REASONFIT: COUPLING VLM STRUCTURAL REASONING WITH DIFFERENTIABLE MULTI-VIEW FITTING FOR PRIMITIVE-BASED RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recovering a part-structured 3D model from multi-view images requires deciding which parts are present and estimating their geometry. These decisions depend on each other: an omitted part can distort its neighbours, while a restrictive primitive can fragment a single component. We present ReasonFit, a closed-loop framework that couples vision-language structural reasoning with differentiable geometric fitting. Within this loop, a vision-language model (VLM) proposes part structure and qualitative shape constraints, while differentiable fitting estimates geometry from multi-view appearance and silhouettes. We use deformable Super-Frustum primitives to capture curved profiles and varying cross-sections within a single part, with VLM shape cues guiding the allowed deformations. In turn, the fitted assembly and unexplained image regions guide the VLM in distinguishing missing parts from fitting errors and revising the structure. The updated assembly is jointly refined while preserving the fitted state of retained parts. Experiments across diverse objects show that ReasonFit produces more compact reconstructions than scaffold-based fitting. Evaluations of geometric fidelity, coverage, and dedicated part recovery further reveal the trade-off between compactness and preserving distinct components.



Figure 1: ReasonFit reconstructs compact, part-structured 3D models from multi-view images. By coupling VLM structural reasoning with differentiable geometric fitting, our method captures diverse object structures and complex part shapes, including curved handles, rounded bodies, and slender supports. Top: example input views and reconstructed assemblies. Below: additional reconstructions illustrating the range of supported shapes and compositions.

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.

1 INTRODUCTION

Reconstructing a 3D object from multi-view images requires more than recovering a surface consistent with its projections. A structured reconstruction must also explain how the object is composed: which parts are present, how they relate, and what geometry belongs to each part. Such an explanation provides a compact representation for understanding and modifying objects at the component level. Parametric primitives offer a natural basis for this representation (Tulsiani et al., 2017; Paschalidou et al., 2019), but recovering a meaningful primitive assembly from images remains difficult. The observations constrain the object's visible appearance without directly specifying its decomposition or the three-dimensional shape of its constituent parts.

This task brings together three interdependent challenges. Recovering part structure requires high-level reasoning: occlusion and ambiguous boundaries make part identities, counts, and relations difficult to establish from local image evidence. Recovering part geometry requires estimating continuous shape parameters without reference meshes, depth maps, or ground-truth part annotations. Even with calibrated views, different geometries and decompositions can explain similar observations. Finally, the representation must accommodate the geometric complexity within a semantic part. A component may bend, taper, or change cross-section along its extent; an overly restrictive primitive forces it into multiple fragments, while unrestricted deformation increases ambiguity during fitting. Successful reconstruction therefore depends jointly on structural interpretation, image-based geometric estimation, and the expressive capacity assigned to each part.

Crucially, errors in these decisions interact. When a part is omitted, neighbouring primitives can expand to explain its pixels; when an extra part is introduced, it can compete with a valid component for the same evidence. Both behaviours can reduce image reconstruction error while moving the assembly away from a meaningful decomposition. Differentiable primitive fitting provides a mechanism for matching multi-view observations (Monnier et al., 2023; Alaniz et al., 2023; Gao et al., 2025), but pixel agreement alone cannot establish whether the part inventory is correct. Conversely, a semantic description can guide decomposition, yet its omissions and mistaken assignments remain unresolved if it is treated as a fixed specification. These dependencies motivate a reconstruction process in which structural hypotheses guide geometric fitting and the fitted geometry supplies evidence for reconsidering those hypotheses.

We introduce ReasonFit, a framework that couples vision-language structural reasoning with differentiable geometric fitting in a closed loop. The central principle is to assign complementary responsibilities to the two processes. A vision-language model (VLM) reasons about part identities, multiplicities, relations, and qualitative shape attributes. Differentiable fitting estimates metric geometry from multi-view appearance and silhouettes. The VLM's description thus defines a structural hypothesis and an admissible space of shapes, while fitting grounds that hypothesis in the observations. The resulting assembly is subsequently returned to the VLM for structural review, making geometric reconstruction an intermediate source of evidence as well as an output.

To connect these levels of inference, a structural compiler grounds the VLM's relational description in a visual hull constructed from foreground masks. It converts qualitative relations into initial primitive placements and extents, without requiring the VLM to predict precise coordinates. We build on the rounded cross-sections, taper, and dilation of SuperFrustum (Ganeshan et al., 2026). For image fitting, we implement a direct longitudinal profile and an object-space bending parameterisation, each with corresponding surface and implicit-field evaluations. These let a rounded body or curved handle be fitted as one component through both appearance and silhouette evidence. The VLM's shape attributes determine which deformations each part can use: a curved component can bend, whereas a flat component need not acquire the same freedom. The representation and permissions define the available shapes; the input views determine their geometry.

During fitting, unexplained foreground regions give rise to candidate primitives. They may indicate missing parts, inaccurate geometry, or rendering gaps. The VLM reviews these candidates and the assembly alongside the input images to revise the inventory. By default, retained primitives carry their geometry and appearance forward. New parts and existing parts judged to have collapsed are initialised from the compiled description, then fitted jointly with the retained assembly.

Experiments across diverse objects show that ReasonFit recovers more compact geometry with improved geometric fidelity over scaffold-based fitting. Figure 1 illustrates the resulting part-structured

assemblies. Beyond these results, the framework offers a way to connect semantic understanding with geometric reconstruction: structural hypotheses guide what to fit, while fitted geometry provides evidence for what to reconsider. Our contributions are:

- A closed-loop reconstruction framework in which VLM structural reasoning and image-based geometric fitting exchange evidence. Inventory revisions carry forward retained states, with selective reinitialisation of collapsed parts before joint continuation.
- A structural compiler that grounds relational part descriptions in silhouette evidence, producing primitive initialisations and part-specific deformation permissions.
- An image-supervised fitting formulation for deformable SuperFrustum assemblies, with longitudinal profiles and object-space bending evaluated through paired surface and implicit paths. RGB rendering and mask-ray constraints update the same geometric parameters.

2 Related Work

Primitive representations and fitting. Cuboids, superquadrics, convex elements, and learned deformable parts provide different compromises between compactness and shape flexibility (Tulsiani et al., 2017; Paschalidou et al., 2019; Deng et al., 2020; Paschalidou et al., 2021). EMS fits point clouds, while Marching Primitives, SuperDec, and Light-SQ use 3D shape information to recover primitive decompositions (Liu et al., 2022; 2023; Fedele et al., 2025; Wang et al., 2025). ResFit introduces SuperFrustum and fits it to target occupancy and surface information (Ganeshan et al., 2026); its implementation already supports arc bending and spherical shapes. SuperFlex also uses bending and tapering for point-cloud decomposition (Tavernini et al., 2026). These representations motivate flexible parts, but do not by themselves resolve image-based fitting. DBW, ISCO, and DPA-Net recover primitive assemblies from images (Monnier et al., 2023; Alaniz et al., 2023; Yu et al., 2024). Gaussian splatting (Kerbl et al., 2023; Huang et al., 2024a) supports appearance modelling alongside geometry: GaussianBlock uses semantic guidance with splitting and fusion (Jiang et al., 2024), and PartGS binds 2D Gaussians to superquadrics (Gao et al., 2025). D-Prism considers dynamic reconstruction, while 3D Superquadric Splatting uses superquadrics as volumetric rendering elements (Yu et al., 2026; MacSwayne et al., 2026). We follow PartGS's surface-bound rendering approach. Our geometric design pairs deformable SF surfaces with implicit mask-ray queries, under part-specific semantic permissions.

Language-guided structured reconstruction. Shape programs and structured assemblies represent relations beyond a set of independent primitives (Tian et al., 2019; Jones et al., 2020; Mo et al., 2019; Paschalidou et al., 2020). CSGNet and PLAD learn program inference, and ShapeCoder discovers reusable abstractions (Sharma et al., 2018; Jones et al., 2022; 2023). Differentiable interpreters connect program structure to continuous geometry (Jones et al., 2020; Cascaval et al., 2022). Language models offer a way to infer such structure: LLM-Primitives generates primitive representations from shape information, Img2CAD separates discrete CAD structure from continuous attributes, and CADTalk labels existing program blocks (Tian et al., 2025; You et al., 2025; Yuan et al., 2024). Cadrille learns multimodal CAD generation, while SpatialBabel studies primitive scene code for spatial reasoning (Kolodiazhnyi et al., 2025; Liu et al., 2026). Kobsik et al. use a VLM and generated segmentations to identify parts before fitting superquadrics to underlying geometry (Kobsik et al., 2026). Our compiler instead grounds qualitative relations in mask-derived support and supplies both initial geometry and deformation permissions for image fitting.

