ORGANIZE PRIMITIVES INTO SEMANTIC PARTS: REINFORCEMENT REASONING FOR 3D SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Primitive-based 3D segmentation offers a compact and explicit alternative to dense surface prediction, naturally supporting structural abstraction and boundary localization. However, geometric decomposition alone does not determine how primitives should be organized into semantic parts: a single part may span multiple primitives, while geometrically similar or touching primitives may belong to different parts. We therefore introduce **RePart** (**Re**inforcement **Part** Reasoning), which formulates primitive-to-part organization as a finite-horizon Markov decision process and learns semantic organization through trajectory-level reinforcement reasoning. RePart constructs a *Composable Primitive Workspace* from fine-grained superquadrics and applies a merge-and-stop policy whose decisions are optimized by their downstream effects on the resulting partition rather than local primitive compatibility. The inferred part identities are then mapped back to the original mesh through *Boundary-Aware Surface Labeling*, preserving accurate surface boundaries beyond the primitive approximation. On PartNet, RePart achieves the strongest results across all four aggregate partition metrics; on 3DCoMPaT++, it obtains the highest RI and SC without target-dataset fine-tuning. These results demonstrate that reinforcement reasoning provides an effective mechanism for organizing geometric primitives into semantic parts while retaining dense segmentation accuracy. Code is available at https://anonymous.4open.science/r/RePart-F01E.

1 INTRODUCTION

3D part segmentation has advanced rapidly through complementary paradigms, including geometric primitive decomposition (Paschalidou et al., 2021; Wu et al., 2022; Monnier et al., 2023; Liu et al., 2023a;c), learned 3D representations (Ma et al., 2025b; Liu et al., 2025a; Hadgi et al., 2026), and semantic transfer from vision–language and multi-view foundation models (Liu et al., 2023b; Umam et al., 2024; Yang et al., 2024; Zhu et al., 2026). These advances have improved the flexibility and generalization of 3D part understanding. Meanwhile, LLM-driven 3D agents and interactive systems (Qi et al., 2024) are broadening the role of segmentation: predicted parts increasingly serve not only as dense outputs, but also as entities for reasoning, selection, editing (Ganeshan et al., 2024), and manipulation (Geng et al., 2023). This shift demands representations that expose compact object structure for high-level reasoning while remaining grounded in the original geometry. Geometric primitives are well suited to these requirements, replacing dense surface elements with spatially coherent units that encode shape, scale, pose, orientation, and volumetric extent.

Primitive-based 3D decomposition has evolved from learned and deformable shape abstractions (Paschalidou et al., 2021; Monnier et al., 2023; Liu et al., 2023a), to explicit primitive fitting (Wu et al., 2022; Liu et al., 2023c), and structure-aware abstraction (Wang et al., 2025; Fedele et al., 2025). These methods naturally support *Structural Abstraction*: dense geometry is represented by compact explicit components at a tractable granularity (Paschalidou et al., 2019). They also support *Boundary Localization*, since each primitive retains spatial support that maps back to the surface (Li et al., 2019). Yet, as parts increasingly serve as meaningful entities rather than dense outputs, a third requirement becomes critical: *Semantic Organization*. Primitive decomposition is mainly governed by geometric fit and structural regularity, which do not determine how primitives compose into semantic parts. A single part may span multiple primitives, while nearby primitives

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

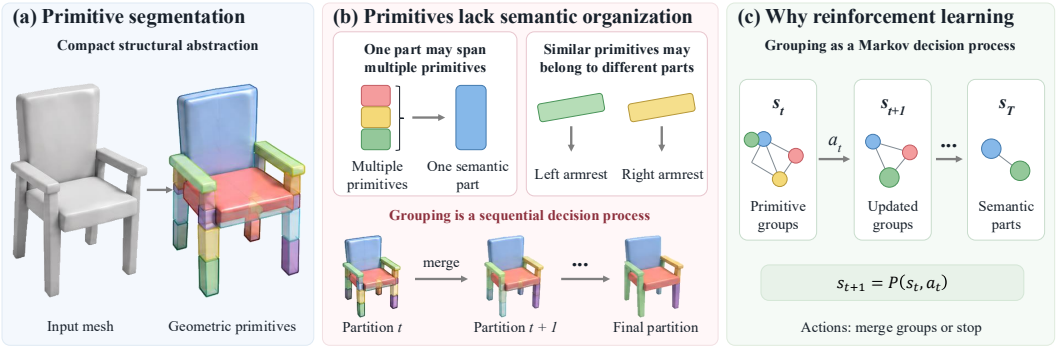

Figure 1: Primitive segmentation exposes compact structural abstractions, while organizing primitives into semantic parts requires sequential decisions, making reinforcement learning a natural fit.

with similar geometry may belong to different parts. Thus, the key unresolved problem is how to organize primitives into semantic parts (Luo et al., 2020).

Semantic Organization cannot be reduced to independent or locally optimal grouping decisions. Pairwise classification assumes that mergeability can be judged in isolation, while greedy agglomeration optimizes only immediate gains and cannot recover from early irreversible mistakes. Supervised sequence learning is also ill-posed, because ground-truth segmentation specifies the desired final partition but not a unique merge trajectory. Starting from an over-segmented primitive partition, each merge changes group geometry, neighborhood relations, and the candidate actions available thereafter, so the value of an early decision is ultimately determined by the partition it enables. We therefore formulate semantic organization as a finite-horizon Markov decision process (Puterman, 1990). This formulation makes reinforcement learning (RL) well matched to the problem: it assigns credit from the quality of the resulting partition back through the sequence of merge-and-stop decisions (Sutton et al., 1998), rather than requiring local pair labels or a prescribed reasoning path.

Building on this formulation, we propose **RePart** (**Re**inforcement **Part** Reasoning), which places *Semantic Organization* at the center of primitive-based 3D segmentation and adapts both the reasoning representation and dense output around it. Composable Primitive Workspace realizes *Structural Abstraction* by constructing a fine-grained SQ workspace with Marching-Primitives, then fixing the primitive geometry and surface correspondence to provide stable, composable units for reasoning without committing to the final part partition. Reinforcement-Learning Part Reasoning realizes *Semantic Organization* by casting SQ grouping as a merge-and-stop decision process and optimizing the policy with PPO, allowing each merge to be evaluated by its downstream effect on the evolving and final partition rather than local compatibility alone. Boundary-Aware Surface Labeling realizes *Boundary Localization* by transferring the inferred part identities to curvature-consistent mesh regions using grouped-SQ surface proximity and volumetric evidence, so that the final segmentation follows the original surface geometry instead of the approximate primitive boundaries.

The contributions of this paper are summarized as follows:

- We formulate primitive-to-part organization as a finite-horizon Markov decision process, enabling semantic grouping without requiring predefined merge sequences.
- We introduce **Reinforcement-Learning Part Reasoning** to optimize merge-and-stop decisions through trajectory-level credit assignment rather than local primitive compatibility.
- We couple this reasoning with a **Composable Primitive Workspace** and **Boundary-Aware Surface Labeling**, bridging compact primitive reasoning and precise mesh segmentation.

2 Related Work

3D Part Segmentation. Recent 3D part segmentation increasingly targets low-shot, class-agnostic, and open-world settings. Vision–language and multi-view foundation models transfer semantic cues to 3D (Liu et al., 2023b; Umam et al., 2024; Yang et al., 2024; Tang et al., 2024), while large-scale 3D supervision and learned representations improve generalization and controllable granularity (Ma

et al., 2025b; Liu et al., 2025a; Zhu et al., 2026; Hadgi et al., 2026; Su et al., 2026; Deng et al., 2026). These methods primarily predict dense part assignments. RePart instead reasons over compact geometric primitives, separating semantic organization from surface labeling.

