ReACT: Reward-informed Autoregressive Decision CAD Transformer

Yijie Ding[1], **Yang Liu**[2,1], **Haobo Jiang**[1], **Jianmin Zheng**[1*]

[1]Nanyang Technological University, Singapore
[2]Beijing Institute of Technology, China
yijie002@e.ntu.edu.sg, liuyang@bit.edu.cn, {haobo.jiang, ASJMZheng}@ntu.edu.sg

Abstract

Reconstructing precise CAD modeling sequences from point clouds remains a challenging task, especially for objects with complex geometry and topology. In this paper, by formulating the CAD sequence reconstruction as a Markov decision process, we introduce *ReACT*, a novel Reward-informed Autoregressive decision Cad Transformer architecture for robust CAD sequence prediction. Beyond previous imitation-only approaches, our key innovation is to frame the CAD Transformer under a reinforcement learning paradigm and thereby integrate reward-inspired heuristic learning into our architecture. This allows *ReACT* to effectively leverage shape-aware long-term reward feedback to guide the inference of (nearly) optimal CAD commands. Specifically, conditioned on past tokens, comprising the historical CAD states, sketch-extrude commands (i.e., actions) and associated geometric rewards, *ReACT* autoregressively outputs the most promising CAD commands in a causal manner. In particular, we develop a novel scaffold-aware CAD state representation that integrates global point-command features with an incrementally constructed surface point scaffold, enabling fine-grained geometric reasoning for subsequent reconstruction prediction. Moreover, an effective local barrel points-guided dense reward function is designed to jointly evaluate surface fidelity and command efficiency for reliable reward guidance. Extensive evaluations on the DeepCAD and Fusion360 benchmarks demonstrate that ReACT can achieve superior CAD reconstruction quality, even for objects with complex shapes.

Introduction

Computer-Aided Design (CAD) is a fundamental pillar of modern industrial design, and CAD models provide a digital representation of 3D products in various forms. This paper considers the problem of inferring CAD modeling sequences from point clouds, which is challenging, particularly for a model with complex topologies and geometry. Point clouds, as raw geometric representations, are easily obtained through 3D scanning devices or computer vision techniques. However, they lack semantic meaning and are not well-suited for high-level editing. In contrast, CAD modeling sequences not only define the final geometric representation of a product but also capture the underlying design process such as sketches, extrusions, and Boolean operations. These sequences play a fundamental role in feature-based modeling (Xu et al. 2021), a widely adopted paradigm in contemporary CAD systems.

Figure 1: CAD models (green), their full surface point clouds (white), and scaffolds (pink) formed by the cumulative LBPs extracted after each sketch loop closure. These scaffolds capture extrusion sidewalls and sketch contours, providing progressive structural context for ReACT's scaffold-aware geometric attention.

Existing work has made substantial progress in generating CAD command sequences from a range of inputs: point clouds (Dupont et al. 2024; Liu et al. 2025), text prompts (Wang et al. 2025b; Khan et al. 2024b) and mixed visual and linguistic cues (Ma et al. 2024; Xu et al. 2024a). Nearly all of these methods, however, are trained purely by imitation: they maximize the likelihood of the ground-truth trace conditioned on the input and then decode tokens to form a command sequence. This objective overlooks two signals inherently encoded in the sequence itself. First, the step-wise ordering records which geometric regions have already been reconstructed and which remain, a progressive context that is vital for guiding subsequent operations. Second, CAD sequence, as a language, reflects human-interpretable structural cues that help regularize long-horizon planning. When both types of evidence are ignored, decoders struggle to allocate tokens across intricate sub-structures and frequently collapse on parts with complex topology or extended design histories.

*Corresponding author: asjmzheng@ntu.edu.sg

To overcome these limitations, we recast CAD sequence reconstruction as a Markov Decision Process (MDP), enabling a shift from pure imitation learning to reward-informed decision making. Drawing inspiration from Decision Transformer (Chen et al. 2021), we formulate CAD modeling as an offline, reward-conditioned sequence modeling problem. In this setting, each sketch-extrude command is treated as an *action*, selected to maximize a dense, geometry-informed *reward* based on a scaffold-aware *state*. This formulation enables three core advantages: (i) it injects local geometric cues and progressive reconstruction context into the prediction process, (ii) it shapes long-horizon planning via a reward signal aligned with desirable modeling structure, and (iii) it turns the temporal credit assignment problem into an autoregressive token prediction task, naturally suited for transformer architectures.

Building on this formulation, we introduce **ReACT**, a **Re**ward-informed **A**utoregressive decision **C**AD **T**ransformer. To the best of our knowledge, ReACT is the first approach to frame CAD reconstruction from point clouds as an offline decision-making process. It autoregressively generates modeling commands conditioned not only on past actions but also on scaffold-aware geometric states and step-wise rewards. This allows the model to plan with structural foresight and geometric precision, enabling robust sequence recovery even for models with intricate topologies and deep design hierarchies. Figure 1 illustrates the scaffolds extracted throughout the reconstruction process, which progressively capture the evolving geometry.

The main contributions of the paper are:

(1) We propose a reward-informed autoregressive decision CAD Transformer, which formulates CAD sequence reconstruction as an offline, reward-conditioned decision-making process and unifies geometric reasoning and autoregressive modeling in a single framework.

(2) We develop a scaffold-aware geometric state representation that integrates local point-wise features with the evolving reconstruction context, enabling fine-grained and progressive guidance for CAD command prediction.

(3) We introduce a Local Barrel Point (LBP)-guided dense reward function that provides step-wise alignment feedback by combining robust surface matching, geometric consistency, and modeling efficiency.

Extensive experiments on the DeepCAD and Fusion360 benchmarks show that ReACT consistently outperforms state-of-the-art baselines in both sequence accuracy and geometric fidelity. Beyond faithfully recovering design intent, ReACT exhibits the ability to generate shorter and more efficient modeling sequences than the ground truth, highlighting its capacity to generalize structural priors and optimize for design efficiency, even outside the training distribution.

Related Work

CAD Modeling Paradigms. Learning-based CAD methods can be grouped by the underlying representations they target: Constructive Solid Geometry (CSG), Boundary Representation (B-Rep), or procedural command sequences.