Feedback and structural refinement. An executed program can reveal errors that are hard to detect in its symbolic description. SIRI combines parameter optimisation with code pruning and grafting, and VPI-Edit learns local program corrections (Ganeshan et al., 2023; Jones et al., 2024). BlenderAlchemy and CAD-Assistant use rendered or evolving scene states to guide edits (Huang et al., 2024b; Mallis et al., 2025); ShapeCraft iteratively refines procedural shape and texture programs (Zhang et al., 2026). Our feedback concerns the part inventory of an image-fitted assembly. The VLM interprets residual candidates and current part geometry, while the revised assembly retains fitted model parameters for subsequent optimisation. This separates structural correction from reconstructing the object anew.

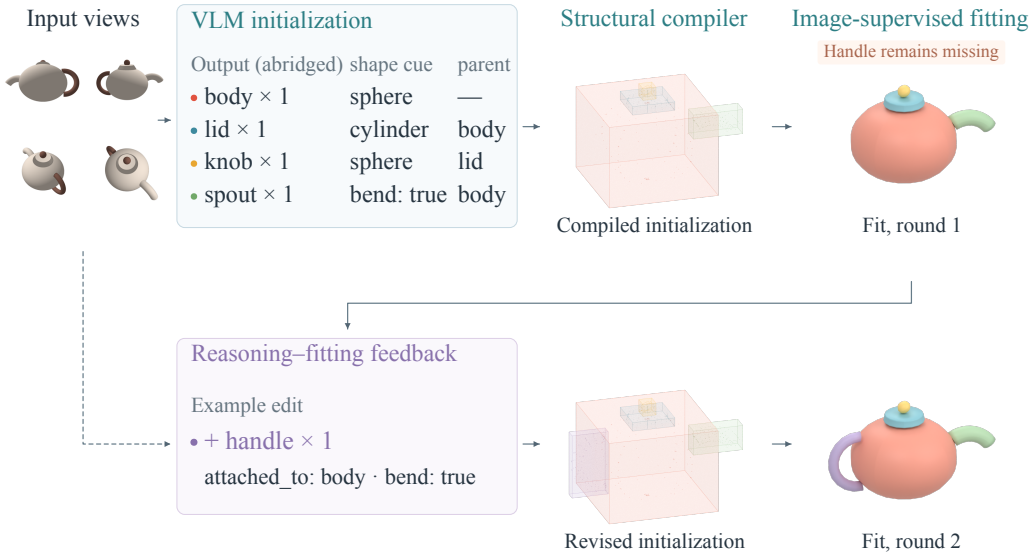


Figure 2: The reasoning–fitting loop. A VLM proposes parts and relations; the compiler grounds them in silhouette evidence. Image fitting refines the assembly and proposes residual candidates for VLM review. Retained and newly initialised parts are then fitted jointly. The missing-handle sequence is schematic.

3 METHOD

Given calibrated images $\mathcal{I} = \{I_n\}_{n=1}^{N}$ and foreground masks $\mathcal{M} = \{M_n\}_{n=1}^{N}$, we recover an assembly $\mathcal{P} = \{(\ell_i, \theta_i)\}_{i=1}^{K}$ of part labels and geometric parameters, using no reference mesh, depth map, or ground-truth part annotation. The VLM first proposes structure; a compiler initialises the corresponding parts, and differentiable fitting estimates their geometry. The fitted assembly then supplies evidence for VLM revision and joint refinement (Figure 2). The VLM makes discrete decisions, while continuous parameters are optimised per object without gradients through the VLM.

3.1 VLM-GUIDED STRUCTURAL INITIALIZATION

To establish which parts should be fitted, the VLM describes selected views in terms of part families, counts, arrangements, attachment relations, and shape cues. A family groups repeated parts; each instance normally receives one primitive, although the `pieces` field permits several. Shape cues specify qualitative attributes rather than metric dimensions. For example, four straight, tapered legs may be arranged radially below a seat, within the lower 30% of the object's height and near its periphery. Here *radial* means distributed around an axis, and each *angular sector* is the portion assigned to one leg.

To turn these relations into geometry, the compiler carves a visual hull (Laurentini, 1994): voxels must project inside every carving-view foreground mask. Boundary voxel centres form the support set $\mathcal{X}$. In the normalised object frame, z is up; axial intervals select a fraction of the hull's height or longest horizontal extent, and radial sectors partition its perpendicular plane. For the four legs, the compiler selects lower peripheral points, restricts them using the seat relation, and divides them into four groups. Each group's points determine initial placement, extent, and orientation. Optional VLM-predicted 2D image anchors guide these assignments by being lifted onto the hull. The assignments initialise geometry and do not constrain its subsequent motion.

The compiler also converts shape cues into binary permissions g_{ij} for primitive i and deformation group j. A sphere cue enables a rounded longitudinal profile; `bend: false` fixes curvature.

Thus, initialisation provides both an assembly and its allowed shape variations. Appendix A.1 details the schema, anchors, and coordinate rules.

3.2 Deformable SuperFrustum Representation

The initial part structure requires a geometric representation that can capture each component without fragmenting it. We use SuperFrustum (SF) (Ganeshan et al., 2026), whose rounded rectangular cross-section, taper, and dilation span box-like, cylindrical, and tapered solids. Figure 3 illustrates these parameters and the rounded and bent forms available in this family. For image fitting, we implement a direct longitudinal profile and object-space bending with paired surface and implicit evaluations. This lets appearance and silhouette evidence jointly fit a rounded body or a curved, non-convex handle as one part. Greater flexibility also increases ambiguity: several deformations may explain the same visible outline. The permissions g_{ij} freeze unused parameter groups to restrict this ambiguity.

Let $F_0(u, v; \eta_i)$ be the straight SF surface, P_{m_i} a longitudinal profile warp, D_{s_i} anisotropic scaling, and B_{κ_i,ϕ_i} bending with curvature κ_i and bend-plane angle ϕ_i. With pose (R_i, t_i),

$$x_i(u, v) = R_i B_{\kappa_i,\phi_i}(D_{s_i} P_{m_i}(F_0(u, v; \eta_i))) + t_i. \tag{1}$$

The profile acts before scaling, and bending acts on the scaled geometry. Implicit queries invert these transforms to test the same solid; the resulting field is sign-correct within the supported domain, not an exact world-space signed distance.

Rounded parts. The warp $P_m(x, y, z) = (m(y/h_y)x, y, m(y/h_y)z)$ contracts cross-sections towards the ends, where h_y is the local half-height. This separates a circular cross-section from a rounded longitudinal outline: a cylinder may still have flat ends. For rounded parts, we tie the end exponents and fix transition, taper, and side-wall bulge parameters to reduce competing freedoms. Axis lengths remain trainable, allowing ellipsoid-like shapes. The inverse radial warp exposes the same profile to mask constraints.

Bent parts. Axial bending (Barr, 1984) lets one primitive follow a curved handle. Curvature is defined after scaling in inverse object-length units. The forward map deforms the surface, and its inverse supports implicit queries. Bounds on folding and total arc angle keep this inverse usable; arc fitting to assigned hull points initialises curvature. Bend changes the central axis, whereas bulge bows the side wall. Pose and scale remain trainable under all permissions. Appendix A.2 gives the transforms and numerical treatment.

3.3 Image-Supervised Geometric Fitting

The compiled assembly provides coarse geometry, which must be refined to match the observations. We couple two differentiable paths through the same SF parameters. Surface-bound 2D Gaussians, as in PartGS (Gao et al., 2025), model appearance: their centres and covariances follow the deformed mesh, so RGB residuals update pose, scale, profile, and curvature. Alongside rendering, we query the primitive's implicit field along calibrated mask rays. Foreground rays should intersect the solid, while background rays should remain outside. This directly constrains primitive geometry rather than a rendered alpha image. Both paths evaluate the same profile and bend transforms: a geometric update changes the surface carrying the Gaussians and the solid tested by mask rays. Appearance and silhouettes therefore constrain a common part shape, rather than separate geometric proxies.

Let Θ denote geometry, Ψ Gaussian appearance and attachment parameters, and $o_i \in [0, 1]$ primitive retention weights. Gaussians bound to primitive i share peak opacity o_i. We minimise

$$\begin{aligned}\mathcal{L}(\Theta, \Psi, o) = \mathcal{L}_{\text{photo}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{occ}}\mathcal{L}_{\text{occ}} \\ + \lambda_{\text{ov}}\mathcal{L}_{\text{overlap}} + \lambda_{\text{par}}\mathcal{L}_{\text{par}} + \lambda_{\text{shape}}\mathcal{L}_{\text{shape}}.\end{aligned} \tag{2}$$

The appearance loss $\mathcal{L}_{\text{photo}}$ combines L_1 and SSIM (Wang et al., 2004). The mask loss $\mathcal{L}_{\text{mask}}$ penalises uncovered foreground and intersected background rays, with occupancy gating to exclude nearly invisible primitives. The ungated intersections train retention through $\mathcal{L}_{\text{occ}}$, so supported primitives can regain weight. The terms $\mathcal{L}_{\text{overlap}}$, $\mathcal{L}_{\text{par}}$, and $\mathcal{L}_{\text{shape}}$ penalise overlap, unnecessary primitives, and unconstrained deformation, respectively (Appendix A.3).