Primitive-Based 3D Abstraction. Primitive abstraction represents 3D geometry with compact, explicit components, ranging from classical superquadric recovery (Leonardis et al., 1997) to learned, probabilistic, deformable, and differentiable formulations (Paschalidou et al., 2021; Wu et al., 2022; Liu et al., 2023a; Monnier et al., 2023). Later work models hierarchical structure or improves geometry-aware decomposition (Niu et al., 2022; Ruan et al., 2025; Liu et al., 2025b), while MPS, Light-SQ, and SUPERDEC further develop SQ-based local abstraction and structural organization (Liu et al., 2023c; Wang et al., 2025; Fedele et al., 2025). GaussianGPT uses explicit Gaussian primitives as discrete tokens for scene generation (von Lützow et al., 2026). RePart instead treats primitives as composable reasoning units and focuses on organizing them into semantic parts.

Sequential Primitive Organization. Classical mesh segmentation and hierarchical decomposition progressively merge geometric regions or components (Attene et al., 2006; Niu et al., 2022; Du et al., 2024; Ye et al., 2025). Learning to Group formulates bottom-up grouping as a contextual bandit with local mergeability rewards (Luo et al., 2020; Brehmer et al., 2020), while reinforcement learning has also been explored for hierarchical clustering and 3D reconstruction (Brehmer et al., 2020; Li et al., 2023). RePart differs by formulating primitive-to-part organization as a finite-horizon merge-and-stop Markov decision process, where irreversible decisions are optimized through their downstream effect on the resulting partition.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW

Given an input triangular mesh $\mathcal{O} = (\mathcal{V}, \mathcal{F})$, our goal is to predict a class-agnostic partition

$$\hat{Y}^{\mathcal{F}} = \{\hat{y}_f^{\mathcal{F}}\}_{f \in \mathcal{F}}$$

of its faces into an unknown number of parts; its numeric labels are defined only up to permutation. During training, we additionally sample a point set $X = \{\boldsymbol{x}_i\}_{i=1}^N$ from the mesh surface and associate each point with a ground-truth part label $Y = \{y_i\}_{i=1}^N$.

Primitive-based part segmentation exposes three coupled gaps across representation, organization, and boundary localization: dense mesh elements preserve surface detail but are unnecessarily fine-grained for global structural reasoning; fitted primitives provide compact geometric units but do not determine part identities; and primitive groups do not directly localize accurate mesh boundaries. RePart addresses these gaps through a *Composable Primitive Workspace* for structural abstraction, *Reinforcement-Learning Part Reasoning* for semantic organization, and *Boundary-Aware Surface Labeling* for boundary localization:

$$\mathcal{O} \xrightarrow{\text{Composable Primitive Workspace}} (\mathcal{W}, \Pi_0) \xrightarrow{\text{RL Part Reasoning}} \Pi_T \xrightarrow{\text{Boundary-Aware Surface Labeling}} \hat{Y}^{\mathcal{F}},$$

where $\mathcal{W}$ denotes the fixed set of superquadrics and Π_t its partition at step t (Sec. 3.2). These components respectively abstract dense geometry into composable structural units, organize them into part-level identities, and localize the inferred identities on the original surface.

3.2 COMPOSABLE PRIMITIVE WORKSPACE

Structural abstraction requires compact units for group-level reasoning while preserving candidate part boundaries for subsequent organization. We therefore construct a fixed, intentionally over-segmented SQ basis whose primitives form a *Composable Primitive Workspace*. SQs provide compact shape, pose, and volumetric support; over-segmentation allows one part to be composed from multiple primitives and preserves candidate boundaries for merge-only organization. Fixing SQ geometry and point correspondence makes alternative part organizations differ only in how the same primitives are grouped, turning structural inference into partition reasoning over a stable geometric basis. To construct this workspace, we normalize the mesh, compute its signed distance field, and

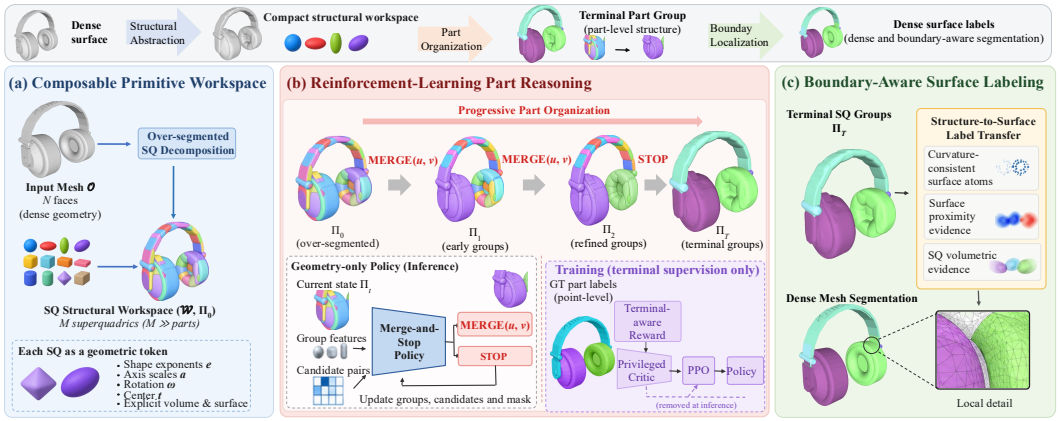

Figure 2: **Overview of RePart.** RePart addresses primitive-based 3D segmentation through three coupled stages that bridge dense geometry, part structure, and surface boundaries. (a) **Composable Primitive Workspace** abstracts the dense surface into compact, composable geometric units for structural reasoning. (b) **Reinforcement-Learning Part Reasoning** organizes these primitives into coherent part-level structure. (c) **Boundary-Aware Surface Labeling** localizes the inferred part identities back onto the original surface to recover dense, boundary-aware segmentation.

initialize an intentionally over-segmented set with Marching-Primitives (Liu et al., 2023c):

$$\mathcal{W} = \{p_1, \ldots, p_M\}, \tag{1}$$

where p_j denotes the j-th fitted SQ primitive.

We next fix the correspondence between surface points and SQ primitives so that different part-organization hypotheses are evaluated over the same geometric basis. Let $d_{ij} = |d_{\mathrm{SQ}}(\boldsymbol{x}_i; p_j)|$, using the standard radial signed-distance approximation defined in Appendix F. We compute soft support and fix each point's primitive assignment as

$$\alpha_{ij} = \frac{\exp(-d_{ij}/\beta)}{\sum_{\ell=1}^{M} \exp(-d_{i\ell}/\beta)}, \qquad z_i = \arg\max_j \alpha_{ij}, \qquad \beta = \mathrm{median}_{i,\ell}\, d_{i\ell}. \tag{2}$$

Both SQ geometry and the point-to-SQ assignments z_i remain fixed throughout reasoning. Starting from $\boldsymbol{g}_0 = (1, \ldots, M)$, the grouping vector $\boldsymbol{g}_t = (g_{t,1}, \ldots, g_{t,M})$ induces

$$\Pi_t = \big\{\{\, j : g_{t,j} = u \,\} : u \in \mathcal{U}_t\big\}, \qquad \mathcal{U}_t = \{g_{t,j}\}_{j=1}^{M}, \tag{3}$$

where $\mathcal{U}_t$ denotes the active group labels and Π_t the induced SQ partition. $\mathcal{W}$ provides fixed composable units, while Π_t captures their evolving part organization; with fixed z_i, all partitions are evaluated under the same point-to-SQ correspondence via $\hat{y}_{t,i} = g_{t,z_i}$.