CSG models 3D shapes by combining primitive solids through Boolean operations, and learning-based variants generate CAD models by synthesizing CSG trees or selecting primitive operations and parameters (Ren et al. 2021; Yu et al. 2021). Domain-specific languages further express programs over primitive sets (Sharma et al. 2018; Mo et al. 2019), but these approaches remain limited by fixed vocabularies and lack the expressiveness needed for real-world CAD geometry. In contrast, *B-Rep* encodes geometry as surfaces and edges with explicit topology, and has seen success in both generative (Xu et al. 2024b; Li, Fu, and Chen 2025; Wu, Xiao, and Zheng 2021) and reconstruction tasks (Zhou, Tang, and Zhou 2023; Liu et al. 2024c; Fu et al. 2023, 2024; Jiang et al. 2023a,b). While highly editable and compatible with industrial tools, B-Rep omits procedural modeling history, making them less suited for applications requiring design interpretability or modification tracking.

Sequence-based modeling instead captures the full procedural history of the design process. Real-world CAD workflows build models by sequentially applying parameterized operations (e.g., sketch and extrude), providing interpretable structure and explicit design intent. Recent works (Xu et al. 2023, 2022) explore this generatively: HNC represents hierarchical sketch trees, SkexGen uses disentangled codebooks for topology and geometry, and DeepCAD employs an autoencoder to generate command sequences. Beyond generation, various methods recover procedural sequences from downstream modalities including point clouds (Khan et al. 2024a; Liu et al. 2025), voxel grids (Li et al. 2024, 2023), rendered images (Chen et al. 2024, 2025), and multimodal inputs (Ma et al. 2023, 2024).

Focusing on reverse-engineering command sequences from point clouds, prior work has explored primitive-specific strategies, such as fitting extruded cylinders (Uy et al. 2022), identifying sketchable regions via barrel points (Wang et al. 2025a), or extracting global sketch structures for sequence enhancement (Khan et al. 2024a). In contrast, we adopt a more holistic use of barrel points—not only as intermediate geometric proxies, but also as reward signals—to guide the model across the entire reconstruction process. Rather than modeling isolated sketch or extrusion steps, our approach fuses local geometric cues with global sequence context to support end-to-end procedural reconstruction.

RL for CAD Modeling. Reinforcement Learning (RL) has recently been introduced into CAD modeling as a means to learn sequential operations in a data-driven manner. For example, RLCAD (Yin et al. 2025) formulates modeling as a sequential decision-making problem using online RL, training agents to learn extrude and revolve operations in a Parasolid-based environment. Similarly, (Zhang et al. 2025) employs Deep Q-Networks to translate 2D drawings into modeling commands. However, these methods rely on handcrafted, simulation-based environments tailored to online training, which restricts scalability and limits the use of large-scale datasets such as DeepCAD (Wu, Xiao, and Zheng 2021) and Fusion 360 Gallery (Willis et al. 2021).

Recent work has applied RL concepts to fine-tune large language models (LLMs). CADCrafter (Chen et al. 2025) uses a code checker as an implicit reward function dur-

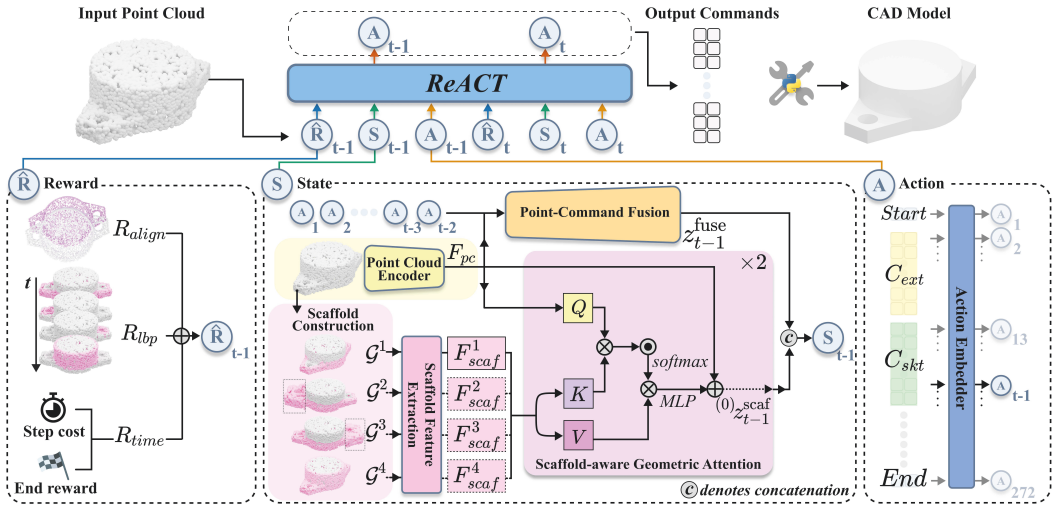

Figure 2: Overview of the ReACT architecture. Given a point cloud, ReACT autoregressively predicts a parametric command sequence that reconstructs the CAD model. At each step, a scaffold-aware geometric state is formed via point-command fusion and scaffold attention, then passed into a causal Transformer conditioned on $\hat{R}_t$ and past trajectories. The output is decoded by a CAD kernel into a B-Rep model. $\mathcal{G}^1, \mathcal{G}^2, \ldots$ denote accumulated scaffolds after each sketch loop closure.

ing LLM tuning. Cadrille (Kolodiazhnyi et al. 2025) adopts GRPO (Shao et al. 2024), an online RL algorithm, for fine-tuning via online feedback.

We take a different approach by proposing an offline, reward-conditioned framework for CAD modeling. To bridge raw point cloud inputs with structured CAD command sequences, we redefine the standard RL formulation (state, action and return), and introduce a dense reward shaping mechanism guided by Local Barrel Points (LBPs), which provides step-wise supervision by scoring each sketch-extrude decision against the ground-truth geometry. This departs from prior works that rely on sparse or terminal rewards (Lin et al. 2020; Yin et al. 2025; Zhang et al. 2025; Kolodiazhnyi et al. 2025; Jiang, Xie, and Yang 2021), enabling fine-grained credit assignment throughout the modeling trajectory. Our method eliminates the need for handcrafted simulators or online interaction, making it scalable to large datasets and diverse operations.

