (i.e., line, arc, or circle) is the lowest-level primitive.
- A circle is defined by four uniformly sampled points on its circumference.
- An arc is defined by three points but with two tokens, where the third point is specified by the next curve (or the first curve when a loop is closed).
- A line is defined by start point.
- A point is represented by two integers which stands for the x and y coordinate, respectively.

2. Hierarchical Composition Rules

- A "sketch" is formed by one or multiple faces.
- A "face" is a 2D area bounded by loops.
- A "loop" is a closed path, consisting of one (i.e., circle) or multiple curves (e.g., line-arc-line).
- A loop with a circle can not contain additional curves since it is already a closed path.
- When a face consists of multiple loops, the first loop defines the external boundary, and the remaining loops define internal loops (i.e., holes).

3. Extrusion Parameters (18-Dimensional)

Each sketch will be followed by an extrude, which is represented by 18 parameters: BVVTTTRRRRRRRRRSOO
- B represents one of the three Boolean operations: add, cut or intersect. It occupies 1 parameter.
- V indicates the displacements of the top and the bottom planes from the reference plane in which a sketch is extruded to form a solid. It occupies 2 parameters.
- T represents 3D translation applied to the extruded solid. It occupies 3 parameters.
- R represents 3D rotation of the extrusion direction. It occupies 9 parameters.
- S represents the uniform scaling factor. It occupies 1 parameter.
- O represents the center of scaling as a 2D coordinate. It occupies 2 parameters.
- Note that every number is an integer.

4. SVG Projection Edge Information (Supplementary Input)

In addition to the multi-view images, you will receive structured edge data extracted from CAD SVG projections (generated via CadQuery's HLR — Hidden Line Removal algorithm). This data provides precise 2D geometric information from four orthographic viewpoints: Iso ('[svg_iso]'), Top ('[svg_top]'), Front ('[svg_front]'), and Right ('[svg_right]').

Each viewpoint contains projection edges in the following format:
- '[svg_view]' marks the start of a viewpoint's edge data
- Each edge is one line: '{vis |hid}_{type},{coordinates} <edge_end>'
- 'vis_' prefix = visible edge (solid outline in the projection)
- 'hid_' prefix = hidden edge (dashed line, occluded by solid body)
- type = 'line' (2 points: x1,y1,x2,y2), 'arc' (3 points: start,mid,end), or 'circle' (4 equidistant points)
*- All coordinates are integers on the same **0–63 grid** as the CAD output sequence*
- '[svg_end]' marks the end of a viewpoint's edge data

*Each SVG projection is independently normalized to the same **0–63 grid resolution** as the CAD output sequence, without a shared cross-view coordinate system. Use this supplementary information to understand the 3D structure, especially the relative edge positions within each view and visible/hidden relationships.*

Input

P_{format}**:**

Output only the sequence using: line,x,y <curve_end> | arc,x1,y1,x2,y2 <curve_end> | circle,... <curve_end> | <loop_end> <face_end> <sketch_end> | add/cut/intersect, 18_params <extrude_end>

D Additional Experiments

Table 5: Leave-one-out ablation of the structure-aware reward formulation. When r_{geo} or r_{ext} is removed, r_{joint} is disabled accordingly because it depends on both components.

Variants	r_{conv}	r_{topo}	r_{geo}	r_{ext}	r_{3D}	r_{joint}	VR ↑	Avg-JSD ↓	Avg-CD ↓	Median-CD ↓
①	✔		✔	✔	✔	✔	91.00	1.18	13.46	1.26
②	✔	✔		✔	✔		**92.40**	<u>1.06</u>	13.16	1.24
③	✔	✔	✔		✔		91.55	1.14	13.06	1.17
④	✔	✔	✔	✔		✔	91.85	1.13	12.72	**1.07**
⑤	✔	✔	✔	✔	✔		91.60	1.13	<u>12.24</u>	1.40
⑥	✔	✔	✔	✔	✔	✔	<u>92.25</u>	**1.03**	**11.82**	<u>1.13</u>

Complementing the progressive ablation study in Sec. 4.4, Tab. 5 evaluates the full reward formulation by removing each component individually. This leave-one-out analysis determines whether each reward remains necessary in the presence of all other supervisory signals, thereby mitigating the influence of the order in which components are introduced. The results demonstrate that the reward terms are complementary rather than interchangeable. Topological supervision constrains the construction hierarchy, whereas the geometry and extrusion rewards capture distinct aspects of the underlying CAD program. The joint reward further couples the geometry and extrusion supervision, while the 3D reward evaluates their combined geometric outcome at the solid level. Although removing a component may improve an individual metric, no ablated formulation achieves the same balance among program validity, distributional consistency, and shape fidelity as the full objective. Together with the progressive ablation results in Tab. 2, these findings support the multi-level reward design adopted in SAR.