3.3 Reinforcement-Learning Part Reasoning

Merge-and-Stop Markov Decision Process. The composable primitive workspace provides compact geometric units, but it does not determine their part-level organization. Because a semantic part may span multiple SQs, the model must determine which current groups should share a part identity and when further consolidation should stop. These decisions are coupled: each irreversible merge changes group geometry and the candidates available later, while supervision specifies the desired terminal partition rather than a unique merge sequence. We therefore formulate part reasoning as a merge-and-stop markov decision process and optimize its trajectories with PPO, allowing decisions to be evaluated through their downstream effects on the resulting organization.

We model part reasoning as a finite-horizon markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, whose actions progressively reorganize the current partition until the policy selects `stop` or reaches the maximum horizon.

State ($\mathcal{S}$). An episode starts from $(\mathcal{W}, \Pi_0)$, with underlying state $\tilde{s}_t = \{X, \mathcal{W}, \boldsymbol{g}_t\}$, where $\boldsymbol{g}_t$ induces the current partition Π_t as defined in Eq. equation 3. The actor observes

$$o_t = \{X_s, \mathcal{W}, \Phi_t^{\mathrm{prim}}, \Phi_t^{\mathrm{global}}, \Phi_t^{\mathrm{pair}}, \boldsymbol{\eta}_t\}, \tag{4}$$

where X_s is sampled without replacement from X at each encoding and $\boldsymbol{\eta}_t$ is a binary action mask, which retains `stop` and valid merge candidates while suppressing unavailable actions. Rather than numeric group IDs, the actor uses geometry-derived descriptors: Φ_t^{prim} describes each primitive and its current group, Φ_t^{global} summarizes grouping progress, and Φ_t^{pair} encodes candidate compatibility. The SQ parameters and point assignments remain fixed, while grouping-dependent descriptors and the candidate mask are recomputed after each merge.

Action ($\mathcal{A}$). We form a dynamic candidate set $\mathcal{C}_t$ by connecting every active group to its κ nearest groups using axis-aligned bounding-box separation, with zero distance for overlapping boxes, and then symmetrizing the relations. The available actions are

$$\mathcal{A}(\tilde{s}_t) = \{\texttt{stop}\} \cup \{\texttt{merge}(u, v) : (u, v) \in \mathcal{C}_t\}. \tag{5}$$

Transition ($\mathcal{P}$). For $a_t = \texttt{merge}(u, v)$, the transition deterministically relabels v as u:

$$g_{t+1,j} = \begin{cases} u, & g_{t,j} = v, \\ g_{t,j}, & \text{otherwise.} \end{cases} \tag{6}$$

Thus, all SQs in group v are reassigned to u, while all other assignments remain unchanged. Group-dependent descriptors, candidates, and the action mask are then recomputed. Resampling X_s changes only the actor observation, not the underlying state. Each merge reduces the group count by one. For $a_t = \texttt{stop}$, we keep $\boldsymbol{g}_{t+1} = \boldsymbol{g}_t$ and terminate the episode; the episode also terminates at the maximum horizon.

Reward ($\mathcal{R}$). Training annotations specify the desired part organization but not a unique sequence of merges. We therefore supervise partition progress rather than individual actions using permutation-invariant overlap and group-count error. For ground-truth part $\mathcal{Y}_k = \{i : y_i = k\}$ and predicted group $\widehat{\mathcal{Y}}_{t,u} = \{i : \hat{y}_{t,i} = u\}$, we define

$$m_t = \frac{1}{K} \sum_{k=1}^{K} \max_{u \in \mathcal{U}_t} \text{IoU}(\mathcal{Y}_k, \widehat{\mathcal{Y}}_{t,u}), E_t = \frac{|G_t - K|}{\max(K, 1)}, \tag{7}$$

where K is the number of ground-truth parts and $G_t = |\Pi_t|$. The overlap term follows permutation-invariant class-agnostic part evaluation (Yang et al., 2024; Liu et al., 2025a).

The transition reward $\mathcal{R}(\tilde{s}_t, a_t, \tilde{s}_{t+1}) = r_t$ is

$$r_t = \underbrace{m_{t+1} - m_t}_{\text{overlap gain}} + \lambda_g \underbrace{(E_t - E_{t+1})}_{\text{count correction}} + \mathbb{I}[a_t = \texttt{stop}] \underbrace{(b_{\text{stop}} - \lambda_s E_{t+1})}_{\text{stop quality}}. \tag{8}$$

The three terms respectively reward a better part organization, movement toward the correct group count, and a well-timed stop. Here, $\lambda_g, \lambda_s \geq 0$ are weighting coefficients and b_{stop} is the base reward for an explicit stop. Labels Y and K provide training rewards but are unavailable to the deployed actor.

Geometry-Only Grouping Policy. Although part annotations supervise training, the deployed actor must infer part organization from geometry alone. We therefore encode evidence at three complementary levels: surface points provide object-level context, SQ descriptors characterize the current groups, and pair descriptors describe the geometry of candidate merges. Centered, radius-normalized points are encoded by a shared MLP with max–mean pooling; separate MLPs encode SQ and primitive descriptors, which are aggregated within each group by masked mean pooling.

As groups evolve, their descriptors aggregate the geometry of all constituent SQs, while a fixed catalog representative maintains consistent action indexing. The merge and stop heads combine the shared geometric context with action-specific descriptors and a geometry-predicted estimate $\hat{\nu}_t$ of the target group-count ratio. The latter is supervised during training but requires no labels at inference. Given catalog logits $h_\theta(o_t)$, the masked policy is

$$\pi_\theta(a_t \mid o_t) = \text{softmax}(h_\theta(o_t) + \log \boldsymbol{\eta}_t). \tag{9}$$

The logarithmic mask assigns $-\infty$ to invalid actions, so the softmax normalizes over $\mathcal{A}(\tilde{s}_t)$ only.

PPO Optimization. We optimize the merge-and-stop policy with PPO (Schulman et al., 2017) using the trajectory rewards above, so each action is credited according to its downstream effect on subsequent partitions and the final stopping decision:

$$\mathcal{L}_{\text{PPO}} = -\mathbb{E}_t \left[\min \left(\rho_t(\theta)\hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{10}$$

where ρ_t is the policy ratio, $\hat{A}_t$ the estimated advantage, and ϵ the clipping radius.

The critic uses stronger training-time supervision than the actor, combining a detached actor representation with label-derived partition statistics to estimate state value and terminal quality. Detachment prevents these privileged signals from entering the actor representation. The full training objective further includes value, entropy, terminal-prediction, and granularity-prediction losses. The resulting terminal partition Π_T encodes the inferred part-level organization and is subsequently passed to Boundary-Aware Surface Labeling in Sec. 3.4.

3.4 Boundary-Aware Surface Labeling

The terminal partition Π_T determines which SQs share the same part identity, but their primitive boundaries only approximate the original surface and therefore do not directly localize precise mesh boundaries. We perform boundary localization on the mesh itself through *Boundary-Aware Surface Labeling*. Curvature-consistent atoms constrain where labels may change, while grouped SQ surface and volumetric evidence determine which part identity each atom receives.