Method

Problem Statement

ReACT aims to reconstruct CAD command sequences from 3D point clouds. Given a point cloud $\mathcal{X} \in \mathbb{R}^{n\times 3}$ sampled from a CAD model surface, our goal is to recover a command sequence $\Phi : \mathcal{X}^{n\times 3} \to C = \{C_1, C_2, \ldots, C_T\}$ such that executing C in a CAD kernel yields a model M that matches the original geometry and topology. Each command $C_t = \{skt, ext\}$ is composed of a 2D sketch and an extrusion operation. Sketches contain closed loops defined by curve primitives (lines, arcs, circles), forming the semantic and geometric basis for solid construction.

Reward-informed Decision CAD Transformer

We model CAD sequence reconstruction as a reward-informed decision-making process and instantiate it with an autoregressive transformer architecture, as shown in Figure 2. At each modeling step t, the model observes a structured *state* $s_t \in \mathcal{S}$, selects an *action* $a_t \in \mathcal{A}$ from a discrete command vocabulary, and receives a *reward* $r_t \in \mathcal{R}$ that scores geometric fidelity and structural efficiency. Together, these elements define an MDP tailored for procedural CAD reconstruction.

Our objective is to learn a policy $\pi_\theta(a|s)$ that maximizes the expected cumulative reward $\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T} r_t\right]$, where the trajectory $\tau = (s_1, a_1, r_1, \ldots, s_T, a_T, r_T)$ spans a fixed horizon T. Following (Chen et al. 2021), we reparameterize the reward using a *return-to-go* (RTG) signal $\hat{R}_t = \sum_{t'=t}^{T} r_{t'}$, which provides a shaped objective reflecting the remaining potential reward at each step.

The model thus processes an interleaved trajectory:

$$\tau = (\hat{R}_1, s_1, a_1, \ldots, \hat{R}_T, s_T, a_T), \tag{1}$$

encoding return-to-go, scaffold-aware states, and command tokens in a sequential fashion. A *causal transformer decoder* autoregressively consumes this sequence to predict the next action token at each step, conditioned on the full history.

Unlike conventional RL environments (e.g., Atari), our setting must reason over unordered point sets and deeply structured, parameter-rich command sequences. To support this, ReACT builds a *scaffold-aware geometric state presentation* on the fly: each s_t encodes point-command features augmented with a geometric scaffold that tracks reconstructed extrusion sidewall patches. This hybrid signal offers both fine-grained local cues and progressive global context for the transformer to attend to. Each predicted sketch-

extrude command is evaluated using a *LBP-guided dense reward* function. This signal combines a robust Chamfer Distance term for surface alignment, a consistency bonus derived from LBP geometry, and a time penalty to encourage succinct modeling traces. Unlike sparse or terminal rewards in prior works (Yin et al. 2025; Kolodiazhnyi et al. 2025; Liu et al. 2024a,b, 2023), this dense feedback enables fine credit assignment throughout the modeling trajectory.

During training, ReACT is optimized via teacher forcing with precomputed rewards. The two discrete parameters (p_t^x, p_t^y) of each command are predicted independently using masked cross-entropy losses with label smoothing. The losses are averaged across time and channels, with padded tokens excluded via key masks. At inference time, ReACT autoregressively predicts actions, updating the state and return-to-go after each step until the end token is reached. The resulting command sequence is executed in a CAD kernel to generate the target B-Rep model.

Scaffold-aware Geometric State Representation

We design the state representation in ReACT to capture the evolving structural context of the reconstructed shape and the alignment between point-level geometry and command tokens. Unlike prior work that only encodes static geometric input, ReACT maintains a dynamic understanding of partially generated geometry to better guide long-horizon planning. To realize this, we introduce a scaffold-aware state representation built from LBPs, which accumulate step-by-step to reflect the evolving reconstruction. These scaffold features act as an intermediate geometric memory that grounds future predictions in previously generated structure.

Scaffold Construction Given an input point cloud $\mathcal{X} = \{x_i\}_{i=1}^n \subset \mathbb{R}^6$ where each point encodes both 3D coordinates and surface normals, we first extract a per-point feature map $F_{pc} \in \mathbb{R}^{n\times d}$ with a point cloud encoder. To provide ReACT with evolving structural context during reconstruction, we extract geometric cues from $\mathcal{X}$ on a per-loop basis and inject them into the model via a dedicated *scaffold-aware geometric attention* module to enhance structural awareness.

At timestep t, we consider the current extrude-sketch pair $C_t = (ext, skt)$, defined by an extrusion axis $k_t \in \mathbb{S}^2$, a plane origin $o_t \in \mathbb{R}^3$, and a height $h_t > 0$. For any point $p \in \mathcal{X}$, we define its projection onto the sketch plane and displacement along the extrusion axis as: $\Pi_t(p) = p - \big((p - o_t)\cdot k_t\big)k_t, u_t(p) = (p - o_t)\cdot k_t$, where Π_t denotes the orthogonal projection of p onto the sketch plane, and $u_t(p)$ is its signed height along the extrusion axis. Let ℓ_t denote the sketch loop at step t. After transforming ℓ_t to global coordinates, we express both the loop and the projected point $\Pi_t(p)$ in the same 2D sketch-plane coordinate frame and uniformly sample a dense set $\mathcal{L}_t = \{q_j\}$ on the 2D contour. The in-plane distance is then $d_{\ell_t}(p) = \min_j \|\Pi_t(p) - q_j\|_2$.

Using two tolerance parameters, ϵ_{skt} (in-plane proximity) and ϵ_h (height margin), we define the LBP set $\mathcal{B}_t$ at step t as

$$\mathcal{B}_t = \left\{ p \in \mathcal{X} \;\middle|\; \begin{array}{l} -\epsilon_h \le u_t(p) \le h_t + \epsilon_h, \\ d_{\ell_t}(p) \le \epsilon_{skt} \end{array} \right\}. \quad (2)$$

A point is tagged as an LBP if it lies within the extrusion slab $[0, h_t]$ with an extra height margin ϵ_h on both sides, and projects within ϵ_{skt} of the loop contour (Figure 3).