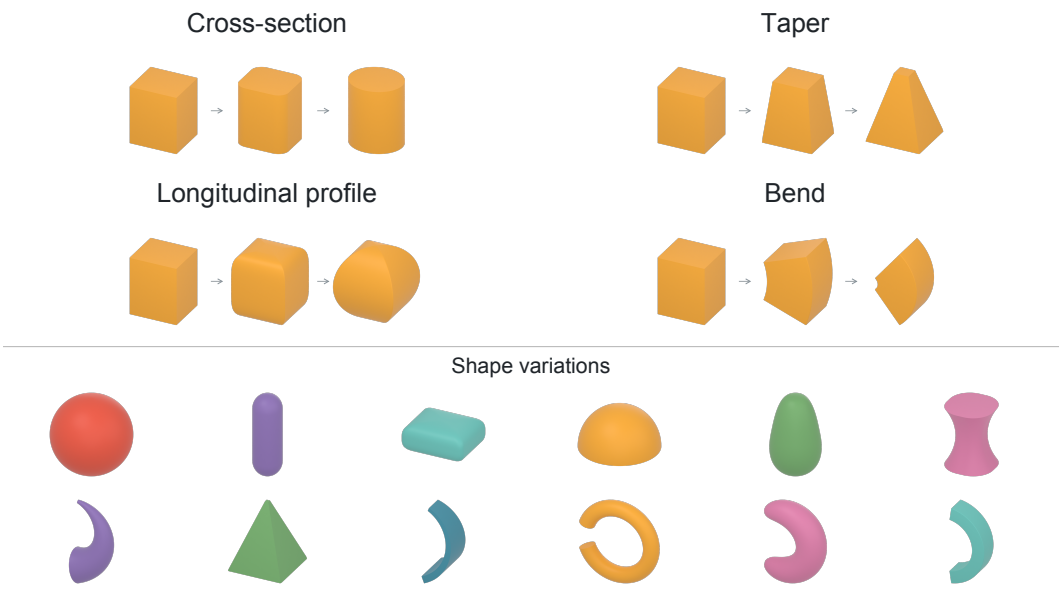


Figure 3: SuperFrustum shape controls. Top: cross-section rounding, taper, longitudinal profiles, and bending. Bottom: their combinations express rounded, tapered, and curved parts.

To cover regions missed by the initial structure, foreground pixels with insufficient rendered coverage vote for voxels outside current primitives. Cross-view mask consistency filters these voxels, and their groups initialise unlabelled candidates with conservative deformation permissions. Birth is restricted to a fixed window and population cap; parsimony and low-occupancy pruning control the count. Since residuals can also reflect fitting errors, these candidates require semantic review.

3.4 Reasoning–Fitting Feedback Loop

To distinguish missing parts from geometric errors, the VLM reviews the fitted assembly, isolated primitives, original images, and initial description. Renderings use stable primitive IDs and colours. The VLM associates semantic instances with fitted primitives, including residual candidates, and identifies missing, redundant, or collapsed parts.

The conservative revision policy adds missing instances and reinitialises collapsed ones when both correspondence queries agree on their identity and collapsed status. These parts use the compiler's initial geometry and appearance. Other primitives retain their geometry, retention weights, appearance, Gaussian attachments, and permissions; ordinary size or shape errors remain for fitting to correct. Deletion is disabled in this setting. We then fit the revised assembly jointly with birth and pruning disabled, preserving fitted evidence while accommodating the accepted structural edits. Appendix A.4 specifies the correspondence and state-transfer rules.

4 Experiments

We compare reconstruction quality and compactness against image-based primitive-fitting methods, then examine how the geometric controls and structural feedback contribute to the resulting assemblies.

4.1 Experimental setup

Data and inputs. We evaluate on Synthetic, ShapeNet (Chang et al., 2015), and Google Scanned Objects (GSO) (Downs et al., 2022). Synthetic contains 41 designed assemblies with 338 labelled parts, enabling evaluation of both geometry and part separation. Its furniture, tools, and toy-like objects combine slender, rounded, and curved components. ShapeNet contains 31 airplane, chair,

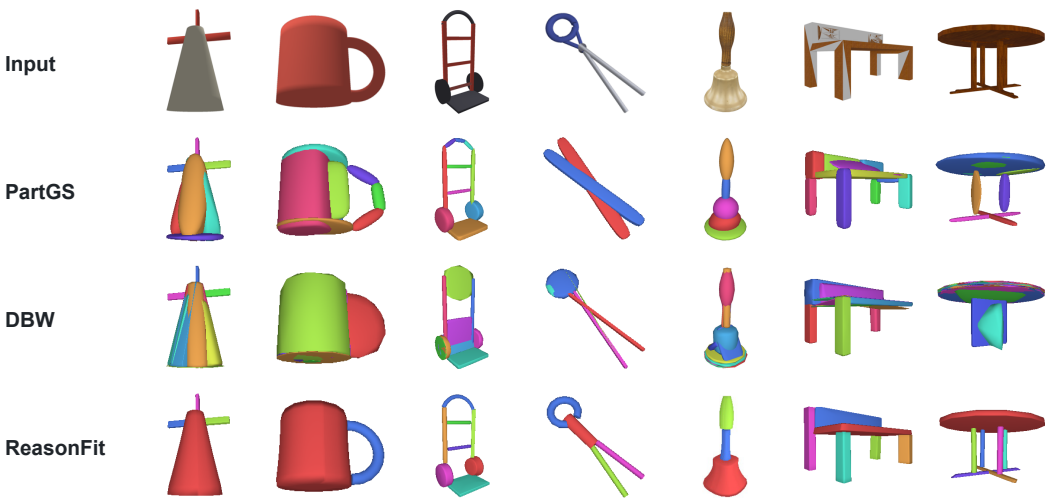


Figure 4: Reconstruction from the same image inputs. Rows show input views, PartGS, DBW, and ReasonFit. Colours identify primitives within each result; curved handles and slender supports reveal differences in shape and decomposition.

and table models; GSO contains 20 scanned household objects, extending the geometry evaluation to more varied shapes. We render 100 calibrated views and foreground masks per object at 800×800 and fit at half resolution. Reference surfaces and part labels are used only for evaluation. Rendering scores use the fitting views.

All main-comparison results on Synthetic, ShapeNet, and GSO use the full pipeline: initial fitting, VLM structural revision, and joint post-revision refinement. The total fitting budget is 30k iterations per object, including refinement, with two VLM correspondence queries for revision. PartGS also uses 30k fitting iterations; DBW follows its original training recipe. Appendices B.1 and B.5 give data, camera, and experiment settings; Table 2 specifies the separate ablation budgets.

VLM configuration. Qwen3.8-27B (Qwen Team, 2026), accessed through OpenRouter at temperature zero, receives six views selected by farthest-point sampling of camera positions. Prompts use anonymous image IDs and a validated part-description schema. Repeated fitting seeds share initial descriptions, measuring optimisation variability conditional on those descriptions (Appendix A.6).

Metrics. Chamfer distance (CD), coverage, and primitive count measure geometric fidelity and compactness. *Dedicated recovery* counts a reference part when a nearby primitive explains at least half its samples and is not dominant for another covered part. It measures whether parts receive separate primitives; multiple primitives are not pooled by semantic label. Definitions and thresholds appear in Appendices B.2 and B.3.

4.2 Part-structured reconstruction

To assess whether semantic reasoning yields compact assemblies without losing geometric detail, we compare with PartGS (Gao et al., 2025) and DBW (Monnier et al., 2023) on the same images, using their original fitting objectives (Table 1). The visual comparisons in Figure 4 show how ReasonFit follows the mug handle as one curved component and preserves slender table supports, while the baselines fragment the handle or merge and distort supports.

On Synthetic, ReasonFit reduces geometric error and represents objects with fewer primitives than either baseline (CD 0.83 and $K/G = 0.92$). It covers more parts than PartGS (85.2% versus 77.8%) and achieves higher dedicated recovery (56.8% versus 51.8%). The gap between coverage and dedicated recovery nevertheless indicates that some components remain merged. On ShapeNet and GSO, ReasonFit also improves CD over both baselines (0.78 and 1.19) while using fewer primitives (7.9 and 5.5). These sets have no part labels in this evaluation.