Each terminal group combines the geometric support of its constituent SQs, allowing one part to span multiple primitives. For $u \in \mathcal{U}_T$, we define the grouped surface samples as $\mathcal{B}_u = \bigcup_{j:g_{T,j}=u} \mathcal{B}(p_j)$, where $\mathcal{B}(p_j)$ denotes samples from the analytic surface of SQ p_j. To localize boundaries without fragmenting coherent surface regions through independent face-wise assignments, we cut high-dihedral adjacencies to form curvature-consistent atoms $\{\Omega_q\}_{q=1}^{Q}$, each of which receives a single group label. For face centroid $\boldsymbol{c}_f$, we compute

$$\delta_{fu} = \min_{\boldsymbol{b}\in\mathcal{B}_u} \|\boldsymbol{c}_f - \boldsymbol{b}\|_2, \qquad \tilde{u}_f = \arg\min_u \delta_{fu}, \qquad \delta_f = \delta_{f\tilde{u}_f}. \tag{11}$$

Each atom then aggregates valid proximity votes:

$$V_{qu} = \sum_{f\in\Omega_q} \mathbb{I}[\tilde{u}_f = u]\mathbb{I}[\delta_f \le \tau_d], \qquad \hat{u}_q = \arg\max_u V_{qu}. \tag{12}$$

We accept this assignment when at least one valid vote exists and at most a fraction τ_o of the atom's faces exceed τ_d. Surface proximity alone can leave regions unresolved that lie within a group's volumetric support but remain far from its fitted surface. For rejected atoms, we therefore evaluate signed radial support using Eq. equation 15:

$$D_u(\boldsymbol{c}_f) = \min_{j:g_{T,j}=u} d_{\text{SQ}}(\boldsymbol{c}_f; p_j). \tag{13}$$

A rejected atom is assigned to group u when at least half of its face centroids satisfy $D_u(\boldsymbol{c}_f) \le \tau_c$; otherwise, we use the group receiving the most unconstrained nearest-surface votes. Each face then inherits its atom label, $\hat{y}_f^{\mathcal{F}} = \hat{u}_q$ for $f \in \Omega_q$. This boundary-aware labeling preserves the inferred part-level organization while allowing final label transitions to follow the original mesh geometry rather than primitive interfaces.

4 Experiments

4.1 Experimental Setup

Datasets and splits. We train RePart on PartNet (Mo et al., 2019), grouping relevant categories into furniture, containers, and daily-use objects and sampling 300 shapes per group. We test on 200 total corresponding-category shapes from PartNet and 200 total from 3DCoMPaT++ (Slim et al., 2025). The latter is used only for testing, without target-dataset fine-tuning.

Training. We first optimize one category-specialized policy per group over 200 epochs. The resulting parameters initialize a joint fine-tuning stage on the union of the three training subsets for 100

Table 1: Category-group mIoU on PartNet and 3DCoMPaT++. RePart uses joint fine-tuning; **bold** and underlined values denote the best and second-best results.

	PartNet			3DCoMPaT++		
Method	Furn.	Cont.	Daily	Furn.	Cont.	Daily
MPS	0.1591	0.5313	0.2169	0.3193	0.1144	0.2008
PartField	0.4098	0.6418	0.4969	0.2438	0.4127	0.4018
P3-SAM	0.3243	0.5663	0.5411	**0.4147**	0.5110	**0.6003**
SAMPart3D	0.2520	0.5111	0.4899	0.3313	0.4544	0.4720
PriMAny	0.2255	0.2067	0.3032	0.2419	0.1964	0.3236
RePart (Ours)	**0.4247**	**0.6456**	**0.5432**	0.3703	**0.5149**	0.4607

epochs. All experiments are run on a single NVIDIA RTX 3090 GPU. Unless otherwise specified, RePart denotes this jointly fine-tuned model. Appendix Figure A.1 reports the training rewards; to prevent reward-scale differences across category groups from joint training, we normalize advantages separately within each category, average its loss, and then macro-average the category losses.

Baselines and metrics. We compare with Marching-Primitives (MPS) (Liu et al., 2023c), PartField (Liu et al., 2025a), P3-SAM (Ma et al., 2025a), SAMPart3D (Yang et al., 2024), and PrimitiveAnything (PriMAny) (Ye et al., 2025). We report mean intersection-over-union (mIoU) (Yang et al., 2024; Liu et al., 2025a) to measure overlap between matched predicted and ground-truth parts; Rand Index (RI) (Rand, 1971) to measure whether pairs of faces are consistently assigned to the same or different parts; Variation of Information (VoI) (Meilă, 2007) to quantify the information difference between two partitions, including splitting and merging errors; and Segmentation Covering (SC) (Arbelaez et al., 2010) to measure how well predicted segments cover ground-truth regions. Following the PartField protocol, we report mIoU across its candidate granularities, with ground-truth labels used only for evaluation-time granularity selection.

4.2 Main Results

Quantitative Results. Tables 1 and 2 show that RePart shows consistently strong *in-domain performance* on PartNet across both category-level mIoU and partition-level metrics. RePart achieves the highest mIoU across all three category groups, reaching 0.4247 on furniture, 0.6456 on containers, and 0.5432 on daily-use objects. It also obtains the strongest aggregate results across all four partition metrics, comparing with the strongest competing results, RePart improves mIoU from 0.5249 to 0.5460 and SC from 0.6502 to 0.6891, while reducing VoI from 1.0213 to 0.8471. These results establish that the complete pipeline produces accurate and structurally coherent partitions across complementary evaluation criteria.

On 3DCoMPaT++, RePart demonstrates cross-dataset transfer without fine-tuning, achieving the highest container mIoU (0.5149), while P3-SAM performs better on furniture and daily-use objects. Across all test shapes, RePart ranks first in RI (0.7575) and SC (0.5790), and second in both mIoU and VoI. Notably, RePart does not lead in transferred mIoU, but retains the strongest RI and SC. The transferred model therefore retains strong pairwise and region-level partition consistency despite the dataset shift. On six PartNet categories excluded from training, RePart further achieves the best VoI and SC and the second-best average mIoU; full results are provided in D.

Qualitative Results. Figure 3 compares segmentation results on five PartNet shapes and five 3DCoMPaT++ shapes. On PartNet, RePart produces coherent part partitions while reducing the strong fragmentation visible in the initial MPS decomposition. Compared with the other baselines, it also preserves localized structures such as thin supports, attachments, and articulated components in the shown examples. On 3DCoMPaT++, RePart retains similarly coherent part organization without target-dataset fine-tuning, consistent with the cross-dataset results in Table 2.

4.3 Ablation Studies

Category specialization and joint fine-tuning. We first compare the three category-specialized policies with the jointly fine-tuned policy. Table 3 reports category-level mIoU. Joint fine-tuning

Table 2: Average partition quality on PartNet / 3DCoMPaT++.

	PartNet				3DCoMPaT++			
Method	mIoU↑	RI↑	VoI↓	SC↑	mIoU↑	RI↑	VoI↓	SC↑
MPS	0.3169	0.5266	1.0213	0.5267	0.2125	0.4371	**1.2043**	0.4371
PartField	0.5249	0.7877	1.0810	0.6502	0.3483	0.6914	1.2998	0.5302
P3-SAM	0.4858	0.6966	1.2238	0.5833	**0.5003**	0.7455	1.3783	0.5606
SAMPart3D	0.4229	0.6999	1.1996	0.5862	0.4144	0.7089	1.5508	0.5304
PriMAny	0.2440	0.5231	2.6506	0.2476	0.2477	0.5932	2.4509	0.2796
RePart (Ours)	**0.5460**	**0.7965**	**0.8471**	**0.6891**	0.4476	**0.7575**	1.2838	**0.5790**

Table 3: Category-level mIoU of category-specific and jointly fine-tuned policies.

