ANASTASIS: RECOVERING CAD MODELING HISTORY FROM MESHES VIA SEARCH OVER FEATURE PROPOSALS

Anonymous authors
Paper under double-blind review

ABSTRACT

CAD reverse engineering is essential for recovering editable models from existing designs. In industrial workflows, meshes are widely used to represent geometry for downstream applications such as simulation. Recovering the underlying parametric CAD structure from meshes is therefore important when the original editable model is unavailable. Existing approaches tackle this inverse problem from different perspectives. Geometry-driven methods directly fit the observed shape but rely on weak design priors, often yielding feature decompositions inconsistent with the original modeling logic, whereas learned models offer stronger priors for feature alignment but struggle with precise parameter recovery. We present Anastasis, a mesh-to-CAD framework that recursively explains the input geometry by recovering executable parametric features from proposals and searching over their composition into a modeling history. Specifically, a learned proposal model identifies feature-level surface evidence and Boolean roles from the input mesh, which are then converted into executable parametric features by a geometry-based recovery stage that recovers their continuous parameters. Finally, a learned feature prior ranks and filters the recovered candidates, after which combinatorial search selects and orders them into a modeling history, while unresolved geometric residuals are recursively processed by the same procedure. Experiments on public CAD reconstruction benchmarks show that Anastasis achieves state-of-the-art performance with 0.981 IoU in-distribution, generalizes well across datasets, and produces modeling histories that better reflect the underlying design process.

1 INTRODUCTION

CAD reverse engineering aims to recover parametric constructions from geometric representations (Kolodiazhnyi et al., 2025; Rukhovich et al., 2025; Xu et al., 2021). Triangle meshes are among the most common representations used in downstream manufacturing workflows, supporting applications such as simulation and additive manufacturing (Livesu et al., 2017; Hu et al., 2018). Recovering a parametric modeling history from a mesh remains a challenging task. Although tessellated geometry is convenient for downstream processing, it introduces a substantial representation gap with respect to parametric CAD sequences (Wu et al., 2021). Furthermore, the mapping from geometry to modeling history is inherently one-to-many (Dupont et al., 2024; Wang et al., 2025), and accurate recovery further requires precise estimation of continuous CAD parameters.

Existing work has explored this inverse problem from different directions. Geometry-driven methods directly fit parametric primitives or features to the observed shape (Uy et al., 2022; Nehme et al., 2026), enabling accurate recovery of the overall geometry. However, their weak design priors make it difficult to resolve fine-grained details and Boolean relationships, often leading to feature decompositions that deviate from the underlying human modeling intent. In contrast, learned approaches better capture modeling priors through direct history prediction (Khan et al., 2024; Ma et al., 2024), but often struggle to recover continuous feature parameters with sufficient geometric precision. These complementary properties motivate a hybrid formulation that exploits learned priors for feature inference while relying on geometric reasoning for precise parameter recovery.

We present Anastasis, a mesh-to-CAD framework that recovers modeling histories through recursive feature proposal, geometric recovery, and combinatorial search. Instead of direct CAD sequence pre-

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.

diction, Anastasis first uses a learned proposal model to generate a set of feature proposals from the input mesh, with each proposal comprising feature-level surface evidence and a Boolean role. These proposals are then converted into executable extrusion or revolution hypotheses through geometric recovery of their continuous parameters. A learned feature prior scores the recovered candidates, followed by a feature sequence search that selects an ordered subset whose execution best matches the target shape. The resulting residuals, consisting of geometry missing from the reconstruction and geometry extending beyond the target, are recursively processed by the same procedure.

Our motivation is that, although tessellation removes the original parametric representation, the resulting surface retains geometric traces of the underlying modeling features that can support accurate recovery of their continuous parameters (Li et al., 2019; Uy et al., 2022). What remains ambiguous is how these traces correspond to modeling features. Anastasis learns how surface traces should be grouped into feature proposals from CAD construction histories, while recovering continuous parameters through geometric fitting. Beyond recovering individual features, reconstructing a modeling history requires feature selection and execution ordering. Anastasis uses a learned feature prior to favor candidates consistent with CAD construction histories, followed by a subset dynamic program (DP) that searches over ordered feature subsets to best reconstruct the target geometry.

In summary, our contributions are as follows: **(1)** We introduce Anastasis, a recursive mesh-to-CAD framework that turns learned feature proposals into geometrically fitted CAD operations and searches over their selection and ordering to recover the modeling history. **(2)** We decouple discrete feature discovery from continuous parameter recovery, using learning to resolve ambiguous feature structure while relying on geometric fitting for precise CAD parameter estimation. **(3)** Extensive experiments show that Anastasis achieves state-of-the-art reconstruction and modeling-history recovery, with 0.981 IoU, the highest operation-level recall, and strong cross-dataset generalization.

2 Related Work

Parametric CAD reconstruction from geometry. Prior work has explored recovering structured CAD representations from geometric observations through primitive-based, boundary-based, and feature-based formulations. SPFN (Li et al., 2019) combines learned point-level properties with differentiable primitive fitting, providing a geometry-grounded alternative to direct parameter prediction. For boundary representation (B-Rep) reconstruction, ComplexGen (Guo et al., 2022) jointly predicts geometric primitives and their relationships with global structural optimization, while Point2CAD (Liu et al., 2024) combines analytic and neural surface fitting with geometric intersection and trimming to reconstruct boundary topology. These methods recover structured boundary representations, but not the modeling features or execution order that produced them. A related line reconstructs shapes using volumetric features: Point2Cyl (Uy et al., 2022) predicts extrusion membership, surface roles, and normals before estimating extrusion parameters through differentiable closed-form formulations, while SECAD-Net (Li et al., 2023) learns implicit sketch profiles and extrusion parameters under self-supervision and combines the resulting solids through Boolean union. However, these approaches recover structured geometric representations at different levels, but do not target the recovery of an ordered modeling history.

Learning CAD modeling sequences. Learning-based methods represent CAD models as construction sequences and learn regularities in their structure and parameters. DeepCAD (Wu et al., 2021) introduces a Transformer-based latent representation of CAD sequences, while SkexGen (Xu et al., 2022) separates topology, geometry, and extrusion into disentangled codebooks for controllable generation. For reconstruction from point clouds, CAD-SIGNet (Khan et al., 2024) uses autoregressive decoding with sketch-instance-guided attention, TransCAD (Dupont et al., 2024) exploits sequence hierarchy together with a loop refiner for sketch parameters, and CAD-Diffuser (Ma et al., 2024) models point-cloud and CAD tokens through multimodal diffusion. More recent approaches leverage pretrained language models and programmatic representations. CAD-Recode (Rukhovich et al., 2025) couples a lightweight point-cloud projector with a language-model decoder to generate Python CAD programs, while Cadrille (Kolodiazhnyi et al., 2025) combines supervised fine-tuning with reinforcement learning using geometric feedback from program execution. These methods learn CAD construction priors together with sequence parameters, leaving geometric accuracy dependent on learned parameter prediction. Anastasis instead uses learning for discrete feature discovery while recovering continuous parameters through geometric fitting.

Search-based CAD history recovery. Program synthesis and search provide an alternative to direct sequence prediction. InverseCSG (Du et al., 2018) combines geometric processing with program synthesis to recover constructive solid geometry trees from meshes. Zone Graphs (Xu et al., 2021) extends history inference to sketch-based CAD features by partitioning an input B-Rep into volumetric zones and using a graph neural network to guide search over extrusion and Boolean operations. Recent mesh-to-CAD methods further integrate learned guidance and execution feedback. CADFit (Nehme et al., 2026) combines a learned sketch prior with geometric feature fitting, forward greedy program assembly, backward pruning, and residual reconstruction, while CADENA (Kabisov et al., 2026) learns a stepwise policy that predicts each feature from the target and current reconstruction. Unlike stepwise or greedy history construction, Anastasis jointly optimizes feature selection and ordering through combinatorial search, enabling candidate features to be evaluated jointly rather than in isolation.