Table 1: Reconstruction quality and compactness. On all three datasets, ReasonFit uses the full reasoning–fitting loop with 30k total fitting iterations, including post-revision refinement, and two revision queries. PartGS uses 30k iterations and DBW its original recipe (Appendix B.5). Ded. and Cov. are percentages of dedicated and covered reference parts; K/G is the primitive-to-part ratio and K the mean primitive count. CD is symmetric unsquared Chamfer distance in % of the reference bounding-box diagonal. Aggregation follows Appendix B.4.

	Synthetic				ShapeNet		GSO	
Method	Ded.↑	Cov.↑	K/G	CD↓	CD↓	K	CD↓	K
ReasonFit (full pipeline)	56.8	85.2	0.92	0.83	0.78	7.9	1.19	5.5
PartGS	51.8	77.8	1.10	0.94	1.00	9.2	1.29	9.4
DBW	48.0	72.0	1.48	1.05	0.87	11.1	1.28	8.7

Table 2: Ablation experiments on the Synthetic dataset. The two sections use different object sets and starting fits (Appendix B.5). Top: 30k initial-fit component ablations, reporting mean and sample standard deviation across seeds 0 and 1. Bottom: a separate 30k+5k revision ablation with paired 5k continuations from shared 30k fits, averaged across seeds 0 and 1. Revision uses two correspondence queries; metrics follow Table 1.

Variant	Ded.↑	Cov.↑	K/G	CD↓
Initial fitting: 30k iterations, two seeds				
All fitting components	40.5 ± 0.0	90.5 ± 3.4	0.71 ± 0.00	0.50 ± 0.00
w/o birth and parsimony	34.5 ± 1.7	86.9 ± 1.7	0.69 ± 0.00	0.63 ± 0.01
w/o shape permissions and initialisers	32.1 ± 1.7	82.1 ± 1.7	0.75 ± 0.05	1.01 ± 0.04
w/o coarse losses	23.8 ± 6.7	47.6 ± 13.5	1.08 ± 0.02	2.67 ± 1.80
Structural revision: two seeds, paired checkpoints				
Before revision (30k)	51.9	82.7	1.10	1.043
No-edit continuation (30k + 5k)	53.8	94.2	1.10	0.746
VLM revision (30k + 5k)	86.5	96.2	1.15	0.655

4.3 Ablations

To assess the roles of candidate generation, shape conditioning, and image constraints, we conduct ablation experiments on the Synthetic dataset. Table 2 summarises the results. Removing birth and parsimony reduces dedicated recovery from 40.5% to 34.5% and raises CD from 0.50 to 0.63: the initial inventory alone misses useful components. Removing shape permissions and their initialisers raises CD to 1.01, showing the importance of conditioning the available deformations and their starting geometry together. Disabling coarse losses degrades CD to 2.67; this removes silhouette and scheduled regularisation terms jointly, rather than isolating mask supervision.

To test whether structural edits help beyond additional optimisation, we use a separate object set and a dedicated 30k+5k ablation schedule: each 30k fit is continued for 5k iterations with and without VLM revision. Each pair shares its starting checkpoint and continuation seed; results are averaged across seeds 0 and 1. Compared with no-edit continuation, revision improves dedicated recovery from 53.8% to 86.5% and reduces CD from 0.746 to 0.655. Coverage increases from 94.2% to 96.2%, while K/G rises from 1.10 to 1.15. These results associate structural revision with better part separation and geometric fidelity at a modest increase in primitive count.

To illustrate this effect, Figure 5 (bottom) shows a table whose rounded feet remain missing after no-edit continuation but are recovered by VLM revision. The pinwheel row independently varies birth and parsimony to show their effects on missing and redundant geometry.

4.4 Primitive expressiveness

To evaluate whether individual primitives can capture rounded and curved parts, we disable the corresponding deformations (Figure 5, top). Without longitudinal profiles, the table feet retain flat

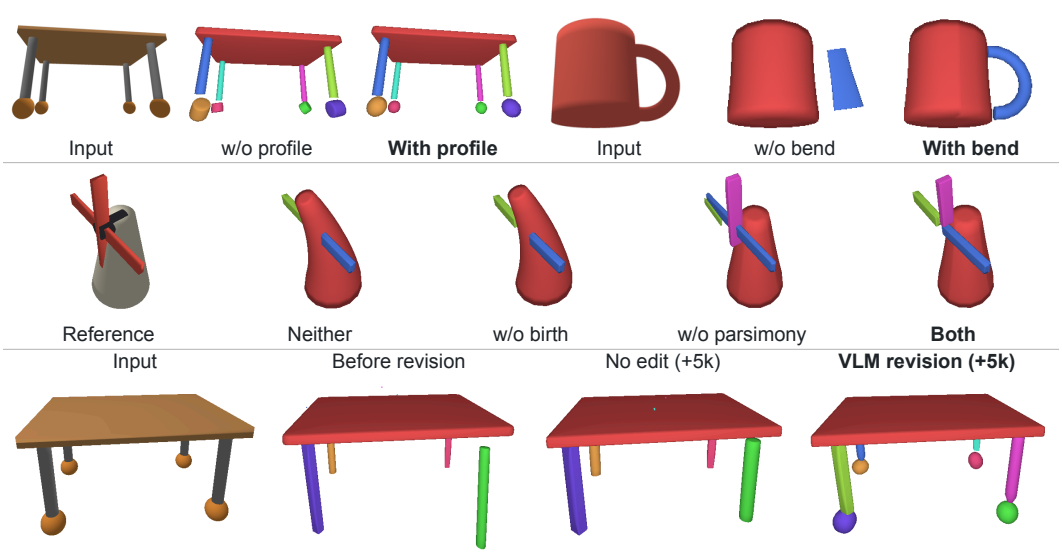


Figure 5: Qualitative ablations. Top: profile and bend controls. Middle: birth and parsimony. Bottom: one table before and after paired 5k ablation continuations, with and without VLM revision. Revision uses two correspondence queries. Colours identify primitives.

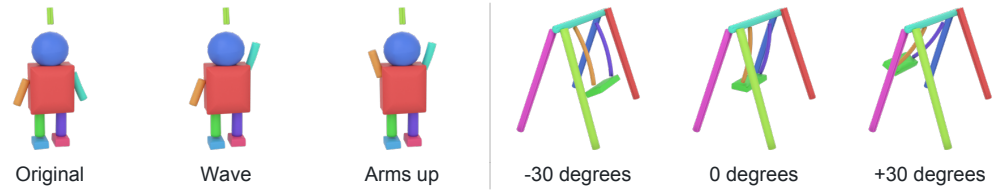


Figure 6: Part-level editing about manually specified pivots. Left: robot arm poses. Right: the swing's seat-and-rod group at $-30°$, $0°$, and $+30°$.

ends; without bending, the mug handle remains straight. Enabling these controls lets one primitive follow each part's outline, connecting the parameter variations in Figure 3 to fitted geometry.

Part-level editing. We edit robot arm poses and rotate the swing's seat-and-rod group about manually specified pivots, preserving the grouped parts' relative arrangement (Figure 6).

5 Discussion and Conclusion

We presented ReasonFit, a closed-loop framework that couples VLM structural reasoning with differentiable geometric fitting for multi-view reconstruction. Expressive SuperFrustum primitives connect semantic parts to continuous geometry, while fitted assemblies provide evidence for revising structural hypotheses. Experiments across diverse objects show more compact reconstructions with improved geometric fidelity, while highlighting that geometric coverage does not guarantee distinct part recovery. More broadly, this work suggests a direction for structured 3D reconstruction: high-level reasoning should both guide geometric optimisation and remain open to correction by the evidence it produces.

Limitations. The method requires calibrated views and foreground masks. Occlusion, competition between primitives, and VLM correspondence errors can leave parts missing, merged, or disconnected. Residual birth can miss parts already explained by neighbours. Permissions reduce deformation ambiguity but cannot resolve these structural errors, and one primitive cannot express every component. Appendix C discusses further cases.

AI USE STATEMENT

Generative AI tools were used only to assist manuscript writing. The authors reviewed and edited the resulting text and take responsibility for the manuscript, implementation, and results. The VLM within the method is described in Section 4.1.

ETHICS STATEMENT

This work studies part-structured reconstruction of objects from images. The predicted parts and shapes are approximations: they do not establish an object's internal structure, mechanical properties, or suitability for manufacturing or robotic manipulation. The experiments use synthetic objects and rendered views of ShapeNet and Google Scanned Objects. Use and redistribution of third-party assets and models are subject to their respective licences.