	PartNet				3DCoMPaT++			
Policy	Furn.	Cont.	Daily	Avg.	Furn.	Cont.	Daily	Avg.
Category-specific	**0.4280**	**0.6515**	0.5075	0.5375	0.3699	0.4517	**0.4646**	0.4255
Staged joint	0.4247	0.6456	**0.5432**	**0.5460**	**0.3703**	**0.5149**	0.4607	**0.4476**

Table 4: Optimization-objective ablations on furniture models. Learned variants use the same SQ state/action representation and actor parameterization.

	PartNet furniture				3DCoMPaT++ furniture			
Policy	mIoU↑	RI↑	VoI↓	SC↑	mIoU↑	RI↑	VoI↓	SC↑
Greedy-BC	0.3757	0.7301	1.4123	0.5284	0.3442	0.7494	1.9009	0.4226
Contextual bandit	0.2591	0.4613	1.5346	0.4614	0.2144	0.3502	1.7428	0.3002
Pairwise classifier	0.3275	0.6362	1.3656	0.4971	0.2465	0.5499	1.7450	0.3700
Myopic-PPO	0.2579	0.4982	1.3813	0.4420	0.1678	0.3354	**1.6703**	0.2980
RePart-S	**0.4280**	**0.7804**	**1.2997**	**0.5751**	**0.3699**	**0.7846**	1.8601	**0.4452**

improves the average from 0.5375 to 0.5460 on PartNet and from 0.4255 to 0.4476 on 3DCoMPaT++. Its gains on PartNet daily-use objects and 3DCoMPaT++ furniture and containers indicate that sharing structural experience across category groups improves transfer.

Reinforcement-Learning Part Reasoning. We compare RePart-S, trained on 300 PartNet furniture shapes, with agglomerative policies sharing the same actor architecture, geometry-only SQ state, merge-and-stop action space, action mask, training data, and environment. Following Learning to Group (Luo et al., 2020), we adapt a contextual bandit and pairwise merge classifier. The former optimizes immediate transition rewards, while the latter learns from counterfactual one-step rewards over legal actions. Myopic-PPO sets $\gamma = 0$ and $\lambda_{\mathrm{GAE}} = 0$, removing temporal bootstrapping and future rewards, while Greedy-BC uses locally optimal merge trajectories as supervision. These controls isolate the effect of optimizing multi-step returns.

Table 4 shows that the contextual bandit and Myopic-PPO struggle to learn grouping under this environment, indicating that immediate rewards alone provide insufficient guidance for the irreversible merge sequence. The pairwise classifier and Greedy-BC make stronger local decisions, but both remain below full PPO. These results support optimizing multi-step returns for SQ grouping rather than treating merges as independent or locally optimal decisions.

Boundary-Aware Surface Labeling. We isolate Boundary-Aware Surface Labeling by fixing the terminal SQ groups and varying only their transfer to mesh faces. Appendix Table C.8 shows that curvature-consistent atoms substantially improve mIoU over direct facewise nearest-SQ assignment across all three PartNet groups, while SQ volumetric evidence further resolves regions left uncertain by surface proximity and reduces fallback usage. These results support separating *Semantic Organization* from *Boundary Localization*.

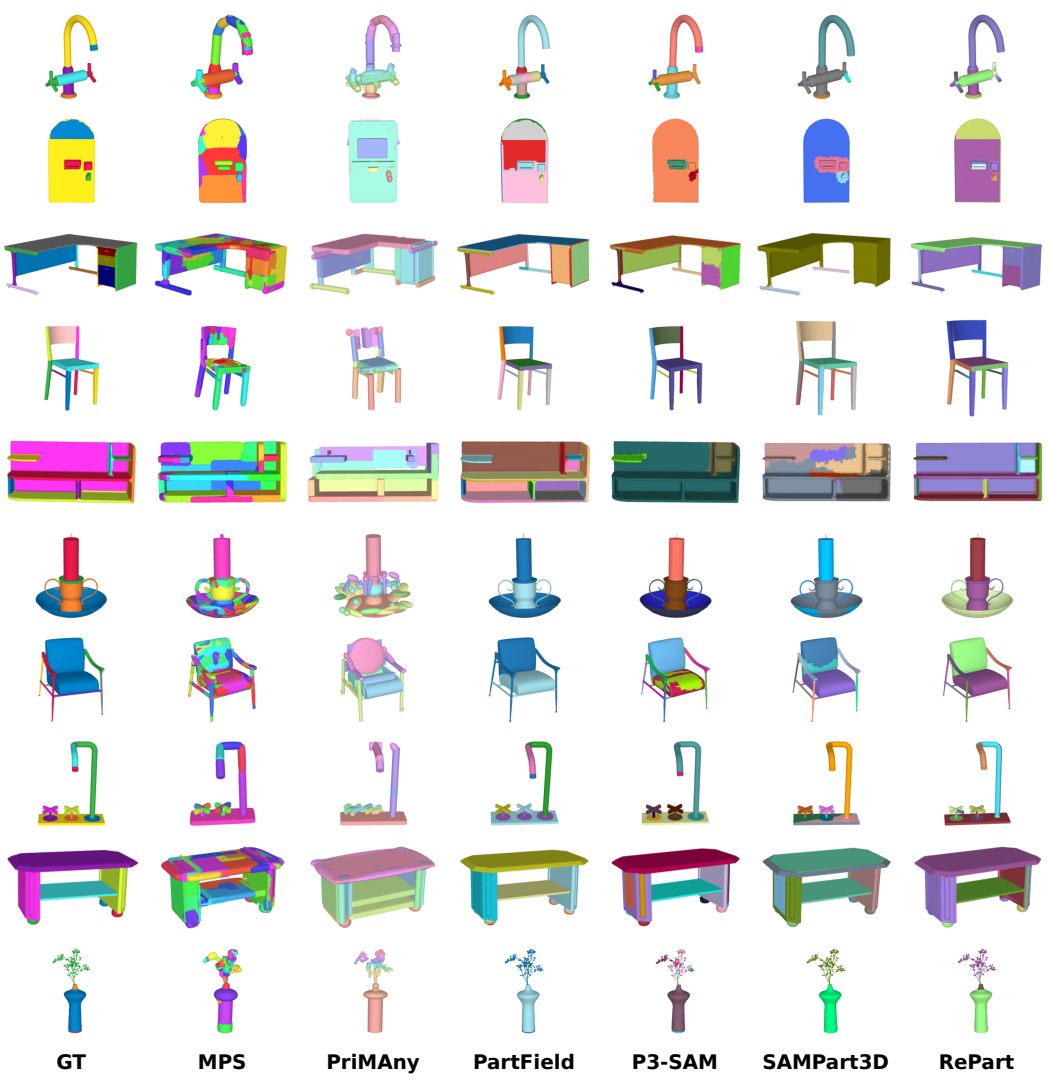

Figure 3: Qualitative comparison on five PartNet shapes (top five rows) and five 3DCoMPaT++ shapes (bottom five rows). RePart reduces the fragmented MPS decomposition and produces coherent part partitions across both datasets.