E MORE QUALITATIVE CASES

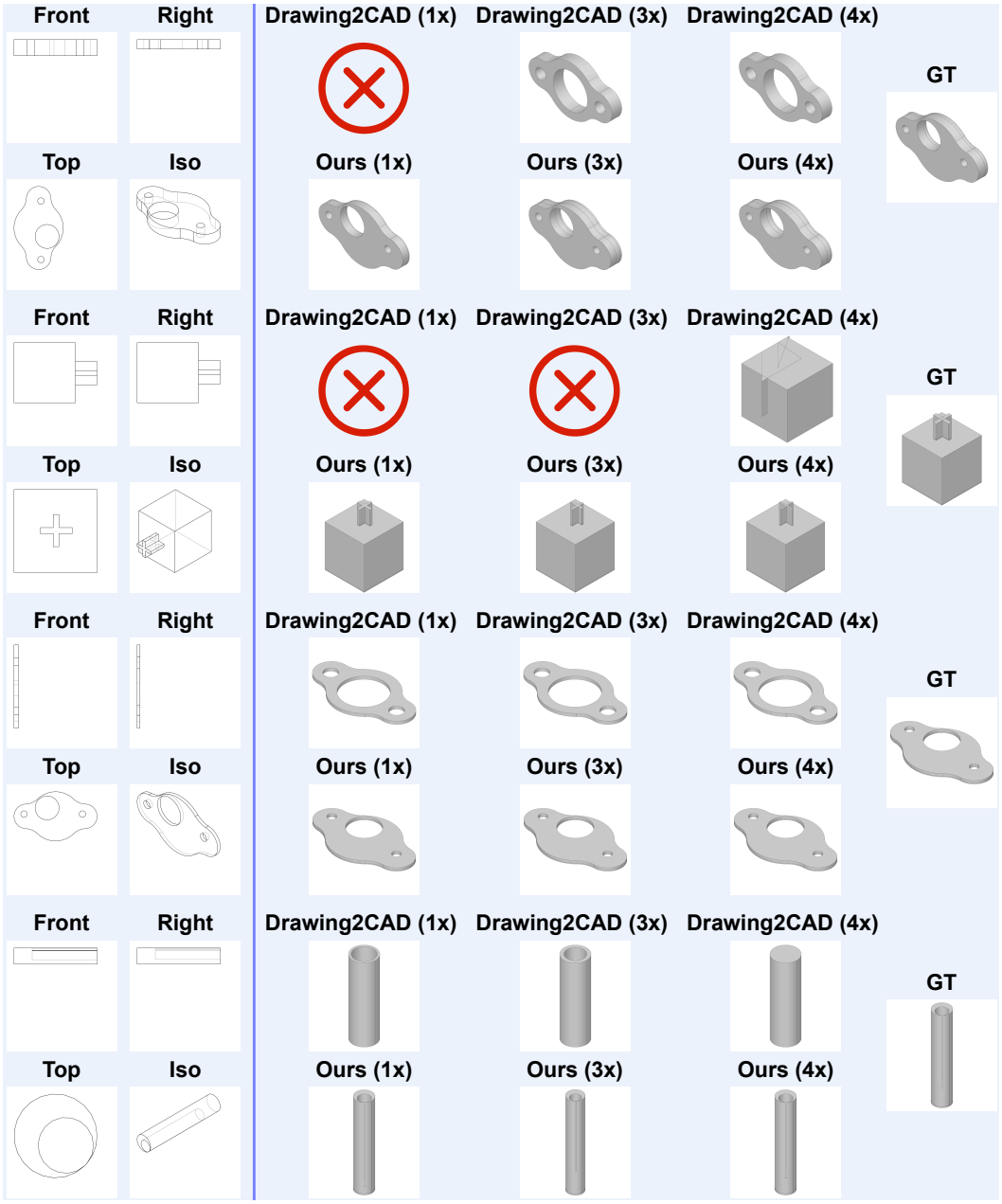

Figure S1: Additional qualitative comparisons across different numbers of input views.

Fig. S1 provides additional comparisons under one-, three-, and four-view inputs. This trend is consistent with prior multi-view reconstruction studies, where additional observations and cross-view consistency reduce uncertainty in the inferred three-dimensional shape (Choy et al., 2016; Tulsiani et al., 2017). The first and third examples reinforce the positional trend observed in the main paper: our method accurately recovers the off-center circular holes, whereas Drawing2CAD exhibits a systematic center bias and moves the holes toward the middle of the part. The second and fourth examples further highlight the benefit of the supplementary SVG tokens during generation. Their explicit edge and primitive cues help the model distinguish fine-grained construction operations, including `add` and `cut`. Consequently, our method reconstructs the small cross-shaped protrusion in the second example and preserves the hollow tube produced by a cut operation in the fourth, while Drawing2CAD frequently generates an invalid program or omits the required feature. These cases