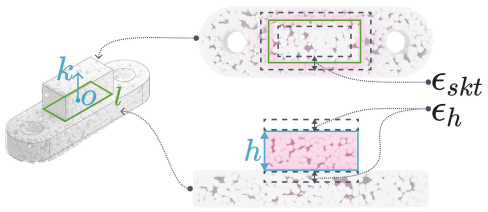

Figure 3: Illustration of LBP extraction at a single decoding step. Points (pink) are selected as LBP if they lie within the sketch-plane-aligned extrusion slab, defined by the height bounds $-\epsilon_h \le u_t(p) \le h_t + \epsilon_h$, and if their projected distance to the sketch loop satisfies $d_{\ell_t}(p) \le \epsilon_{skt}$.

To represent the evolving reconstruction state, we define the *scaffolding regions* as the union of all LBPs collected up to the current timestep: $\mathcal{G}_t = \bigcup_{i=1}^{t-1} \mathcal{B}_i$. This cumulative structure captures the progressively reconstructed geometry and acts as a conditioning signal for downstream decoding.

Scaffold-Aware Geometric Attention To inject geometric priors from the scaffolding regions, we gather point-wise features from $\mathcal{G}_t$ via the global feature map $F_{pc} \in \mathbb{R}^{n\times d}$ and introduce a scaffold attention module comprising multiple cross-attention (CA) layers (Vaswani et al. 2017), where the full feature set F_{pc} serves as queries, and scaffold features $F_t^{scaf} = \{f_i \in F_{pc} \mid x_i \in \mathcal{G}_t\}$ serve as keys and values. The cross-attention operation is defined as:

$$\begin{aligned} A_t &= \mathrm{softmax}\left(\phi_Q(F_{pc}) \cdot \phi_K(F_t^{scaf})^\top / \sqrt{d_k}\right), \\ \mathrm{CA}(F_{pc}, F_t^{scaf}) &= A_t \cdot \phi_V(F_t^{scaf}), \end{aligned} \quad (3)$$

where ϕ_Q, ϕ_K, and ϕ_V denote learnable linear projections and d_k denotes the feature dimension of each head. This formulation enables each point to selectively attend to scaffold geometry, enriching the global features with structural awareness. The resulting attended feature is then passed through a feed-forward network (FFN) with a residual connection to produce Z_t^{scaf}, from which we apply row-wise max pooling to extract a global highlight token summarizing the reconstructed scaffold geometry.

Point-Command Fusion To couple command tokens with the underlying geometry, we introduce a *point-command fusion* decoder. Each CAD command at step t is quantized to a two-dimensional token (p_t^x, p_t^y); the parameters are one-hot encoded, embedded, and augmented with a learnable positional encoding $e_{pos}(t)$, yielding

$$z_t = e^x(p_t^x) + e^y(p_t^y) + e_{pos}, \quad Z_0 = \big[z_1, \ldots, z_T\big]. \quad (4)$$

A stack of D_{fuse} Transformer-decoder layer refines Z_0. At layer l, token features first undergo masked self-attention $Z_l^{msa} = \mathrm{MSA}\big(Z_{l-1}; M_{cmd}\big) + Z_{l-1}$, where M_{cmd} is the standard autoregressive mask. From a designated depth $D_{ca} \le D_{fuse}$ onward, token features additionally attend to the point cloud feature F_{pc} through CA: $\tilde{Z}_l^{ca} =$

$\mathrm{CA}\big(Z_l^{msa}, F_{pc}\big)$. Earlier layers $l < D_{ca}$ skip CA, allowing intra-sequence patterns to form before geometry is introduced. Each block then passes $\tilde{Z}_l^{ca}$ through an FFN with residual connections. The output from the final layer, Z_t^{fuse}, encodes geometry-aware command features. To construct the scaffold-aware geometric state used in trajectory learning, we concatenate the fusion feature Z_t^{fuse} with the scaffold token Z_t^{scaf}, followed by a projection:

$$s_t = \text{Linear}(\text{concat}(Z_t^{\text{fuse}}, Z_t^{\text{scaf}})). \tag{5}$$

LBP-guided Dense Reward Shaping

To provide human-interpretable and geometry-aware intermediate feedback during autoregressive reconstruction, we design a dense reward function guided by LBPs. This reward evaluates reconstruction quality from two perspectives: *spatial fidelity* and *time efficiency*:

$$R = \underbrace{R_{\text{align}} + R_{\text{lbp}}}_{\text{spatial}} + R_{time} \tag{6}$$

where the spatial quality assessment contains a global alignment score R_{align} on a per-sketch basis and an R_{lbp} LBP coverage reward on a per-loop basis.

Spatial Quality To assess the geometric fidelity of the partially reconstructed shape $\mathcal{Y}_t$ with respect to the target point cloud $\mathcal{X}$, we adopt the *Robust Chamfer Distance* (RCD) metric proposed in (Jiang et al. 2021), which mitigates outlier effects due to incomplete or partial geometry. Let the nearest-neighbor distance be $d(p, \mathcal{Q}) = \min_{q\in\mathcal{Q}} \|p - q\|_2$. We then define a soft inlier kernel $\psi(d; \epsilon) = max(0, 1-d/\epsilon)$ where $\epsilon > 0$ is an inlier tolerance. This kernel down-weights points farther from their nearest matches, assigning zero to those outside the tolerance band:

$$\begin{aligned} R_{align}(\mathcal{Y}_t, \mathcal{X}; \epsilon) = \frac{1}{|\mathcal{Y}_t|} \sum\nolimits_{y\in\mathcal{Y}_t} \psi\big(d(y, \mathcal{X}); \epsilon\big) \\ + \frac{1}{|\mathcal{X}|} \sum\nolimits_{x\in\mathcal{X}} \psi\big(d(x, \mathcal{Y}_t); \epsilon\big). \end{aligned} \tag{7}$$

Compared to standard Chamfer Distance, this clipped formulation provides a more stable signal for intermediate predictions by avoiding large penalties from outliers or extrusions still under construction.