5 Conclusion and Discussion

We introduced **RePart**, a framework for *Semantic Organization* in 3D part segmentation that treats geometric primitives as reasoning units rather than final parts. RePart formulates primitive grouping as a finite-horizon merge-and-stop process, using trajectory-level reinforcement learning to optimize decisions by their downstream effects on the final partition, while Boundary-Aware Surface Labeling maps inferred part identities back to the mesh. Experiments on PartNet and 3DCoMPaT++ support our core claim that multi-step reinforcement reasoning effectively organizes a fixed primitive workspace into coherent part-level structure, rather than primitive decomposition alone yielding semantic parts. Comparisons with local and myopic objectives further show the benefit of optimizing sequential decisions. Compact geometric primitives thus provide an intermediate reasoning space between dense geometry and semantic structure. RePart separates structural abstraction, part organization, and boundary localization. The fixed MPS-derived workspace provides a stable basis for sequential reasoning, but also ties performance to the quality of the initial decomposition. Our current study focuses on 900 shapes across three category groups; extending the action space and training coverage may further improve flexibility and generalization.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, manuscript organization, and improving clarity and readability, as well as with modifying and debugging research code. All research decisions, including the problem formulation, method specification, experimental protocol, result interpretation, and scientific conclusions, were made and validated by the authors. All AI-assisted text and code were reviewed and verified by the authors. The authors take full responsibility for the final manuscript, implementation, claims, and reported experimental results.

ETHICS STATEMENT

This work studies class-agnostic 3D part segmentation from geometric data and does not involve human subjects, private personal data, or safety-critical deployment. Our experiments use existing 3D asset datasets, including PartNet and 3DCoMPaT++, for research purposes. We acknowledge that the use and redistribution of 3D assets and derived results should respect the provenance, licenses, and usage terms of the corresponding datasets and resources. Potential downstream applications of 3D understanding systems should likewise consider appropriate data governance and responsible use.

REPRODUCIBILITY STATEMENT

We provide the implementation and experimental details needed to reproduce RePart in the main paper and appendix. These include dataset splits and evaluation protocols, Composable Primitive Workspace construction and point-to-primitive correspondence, Reinforcement-Learning Part Reasoning with its merge-and-stop markov decision process and reward formulation, actor and critic inputs, PPO training settings, Boundary-Aware Surface Labeling, optimization hyperparameters, and hardware configuration. The appendix further reports detailed state representations, training-only privileged signals, parameter and feature ablations, Boundary-Aware Surface Labeling settings, and the SQ distance formulation used by the method. We plan to release the source code and necessary implementation resources upon publication to facilitate reproducibility and future research.

REFERENCES

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

Marco Attene, Bianca Falcidieno, and Michela Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006. doi: 10.1007/s00371-006-0375-x.

Alan H Barr. Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications*, 1(01):11–23, 1981.

Johann Brehmer, Sebastian Macaluso, Duccio Pappadopulo, and Kyle Cranmer. Hierarchical clustering in particle physics through reinforcement learning. In *Machine Learning and the Physical Sciences Workshop at NeurIPS*, 2020. URL https://arxiv.org/abs/2011.08191.

Ken Deng, Yunhan Yang, Jingxiang Sun, Xihui Liu, Yebin Liu, Ding Liang, and Yan-Pei Cao. Geosam2: Unleashing the power of sam2 for 3d part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6367–6376, 2026.

Bi'an Du, Xiang Gao, Wei Hu, and Renjie Liao. Generative 3d part assembly via part-whole-hierarchy message passing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20850–20859. IEEE, 2024.

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, Ryan Huang, Xianghao Xu, R Kenny Jones, and Daniel Ritchie. Parsel: Parameterized shape editing with language. *ACM Transactions on Graphics (TOG)*, 43(6):1–14, 2024.

Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7081–7091. IEEE, 2023.

Souhail Hadgi, Bingchen Gong, Ramana Sundararaman, Emery Pierson, Lei Li, Peter Wonka, and Maks Ovsjanikov. Patchalign3d: Local feature alignment for dense 3d shape understanding. *arXiv preprint arXiv:2601.02457*, 2026.

Aleš Leonardis, Aleš Jaklič, and Franc Solina. Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1289–1295, 1997. doi: 10.1109/34.632988.

Lan Li, Fazhi He, Rubin Fan, Bo Fan, and Xiaohu Yan. 3d reconstruction based on hierarchical reinforcement learning with transferability. *Integrated Computer-Aided Engineering*, 30(4):327–339, 2023.

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.

Di Liu, Xiang Yu, Meng Ye, Qilong Zhangli, Zhuowei Li, Zhixing Zhang, and Dimitris N Metaxas. Deformer: Integrating transformers with deformable models for 3d shape abstraction from a single image. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14190–14200. IEEE, 2023a.

Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21736–21746. IEEE, 2023b.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9704–9715. IEEE, 2025a.

Richard Liu, Daniel Fu, Noah Tan, Itai Lang, and Rana Hanocka. Wir3d: Visually-informed and geometry-aware 3d shape abstraction. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14810–14821. IEEE, 2025b.

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, 2023c.

Tiange Luo, Kaichun Mo, Zhiao Huang, Jiarui Xu, Siyu Hu, Liwei Wang, and Hao Su. Learning to group: A bottom-up framework for 3d part discovery in unseen categories. 2020. URL `https://iclr.cc/virtual_2020/poster_rkl8dlHYvB.html`.

Changfeng Ma, Yang Li, Xinhao Yan, Jiachen Xu, Yunhan Yang, Chunshi Wang, Zibo Zhao, Yanwen Guo, Zhuo Chen, and Chunchao Guo. P3-sam: Native 3d part segmentation. *arXiv preprint arXiv:2509.06784*, 2025a.

Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025b.

Marina Meilă. Comparing clusterings—an information based distance. *Journal of multivariate analysis*, 98(5):873–895, 2007.

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 909–918. IEEE, 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.

Chengjie Niu, Manyi Li, Kai Xu, and Hao Zhang. Rim-net: Recursive implicit fields for unsupervised learning of hierarchical shape structures. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11769–11778, 2022.

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, 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.

Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *European Conference on Computer Vision*, pp. 214–238. Springer, 2024.

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

Xiaoqian Ruan, Pei Yu, Dian Jia, Hyeonjeong Park, Peixi Xiong, and Wei Tang. Learning partonomic 3d reconstruction from image collections. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26734–26744. IEEE, 2025.

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

Habib Slim, Xiang Li, Yuchen Li, Mahmoud Ahmed, Mohamed Ayman, Ujjwal Upadhyay, Ahmed Abdelreheem, Arpit Prajapati, Suhail Pothigara, Peter Wonka, et al. 3dcompat++: An improved large-scale 3d vision dataset for compositional recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Franc Solina and Ruzena Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE transactions on pattern analysis and machine intelligence*, 12(2):131–147, 1990.

Han Su, Tianyu Huang, Zichen Wan, Xiaohe Wu, and Wangmeng Zuo. S^2am3d: Scale-controllable part segmentation of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14357–14366, 2026.

Richard S Sutton, Andrew G Barto, and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

George Tang, William Zhao, Logan Ford, David Benhaim, and Paul Zhang. Segment any mesh. *arXiv preprint arXiv:2408.13679*, 2024.

Ardian Umam, Cheng-Kun Yang, Min-Hung Chen, Jen-Hui Chuang, and Yen-Yu Lin. Partdistill: 3d shape part segmentation by vision-language model distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3470–3479. IEEE, 2024.

Nicolas von Lützow, Barbara Rössle, Katharina Schmid, and Matthias Nießner. Gaussiangpt: Towards autoregressive 3d gaussian scene generation. *arXiv preprint arXiv:2603.26661*, 2026.

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.