3 METHOD

3.1 PROBLEM DEFINITION AND FRAMEWORK OVERVIEW

Problem definition. Given an input mesh M representing a target shape S, we aim to recover a parametric modeling sequence whose execution reconstructs the input geometry. We represent the recovered sequence using a feature tree T, which decouples Boolean composition from parametric modeling features: Boolean relations are encoded by the tree structure, while each leaf corresponds to an individual parametric feature. In this work, we assume an input mesh tessellated from a Boundary Representation (B-Rep) model, and consider extrusion and revolution as the supported parametric feature types. Since multiple modeling sequences can yield the same geometry, our goal is to recover a valid and executable sequence that reconstructs the input mesh.

Feature tree representation. We model a CAD construction as an ordered feature tree T. Each leaf corresponds to a parametric modeling feature $f = (P, \kappa, \theta)$, where P is a sketch, $\kappa \in \{\text{extrude}, \text{revolve}\}$ is the feature type, and θ denotes its associated parameters. Executing a leaf produces a solid, denoted by $\text{eval}(f)$. Each internal node v contains an ordered list of children $(v_1, \ldots, v_m)$, where each child v_i is associated with a Boolean role $\rho_i \in \{\text{Union}, \text{Difference}\}$. Let R_i denote the intermediate solid after applying the first i children, with $R_0 = \varnothing$. The children are evaluated sequentially, with $R_i = R_{i-1} \cup \text{eval}(v_i)$ if $\rho_i = \text{Union}$ and $R_i = R_{i-1} \setminus \text{eval}(v_i)$ if $\rho_i = \text{Difference}$, and the resulting shape of the node is $\text{eval}(v) = R_m$. The complete feature tree is evaluated recursively from the leaves to the root: parametric features first instantiate the leaf solids, and each internal node then combines its children in order. This recursive evaluation induces the corresponding linear modeling sequence, with $\text{eval}(v_{\text{root}})$ as the final shape.

Method overview. Anastasis constructs the feature tree T recursively. At each node v, the proposal network takes the current target shape S_v and produces a set of surface proposals $\{(E_k, \rho_k)\}$, each consisting of feature-level surface evidence E_k and a Boolean role ρ_k. The recovery stage then converts these proposals into a set of executable parametric features $F_v = \{(f_k, \rho_k)\}$ by recovering their geometric parameters from the observed evidence. A learned feature prior evaluates the quality of candidates in F_v conditioned on S_v and retains the top-n hypotheses, denoted by F'_v. The sequence search then determines an ordered subset $\pi'_v = \text{search}(S_v, F'_v)$, whose execution produces the reconstruction $R_v = \text{eval}(\pi'_v)$. The selected features form the local construction at node v. The missing residual $S_v \setminus R_v$ and excess residual $R_v \setminus S_v$ are recursively solved by the same procedure and attached through Union and Difference relations, respectively. Repeating this process yields the complete feature tree T, whose evaluation provides the final reconstructed shape. The feature proposal, geometric recovery, and feature sequence search modules are detailed in Secs. 3.2, 3.3, and 3.4, respectively.

3.2 LEARNING FEATURE PROPOSALS

The proposal stage infers latent modeling features from surface evidence observed on the input mesh. We decompose feature proposal into local surface organization and global feature grouping, first consolidating neighboring faces into coherent surface units and then grouping these units into

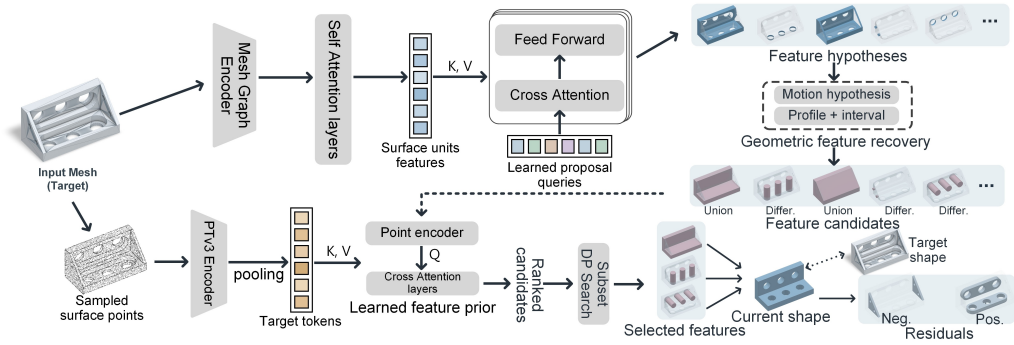

Figure 1: Overview of Anastasis. Given an input mesh, we infer surface-based feature hypotheses and recover their geometric parameters to obtain executable Boolean feature candidates. A learned feature prior assesses the candidates in the context of the target geometry, guiding subset dynamic programming to jointly select and order features. The resulting shape is compared with the target to identify missing and excess material, which serve as targets for recursive reconstruction.

individual feature hypotheses, thereby reducing the fragmentation and boundary instability caused by reasoning directly over mesh triangles.

Mesh encoding and surface unit formation. We represent the input mesh as a face-adjacency graph, where each node corresponds to a triangle and edges connect faces sharing a mesh edge. Rotation-invariant geometric descriptors (Hanocka et al., 2019), including dihedral geometry and multiscale normal variation, are embedded into face and edge features, which are propagated over the graph by a message-passing network (Gilmer et al., 2017) to produce contextualized face embeddings $\mathbf{h}_a$ and edge embeddings $\mathbf{e}_{ab}$. We predict a symmetric affinity for every adjacent face pair and extract surface units as connected components:

$$b_{ab} = \phi_{\text{aff}}([\mathbf{h}_a + \mathbf{h}_b;\, |\mathbf{h}_a - \mathbf{h}_b|;\, \mathbf{e}_{ab}])\,, \qquad \{U_j\}_{j=1}^{K} = \text{CC}(\{(a,b) \in A : \sigma(b_{ab}) > \tau\})\,. \quad (1)$$

Here b_{ab} is the affinity logit, σ denotes the sigmoid function, A denotes face adjacency, and τ is a fixed probability threshold; CC extracts connected components after thresholding, and U_j denotes the resulting surface units.

In parallel, an MLP ϕ_z projects each face embedding to an auxiliary discriminative embedding $\mathbf{z}_a = \phi_z(\mathbf{h}_a)$, supervised to cluster faces within the same surface unit and separate those across different units. We also compute explicit geometric statistics $\mathbf{g}_j$ for every surface unit, including area, spatial extent, normal statistics, and boundary geometry. The unit token $\mathbf{u}_j$ is formed by combining an area-weighted mean and max pool of $\mathbf{h}_a$, an area-weighted mean of $\mathbf{z}_a$, and $\mathbf{g}_j$.

Feature hypotheses. The surface-unit tokens are first contextualized by self-attention and then read by a set of learned feature queries through a Transformer decoder (Carion et al., 2020). Each learned query forms a feature hypothesis by associating the surface units that may originate from the same modeling feature. For a decoded query $\mathbf{q}_k$, we predict a selection probability over every surface unit, together with a Boolean role:

$$m_{kj} = \text{sigmoid}\Big(\phi_q(\mathbf{q}_k)^\top \phi_u(\widetilde{\mathbf{u}}_j)/\sqrt{d_{\text{m}}}\Big)\,, \qquad \mathbf{p}_k^{\text{role}} = \text{softmax}(\mathbf{W}_{\text{r}}\mathbf{q}_k), \quad (2)$$

Here $\widetilde{\mathbf{u}}_j$ is the contextualized token of surface unit U_j, m_{kj} is its selection probability for query k, and d_{m} the mask-embedding dimension. The projections ϕ_q and ϕ_u define the mask-embedding space, while $\mathbf{W}_{\text{r}}$ parameterizes the Boolean-role head. For each query k, we retain surface units with $m_{kj} > \tau$, and the retained units define the corresponding surface proposal E_k. The unit selections use independent sigmoid activations, allowing multiple hypotheses to share the same surface evidence and preserving proposal diversity. Feature type and continuous geometric parameters are deferred to the subsequent geometric recovery stage in Sec. 3.3.