While R_{align} captures global consistency, it lacks the ability to guide fine-grained actions such as loop placement. We therefore introduce a complementary *LBP coverage reward* that directly reflects how well an extrude-loop step explains nearby surface regions: $R_{lbp} = |\mathcal{B}_t|/|\mathcal{X}|$ where $\mathcal{B}_t$ is the set of local barrel points at step t, as defined in Equation 2. This term promotes actions whose projected footprint tightly overlaps with the target geometry. Importantly, we compute R_{lbp} using only the *current* LBP set $\mathcal{B}_t$, rather than the cumulative scaffold region $\mathcal{G}_t$, to ensure that reward feedback reflects immediate contributions from each modeling step. This localized supervision encourages the policy to carve out the correct region of interest precisely.

Time Efficiency While spatial rewards focus on geometric fidelity, they do not regulate the efficiency of the reconstruction process. To address this, we introduce a time cost term R_{time}, which comprises a uniform step penalty and a terminal bonus. The step penalty uniformly penalizes every generated token, discouraging unnecessarily long command traces and implicitly favoring more concise modeling programs. In contrast, the terminal bonus is issued when the decoder emits the special `END` token at an appropriate point, incentivizing the model to terminate reconstruction precisely when the geometry is complete. Together, these terms encourage ReACT to balance completeness with brevity, guiding the policy toward efficient command sequences.

Experiments

Experimental Settings

Dataset. We train and evaluate on DeepCAD (Wu, Xiao, and Zheng 2021) and report cross-dataset results on the Fusion360 Reconstruction benchmark (Willis et al. 2021). Following (Khan et al. 2024a; Dupont et al. 2024), we delete near-duplicate models detected by geometric similarity and discard sequences whose generated geometry diverges strongly from the reference (large CD) to obtain a clean split. For every cleaned sample, we quantize all instruction parameters to 8-bit integers; the command trace is represented as an extrusion-sketch sequence.

Implementation Details. ReACT operates in a 256-dimensional latent space. We uniformly sample 4,096 surface points from each CAD model and encode them using a modified DGCNN (Wang et al. 2019) with $k = 60$ neighbors. The *point-command fusion* decoder uses 8 Transformer layers with CA to point features from layer 2 onward, while the *Trajectory Decoder* applies 4 SA layers conditioned on RTG-state-action embeddings. *Scaffold-aware geometric attention* module comprises 2 CA blocks. Further training settings and hyperparameters are detailed in the appendix.

Evaluation Details. For quantitative evaluation, we report three complementary metrics. First, geometric fidelity is measured by the bidirectional CD: we uniformly sample 8,192 points from both the predicted and ground-truth meshes, compute the average nearest-neighbor distance in both directions, scale the result by 10^3 for readability, and report both mean and median values. Second, to assess structural consistency of the command trace, we compute an F1 score over extrude-sketch primitives: following (Khan et al. 2024b,a), sketch loops are paired with their ground-truth counterparts via the Hungarian matching algorithm (Kuhn 2010) while extrusion primitives are compared by occurrence counts. Finally, robustness is captured by the invalid-rate (IR), defined as the percentage of predicted sequences that cannot be converted into a valid B-Rep by PythonOCC (Paviot 2025) kernel. Taken together, CD reflects surface accuracy, F1 gauges sequence correctness, and IR penalizes syntactically or geometrically infeasible outputs.

Baseline Methods. Because our contribution is to move from pure imitation learning to reward-conditioned decision making, we evaluate ReACT against several representative imitation-learning baselines in both modeling regimes. The feed-forward group contains DeepCAD (Wu, Xiao, and Zheng 2021) and TransCAD (Dupont et al. 2024), which map a point cloud to an entire command trace in a single,

Methods	CD ($\times 10^3$) ↓ Mean	Median	IR ↓
DeepCAD	69.51	14.44	11.95
TransCAD	36.94	6.03	5.13
Text2CAD*	9.07	0.64	4.81
ReACT (Ours)	**3.92**	**0.38**	**2.68**

Table 1: Quantitative comparison of reconstruction performance on the DeepCAD dataset. ReACT achieves the lowest CD and IR, indicating superior geometric fidelity and successful recovery of valid command sequences.

reward-agnostic forward pass. To supply an autoregressive yet still reward-free reference, we modify Text2CAD (Khan et al. 2024b) to let it process point clouds as inputs, trained solely by teacher forcing on ground-truth commands. All methods are fine-tuned using their original implementation details to ensure fair comparisons and robust performance. More implementation details are included in the appendix.

Experimental Results

Quantitative Results. Tables 1 and 2 reveal that ReACT outperforms all imitation-learning baselines on DeepCAD across every metric. It achieves the lowest mean and median CD, indicating superior surface fidelity, while simultaneously obtaining the smallest IR, demonstrating the syntactic soundness of its generated command traces. At the primitive level, ReACT attains the highest F1 scores for lines, arcs, circles, and extrusions, with especially large gains on curved sketches, which are traditionally the most challenging primitives to recover. Together, these results show that conditioning sequence generation on trajectory-level rewards and scaffold-aware states yields solutions that are both geometrically precise and structurally consistent with the design intent in the ground-truth trace.

The same trend holds when the models are evaluated on Fusion360 (Table 3). Following prior works (Dupont et al. 2024; Khan et al. 2024a; Ma et al. 2023), median CD and IR are reported. ReACT maintains the lowest median CD and IR under this distribution shift, demonstrating stronger generalization than both feed-forward and autoregressive imitation baselines. Altogether, the results confirm that ReACT is able to produce valid, high-fidelity CAD reconstructions that remain robust beyond the training distribution.

Methods	F1 Score ↑ Line	Arc	Circle	Extrusion
DeepCAD	66.30	12.65	60.07	86.31
TransCAD	78.25	27.94	61.54	90.85
Text2CAD*	76.50	30.36	64.52	90.99
ReACT (Ours)	**79.14**	**42.00**	**70.53**	**91.37**

Table 2: F1 scores for primitive-specific accuracy on DeepCAD. ReACT consistently outperforms all baselines across lines, arcs, circles, and extrusions, demonstrating strong geometric reasoning and structural awareness.