Yuwei Wu, Weixiao Liu, Sipu Ruan, and Gregory S Chirikjian. Primitive-based shape abstraction via nonparametric bayesian inference. In *European Conference on Computer Vision*, pp. 479–495. Springer, 2022.

Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y Lam, Yan-Pei Cao, and Xihui Liu. Sampart3d: Segment any part in 3d objects. *arXiv preprint arXiv:2411.07184*, 2024.

Jingwen Ye, Yuze He, Yanning Zhou, Yiqin Zhu, Kaiwen Xiao, Yong-Jin Liu, Wei Yang, and Xiao Han. Primitiveanything: Human-crafted 3d primitive assembly generation with auto-regressive transformer. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2025.

Zhe Zhu, Le Wan, Rui Xu, Yiheng Zhang, Honghua Chen, Zhiyang Dou, Cheng Lin, Yuan Liu, and Mingqiang Wei. Partsam: A scalable promptable part segmentation model trained on native 3d data. In *International Conference on Learning Representations*, volume 2026, pp. 132469–132493, 2026.

A Additional Experimental Details and Parameter Analysis

A.1 Training Configuration

We train one category-specific policy on 300 shapes from each of the furniture, containers, and daily-use groups. The joint model uses the union of all 900 shapes and is initialized by parameter averaging of the three category-specific checkpoints before another 100 epochs of PPO fine-tuning. The joint stage uses a smaller learning rate to preserve the behaviors learned by the specialist policies. As described in the main paper, advantages are normalized within each category, and the losses are first averaged per category and then macro-averaged. Tables A.2 and A.3 list the remaining settings and their roles.

Staged versus direct mixed-category training. Figure A.1 compares the optimization trajectories: joint fine-tuning from specialized policies reaches a reward near 0.44 within 100 epochs, whereas direct training on all 900 shapes reaches 0.42 after 200 epochs. Table A.1 likewise shows that staged training raises average mIoU from 0.5308 to 0.5460 on PartNet and from 0.4236 to 0.4476 on 3DCoMPaT++. Direct training remains applicable without category-specific initialization.

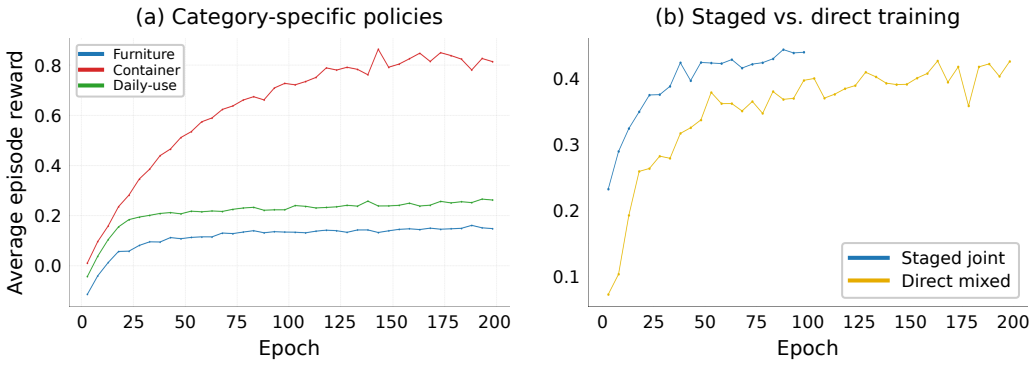

Figure A.1: PPO training rewards for category-specific training and mixed-category training. Each point is averaged over a non-overlapping window of five epochs.

Table A.1: Category-level mIoU under direct mixed-category and staged joint training. Bold denotes the preferred result.

	PartNet				3DCoMPaT++			
Policy	Furn.	Cont.	Daily	Avg.	Furn.	Cont.	Daily	Avg.
Direct mixed	0.4186	0.6314	0.5263	0.5308	0.3558	0.4733	0.4477	0.4236
Staged joint	**0.4247**	**0.6456**	**0.5432**	**0.5460**	**0.3703**	**0.5149**	**0.4607**	**0.4476**

Unless stated otherwise, PPO uses discount factor $\gamma = 0.99$, GAE parameter $\lambda = 0.95$, clipping radius $\epsilon = 0.2$, entropy coefficient 0.01, and value-loss coefficient 0.5. All parameter studies below use the same 300 PartNet furniture shapes and change only the stated setting.

The default maximum horizon is 200 action steps per episode, with earlier termination when the policy selects `stop`; joint training inherits the largest horizon among its source configurations. We clip the global gradient norm to 1.0. Training actions are sampled from the masked policy, whereas evaluation selects the highest-logit valid action. Point sampling remains stochastic during evaluation.

A.2 Parameter Ablations

Figure A.2 reports the five-epoch mean training reward for the default setting and four variants. The default uses $\kappa = 4$, both group-count shaping terms, and PPO from a random actor initialization.

Table A.2: Data, schedule, and PPO settings for category-specific and joint training.

Parameter	Category-specific	Joint	Role
num_samples	300	900	Number of PartNet training shapes. The joint set is the union of the three category groups.
train_epochs	200	100	Number of outer training epochs. Each epoch collects fresh trajectories before PPO updates.
lr	1×10^{-4}	5×10^{-5}	Adam learning rate. The lower joint rate reduces changes to the averaged specialist initialization.
rollout_episodes_per_epoch	304	304	Number of complete merge-and-stop episodes collected in each outer epoch.
rollout_workers	16	16	Parallel workers used to collect trajectories. This changes throughput, not the objective.
ppo_epochs	4	4	Number of optimization passes over each collected rollout buffer.
minibatch_size	128	128	Number of rollout transitions in each PPO update minibatch.
policy_temperature	1.0	1.0	Temperature applied to masked action logits during sampling; 1.0 leaves their scale unchanged.
seed	42	42	Random seed for data sampling, rollouts, and model initialization.
heartbeat_sec	30	30	Progress-reporting interval used by MPS preprocessing; it does not affect learning.

Table A.3: Representation, supervision, reward, and output settings shared by both training stages.

Parameter	Category-specific	Joint	Role
bootstrap_source	MPS	MPS	Uses the MPS decomposition as the initial oversegmented SQ proxy.
point_sample_size	512	512	Number of mesh-surface points sampled whenever the geometry observation is encoded.
merge_neighbor_k	4	4	Restricts each group to merge candidates among its four nearest groups.
sq_label_method	coverage	coverage	Assigns training-time SQ labels using geometric coverage rather than point-wise majority alone.
final_mesh_mapping_mode	atom	atom	Maps grouped SQs by atom voting, SQ volumetric rescue, and a final nearest-surface fallback.
aux_part_loss_w	0.5	0.5	Weight for the critic's auxiliary prediction of terminal part mIoU.
aux_group_loss_w	0.25	0.25	Weight for the critic's auxiliary prediction of terminal normalized group-count error.
aux_target_count_loss_w	0.1	0.1	Weight for predicting the target group-count ratio from actor-visible geometry during training.
group_excess_penalty_w	0.02	0.02	Dense reward weight for reducing the normalized difference between predicted and target group counts.
stop_reward_bonus	0.005	0.005	Base reward b_{stop} for an explicit stop.
stop_group_penalty_w	0.1	0.1	Penalizes stopping while a group-count mismatch remains.