REPRODUCIBILITY STATEMENT

Section 3 describes the reconstruction stages, and Appendix A gives the primitive parameterisation, fitting objective, and state transfer across structural edits. Appendix B defines the camera protocol, evaluation metrics, and paired comparisons. Appendix A.6 describes the part-description schema and validation interface. Repeated fitting runs use cached VLM descriptions, separating optimisation variability from variation across VLM calls.

REFERENCES

Stephan Alaniz, Massimiliano Mancini, and Zeynep Akata. Iterative superquadric recomposition of 3d objects from multiple views. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17967–17977. IEEE, 2023.

Alan H Barr. Global and local deformations of solid primitives. *ACM Siggraph Computer Graphics*, 18(3):21–30, 1984.

Dan Cascaval, Mira Shalah, Phillip Quinn, Rastislav Bodik, Maneesh Agrawala, and Adriana Schulz. Differentiable 3d cad programs for bidirectional editing. In *Computer Graphics Forum*, volume 41, pp. 309–323. Wiley Online Library, 2022.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Daniel Cremers and Kalin Kolev. Multiview stereo and silhouette consistency via convex functionals over convex domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6): 1161–1174, 2010.

Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31–41. IEEE, 2020.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 international conference on robotics and automation (ICRA)*, pp. 2553–2560. Ieee, 2022.

Elisabetta Fedele, Boyang Sun, Leonidas Guibas, Marc Pollefeys, and Francis Engelmann. Superdec: 3d scene decomposition with superquadric primitives. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24625–24635. IEEE, 2025.

Aditya Ganeshan, R Kenny Jones, and Daniel Ritchie. Improving unsupervised visual program inference with code rewriting families. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15745–15755. IEEE, 2023.

Aditya Ganeshan, Matheus Gadelha, Thibault Groueix, Zhiqin Chen, Siddhartha Chaudhuri, Vladimir Kim, Wang Yifan, and Daniel Ritchie. Residual primitive fitting of 3d shapes with superfrusta. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7404–7413, 2026.

Zhirui Gao, Renjiao Yi, Yuhang Huang, Wei Chen, Chenyang Zhu, and Kai Xu. Self-supervised learning of hybrid part-aware 3d representations of 2d gaussians and superquadrics. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9649–9659. IEEE, 2025.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 2024a.

Ian Huang, Guandao Yang, and Leonidas Guibas. Blenderalchemy: Editing 3d graphics with vision-language models. In *European Conference on Computer Vision*, pp. 297–314. Springer, 2024b.

Shuyi Jiang, Qihao Zhao, Hossein Rahmani, De Wen Soh, Jun Liu, and Na Zhao. Gaussianblock: Building part-aware compositional and editable 3d scene by primitives and gaussians. *arXiv preprint arXiv:2410.01535*, 2024.

R Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J Mitra, and Daniel Ritchie. Shapeassembly: Learning to generate programs for 3d shape structure synthesis. *ACM Transactions on Graphics (TOG)*, 39(6):1–20, 2020.

R Kenny Jones, Homer Walke, and Daniel Ritchie. Plad: Learning to infer shape programs with pseudo-labels and approximate distributions. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9861–9870. IEEE, 2022.

R Kenny Jones, Paul Guerrero, Niloy J Mitra, and Daniel Ritchie. Shapecoder: Discovering abstractions for visual programs from unstructured primitives. *ACM Transactions on Graphics (TOG)*, 42(4):1–17, 2023.

R Kenny Jones, Renhao Zhang, Aditya Ganeshan, and Daniel Ritchie. Learning to edit visual programs with self-supervision. *Advances in Neural Information Processing Systems*, 37:109407–109429, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Gregor Kobsik, Tim Elsner, and Leif Kobbelt. Harnessing generative image models for training-free primitive shape abstraction. *arXiv preprint arXiv:2607.05568*, 2026.

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

Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000.

Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on pattern analysis and machine intelligence*, 16(2):150–162, 1994.

Aldo Laurentini. How far 3d shapes can be understood from 2d silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):188–195, 1995.

Junze Liu, Kun Qian, Florian Dubost, Kai Zhong, Arvind Srinivasan, Nan Chen, Anping Wang, Sam Zhang, Alejandro Mottini, Qingjun Cui, et al. 3d primitives are a spatial language for vlms. *arXiv preprint arXiv:2605.12586*, 2026.

Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Robust and accurate superquadric recovery: A probabilistic approach. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2666–2675. IEEE, 2022.

Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Marching-primitives: Shape abstraction from signed distance function. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8771–8780. IEEE, 2023.

Daniel MacSwayne, Ales Leonardis, and Jianbo Jiao. 3d superquadric splatting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5154–5163, 2026.

Dimitrios Mallis, Ahmet Serda Karadeniz, Sebastian Cavada, Danila Rukhovich, Niki Foteinopoulou, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. Cad-assistant: Tool-augmented vllms as generic cad task solvers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7284–7294. IEEE, 2025.

Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019.

Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei Efros, and Mathieu Aubry. Differentiable blocks world: Qualitative 3d decomposition by rendering primitives. *Advances in Neural Information Processing Systems*, 36:5791–5807, 2023.

Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10336–10345. IEEE, 2019.

Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1060–1070, 2020.

Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3203–3214. IEEE, 2021.

Qwen Team. Qwen3.8-27B model card, 2026. URL `https://huggingface.co/Qwen/Qwen3.8-27B`. Accessed September 25, 2026.

Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International journal of computer vision*, 35(2):151–173, 1999.

Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. Csgnet: Neural shape parser for constructive solid geometry. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5515–5523, 2018.

Gabriel Tavernini, Elisabetta Fedele, Tiago Novello, Leonidas Guibas, Marc Pollefeys, and Francis Engelmann. Superflex: Deformable superquadrics for point cloud decomposition. In *European Conference on Computer Vision*, pp. 457–474. Springer, 2026.

Kuan Tian, Zhihao Hu, Yonghang Guan, and Jun Zhang. Llm-primitives: Large language model for 3d reconstruction with primitives. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Learning to infer and execute 3d shape programs. *arXiv preprint arXiv:1901.02875*, 2019.

Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1466–1474. IEEE, 2017.

Yuhan Wang, Weikai Chen, Zeyu Hu, Runze Zhang, Yingda Yin, Ruoyu Wu, Keyang Luo, Shengju Qian, Yiyan Ma, Hongyi Li, et al. Light-sq: Structure-aware shape abstraction with superquadrics for generated meshes. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Yang You, Mikaela Angelina Uy, Jiaqi Han, Rahul Thomas, Haotong Zhang, Yi Du, Hansheng Chen, Francis Engelmann, Suya You, and Leonidas Guibas. Img2cad: Reverse engineering 3d cad models from images through vlm-assisted conditional factorization. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

Fenggen Yu, Yiming Qian, Xu Zhang, Francisca Gil-Ureta, Brian Jackson, Eric Bennett, and Hao Zhang. Dpa-net: Structured 3d abstraction from sparse views via differentiable primitive assembly. In *European Conference on Computer Vision*, pp. 454–471. Springer, 2024.

Xingyuan Yu, Yijin Li, Chong Zeng, Yuhang Ming, Hujun Bao, and Guofeng Zhang. D-prism: Differentiable primitives for structured dynamic modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7556–7566, 2026.

Haocheng Yuan, Jing Xu, Hao Pan, Adrien Bousseau, Niloy J Mitra, and Changjian Li. Cadtalk: An algorithm and benchmark for semantic commenting of cad programs. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3753–3762. IEEE, 2024.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

Shuyuan Zhang, Chenhan Jiang, Zuoou Li, and Jiankang Deng. Shapecraft: Llm agents for structured, textured and interactive 3d modeling. *Advances in Neural Information Processing Systems*, 38:65116–65144, 2026.

A Implementation Details

A.1 Part description and structural compiler

The description records part families, instance counts, arrangements, attachment relations, and qualitative shape cues. Bilateral arrangements separate opposite sides of the reference frame, radial arrangements divide angular sectors, and stacked arrangements divide axial intervals. Attachment and relative-position relations further restrict the candidate hull points. The compiler records unassigned points and slots with little geometric support. These are initialisation diagnostics rather than part-level supervision. Each family has a unique name, a positive instance count, and an arrangement chosen from single, bilateral, radial, or stacked. Shape cues distinguish box, slab, cylinder, frustum, sphere, and other shapes, with separate flags for circular cross-sections and bending. An axial interval lies in $[0, 1]$; a radial category distinguishes core from peripheral parts. A child can name its parent, attachment site, and qualitative relative size, position, and orientation. The schema treats counts as inventory proposals rather than reference labels.

Validation normalises names and checks the categorical fields, count ranges, and interval endpoints. The compiler converts family counts into instance slots in an object-level frame and assigns hull points to each slot. These points determine the initial boxes; the language description does not supply their metric coordinates. Fitting can move primitives beyond these initial assignments. Paired fitting runs share the compiled boxes as well as the part descriptions.