Training. We train the proposal network in two stages. The first stage learns surface grouping from ground-truth surface units, defined as connected components of triangles sharing the same B-Rep face provenance obtained by replaying the modeling history. We then freeze the grouping

module and train the feature-query decoder using ground-truth feature masks and Boolean roles. The objectives are

$$\mathcal{L}_{\mathrm{I}} = \mathcal{L}_{\mathrm{aff}} + \beta\mathcal{L}_{\mathrm{emb}}, \qquad \mathcal{L}_{\mathrm{II}} = \frac{1}{|H|}\sum_{(k,g)\in H}\left(\mathcal{L}_{kg}^{\mathrm{mask}} + \lambda_{\mathrm{r}}\mathcal{L}_{kg}^{\mathrm{role}}\right) + \lambda_{\mathrm{a}}\mathcal{L}_{\mathrm{aux}}, \tag{3}$$

where $\mathcal{L}_{kg}^{\mathrm{mask}} = \lambda_{\mathrm{b}}\mathcal{B}_{kg} + \lambda_{\mathrm{i}}\mathcal{I}_{kg}$. For surface grouping, let A be the set of adjacent face pairs and $y_{ab} \in \{0,1\}$ indicate whether faces a and b belong to the same surface unit. We use weighted binary cross-entropy $\mathcal{L}_{\mathrm{aff}} = -|A|^{-1}\sum_{(a,b)\in A}[\lambda_+ y_{ab}\log\sigma(b_{ab}) + (1-y_{ab})\log(1-\sigma(b_{ab}))]$, where b_{ab} is the affinity logit and λ_+ balances the positive class. The embedding loss $\mathcal{L}_{\mathrm{emb}}$ follows De Brabandere et al. (2017), promoting intra-unit compactness and inter-unit separation. For the proposal decoder, Hungarian matching defines a one-to-one assignment H between queries and target features. For a matched pair (k,g), let $y_{gj} \in \{0,1\}$ indicate whether surface unit U_j belongs to feature g, and let w_j be its normalized surface area. The mask loss combines area-weighted binary cross-entropy $\mathcal{B}_{kg} = -\sum_j w_j[y_{gj}\log m_{kj} + (1-y_{gj})\log(1-m_{kj})]$ and area-weighted IoU loss $\mathcal{I}_{kg} = 1 - \frac{\sum_j w_j m_{kj} y_{gj}}{\sum_j w_j(m_{kj}+y_{gj}-m_{kj}y_{gj})}$, providing per-unit classification and global mask-overlap supervision, respectively. Boolean roles are supervised by $\mathcal{L}_{kg}^{\mathrm{role}} = -\log \mathbf{p}_k^{\mathrm{role}}[\rho_g]$. Since one-to-one matching gives positive supervision to only one query per target, we additionally supervise high-overlap unmatched pairs H_{aux} with $\mathcal{L}_{\mathrm{aux}} = |H_{\mathrm{aux}}|^{-1}\sum_{(k,g)\in H_{\mathrm{aux}}}\mathcal{L}_{kg}^{\mathrm{mask}}$. This auxiliary supervision encourages proposal diversity without introducing additional Boolean-role labels.

3.3 Feature Recovery

Given a surface proposal E_k and its predicted Boolean role, we recover an executable extrusion or revolution from its 2D profile, generating motion, and sweep interval.

Motion recovery. The lateral surface of a swept feature is tangent to its generating motion. For each triangle i, let $\mathbf{x}_i$ be its centroid, $\mathbf{n}_i$ its unit outward normal, and a_i its area. We recover extrusion and revolution motions by minimizing the normal component of their local velocities:

$$\mathbf{d}' = \arg\min_{\|\mathbf{d}\|=1}\sum_i a_i(\mathbf{n}_i^\top\mathbf{d})^2, \quad (\boldsymbol{\omega}', \mathbf{v}_0') = \arg\min_{\|\boldsymbol{\omega}\|^2+\|\mathbf{v}_0\|^2=1}\sum_i a_i\left[\mathbf{n}_i^\top(\mathbf{v}_0 + \boldsymbol{\omega}\times\mathbf{x}_i)\right]^2. \tag{4}$$

Here $\mathbf{d}$ is the extrusion direction, while $\mathbf{v}(\mathbf{x}) = \mathbf{v}_0 + \boldsymbol{\omega}\times\mathbf{x}$ describes rotational motion. The two objectives are symmetric eigenvalue problems. Extrusion uses the minimum eigenvector of the 3×3 matrix $\mathbf{G}_{\mathrm{ext}} = \sum_i a_i\mathbf{n}_i\mathbf{n}_i^\top$, while revolution uses the minimum eigenvector of the 6×6 slippage matrix $\mathbf{G}_{\mathrm{rev}} = \sum_i a_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i^\top$, with $\boldsymbol{\xi}_i = [\mathbf{x}_i\times\mathbf{n}_i; \mathbf{n}_i]$ (Lee et al., 1999; Gelfand & Guibas, 2004).

Since cap surfaces do not satisfy the lateral self-sliding constraint, we generate multiple motion hypotheses from complementary geometric cues instead of fitting all proposal triangles at once. Planar cap normals provide extrusion directions; for partial revolutions, the intersection of two angular cap planes provides a rotation axis. We also solve Eq. 4 on the full evidence and on subsets with dominant planar regions removed. Each motion hypothesis is evaluated by how well it explains the proposal triangles. For triangle i, we measure the self-sliding residual $\varepsilon_i(\eta) = |\mathbf{n}_i^\top\widehat{\mathbf{v}}_\eta(\mathbf{x}_i)|$, where $\widehat{\mathbf{v}}_\eta(\mathbf{x}_i)$ is the normalized local motion induced by hypothesis η at $\mathbf{x}_i$. Residuals near zero indicate lateral surfaces, those near one indicate caps, and the remaining triangles are treated as outliers. We select the motion hypothesis that explains the largest fraction of the proposal surface area.

Profile and interval recovery. Once the motion is recovered, we recover the two-dimensional profile P and its sweep interval from the identified cap and lateral surfaces. Cap regions provide filled regions of P. For lateral surfaces, we intersect the triangles with several cross-sections at fixed sweep positions and stitch the resulting segments into ordered boundary curves using mesh connectivity. For extrusion, the section is a plane perpendicular to the extrusion direction; for revolution, it is a half-plane containing the rotation axis, represented in signed meridian coordinates.

Complete cap evidence directly determines the profile P, while a closed lateral section determines its boundary. When the observed profile is incomplete, we apply a fixed completion hierarchy, preferring local extensions and short boundary bridges before broader geometric fallbacks, such as convex or bounding-shape completions. The sweep interval is recovered from the same evidence. Caps fix

interval endpoints, while lateral triangles determine the observed sweep extent. For extrusion, this extent is $I_{\text{lat}} = [\min_{\mathbf{x}\in X_{\text{lat}}} s(\mathbf{x}), \max_{\mathbf{x}\in X_{\text{lat}}} s(\mathbf{x})]$, where X_{lat} contains the lateral-triangle vertices and $s(\mathbf{x})$ is their coordinate along the extrusion direction. For revolution, we collect the angular intervals covered by the lateral triangles and remove the largest uncovered angular gap, yielding the smallest arc that covers the observed evidence.

For Boolean difference, the part of the tool solid outside the remaining material is unobservable. We therefore extend an open lateral profile outward until it clears the surrounding material and close it with a straight segment. If a sweep endpoint is likewise unconstrained, we use a cut-through extent beyond the surrounding material. The recovered feature is instantiated by the CAD kernel and retained only if a sufficient fraction of its supporting surface evidence lies within a prescribed distance of its boundary. Additional details are provided in Appendix B.