Methods	Median CD ($\times 10^3$) ↓	IR ↓
DeepCAD	92.03	28.75
TransCAD	42.34	5.68
Text2CAD*	1.79	5.90
ReACT (Ours)	**0.66**	**2.08**

Table 3: Generalization results on the Fusion360 dataset. ReACT significantly outperforms prior methods in median CD and IR, highlighting its strong cross-dataset robustness.

Qualitative Results. We next examine the qualitative performance of our proposed framework. Figure 4 visualizes reconstruction results on the DeepCAD dataset. Across a variety of shapes, from the concatenation of extrusion cylinders to parts with thin walls, holes, and nested features, ReACT reproduces both the global silhouette and the local geometry more faithfully than other baselines. DeepCAD, in contrast, often emits invalid command traces, indicated by the *invalid* labels; even when a solid is produced, small features and curved surfaces are lost. TransCAD likewise struggles with completeness, frequently omitting entire sub-features or returning blocky shapes that diverge from the reference. Text2CAD* performs better than feed-forward models but still discards delicate details, yielding coarse approximations of the ground truth. Figure 5 reports the comparison on Fusion360, where all models are evaluated without retraining. Despite the domain shift in part style and surface finish, ReACT again delivers reconstructions that closely match the ground truth, while the baselines degenerate into invalid sketches or overly simplified solids. These visual observations reinforce the quantitative advantages reported earlier and highlight ReACT's ability to maintain both structural soundness and geometric fidelity beyond its training distribution. Additional examples are provided in the appendix.

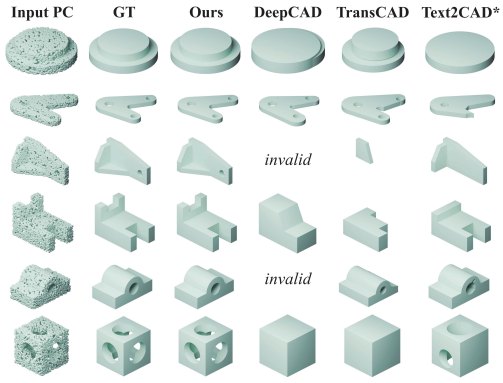

Figure 4: Qualitative results on the DeepCAD dataset. ReACT recovers complete and detailed geometry, whereas baselines often yield coarse or invalid results.

Ablation Study. We perform ablation experiments to evaluate the contributions of key components in ReACT. Table 4 reports the results across sketch and extrusion prediction accuracy (F1 Score), geometric fidelity (CD), and syntactic validity (IR). We first examine the role of deci-

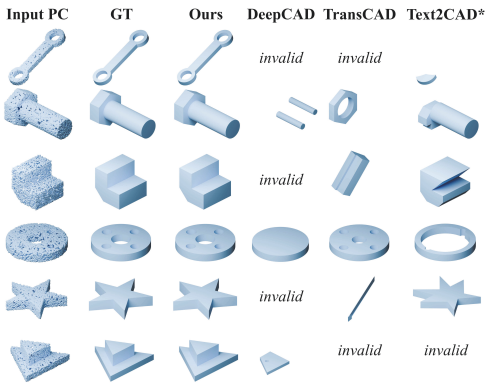

Figure 5: Cross-dataset comparison on Fusion360 dataset. ReACT maintains high-fidelity reconstructions and valid command sequences while imitation-learning baselines frequently produce invalid or incomplete models.

Model Variants	F1 Score ↑		CD ($\times 10^3$) ↓		IR ↓
	Skt.	Ext.	Mean	Median	
w/o RTG cond.	60.19	82.17	9.967	0.896	5.51
w/o R_{align}	71.77	89.19	8.123	0.576	4.66
w/o R_{lbp}	67.49	82.96	9.512	0.743	5.09
w/o R_{time}	69.95	87.16	8.251	0.583	4.14
w/o Scaf. Enc.	73.98	90.98	6.110	0.397	2.96
ReACT (Ours)	**75.76**	**91.37**	**3.929**	**0.385**	**2.68**

Table 4: Ablation study: Removing reward conditioning, reward terms, or structural encoding leads to performance degradation, validating their respective contributions.

sion formulation by removing the RTG conditioning. This effectively reduces ReACT to a pure imitation learner without reward guidance. As shown in the first row of Table 4, this variant suffers the largest performance drop across all metrics, confirming the importance of reward conditioning in enabling temporally structured planning and dynamic trajectory shaping. We next evaluate the impact of ReACT's dense reward design by ablating its individual components. Removing R_{align} term leads to noticeably higher CD and IR, reflecting reduced global alignment and weaker gradient signals for shaping overall geometry. Excluding R_{lbp} causes the sharpest drop in sketch and extrusion F1, highlighting the importance of fine-grained supervision along local surface boundaries for accurate primitive recovery. When R_{time} is removed, the model still maintains moderate geometric fidelity but shows increased invalidity and lower sketch accuracy, suggesting that temporal constraints play a regularizing role by discouraging redundant command sequences. Finally, we assess the role of structural conditioning by disabling the scaffold-aware encoding module, which provides geometric context from intermediate reconstructions. The absence of this module results in consistent degradation across all metrics, confirming its effectiveness in enriching the decoder's representation with spatially grounded cues.

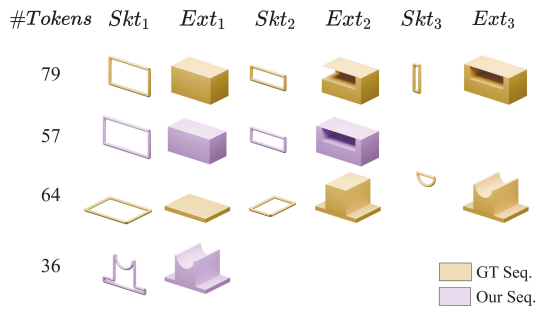

Figure 6: Comparison between ground-truth (gold) and ReACT-generated (purple) sequences. ReACT produces shorter, yet structurally faithful modeling sequences.

Sequence Efficiency. Beyond accurately replicating the ground-truth sequences, ReACT is able to discover more efficient command sequences than those seen during training. Figure 6 visualizes two examples where ReACT produces valid and structurally faithful models using fewer tokens than the corresponding ground-truth sequences. These improvements often emerge in cases involving nested or redundant sketches, where the model learns to consolidate operations while preserving the intended geometry. This behavior reflects ReACT's ability to generalize structural priors and, guided by the dense reward function, actively favor more concise and efficient modeling programs. Additional examples are provided in the appendix.