Carving and coordinate conventions. The normalised working volume is the same cube $[-0.55, 0.55]^3$ for every object. The default carver uses a 200^3 grid and every second input view. An RGBA input pixel is foreground when its 8-bit alpha exceeds 12. Projection tests use positive camera depth and in-frame pixel coordinates; a voxel must pass the mask test in every selected view. We use occupied voxels without a fully occupied six-neighbourhood as hull surface support. This fixed volume is a dataset coordinate convention, not an object-specific bound estimated from its reference geometry.

Image-anchor acquisition. Anchors are produced by a separate VLM localisation pass after the part inventory has been generated and before compilation. The localisation prompt gives only the family names and expected counts, together with calibrated input views, and asks for one point near the centre of each visible instance. It requests no point for an occluded instance. Each response records a family name, a view-local instance index, a normalised image coordinate (u, v), and confidence. The origin is the upper-left corner; u increases rightward and v downward. Instance indices are not assumed to correspond across views.

Each displayed view is associated with its calibrated camera and original image dimensions. The compiler converts (u, v) to pixels using those dimensions and projects hull points with the corresponding camera. It gathers support in a window of half-width $0.02 \max(W, H)$ around the point, takes a depth-filtered median position, and clusters same-family positions across views. The resulting 3D anchors guide spatial assignment and frame selection; they are not metric predictions by the VLM. When image anchors are unavailable, the compiler retains the declared axis and resolves its horizontal sign from hull spread in the peripheral-part intervals.

The semantic frame takes z as up. A horizontal-axis description selects the larger of the hull's x and y extents. Image anchors are lifted to hull points, and the horizontal sign is chosen by counting anchors within their declared axial intervals, weighted by the number of supporting views. Ties are resolved by transverse hull spread in peripheral-part intervals. The compiler switches to the vertical axis only when its anchor score is at least 1.5 times the best horizontal score and covers at least 80% of anchor-view support. A vertical axis runs from bottom to top. Cached initial boxes fix these choices for all fitting seeds. Radial directions are measured around the axis through the median transverse hull position. The four-leg example in Section 3.1 illustrates how a VLM description specifies these relations.

A.2 Primitive parameterisation

The base SuperFrustum supplies a rounded cross-section, taper, and an outward offset (Ganeshan et al., 2026). Its public implementation also contains an arc mapping, called `map_arc_bulge`, and

a `SolidSF` sphere branch that suppresses the base extents and forms a sphere through dilation. We retain the base cross-section construction but use the longitudinal profile and object-space bending transform below. Throughout this paper, *roundness* means cross-section corner rounding, *profile* means longitudinal end shape, *bulge* means side-wall bowing, and *bend* means curvature of the central axis. Our bulge parameter is distinct from the same-named arc control in ResFit.

Base implicit field. In the primitive's undeformed local frame, let h_x, h_y, h_z be its half-extents, $\rho = \min(h_x, h_z)$, and $r = c\rho$ for cross-section roundness $c \in [0, 1]$. For a query point (x, y, z), define

$$q = (|x|, |z|) - (h_x, h_z) + (r, r), \tag{3}$$

$$d_2(x, z) = \| \max(q, 0)\|_2 + \min(\max(q_1, q_2), 0) - r. \tag{4}$$

This is the signed distance to the rounded rectangular cross-section. Let a control taper and let $\mathcal{T}_a$ be the trapezoid with vertices $(-\rho, h_y)$, $(-\rho, -h_y)$, $(0, -h_y)$, and $((a-1)\rho, h_y)$. The base field used for implicit queries is

$$f_0(x, y, z) = \mathrm{sd}_{\mathcal{T}_a}\big(d_2(x, z), y\big) - \delta, \tag{5}$$

where $\mathrm{sd}_{\mathcal{T}_a}$ is signed distance to this 2D trapezoid and δ is the outward offset. Negative values denote the interior. When bulge b is enabled, d_2 is replaced by $d_2 - b\rho(1-\hat{y}^2)$, with $\hat{y} = \mathrm{clip}(y/h_y, -1, 1)$. This term moves the side wall while leaving its end rings unchanged. After deformation, the field is used for inside/outside queries and is not assumed to be an exact Euclidean signed distance.

Longitudinal profile. Let $\xi = y/h_y \in [-1, 1]$ denote normalised height along the straight primitive's local axis. The longitudinal profile scales the cross-section by

$$u_t(\xi) = \mathrm{clip}\left(\frac{\xi - r_t}{1 - r_t}, 0, 1\right), \qquad u_b(\xi) = \mathrm{clip}\left(\frac{r_b - \xi}{1 + r_b}, 0, 1\right), \tag{6}$$

$$m(\xi) = \Big(1 - u_t(\xi)^{2/e_t}\Big)^{e_t/2} \Big(1 - u_b(\xi)^{2/e_b}\Big)^{e_b/2}. \tag{7}$$

Here $e_t, e_b > 0$ control the end profiles and $r_t, r_b \in (-1, 1)$ delimit their extent. The surface map applies $(x, y, z) \mapsto (m(\xi)x, y, m(\xi)z)$. For an untapered circular cross-section with no bulge or offset, symmetric unit exponents and zero reach give the ideal spherical profile when the canonical radial and axial half-extents agree. Anisotropic scaling then permits ellipsoids; a sphere cue alone does not set equal axis lengths or fix the shared exponent to one. The implementation clamps normalised height to $[-0.999, 0.999]$ and floors power bases and the inverse-warp denominator near the poles. It therefore uses small end rings rather than exact singular poles, and the tessellated surface approximates the ideal profile. For implicit queries, the profile is applied by the inverse radial warp

$$f_{\mathrm{prof}}(x, y, z) = m(y/h_y) f_0\left(\frac{x}{m(y/h_y)}, y, \frac{z}{m(y/h_y)}\right), \tag{8}$$

The multiplier preserves the zero set and provides a local distance scaling; it does not restore exact Euclidean distances away from the surface.

Bending. Bending follows an axial deformation (Barr, 1984). In the bend plane, for curvature κ and $\beta = \kappa y$, the transform is

$$B_\kappa(x, y, z) = \left(x,\ \frac{\sin\beta}{\kappa} - z\sin\beta,\ \frac{1 - \cos\beta}{\kappa} + z\cos\beta\right). \tag{9}$$

Its continuous limit at $\kappa = 0$ is the identity. The forward map uses small-angle expressions for $\sin\beta/\beta$ and $(1 - \cos\beta)/\beta$. Bending follows local anisotropic scaling, so κ has inverse object-length units; rotations about the local axis select the bend plane. For nonzero curvature, the implicit path recovers the straight coordinates from the signed radial distance to the arc centre and an $\mathrm{atan2}$ angle. For $|\kappa| < 10^{-4}$, that inverse uses the identity approximation: curvature gradients in this interval come from the surface-rendering path, not from the implicit inverse. The inverse is restricted to the non-folding domain below and is not globally unique for arbitrary curvature or overlapping turns.

The VLM-predicted shape cues gate these parameter groups separately. In particular, a circular cross-section cue does not imply a rounded longitudinal profile. The mask of enabled parameters is carried with each primitive through fitting and revision. The permission-aware compiler enables longitudinal profiles for sphere or ellipsoid cues and bending for an explicit bend cue. A circular cylinder therefore need not receive a rounded longitudinal profile. Taper cues and assigned end cross-sections initialise tapered parts. The simplified rounded part configuration shares the two profile exponents, fixes their transition positions at zero, and fixes taper and bulge at their neutral values. Scale and pose remain trainable; a sphere cue alone does not enforce equal scales.

Profile exponents lie in $(0.02, 1)$ and transition positions in $(-0.995, 0.995)$. The signed bulge parameter uses a scaled hyperbolic tangent with magnitude bound 0.35. In the object-space bending mode, curvature is bounded by both the cross-section radius and the half-length: $|\kappa| < \min(12, 0.95/r_{\max}, 0.98\pi/h)$, where $r_{\max}$ includes outward dilation and positive bulge. These bounds limit cross-section folding and the nominal arc span; the inverse is used on its non-folding branch, not as a globally bijective map. In the reported experiments, the local bend plane is fixed after initialisation; the primitive's overall pose remains optimisable. The reported main configuration uses solid primitives; cavity and through-hole templates are separate configurations.

The solid surface template has 12 longitudinal rings and 36 vertices around each ring: five samples per straight cross-section segment and four per corner arc. Each end cap uses a shared boundary ring, an inner ring, and a centre vertex. Vertex positions are differentiable functions of the shape parameters, while triangle connectivity is fixed. The implicit field and rendered mesh implement the same deformation model but are not numerically identical surfaces: the latter is a finite tessellation, and the former is not an exact world-space signed distance after deformation.