3.4 Search with Learned Feature Priors

Feature recovery produces a set of executable feature hypotheses. We then use a learned feature prior to rank these hypotheses, explicit geometric search to select and order them into a construction sequence, and residual recursion to recover geometry not explained by the current sequence.

Learning feature prior. We sample the input mesh surface and encode the sampled points with a pretrained Point Transformer V3 backbone (Wu et al., 2024). Multi-stage features are aggregated and locally pooled to form the mesh tokens $\mathbf{T} = \{\mathbf{t}_j\}_{j=1}^{N_T}$. For each recovered feature i, we sample boundary points $\{(\mathbf{x}_{ip}, \mathbf{n}_{ip})\}_{p=1}^{N_{\text{pt}}}$, where $\mathbf{x}_{ip}$ and $\mathbf{n}_{ip}$ denote the sample position and normal. Each sample is augmented with a relation descriptor $\mathbf{r}_{ip}$ encoding its distance to the input mesh, inside-outside state, and a surface-support indicator that marks whether the sample lies on the observed mesh surface. For supported samples, we attach the backbone feature $\mathbf{b}_{ip}$ of the nearest mesh point; otherwise, $\mathbf{b}_{ip}$ is replaced by a learned embedding. The resulting point feature $\mathbf{u}_{ip} = [\mathbf{x}_{ip}; \mathbf{n}_{ip}; \mathbf{r}_{ip}; \mathbf{b}_{ip}]$ is first projected by a shared MLP into a per-point embedding $\mathbf{h}_{ip}$. We then select surface centers by farthest-point sampling and collect a local neighborhood N_{iu} around each center $\mathbf{c}_{iu}$. Each neighborhood is aggregated as $\mathbf{q}_{iu} = \text{MaxPool}_{p\in N_{iu}} \phi_{\text{local}}([\mathbf{h}_{ip}; \mathbf{x}_{ip} - \mathbf{c}_{iu}])$, yielding local surface tokens $\mathbf{Q}_i = \{\mathbf{q}_{iu}\}_{u=1}^{N_P}$, where ϕ_{local} is a shared point-wise MLP following PointNet (Qi et al., 2017). A set of learned feature queries and a summary query attend to $\mathbf{Q}_i$, producing local feature representations $\mathbf{H}_i^0 = \{\mathbf{h}_{ik}^0\}$ and a global summary $\mathbf{c}_i^0$, respectively.

These representations are refined through L_{att} interaction layers. Let $\mathbf{C}^t = \{\mathbf{c}_i^t\}_{i=1}^{N_C}$ denote the summaries of all recovered features. At layer t,

$$\widetilde{\mathbf{C}}^t = \text{SelfAttn}(\mathbf{C}^t), \qquad \mathbf{H}_i^{t+1} = \text{CrossAttn}(\mathbf{H}_i^t, \mathbf{T}), \qquad \mathbf{c}_i^{t+1} = \text{CrossAttn}(\widetilde{\mathbf{c}}_i^t, \mathbf{H}_i^{t+1}), \quad (5)$$

where $\widetilde{\mathbf{c}}_i^t$ is the i-th token of $\widetilde{\mathbf{C}}^t$. Thus, the local representations gather evidence from the input mesh, while the summary also incorporates context from other recovered features. After L_{att} layers, an MLP followed by a sigmoid maps $\mathbf{c}_i^{L_{\text{att}}}$ to a feature score s_i.

Training targets are obtained by replaying the ground-truth modeling history. We match each recovered feature to role-compatible ground-truth features using IoU over the volumes that actually change the material state. The resulting agreement is converted to a soft target and optimized with binary cross-entropy.

Feature sequence search. Let S be the current target and $C_i = \text{eval}(f_i)$ the solid instantiated by candidate i with Boolean role ρ_i. We retain the top-n candidates according to s_i and seek an ordered subset π whose execution $R(\pi)$ best matches S. We maximize $J(\pi) = \text{Vol}(R(\pi)\cap S) - \text{Vol}(R(\pi)\setminus S)$, which rewards correctly reconstructed material while penalizing excess material.

Our key observation is that the final state of a point is determined by the last operation containing it. Let B be the candidates executed after i. Since later operations determine the final state in overlapping regions, candidate i only affects $D_i(B) = C_i \setminus \cup_{j\in B} C_j$. Since $D_i(B)$ depends only on which candidates follow i, not their order, it defines a valid subset-DP state. The contribution of candidate i is $\Delta_i(B) = [\rho_i = \text{Union}](\text{Vol}(D_i(B) \cap S) - \text{Vol}(D_i(B) \setminus S))$. For a Difference candidate, $\Delta_i(B) = 0$, but it can improve the objective by removing excess material introduced by earlier Unions. Let $V(B)$ denote the best objective over all orderings of B. Placing i before the

remaining candidates gives $V(B) = \max_{i \in B}\{V(B \setminus \{i\}) + \Delta_i(B \setminus \{i\})\}$, with $V(\varnothing) = 0$. The optimal subset is $\arg\max_B V(B)$, and backtracking recovers its order π'. The DP runs in $O(n2^n)$ time. We estimate volumes using sampled points, each represented by the n-bit pattern of candidates containing it. A subset-sum transform reduces each transition to two table lookups. The recovered sequence π' defines the node's ordered children, and CAD-kernel evaluation yields $R' = R(\pi')$.

Residual recursion. The residual geometry, consisting of missing material $S \setminus R'$ and excess material $R' \setminus S$, is recursively reconstructed by the same pipeline and incorporated through Union and $\mathrm{Difference}$ relations, respectively. This recursion yields the feature tree T defined in Sec. 3.1.

4 Experiments

4.1 Experimental Setups

Datasets and baselines. We evaluate on the test sets of DeepCAD (Wu et al., 2021), FutureCAD (Li et al., 2026), and Fusion360 (Willis et al., 2021). We compare Anastasis with Point2Cyl (Uy et al., 2022), CAD-Recode (Rukhovich et al., 2025), cadrille (Kolodiazhnyi et al., 2025), CADFit (Nehme et al., 2026), and CADENA (Kabisov et al., 2026). Since several baselines are restricted to sketch-extrude constructions and cannot represent revolve operations, Table 1 uses the revolve-free subset of FutureCAD for a fair comparison. Models containing revolve operations are evaluated separately in Table 2 using methods that support them.

Metrics. To ensure a consistent comparison across shapes, predictions and ground-truth models are independently normalized by centering their bounding boxes, scaling the longest side to unit length, and translating them into $[0, 1]^3$. We report volumetric **IoU** and **Chamfer Distance (CD)**, where CD is the sum of the two directional mean squared nearest-neighbor distances between 8,192 uniformly sampled surface points and is scaled by $1{,}000$; its mean and median are reported as CD_μ and CD_{med}. We additionally report **GMS** (Kabisov et al., 2026) for geometric similarity. To evaluate modeling-history recovery, predicted and reference operations are matched by Hungarian assignment using volumetric IoU of their material changes, and **R@.5** denotes the recall of reference operations matched at IoU ≥ 0.5. We report the **invalidity ratio (IR)** as the fraction of outputs that fail to build or are not watertight, together with mean and median end-to-end runtime, t_μ and t_{med}.

Implementation details. The proposal network uses a 12-layer mesh encoder followed by 12 self-attention layers and a 4-layer Transformer decoder with 48 learned queries; surface grouping and proposal decoding are trained sequentially for 25 and 40 epochs, respectively. The learned feature prior builds on a pretrained Point Transformer V3 backbone (Wu et al., 2024) with a 6-layer interaction module, using 256 mesh tokens and 16 local queries per recovered feature, and is trained for 20 epochs with a lower learning rate for the pretrained backbone. All models are trained with AdamW on 8 NVIDIA A100 GPUs; further implementation details are provided in Appendix A.