Discussions

We have presented *ReACT*, a reward-informed Decision CAD Transformer that integrates scaffold-aware geometric reasoning with LBP-guided dense reward supervision for robust CAD sequence prediction from point clouds. By framing the task as an offline decision-making problem, ReACT goes beyond imitation learning and demonstrates strong generalization to diverse and challenging shapes.

Despite its performance, ReACT remains constrained by the size and structural diversity of available CAD datasets. For highly intricate or unstructured models, especially those involving irregular sketches or multi-part assemblies, the model may fail to produce valid or complete sequences. We include representative failure cases and qualitative analyses in the appendix. Future work will explore scaling ReACT to larger and more diverse modeling corpora, incorporating richer operation vocabularies, and improving its capacity for advanced geometric reasoning. These extensions would enhance its real-world applicability and support broader reverse-engineering tasks across design domains.

Acknowledgments

This research is partially supported by the GreenTRI Grant from Jinan-NTU Green Technology Research Institute, as well as the China Scholarship Council (Grant 202406030027).

References

Chen, C.; Wei, J.; Chen, T.; Zhang, C.; Yang, X.; Zhang, S.; Yang, B.; Foo, C.; Lin, G.; Huang, Q.; and Liu, F. 2025. CADCrafter: Generating Computer-Aided Design Models from Unconstrained Images. *CoRR*, abs/2504.04753.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *NeurIPS*, 15084–15097.

Chen, T.; Yu, C.; Hu, Y.; Li, J.; Xu, T.; Cao, R.; Zhu, L.; Zang, Y.; Zhang, Y.; Li, Z.; and Sun, L. 2024. Img2CAD: Conditioned 3D CAD Model Generation from Single Image with Structured Visual Geometry. *CoRR*, abs/2410.03417.

Dupont, E.; Cherenkova, K.; Mallis, D.; Gusev, G.; Kacem, A.; and Aouada, D. 2024. TransCAD: A Hierarchical Transformer for CAD Sequence Inference from Point Clouds. In Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; and Varol, G., eds., *ECCV*, volume 15119 of *Lecture Notes in Computer Science*, 19–36. Springer.

Fu, R.; Li, Q.; Wen, C.; An, N.; and Tang, F. 2024. A Novel Framework for Learning Bézier Decomposition from 3D Point Clouds. *IEEE Transactions on Circuits and Systems for Video Technology*.

Fu, R.; Wen, C.; Li, Q.; Xiao, X.; and Alliez, P. 2023. BPNet: Bézier Primitive Segmentation on 3D Point Clouds. In Elkind, E., ed., *IJCAI*, 754–762.

Jiang, H.; Dang, Z.; Wei, Z.; Xie, J.; Yang, J.; and Salzmann, M. 2023a. Robust Outlier Rejection for 3D Registration With Variational Bayes. In *CVPR*, 1148–1157.

Jiang, H.; Salzmann, M.; Dang, Z.; Xie, J.; and Yang, J. 2023b. Se (3) diffusion model-based point cloud registration for robust 6d object pose estimation. *NeurIPS*, 36: 21285–21297.

Jiang, H.; Shen, Y.; Xie, J.; Li, J.; Qian, J.; and Yang, J. 2021. Sampling Network Guided Cross-Entropy Method for Unsupervised Point Cloud Registration. In *ICCV*, 6108–6117.

Jiang, H.; Xie, J.; and Yang, J. 2021. Action candidate based clipped double q-learning for discrete and continuous action tasks. In *AAAI*.

Khan, M. S.; Dupont, E.; Ali, S. A.; Cherenkova, K.; Kacem, A.; and Aouada, D. 2024a. CAD-SIGNet: CAD Language Inference from Point Clouds Using Layer-Wise Sketch Instance Guided Attention. In *CVPR*, 4713–4722.

Khan, M. S.; Sinha, S.; Sheikh, T. U.; Stricker, D.; Ali, S. A.; and Afzal, M. Z. 2024b. Text2CAD: Generating Sequential CAD Designs from Beginner-to-Expert Level Text Prompts. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *NeurIPS*.

Kolodiazhnyi, M.; Tarasov, D.; Zhemchuzhnikov, D.; Nikulin, A.; Zisman, I.; Vorontsova, A.; Konushin, A.; Kurenkov, V.; and Rukhovich, D. 2025. cadrille: Multimodal CAD Reconstruction with Online Reinforcement Learning. *arXiv preprint arXiv:2505.22914*.

Kuhn, H. W. 2010. The Hungarian Method for the Assignment Problem. In Jünger, M.; Liebling, T. M.; Naddef, D.; Nemhauser, G. L.; Pulleyblank, W. R.; Reinelt, G.; Rinaldi, G.; and Wolsey, L. A., eds., *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, 29–47. Springer.

Li, J.; Fu, Y.; and Chen, F. 2025. DTGBrepGen: A Novel B-rep Generative Model through Decoupling Topology and Geometry. In *CVPR*, 21438–21447.

Li, P.; Guo, J.; Li, H.; Benes, B.; and Yan, D. 2024. SfmCAD: Unsupervised CAD Reconstruction by Learning Sketch-based Feature Modeling Operations. In *CVPR*, 4671–4680.

Li, P.; Guo, J.; Zhang, X.; and Yan, D. 2023. SECAD-Net: Self-Supervised CAD Reconstruction by Learning Sketch-Extrude Operations. In *CVPR*, 16816–16826.

Lin, C.; Fan, T.; Wang, W.; and Nießner, M. 2020. Modeling 3D Shapes by Reinforcement Learning. In *ECCV*, 545–561.

Liu, S.; Luo, W.; Zhou, Y.; Chen, K.; Zhang, Q.; Xu, H.; Guo, Q.; and Song, M. 2024a. Transmission Interface Power Flow Adjustment: A Deep Reinforcement Learning Approach Based on Multi-Task Attribution Map. *IEEE Transactions on Power Systems*, 39(2): 3324–3335.