A.3 Fitting objective and optimisation

For a rendered image $\hat{I}$ and an observed image I, the photometric term is

$$\mathcal{L}_{\text{photo}} = (1 - \lambda_s) w_{\text{rgb}} \|\hat{I} - I\|_1 + \lambda_s \big(1 - \text{SSIM}(\hat{I}, I)\big), \tag{10}$$

where pixel and channel errors are averaged. SSIM follows Wang et al. (2004). Removing appearance supervision requires disabling both terms in Equation 10, not just its L_1 coefficient.

Let $f_i(x)$ be primitive i's implicit field after inverse pose, bending, scale, and profile transforms, negative inside. The ray hinges use $\bar{f}_i(x) = a_i f_i(x)$, where a_i is the detached mean of the three object-space scale factors. This positive factor adjusts the relative units of different primitives without changing their inside/outside tests; it is not an exact distance correction under anisotropic deformation. The occupancy-gated field and ray minimum are

$$\widetilde{f}_i(x) = \bar{f}_i(x) + C\,\mathbf{1}[\text{sg}(o_i) < \tau_o], \qquad q(r) = \min_{i,j} \widetilde{f}_i(x_{rj}), \tag{11}$$

where x_{rj} are samples on ray r, sg stops gradients, $\tau_o = 0.1$, and $C = 10^3$. The positive offset suppresses low-occupancy primitives in both foreground and background geometry losses. With foreground and background ray sets $\mathcal{R}_f, \mathcal{R}_b$, the mask losses are

$$\mathcal{L}_{\text{fg}} = \frac{1}{|\mathcal{R}_f|} \sum_{r \in \mathcal{R}_f} [q(r) - \epsilon]_+, \qquad \mathcal{L}_{\text{bg}} = \frac{1}{|\mathcal{R}_b|} \sum_{r \in \mathcal{R}_b} [-q(r) - \epsilon]_+. \tag{12}$$

Here $\epsilon = 10^{-3}$, and empty ray sets contribute zero. Separate weights scale the foreground and background terms. These losses depend on primitive geometry and the detached occupancy gate, not on rendered Gaussian alpha. A low-opacity primitive therefore cannot satisfy a foreground ray merely by placing invisible geometry along it. This gate is a finite field offset, not deletion; even if every primitive is below threshold, the foreground hinge remains active.

Occupancy supervision is evaluated separately, without this field offset. Define $H_i(r) = \mathbf{1}[\min_j \bar{f}_i(x_{rj}) \le 10^{-8}]$ and $p(r) = \max_i o_i H_i(r)$. Binary cross-entropy compares $p(r)$ with the mask label, using only intersected foreground rays and all sampled background rays. Foreground rays are weighted inversely by the number claimed by the maximising primitive and averaged over claiming primitives; background rays are averaged per ray. The two averages receive equal weight. The hard intersection test does not differentiate geometry: this loss trains o_i, whereas the ray hinges train geometry. Because H_i is not occupancy-gated, a suppressed primitive with foreground support

can regain weight. For solid primitives, o_i is also the peak opacity shared by their Gaussians and the quantity used for pruning. The parsimony term is $\mathcal{L}_{\text{par}} = K^{-1} \sum_i \sqrt{o_i}$. Overlap is penalised through soft occupancy of sampled points, and shape regularisers act on the enabled deformation parameters. These terms operate on the predicted assembly; no reference occupancy or signed-distance field is supplied to the image-only fits.

We optimise with Adam (Kingma & Ba, 2014). The supervision cutoff gates mask, occupancy, parsimony, and several deformation terms together, whereas the overlap term has its own weight outside this gate. The gate is active while the iteration index is less than the configured cutoff. The reported coarse-loss ablation sets this cutoff to zero, removing all gated terms; it does not isolate the mask loss. Revision and no-edit continuation use the same cutoff and learning-rate policy. Experiment budgets are listed in Appendix B.5. The supervision cutoff is 30k, so the gated terms stay active throughout the initial fit; parsimony has weight 10^{-3}, the occupancy-entropy term has weight 0.02 from iteration 10k, and the overlap penalty uses the soft-occupancy form with its normalisation constant fixed to 1.95 and the opacity factor detached. Residual birth is enabled between iterations 5k and 20k with a population cap of $n + \max(3, \lfloor n/2 \rfloor)$ for an inventory of n parts; primitives whose occupancy falls below 0.1 are pruned at the end, and block scales are bounded below by 0.02. Deformation parameters use learning rate 10^{-3} with a 0.05 multiplier on the bend parameters; all other learning rates are the PartGS defaults. Continuation runs restart the iteration counter at zero and keep the structure frozen.

A.4 Correspondence, edits, and continuation

Correspondence is estimated from the current fitted geometry and does not use the initial name as a semantic ground-truth label. By default, retained primitives transfer their fitted geometry and appearance across the edit boundary. New instances and accepted shrunk instances use the compiled slot's initialisation and shape permissions. Continuous parameters remain trainable during continuation, including those of retained parts. The no-edit control follows the same transfer path with no changes to the inventory. The retained state includes pose, scale, shape parameters, occupancy, appearance coefficients, and Gaussian attachment parameters. Adam moments are reset, and the iteration counter restarts at zero in both the revision and no-edit arms.

Repeated correspondence queries are reconciled at the part-family level, so permuting otherwise equivalent leg instances is not treated as a disagreement. A missing-family count uses the minimum supported across queries. Disputed assignments are retained rather than treated as evidence for deletion. Primitive identifiers come from the fitted checkpoint, including born candidates, rather than assuming the original slot order is unchanged. The two-query configuration accepts a missing-family count only when both valid responses support it, using the smaller count. Invalid responses are retried up to three times per requested response; exhausting this budget is treated as a failed revision. The model, prompt, and validator remain fixed across retries. We accept the first valid response without inspecting subsequent fitting scores.

The prompt defines *shrunk* as a primitive that has collapsed to a sliver and covers almost none of its intended part. The VLM makes this judgement from the input images, fitted views, isolated primitives, and reported sizes; there is no additional numerical size or occupancy threshold. Reinitialisation requires a valid assignment to an existing slot. With repeated queries, the same primitive must be assigned to the same family and labelled shrunk by every response; a status disagreement preserves its fitted state. An accepted shrunk instance uses the slot's compiled initialisation instead of copying any learned state from that primitive. This resets translation, rotation, scale, deformation parameters, retention weight, appearance coefficients, and Gaussian attachment and scale/orientation parameters. The slot's deformation permissions are applied again. Ordinary *wrong size*, *wrong shape*, and *misplaced* judgements preserve the learned state; those errors are left to continuous fitting.

Deletion is disabled in this conservative configuration: disputed primitives are retained, and accepted additions are fitted alongside them. The main comparisons and revision ablation use this two-query agreement rule. Experiment-specific budgets are listed in Appendix B.5.

A.5 RECONSTRUCTION PROCEDURE

The procedure below describes one structural revision. The state of a primitive includes its stable identifier, geometric parameters, occupancy, appearance, Gaussian attachments, and deformation permissions.

1. Obtain a part description from the selected input views. Validate the description, construct the visual hull, and compile the initial assembly.

2. Fit the assembly for the initial budget. Within the configured birth window, propose primitives from residual coverage. Apply the configured occupancy penalty and pruning rule, then save the fitted checkpoint.

3. Render the whole assembly and individual primitives for VLM inspection. Associate semantic instances with current primitive identifiers and reconcile the correspondence queries under the selected edit policy.

4. Build the revised inventory. Copy retained primitive states from the checkpoint, and initialise added instances and accepted shrunk instances from their compiled slots. Naming a candidate alone does not reset its fitted geometry.

5. Create the continuation optimiser and fit for the continuation budget with birth and pruning disabled. Retained and new parts are both trainable. The transfer copies model parameters, not Adam moments.

In the full pipeline, the fitting budgets in Steps 2 and 5 sum to 30k iterations, and Step 3 uses two correspondence queries. The 30k+5k schedule is specific to the structural-revision ablation (Appendix B.5). For the ablation's no-edit control, replace Step 3 with an identity correspondence and apply no additions, deletions, or reinitialisations in Step 4. All other continuation settings, including the optimiser reset and iteration-counter policy, are shared. This control measures whether edits help beyond refitting the same state. The budgets and query counts for these steps follow Appendix B.5.