4.2 Main Results

Geometric reconstruction. Table 1 evaluates reconstruction quality on DeepCAD together with cross-dataset evaluation on FutureCAD and Fusion360. Anastasis achieves the best IoU, mean CD, and GMS across all three datasets, indicating consistent gains in both volumetric and surface-level accuracy. The advantage is particularly evident on FutureCAD, whose models involve more complex modeling histories: IoU improves from 0.918 to 0.971, while mean CD drops from 1.104 to 0.294. The consistent performance on the two cross-dataset benchmarks indicates that our method maintains high geometric accuracy across different data distributions. In terms of efficiency, Anastasis provides a practical balance between reconstruction accuracy and runtime. Compared with geometry-optimization methods such as CADFit, it substantially reduces computation time while remaining practical across all three datasets.

Modeling-history recovery. Table 1 further evaluates whether the reconstructed geometry is supported by modeling operations consistent with the reference histories. Anastasis achieves the highest R@.5 on all three datasets, with particularly large gains on FutureCAD and Fusion360, improving from 0.647 to 0.803 and from 0.700 to 0.810, respectively. This suggests that the improvement extends beyond shape accuracy to closer feature-level agreement with the reference modeling history.

Table 1: Reconstruction of CAD modeling histories on DeepCAD, FutureCAD, and Fusion360. FutureCAD and Fusion360 are used for cross-dataset evaluation. For fair comparison, FutureCAD uses the revolve-free subset; models containing revolve operations are reported separately in Table 2. † denotes point-cloud-based methods; the remaining methods use mesh input.

Method	IoU ↑	CD_μ ↓	CD_{med} ↓	IR ↓	GMS ↑	R@.5 ↑	t_μ (s) ↓	t_{med} (s) ↓
DeepCAD (Wu et al., 2021) (In-Distribution)								
Point2Cyl† (Uy et al., 2022)	0.426	14.388	9.527	3.00%	28.30	0.383	34.29	36.96
CAD-Recode† (Rukhovich et al., 2025)	0.905	0.678	0.175	6.30%	91.91	0.830	**2.21**	**1.76**
cadrille† (Kolodiazhnyi et al., 2025)	0.904	0.623	0.176	0.15%	91.49	0.822	2.75	2.35
CADFit (Nehme et al., 2026)	0.935	1.995	0.165	2.40%	91.90	0.811	208.57	98.64
CADENA (Kabisov et al., 2026)	0.954	0.949	**0.148**	0.90%	96.34	0.831	13.80	10.08
Anastasis (*Ours*)	**0.981**	**0.313**	0.152	**0.00%**	**96.94**	**0.877**	9.79	4.37
FutureCAD (Li et al., 2026) (Out-of-Distribution)								
Point2Cyl† (Uy et al., 2022)	0.289	20.136	13.645	3.20%	18.61	0.153	43.25	39.38
CAD-Recode† (Rukhovich et al., 2025)	0.835	1.466	0.213	10.45%	82.73	0.636	**3.32**	**2.77**
cadrille† (Kolodiazhnyi et al., 2025)	0.837	1.104	0.215	0.69%	82.66	0.625	3.59	3.27
CADFit (Nehme et al., 2026)	0.854	3.421	0.179	1.90%	83.34	0.573	264.56	152.43
CADENA (Kabisov et al., 2026)	0.918	1.377	0.150	2.65%	93.15	0.647	21.16	16.70
Anastasis (*Ours*)	**0.971**	**0.294**	**0.145**	**0.00%**	**96.81**	**0.803**	10.10	4.06
Fusion360 (Willis et al., 2021) (Out-of-Distribution)								
Point2Cyl† (Uy et al., 2022)	0.360	14.487	8.954	4.00%	28.01	0.285	40.84	38.46
CAD-Recode† (Rukhovich et al., 2025)	0.847	0.842	0.178	8.75%	82.32	0.699	**2.91**	**2.32**
cadrille† (Kolodiazhnyi et al., 2025)	0.858	0.793	0.176	0.75%	82.12	0.689	3.19	2.76
CADFit (Nehme et al., 2026)	0.857	2.915	0.160	3.20%	83.05	0.655	263.08	139.81
CADENA (Kabisov et al., 2026)	0.925	0.599	0.126	1.97%	90.31	0.700	18.53	13.76
Anastasis (*Ours*)	**0.967**	**0.400**	**0.124**	**0.00%**	**92.30**	**0.810**	18.61	6.61

Table 2: Reconstruction on the FutureCAD models containing revolve operations.

Method	IoU ↑	CD_μ ↓	IR ↓	GMS ↑	R@.5 ↑
CADFit	0.782	5.167	6.65%	70.46	0.456
CADENA	0.869	2.384	3.47%	81.44	0.572
Anastasis (*Ours*)	**0.928**	**0.828**	**1.18%**	**88.94**	**0.668**

Evaluation on Broader Feature Types. Table 2 evaluates reconstruction when the modeling space is extended beyond extrusion-only constructions to include revolution features. For this evaluation, Anastasis is trained on FutureCAD, whereas CADENA relies on large-scale procedurally generated training data and CADFit is optimization-driven with a learned sketch prior. Anastasis achieves clear gains over the strongest baseline in both geometric accuracy and operation-level recovery. These consistent improvements across geometric and history metrics indicate that the proposed framework can accommodate a broader range of feature types without being tied to extrusion-only modeling. Meanwhile, its invalidity ratio decreases to 1.18%, compared with 3.47% for CADENA, suggesting that this broader feature support does not come at the expense of reconstruction validity.

Qualitative results. Figure 2 shows that Anastasis recovers both coherent feature structures and fine geometric details. In contrast, geometry-driven and purely learned methods often struggle with faithful recovery of local details.

4.3 Ablation Studies

Table 3 evaluates the main components of Anastasis across different levels of modeling complexity. We consider three variants: *w/o surface units* operates directly on individual mesh triangles during feature proposal; *w/o learned prior* retains all valid recovered candidates; and *w/o subset DP* replaces subset dynamic programming with greedy selection that visits candidates in descending learned score and keeps each one only if it reduces the residual error. We further discuss representative failure cases of Anastasis in Appendix C.

Surface units. Removing surface units lowers IoU and GMS across all three complexity groups. The difference is more evident on Medium models, where mean CD increases from 0.306 to 0.602 and R@.5 decreases from 0.609 to 0.527. The results suggest that surface-unit grouping provides a more effective representation for feature proposal than operating directly on mesh triangles, particularly on models with more operations. On Simple models, however, the gap is small and triangle-level prediction achieves a slightly lower mean CD.

Learned feature prior. The effect of removing the learned feature prior is small for Simple models but grows with construction complexity. IoU decreases from 0.961 to 0.949 on Medium models and from 0.921 to 0.894 on Complex models. This suggests that the learned prior improves the quality

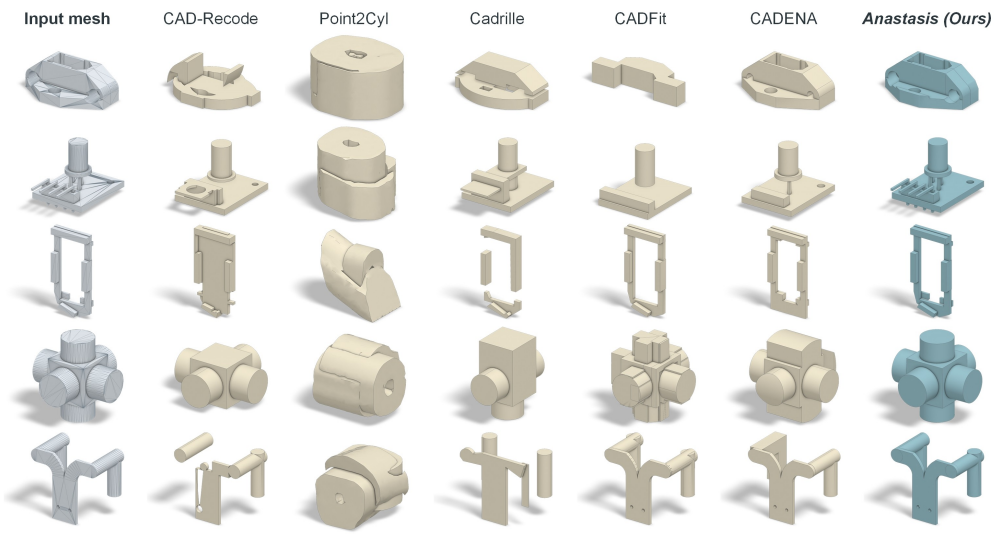

Figure 2: Qualitative comparison of mesh-to-CAD reconstruction. Anastasis faithfully preserves fine geometric details and coherent feature structures, closely reproducing both the local geometry and overall shape of the input meshes. These results illustrate the complementary strengths of precise geometric recovery and learned modeling priors.