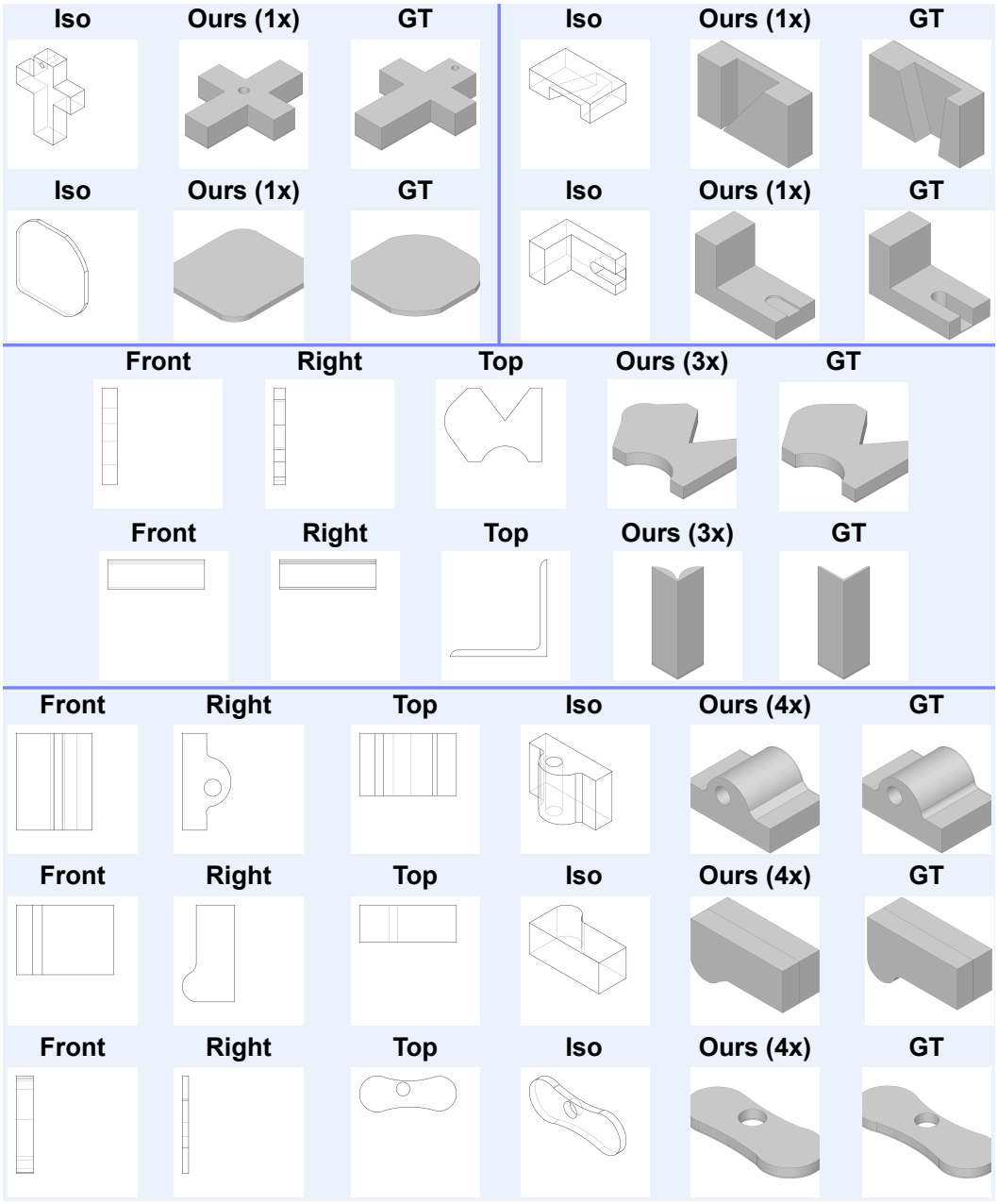

Figure S2: Representative failure cases of our method under incomplete or geometrically ambiguous visual evidence.

show that the proposed representation improves both spatial localization and the recovery of detailed Boolean construction steps across different numbers of input views.

Fig. S2 illustrates the remaining ambiguity of reconstructing a parametric solid from incomplete visual evidence. Such ambiguity is intrinsic to reconstruction from incomplete visual evidence, for which a model may rely on learned category-level priors instead of recovering the instance-specific geometry (Tatarchenko et al., 2019). In the two examples in the first row, perspective and occlusion do not provide sufficient constraints to determine the hole locations unambiguously. Unlike a human designer, who may infer the intended placement from design conventions or prior experience, the model relies on the observed evidence and therefore places the holes incorrectly. A second failure mode arises for consecutive, small-radius curves. This behavior appears in the first example of the second row and recurs throughout the later curve-dominated cases: subtle differences between curve primitives and curvatures occupy only a few pixels or quantized SVG coordinates, so the

model may confuse the curve type or estimate an incorrect radius, leading to visible deviations from the ground-truth boundary. The second example in the second row further suggests that perspective ambiguity can propagate beyond the sketch: the model misinterprets the final extrusion operation and produces an incorrect three-dimensional extent. Overall, these cases indicate that the principal failures occur when the available views underconstrain feature placement or provide insufficient resolution to distinguish fine curved geometry and its associated extrusion parameters.