A.6 PROMPT AND VALIDATION INTERFACE

Initial reasoning receives anonymised object views and a shared JSON schema. The schema records a family name, count, arrangement, shape cues, an axial interval, and parent-relative relations. It permits dimensionless axial intervals but does not ask the VLM for metric coordinates or dimensions. Validation checks names, allowed vocabulary, positive counts, parent references, and interval bounds. In condensed form, the localisation prompt asks: "For each view, click once near the centre of every visible instance of the listed part families; skip occluded instances." Its per-view response has the following form:

```
{"part": "leg", "local_instance": 0,
 "click": [0.42, 0.61], "confidence": 0.9}
```

Each view is associated with its calibrated camera, as described in Appendix A.1. The revision prompt includes conditional instructions for residual candidates and explains that display colours are arbitrary primitive identifiers. Revision responses associate semantic instances with checkpoint primitive identifiers and list missing instances and proposed additions. Validation checks that referenced identifiers exist, counts are nonnegative, and added parts have complete initialisation records. Each per-primitive field must have one row per output slot, and pairwise relation fields must have one row and column per slot. Reindexing preserves retained rows; added rows use the compiler's initialisation and the field's default values where unspecified. Born primitives are included in the correspondence views, so they can receive a semantic assignment without being recreated. Validation failures are not interpreted as a no-edit decision.

B Evaluation Details

B.1 Data and camera protocol

The synthetic objects are built from hand-authored templates of recognisable assemblies. Parts use boxes, cylinders, spheres or ellipsoids, frusta, circular arc tubes, and container meshes. The template specifies each part's type and count, dimensions, orientation, and position. Rods and curved attachments can instead be specified by endpoints on neighbouring parts, with a small overlap at the joint. Dimensions, ellipsoid semi-axes, tube radii, arc sweeps, and repetition counts differ across templates; they are fixed within each object. Each template contributes one fixed assembly to the main set.

The generator samples approximately 10^5 surface points per object, allocates samples by part surface area with a minimum of 500 per part, and stores their part labels. Attachment-based templates are checked for connectivity using surface-sample distances before rendering. The generating parts define a reference decomposition; other semantic partitions may also be valid.

The rendered-object camera setup uses 100 views sampled on a Fibonacci sphere, elevations of approximately -20° to 72°, radii in $[1.3, 1.5]$, a 60° field of view, and 800×800 images. Fitting uses half-resolution images. All 100 views are used for fitting; no camera is withheld.

Main-comparison fits use fitting seed 0; the ablation seed counts are reported in Table 2. PartGS is run from its public repository at the author-recommended settings (20 blocks, block-to-scene ratio 0.15, overlap 0.1, parsimony 10^{-3}) with a 30k fitting budget and the same per-ray near/far bounds. Runs were executed on NVIDIA RTX 4090 and RTX 3090 GPUs. Ablations are conducted on the Synthetic dataset.

B.2 Part-recovery metrics

Each labelled reference surface sample is assigned to its nearest predicted primitive surface if its distance is at most 2% of the object diagonal. More distant samples remain unassigned. Distances are computed to triangles, not mesh vertices. For reference part g, let $d(g)$ be the most frequent assigned primitive and $s(g)$ the fraction of all samples of g assigned to it. A part is *covered* if $s(g) \geq \frac{1}{2}$ and its median distance to the nearest surface in the predicted assembly is within the threshold. It is *dedicated* if it is covered and $d(g)$ is not dominant for any other covered reference part.

A predicted primitive is *merged* if it dominates multiple covered reference parts. It is *starved* if it receives fewer than 0.5% of all reference samples under the distance-gated assignment. Parts without a majority assignment may be fragmented or insufficiently covered; this count is therefore not treated as a pure splitting metric. Dedicated recovery measures dominant one-to-one correspondence and does not guarantee complete surfaces or connected joints. A semantic instance represented by several primitives is scored without merging those primitives. It can still count as dedicated if one primitive receives at least half of its samples and does not dominate another covered part; otherwise, even a geometrically accurate multi-piece reconstruction does not receive dedicated credit. The score therefore measures a compact dominant correspondence, not arbitrary many-to-one semantic recovery.

B.3 Geometric accuracy

Let P and Q be surface samples from the predicted mesh and reference geometry, and let D be the reference bounding-box diagonal. We use the symmetric mean of unsquared Euclidean distances,

$$\mathrm{CD}(P, Q) = \frac{1}{2D} \left(\frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p - q\|_2 + \frac{1}{|Q|} \sum_{q \in Q} \mathrm{dist}(q, \mathcal{S}_P) \right), \tag{13}$$

where $\mathcal{S}_P$ is the predicted triangle surface, and display $100\,\mathrm{CD}$ in the tables. The prediction and reference share the calibrated coordinate frame; no per-result ICP or scale alignment is applied. The reference set is the 10^5 surface samples of each object; the prediction set is 10^5 area-uniform samples of the output mesh drawn with evaluation seed 0. Reference-to-prediction distances are exact point-

to-triangle distances to the output mesh, and prediction-to-reference distances are nearest-neighbour distances to the reference samples. The same metric is used for all methods.

B.4 Aggregation and comparison protocol

For each seed, dedicated recovery and coverage are computed by summing the corresponding part counts across scenes and dividing by the total reference part count. Compactness is $\sum_s K_s / \sum_s G_s$, where K_s and G_s are the predicted and reference counts for scene s. CD and mean primitive count are averaged over scenes. Repeated-seed component ablations report the mean and sample standard deviation of these dataset-level quantities. The structural-revision comparison in the lower half of Table 2 reports only the mean across seeds. A single-seed result has no uncertainty estimate.

Within each comparison group, paired ablations use the same scenes, initial descriptions, boxes, and seed identifiers. Revision and no-edit continuation additionally share the exact pre-revision checkpoints and transfer procedure. The paired change is computed before aggregation. All objects in each evaluation set contribute to recovery, including poor or empty reconstructions; CD is undefined for an empty output mesh. Initial VLM descriptions are shared across fitting seeds, so these comparisons measure fitting variability conditional on the initial structure. Fitting budgets and query counts are listed in Appendix B.5.

B.5 Experiment settings

Table 3 collects the fitting and revision settings. Query counts refer to valid correspondence responses and exclude initial structural reasoning, localisation, and validation retries. Deletion is disabled during revision.

Table 3: Fitting budgets and structural-revision settings. All main-comparison datasets use the full reasoning–fitting loop within a total budget of 30k fitting iterations, including post-revision refinement. Queries count valid correspondence responses only; initial reasoning, localisation, and validation retries are excluded.

Experiment	Fitting budget	Revision queries	Seed IDs
ReasonFit full pipeline (all datasets)	30k total	2	0
Initial-fit ablations	30k	—	0, 1
Revision / no-edit ablation	30k + 5k	2 / 0	0, 1
Qualitative ablation (Fig. 5, bottom)	+5k per arm	2 / 0	0
PartGS	30k	—	0
DBW	1,800 epochs	—	227391

The component ablations use airplane_toy, office_chair, pinwheel, potted_plant, and scissors; their initial descriptions and boxes are shared across variants. The revision comparison instead uses archived 30k fits of ball_table, crab, and mug, so its starting scores are not those of the component-ablation baseline. Both comparisons use fitting seeds 0 and 1. For each revision pair, no-edit and edited continuations use the same checkpoint, state-transfer procedure, and 5k fitting schedule; revision uses two correspondence queries. The lower half of Table 2 reports the mean across these continuation seeds. Longitudinal-profile permissions are disabled in the component-ablation inputs. The *w/o shape permissions and initialisers* arm strips the profile, bend, and taper permission/initialisation fields and disables the bend and taper switches; other SF parameters remain optimisable. The *w/o coarse losses* arm sets the coarse-stage cutoff to zero, disabling all losses governed by that cutoff rather than only the mask term.

The main table reports the final post-revision reconstructions on the evaluation sets in Section 4.1, with fitting seed 0 per object. DBW uses its original 1,800-epoch recipe, 20 initial blocks, seed 227391, and parsimony weight 0.001.

C Failure Modes and Limitations

The failure modes below concern both the structure of the assembly and the geometry of individual primitives. They explain why a low rendering loss or a compact inventory is not sufficient evidence of successful part recovery.

Geometry and visibility. Silhouettes constrain the visual hull but cannot determine concavities that leave the observed outlines unchanged (Laurentini, 1994; 1995). Photo-consistency can supply additional constraints (Seitz & Dyer, 1999; Kutulakos & Seitz, 2000; Cremers & Kolev, 2010), although geometry–appearance ambiguity remains (Zhang et al., 2020).

Optimisation and structural revision. Small parameter derivatives, competing primitives, and early sampling differences can lead to different fitted assemblies. Short, thick bends can be difficult to initialise and measure. VLM correspondence can also vary across calls, and agreement does not establish correctness. Residual birth may fragment a part or miss one already covered by a neighbour. Finally, fixing the inventory during continuation prevents birth and pruning from undoing an erroneous accepted edit.