Table 3: Ablation study on FutureCAD across different levels of part complexity. *w/o surface units* removes surface-unit grouping and operates directly on individual mesh triangles; *w/o learned prior* retains all valid recovered candidates; *w/o subset DP* replaces subset dynamic programming with greedy selection in descending prior order.

	Simple (ops ≤ 2)				Medium ($2 <$ ops ≤ 5)				Complex (ops > 5)			
Variant	IoU ↑	CD_μ ↓	GMS ↑	R@.5 ↑	IoU ↑	CD_μ ↓	GMS ↑	R@.5 ↑	IoU ↑	CD_μ ↓	GMS ↑	R@.5 ↑
Anastasis (*Full*)	**0.988**	0.216	**98.00**	0.884	**0.961**	**0.306**	**95.73**	**0.609**	**0.921**	**0.381**	**91.61**	**0.440**
Feature proposal												
w/o surface units	0.976	**0.198**	97.41	0.862	0.958	0.602	94.76	0.527	0.910	0.470	89.28	0.423
Selection												
w/o learned prior	0.987	0.251	97.20	0.892	0.949	0.420	93.23	0.591	0.894	0.430	90.38	0.425
w/o subset DP	0.979	0.209	97.31	**0.909**	0.939	0.423	93.24	0.571	0.872	0.707	89.18	0.393

of candidates entering the subsequent search, with greater benefit on more complex constructions where modeling relationships among features become more important.

Subset dynamic programming. The importance of combinatorial search is most evident on Complex models, where replacing subset DP with greedy selection reduces IoU from 0.921 to 0.872 and increases mean CD from 0.381 to 0.707. In contrast, greedy selection remains competitive on Simple models and even achieves a higher R@.5. This contrast reflects the increasing interaction among features: when only a few operations are present, local ranking is often sufficient, whereas longer constructions require joint selection and ordering to resolve overlapping additions and cuts.

5 Conclusion

We presented Anastasis, a mesh-to-CAD framework that reconstructs parametric modeling histories by turning learned feature proposals into geometrically fitted CAD operations and searching over their selection and ordering. By separating discrete feature discovery from continuous parameter recovery, Anastasis combines learned modeling priors with precise geometric estimation. A learned feature prior guides candidate selection, while subset dynamic programming resolves feature selection and execution order, with unresolved geometric residuals handled recursively. Experiments demonstrate state-of-the-art reconstruction and modeling-history recovery across datasets, together with strong cross-dataset generalization. Future work will extend the framework beyond extrusion and revolution to support a broader range of modeling operations.

AI USE STATEMENT

The research methodology, experimental design, and primary implementation were developed by the authors without AI assistance. Generative AI tools were used for language polishing and limited assistance with code development. All AI-assisted code was reviewed and verified for correctness by at least two authors, and all language revisions were reviewed to ensure that the intended scientific meaning was preserved.

REPRODUCIBILITY STATEMENT

The proposed method and training objectives are described in Sec. 3, and the datasets and evaluation protocol are specified in Sec. 4.1. Appendix A provides the network configurations, training settings, and search parameters. Appendix B gives additional details of the geometric recovery procedure. We will publicly release our code upon acceptance to facilitate reproducibility.

REFERENCES

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017.

Tao Du, Jeevana Priya Inala, Yewen Pu, Andrew Spielberg, Adriana Schulz, Daniela Rus, Armando Solar-Lezama, and Wojciech Matusik. Inversecsg: Automatic conversion of 3d models to csg trees. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018.

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

Natasha Gelfand and Leonidas J Guibas. Shape segmentation using local slippage analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 214–223, 2004.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. Pmlr, 2017.

Haoxiang Guo, Shilin Liu, Hao Pan, Yang Liu, Xin Tong, and Baining Guo. Complexgen: Cad reconstruction by b-rep chain complex generation. *ACM Transactions on Graphics (TOG)*, 41(4): 1–18, 2022.

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (ToG)*, 38(4):1–12, 2019.

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. Tetrahedral meshing in the wild. *ACM Trans. Graph.*, 37(4):60, 2018.

Soslan Kabisov, Gennadiy Savrasov, Maksim Elistratov, Antonio Rodriguez, Daniil Ignatiev, Nikita Gavrilov, Rustam Uzdenov, Alexey I Boyko, Igor Pasechnik, Anton Konushin, et al. Cadena: Stepwise cad reverse engineering. *arXiv preprint arXiv:2608.00799*, 2026.

Mohammad Sadil Khan, Elona Dupont, Sk Aziz Ali, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. Cad-signet: Cad language inference from point clouds using layer-wise sketch instance guided attention. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4713–4722. IEEE, 2024.

Maksim Kolodiazhnyi, Denis Tarasov, Dmitrii Zhemchuzhnikov, Alexander Nikulin, Ilya Zisman, Anna Vorontsova, Anton Konushin, Vladislav Kurenkov, and Danila Rukhovich. cadrille: Multimodal cad reconstruction with reinforcement learning. *arXiv preprint arXiv:2505.22914*, 2025.

In-Kwon Lee, Johannes Wallner, and Helmut Pottmann. Scattered data approximation with kinematic surfaces. *SAMPTA*, 99(72-77):46–49, 1999.

Jiahao Li, Qingwang Zhang, Qiuyu Chen, Guozhan Qiu, Yunzhong Lou, and Xiangdong Zhou. Towards high-fidelity CAD generation via LLM-driven program generation and text-based b-rep primitive grounding. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=jfOhTGs5G5.

Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 2647–2655. IEEE, 2019.

Pu Li, Jianwei Guo, Xiaopeng Zhang, and Dong-Ming Yan. Secad-net: Self-supervised cad reconstruction by learning sketch-extrude operations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16816–16826. IEEE, 2023.

Yujia Liu, Anton Obukhov, Jan Dirk Wegner, and Konrad Schindler. Point2cad: Reverse engineering cad models from 3d point clouds. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3763–3772. IEEE, 2024.

Marco Livesu, Stefano Ellero, Jonàs Martínez, Sylvain Lefebvre, and Marco Attene. From 3d models to 3d prints: an overview of the processing pipeline. In *Computer Graphics Forum*, volume 36, pp. 537–564. Wiley Online Library, 2017.

Weijian Ma, Shuaiqi Chen, Yunzhong Lou, Xueyang Li, and Xiangdong Zhou. Draw step by step: Reconstructing cad construction sequences from point clouds via multimodal diffusion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27144–27153. IEEE, 2024.

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

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

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

Mikaela Angelina Uy, Yen-Yu Chang, Minhyuk Sung, Purvi Goel, Joseph G Lambourne, Tolga Birdal, and Leonidas J Guibas. Point2cyl: Reverse engineering 3d objects from point clouds to extrusion cylinders. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11850–11860, 2022.

Ruiyu Wang, Yu Yuan, Shizhao Sun, and Jiang Bian. Text-to-CAD generation through infusing visual feedback in large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 65326–65345. PMLR, 13–19 Jul 2025.