Liu, S.; Song, J.; Zhou, Y.; Yu, N.; Chen, K.; Feng, Z.; and Song, M. 2024b. Interaction Pattern Disentangling for Multi-Agent Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 8157–8172.

Liu, S.; Zhou, Y.; Song, J.; Zheng, T.; Chen, K.; Zhu, T.; Feng, Z.; and Song, M. 2023. Contrastive Identity-Aware Learning for Multi-Agent Value Decomposition. In *AAAI*, volume 37, 11595–11603.

Liu, Y.; Obukhov, A.; Wegner, J. D.; and Schindler, K. 2024c. Point2CAD: Reverse Engineering CAD Models from 3D Point Clouds. In *CVPR*, 3763–3772.

Liu, Y.; Ren, D.; Ding, Y.; Zheng, J.; and Deng, F. 2025. Learning CAD Modeling Sequences via Projection and Part Awareness. In *NeurIPS*.

Ma, W.; Chen, S.; Lou, Y.; Li, X.; and Zhou, X. 2024. Draw Step by Step: Reconstructing CAD Construction Sequences from Point Clouds via Multimodal Diffusion. In *CVPR*, 27144–27153.

Ma, W.; Xu, M.; Li, X.; and Zhou, X. 2023. MultiCAD: Contrastive Representation Learning for Multi-modal 3D Computer-Aided Design Models. In Frommholz, I.; Hopfgartner, F.; Lee, M.; Oakes, M.; Lalmas, M.; Zhang, M.; and Santos, R. L. T., eds., *ICML*, 1766–1776.

Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N. J.; and Guibas, L. J. 2019. StructureNet: hierarchical graph networks for 3D shape generation. *ACM Transactions on Graphics (TOG)*, 38(6): 242:1–242:19.

Paviot, T. 2025. Pythonocc. https://dev.opencascade.org/project/pythonocc [Accessed: (2025-02-20)].

Ren, D.; Zheng, J.; Cai, J.; Li, J.; Jiang, H.; Cai, Z.; Zhang, J.; Pan, L.; Zhang, M.; Zhao, H.; and Yi, S. 2021. CSG-Stump: A Learning Friendly CSG-Like Representation for Interpretable Shape Parsing. In *ICCV*, 12458–12467.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Zhang, M.; Li, Y.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models.

Sharma, G.; Goyal, R.; Liu, D.; Kalogerakis, E.; and Maji, S. 2018. CSGNet: Neural Shape Parser for Constructive Solid Geometry. In *CVPR*.

Uy, M. A.; Chang, Y.; Sung, M.; Goel, P.; Lambourne, J.; Birdal, T.; and Guibas, L. J. 2022. Point2Cyl: Reverse Engineering 3D Objects from Point Clouds to Extrusion Cylinders. In *CVPR*, 11840–11850.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *NeurIPS*, 5998–6008.

Wang, C.; Ma, X.; Wang, B.; Tang, S.; Meng, Y.; and Jiang, P. 2025a. Point2Primitive: CAD Reconstruction from Point Cloud by Direct Primitive Prediction. *ArXiv*, abs/2505.02043.

Wang, R.; Yuan, Y.; Sun, S.; and Bian, J. 2025b. Text-to-CAD Generation Through Infusing Visual Feedback in Large Language Models. *CoRR*, abs/2501.19054.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics (TOG)*, 38(5): 146:1–146:12.

Willis, K. D. D.; Pu, Y.; Luo, J.; Chu, H.; Du, T.; Lambourne, J. G.; Solar-Lezama, A.; and Matusik, W. 2021. Fusion 360 Gallery: A Dataset and Environment for Programmatic CAD Construction from Human Design Sequences. *ACM Transactions on Graphics (TOG)*, 40(4).

Wu, R.; Xiao, C.; and Zheng, C. 2021. DeepCAD: A Deep Generative Network for Computer-Aided Design Models. In *ICCV*, 6752–6762.

Xu, J.; Wang, C.; Zhao, Z.; Liu, W.; Ma, Y.; and Gao, S. 2024a. CAD-MLLM: Unifying Multimodality-Conditioned CAD Generation With MLLM. *CoRR*, abs/2411.04954.

Xu, X.; Jayaraman, P. K.; Lambourne, J. G.; Willis, K. D.; and Furukawa, Y. 2023. Hierarchical Neural Coding for Controllable CAD Model Generation. In *ICML*, 38443–38461.

Xu, X.; Lambourne, J. G.; Jayaraman, P. K.; Wang, Z.; Willis, K. D. D.; and Furukawa, Y. 2024b. BrepGen: A B-rep Generative Diffusion Model with Structured Latent Geometry. *ACM Transactions on Graphics (TOG)*, 43(4): 119:1–119:14.

Xu, X.; Peng, W.; Cheng, C.; Willis, K. D. D.; and Ritchie, D. 2021. Inferring CAD Modeling Sequences Using Zone Graphs. In *CVPR*, 6062–6070.

Xu, X.; Willis, K. D. D.; Lambourne, J. G.; Cheng, C.; Jayaraman, P. K.; and Furukawa, Y. 2022. SkexGen: Autoregressive Generation of CAD Construction Sequences with Disentangled Codebooks. In *ICML*, volume 162, 24698–24724.

Yin, X.; Lu, X.; Shen, J.; Ni, J.; Li, H.; Tong, R.; Tang, M.; and Du, P. 2025. RLCAD: Reinforcement Learning Training Gym for Revolution Involved CAD Command Sequence Generation. *CoRR*, abs/2503.18549.

Yu, F.; Chen, Z.; Li, M.; Sanghi, A.; Shayani, H.; Mahdavi-Amiri, A.; and Zhang, H. 2021. CAPRI-Net: Learning Compact CAD Shapes with Adaptive Primitive Assembly. *CoRR*, abs/2104.05652.

Zhang, C.; Polette, A.; Pinquié, R.; Iida, M.; Charnace, H.; and Pernot, J.-P. 2025. Reinforcement Learning-Based Parametric Cad Models Reconstruction from 2d Orthographic Drawings. *ArXiv*.

Zhou, S.; Tang, T.; and Zhou, B. 2023. CADParser: A Learning Approach of Sequence Modeling for B-Rep CAD. In *IJCAI*, 1804–1812.