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

Rundi Wu, Chang Xiao, and Changxi Zheng. Deepcad: A deep generative network for computer-aided design models. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6752–6762. IEEE, 2021.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851. IEEE, 2024.

Xiang Xu, Karl DD Willis, Joseph G Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. Skexgen: Autoregressive generation of cad construction sequences with disentangled codebooks. In *International Conference on Machine Learning*, pp. 24698–24724. PMLR, 2022.

Xianghao Xu, Wenzhe Peng, Chin-Yi Cheng, Karl D. D. Willis, and Daniel Ritchie. Inferring cad modeling sequences using zone graphs. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6058–6066. IEEE, 2021.

A Implementation Details

Table 4 summarizes the network configurations. The following sections describe the training and search settings.

Table 4: Network configurations.

Proposal network		**Learned feature prior**	
Face input dimension	13	Target surface samples	8,192
Edge input dimension	23	Mesh tokens	256
Face embedding dimension	512	Samples per feature	512
Edge embedding dimension	128	Surface centers	32
Mesh encoder layers	12	Neighbors per center	32
Auxiliary embedding dimension	32	Local queries	16
Unit geometry dimension	21	Summary queries	1
Unit token dimension	512	Interaction dimension	768
Unit attention layers	12	Interaction layers	6
Unit attention heads	8	Interaction heads	12
Decoder layers	4	Feed-forward width	3,072
Learned feature queries	48		
Feed-forward width	3,072		
Mask embedding dimension	128		

A.1 Proposal Network

Encoding and decoding. The mesh encoder embeds face and edge descriptors with separate MLPs. Messages combine adjacent face and edge embeddings and are aggregated by mean and max pooling. Residual MLP updates include a global mean-pooled feature every three layers. Unit tokens combine pooled face embeddings, auxiliary embeddings, and geometric statistics. Auxiliary embeddings do not determine connected components. The threshold τ in Sec. 3.2 is $1/(1+e^{-1})$ for surface grouping and 0.5 for unit selection. Self-attention contextualizes the units before queries predict masks and Boolean roles. Independent masks allow proposals to overlap. Feature types and parameters are recovered geometrically.

Supervision and optimization. History replay provides face provenance and feature labels. Connected triangles from the same B-Rep face define grouping targets. Grouping is trained for 25 epochs and frozen during 40 epochs of decoder training. Mask weights are unit areas normalized by the total mesh area. In addition to Hungarian matching, up to three unmatched queries with high overlap receive only mask supervision per target. Each query is assigned to at most one target. The loss weights are $\lambda_+ = 0.8$, $\beta = 0.3$, $\lambda_\mathrm{b} = 4$, $\lambda_\mathrm{i} = 8$, $\lambda_\mathrm{r} = 1$, and $\lambda_\mathrm{a} = 0.5$. The embedding loss uses margins of 0.5 for intra-unit compactness and 1.5 for inter-unit separation, with a mean-embedding regularization weight of 10^{-3}. Both stages use AdamW with moments $(0.9, 0.95)$, weight decay 0.05, gradient clipping at 1.0, and 800 warmup steps followed by cosine decay. Grouping uses a learning rate of 6.0×10^{-4} and an effective batch size of 512. The decoder uses 5.1×10^{-4} and an effective batch size of 384. Both stages use eight A100 GPUs and BF16 precision. Evaluation uses exponential moving average weights with decay 0.999.

A.2 Learned Feature Prior

Input and interaction. All candidates share the target's normalization, which translates its bounding-box center to the origin and scales its longest side to one. Point Transformer V3 encodes target points sampled in proportion to surface area. Features from multiple stages are pooled into mesh tokens. Candidate points use the descriptors defined in Sec. 3.4. Supported points receive the nearest target point's backbone feature, while others use a learned embedding. Pooled candidate features are processed by the query and interaction layers to obtain scores s_i.

Training. Candidates are matched to ground-truth features with the same Boolean role using IoU over the volumes that change the material state. Soft targets are zero for IoU at or below 0.5 and

one at or above 0.9, with linear interpolation in between. Padded candidates are excluded from the binary cross-entropy loss. We train for 20 epochs with a backbone learning rate of 10^{-5} and 10^{-4} for newly added layers. AdamW uses moments $(0.9, 0.999)$, weight decay 0.05, and gradient clipping at 1.0. Training uses BF16 precision and 1,000 warmup steps followed by cosine decay to 10^{-6}. The backbone is frozen for the first 2,000 steps.

A.3 Subset Search and Residual Recursion

Budget and volume lookup. Each node retains at most 16 candidates and samples $N_{\text{vol}} = 16{,}384$ points uniformly in a box Ω containing the target and all Union candidates. A residual is considered negligible when its volume is below 1% of the root target volume.

Samples are grouped by the subset B of candidates containing them. For each group, $h_{\text{vol}}(B)$ counts samples inside the target minus those outside it. The subset-sum transform is

$$z_{\text{vol}}(B) = \sum_{B' \subseteq B} h_{\text{vol}}(B'). \tag{6}$$

For suffix B, let $\bar{B} = \{1, \dots, n\} \setminus B$. The sampled estimate of $\Delta_i(B)$ in Sec. 3.4, for $i \notin B$, is

$$\widehat{\Delta}_i(B) = [\rho_i = \text{Union}] \frac{\text{Vol}(\Omega)}{N_{\text{vol}}} \big[z_{\text{vol}}(\bar{B}) - z_{\text{vol}}(\bar{B} \setminus \{i\}) \big]. \tag{7}$$

The two lookups isolate samples in C_i outside all suffix candidates. Preprocessing takes $O(n2^n)$ time. The DP stores one value and one backtracking choice per subset, requiring $O(2^n)$ memory, and is exact for the sampled objective.

Execution. The recovered sequence π' is executed by the CAD kernel to obtain $R' = R(\pi')$. Missing and excess residuals, $S \setminus R'$ and $R' \setminus S$, are processed by the same pipeline. The Boolean role connecting a child to its parent is independent of the roles within that child. Recursion stops when the residual is negligible, the depth reaches 5, or the number of steps reaches 30. A child result is retained only if it reduces the parent's error.

B Parameter Recovery Details

Given surface evidence E_k and Boolean role ρ_k, we recover a feature $f_k = (P, \kappa, \theta)$ using the mesh geometry and connectivity. For a single feature, we write $f = f_k$ and $\rho = \rho_k$ and denote its sweep interval by I. Analytic B-Rep surfaces and reference modeling histories are not required at inference.

B.1 Motion Hypotheses and Surface Roles

Extrusion direction. Lateral normals are orthogonal to the extrusion direction. The first objective in Eq. 4 is therefore minimized by an eigenvector associated with the smallest eigenvalue of $\mathbf{G}_{\text{ext}}$. A single planar patch does not determine a unique tangent direction, so we also use planar cap normals to generate hypotheses. Opposite directions describe the same extrusion axis, with orientation determined by the coordinate frame and sweep interval.

Revolution axis. Using $\mathbf{n}_i^\top(\boldsymbol{\omega} \times \mathbf{x}_i) = \boldsymbol{\omega}^\top(\mathbf{x}_i \times \mathbf{n}_i)$, an eigenvector associated with the smallest eigenvalue of $\mathbf{G}_{\text{rev}}$ gives $(\boldsymbol{\omega}', \mathbf{v}_0')$. For nonzero angular motion, the unit axis and its closest point to the origin are

$$\mathbf{a} = \frac{\boldsymbol{\omega}'}{\|\boldsymbol{\omega}'\|}, \qquad \mathbf{o} = \frac{\boldsymbol{\omega}' \times \mathbf{v}_0'}{\|\boldsymbol{\omega}'\|^2}. \tag{8}$$

Pure rotation satisfies $\mathbf{v}_0' = -\boldsymbol{\omega}' \times \mathbf{o}$. Because the slippage matrix contains both positions and normals, the fit is sensitive to coordinate scale. We center the evidence and scale it to a unit ball before fitting, then map the recovered axis back to the original coordinates.

Hypothesis generation and selection. Cap triangles do not satisfy the tangency constraint for lateral surfaces. We generate hypotheses from planar cap normals for extrusion and from intersections of angular cap planes for partial revolution. We also solve Eq. 4 using the full evidence and subsets with dominant planar regions removed. The residual $\varepsilon_i(\eta)$ identifies lateral surfaces, caps, and outliers. Hypotheses are ranked by the fraction of proposal area they explain. The selected motion must yield a nondegenerate profile and an executable sweep.

B.2 Extrusion Profile and Interval

Local coordinates. With origin $\mathbf{o}$ and orthonormal vectors $\mathbf{e}_1, \mathbf{e}_2$ perpendicular to $\mathbf{d}'$, the sweep is

$$\mathbf{x}(u, v, t) = \mathbf{o} + u\mathbf{e}_1 + v\mathbf{e}_2 + t\mathbf{d}', \qquad (u, v) \in P, \quad t \in I. \tag{9}$$

Profile coordinates are projections relative to $\mathbf{o}$ onto $\mathbf{e}_1, \mathbf{e}_2$, and the sweep coordinate is $s(\mathbf{x}) = (\mathbf{x} - \mathbf{o})^\top \mathbf{d}'$. Changing the origin preserves the generated solid when the profile and interval are transformed accordingly.

Cap regions. Connected coplanar triangles classified as caps are grouped into patches. A patch boundary consists of mesh edges incident to exactly one triangle of that patch. Walking these edges through shared mesh vertices gives ordered boundary loops, which are projected into the profile plane. Loop containment distinguishes outer boundaries from holes and allows a profile to contain multiple disconnected regions.

A cap fragment may be incomplete because later operations have removed part of the original profile. We therefore check whether a cap boundary edge is supported by lateral evidence belonging to the same proposal. Supported boundaries constrain the feature profile. Other boundaries may have been introduced by subsequent operations, so the cap fragment is treated as evidence for only part of the profile.

Lateral cross-sections. The lateral triangles are intersected with planes at several fixed sweep coordinates. Each intersected triangle contributes a segment. Segment endpoints are associated with the mesh edges they intersect, allowing adjacent triangles to be stitched through mesh connectivity. Walking the resulting graph produces closed loops and open chains in the profile plane. Multiple sections provide complementary evidence when some parts of the lateral surface have been removed. Closed loops define profile regions directly. Open chains are passed to the completion procedure below.

Sweep interval. Cap patches constrain interval endpoints through their sweep coordinates. The extent observed on the lateral surface is measured from its vertices, as in I_{lat} in the main paper. Using only triangle centroids can underestimate the sweep length. When cap evidence fixes an endpoint, it takes precedence over an interior lateral sample. Otherwise, the observed lateral extent supplies the endpoint whenever it is available. Unconstrained ends of a Difference feature are handled by the cut-through construction described below.

B.3 Revolution Profile and Angular Interval

Meridian coordinates. For axis $(\mathbf{o}, \mathbf{a})$ and orthonormal basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{a})$,

$$\mathbf{x}(r, z, \alpha) = \mathbf{o} + z\mathbf{a} + r\big(\cos\alpha\, \mathbf{e}_1 + \sin\alpha\, \mathbf{e}_2\big), \qquad (r, z) \in P, \quad \alpha \in I. \tag{10}$$

Here $z = (\mathbf{x} - \mathbf{o})^\top \mathbf{a}$. As described in Sec. 3.3, the profile lies in a half-plane containing the rotation axis. During section assembly, we retain signed radial coordinates to preserve connectivity on opposite sides of the axis. Taking absolute radii before closing the region can fold a valid loop back onto itself. The assembled region is then restricted to the half-plane $r \geq 0$.

Angular support. Each lateral triangle contributes an angular coverage interval. We account for periodicity by joining intervals across the boundary between 0 and 2π when necessary. The complement of the largest uncovered angular gap gives the smallest arc containing the observed lateral evidence. If the evidence covers the complete angular period, the sweep interval spans 2π.

Angular cap planes provide endpoint constraints for partial revolutions. Two such planes alone do not determine which of the two complementary arcs was swept. Lateral angular coverage resolves this ambiguity, and the interval endpoints are aligned with the available cap evidence. This combines the precise endpoint information of caps with the coverage information of lateral triangles.

B.4 Incomplete Profiles and Difference Features

Completion order. Complete cap regions and closed lateral sections are used whenever available. For incomplete profiles, we first try local tangent extensions, followed by short boundary bridges and then convex or bounding-shape completions. Holes and disconnected regions supported by the evidence are preserved. Each completed profile must pass geometric verification. Completion does not use a separate learned model.

Unobserved portions of cutting tools. A Difference feature is not constrained by the observed surface outside the surrounding material. Its open profile is extended outward until it clears that material and is closed by a straight segment. The current target's cross-section supplies the local material extent. Similarly, a sweep endpoint without cap or lateral support uses a cut-through extent. Observed lateral endpoints are retained because extending beyond them could remove material that should remain.

B.5 Feature Construction and Verification

Executable construction. The CAD kernel sweeps the recovered profile regions, including holes and disconnected components, to obtain $C = \mathrm{eval}(f)$. Degenerate profiles, zero-volume sweeps, and failed constructions are rejected. Parameters are stored with each feature for regeneration during sequence execution.

Evidence absorption. Let $\mathbf{x}_\ell, \mathbf{n}_\ell$ be evidence samples and outward normals, with represented areas w_ℓ and locally scaled distance tolerances δ_ℓ. Let $\mathbf{n}_C(\mathbf{x}_\ell)$ denote the candidate normal at the closest boundary point. The absorption score is

$$\mathrm{Abs}(C, E_k) = \frac{\sum_\ell w_\ell \big[d(\mathbf{x}_\ell, \partial C) \le \delta_\ell\big] \big[\varsigma_\rho \, \mathbf{n}_\ell^\top \mathbf{n}_C(\mathbf{x}_\ell) \ge \cos 20^\circ\big]}{\sum_\ell w_\ell}, \tag{11}$$

where brackets are indicators and $\varsigma_\rho = 1$ for Union or -1 for Difference. The sign accounts for the opposite outward normals of the tool and remaining material along a cut surface. The normal test also excludes nearby surfaces with incompatible orientations. Candidates with absorption above τ_{abs} are retained. Distance tolerances account for the local tessellation scale, and area weights prevent small triangles from contributing disproportionately. Absorption checks whether an individual feature explains its surface evidence. Its compatibility with other features is evaluated by the learned prior and sequence search.

C FAILURE CASES

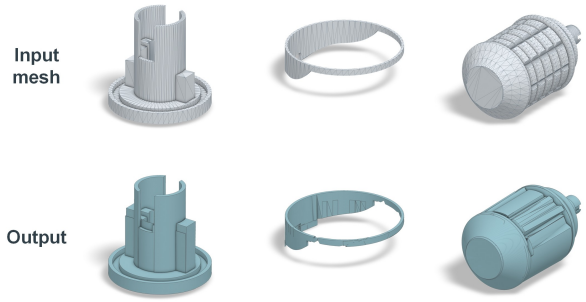

Figure 3: Representative failure cases. The top and bottom rows show input meshes and reconstructions, respectively. Poor proposal quality causes local reconstruction errors in the first column. In the middle and right columns, only some features involved in complex interactions are correctly proposed in the same round, causing errors to accumulate through residual recursion.

Figure 3 illustrates two sources of reconstruction errors. In the first column, poor proposal quality causes local details to be missing or distorted despite recovery of the main structure. The middle and right columns involve complex interactions among features that must be proposed correctly in the same round to obtain a satisfactory construction. If only some are available, the search cannot evaluate their complete Boolean interaction and may select an incomplete or distorted construction. The resulting residuals contain both unexplained geometry and errors introduced by that construction. As these residuals become the targets of subsequent rounds, early errors affect later proposals and accumulate through recursion.