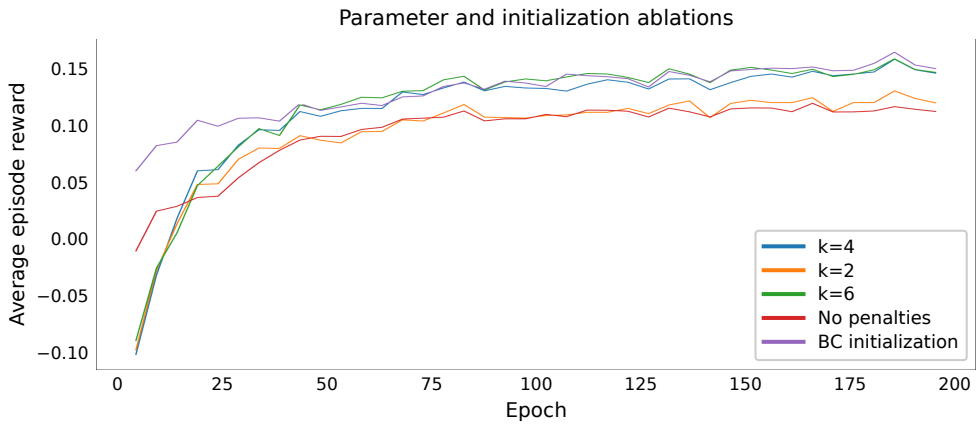

Figure A.2: PPO reward curves on the 300-shape furniture training set. Curves are averaged over every five epochs. The default cold-start PPO setting uses $\kappa = 4$ and both group-count penalties.

Candidate neighborhood size. The parameter `merge_neighbor_k` controls how many geometrically nearest groups are considered for each current group. With $\kappa = 2$, the candidate set is often too narrow to include a useful merge, and the reward curve remains clearly below the default. Increasing the neighborhood to $\kappa = 6$ produces a curve that is nearly identical to $\kappa = 4$, indicating little benefit from the additional candidates. However, the larger candidate set increases the masked action space and the cost of evaluating merge features. We therefore use $\kappa = 4$ as the best balance between candidate coverage and training efficiency.

Group-count reward shaping. Let G_t be the number of current groups and K the ground-truth part count. As in the main method, their normalized mismatch is $E_t = |G_t - K|/\max(K, 1)$, which treats over- and under-segmentation symmetrically. The dense term weighted by `group_excess_penalty_w` rewards a transition when it reduces this mismatch, while `stop_group_penalty_w` penalizes stopping with a nonzero mismatch. Setting both weights to zero removes the positive shaping signal produced by eliminating redundant groups. PPO must then assign credit mainly through changes in partition overlap, making useful merge sequences harder to learn. Accordingly, the `no_penalties` curve converges more slowly and remains below the default, while its mIoU also decreases on both test sets in Table A.4. For a direct measure of grouping granularity, we additionally report the mean absolute terminal count error $|G_T - K|$. Removing the penalties increases this error from 4.1500 to 9.9667 on PartNet and from 5.9167 to 11.0833 on 3DCoMPaT++. Since the SQ workspace is intentionally over-segmented and the policy progressively merges groups, the substantially lower error of the default setting shows that the count-aware rewards effectively control residual over-segmentation.

Behavior-cloning initialization. We also tested whether a greedy expert could provide a useful warm start for the large merge action space. The expert scores each legal merge by its immediate post-merge mIoU, chooses the best one, and stops when no candidate gives a positive gain. We apply behavior cloning to expert trajectories from a subset of the training shapes until the cloning loss falls below 0.1, and then continue with the same PPO training used by the default model. This setting differs from the Greedy-BC baseline in the main paper, which is evaluated without subsequent PPO training. As Figure A.2 shows, the warm-started policy begins with a much higher reward, but its advantage gradually disappears and its late-stage curve overlaps the cold-start PPO curve. The expert initialization narrows early exploration without improving the final policy, adds a separate training stage, and gives lower test mIoU in Table A.4. We therefore use cold-start PPO in RePart. Behavior cloning may still be useful for more complex tasks with much larger action spaces or rewards that are difficult to discover through unguided exploration.

Table A.4: Furniture results for reward-shaping and behavior-cloning variants. Count error is the mean absolute difference between the terminal group count and the ground-truth part count; lower is better. All models use the same 300 PartNet furniture training shapes.

	mIoU↑		$\lvert G_T - K \rvert \downarrow$	
Training setting	PartNet	3DCoMPaT++	PartNet	3DCoMPaT++
Cold-start PPO (default)	**0.4280**	**0.3699**	**4.1500**	**5.9167**
BC warm start + PPO	0.4165	0.3566	–	–
No group-count penalties	0.4130	0.3558	9.9667	11.0833

A.3 Sensitivity to Random Seeds

We repeat the category-specific furniture experiment with five random seeds, using 300 PartNet furniture shapes for training in each run. Table A.5 reports the results on the PartNet furniture test set. The final row gives the mean and sample variance across the five runs, with variance computed using a denominator of $5 - 1$. The mIoU ranges from 0.4247 to 0.4311, with a mean of 0.4271, indicating limited sensitivity to the random seed in this setting. The other partition metrics show some variation, particularly RI, but the results remain broadly consistent across runs rather than depending on a single favorable seed.

Table A.5: Random-seed sensitivity of RePart-S trained on 300 PartNet furniture shapes. The final row reports the mean, with sample variance in parentheses.

Seed	mIoU↑	RI↑	VoI↓	SC↑
42	0.4280	0.7804	1.2997	0.5751
91161	0.4266	0.7669	1.3310	0.5649
926	0.4311	0.7967	1.3198	0.5794
339	0.4247	0.7240	1.3500	0.5496
32746	0.4253	0.7551	1.3148	0.5646
Mean (variance)	0.4271 (6.513×10^{-6})	0.7646 (7.560×10^{-4})	1.3231 (3.533×10^{-4})	0.5667 (1.330×10^{-4})

B State Representation and Feature Ablations

B.1 State Representation

At each decision step, the actor receives only quantities computed from the input geometry and current grouping in the Composable Primitive Workspace, which provides *Structural Abstraction*. Table B.6 gives the exact organization of these inputs. Here $P = |X_s|$ is the number of points sampled for one actor observation, while $N = |X|$ denotes the full point set in the main method. We report related dimensions as feature groups instead of listing every scalar separately, while retaining the component order needed to reproduce the state. Distances and spatial extents are normalized by the scene radius unless stated otherwise; ratios are normalized by the relevant point, primitive, group, or step count. SQ parameters remain fixed after MPS initialization, whereas grouping-dependent descriptors are recomputed after every merge.

The stored point state is $P\times3$; the encoder constructs the seven-dimensional point token in Table B.6 before applying a shared MLP and concatenating max and mean pooled features. SQ parameters and the 17-dimensional group descriptors are encoded per token and pooled with `sq_mask`. Their pooled features and the eight global descriptors form the actor context. The stop head scores this context directly. For a merge, the actor additionally uses the two canonically ordered tokens, their absolute difference, and the 17-dimensional pair descriptor. The routing tensors—`action_mask` (A), `merge_pair_indices` $(C\times2)$, `merge_pair_action_ids` (C), and `merge_pair_mask` (C)—only map valid candidates to action logits and exclude padded or unavailable actions; they do not provide geometric evidence.

Table B.6: Geometry-only state used by the RePart actor. Dimensions use M SQ slots, P sampled points, and C candidate pairs. Subgroups follow their order in the implementation.

Feature group	Dim.	Components	Update and scaling
Point token	$P \times 7$	Raw xyz, centered and radius-normalized xyz, and normalized radial distance.	Points are resampled per state; center and radius are recomputed by the encoder.
SQ token	$M \times 11$	Two exponents, three axis lengths, three Euler angles, and three translations.	Fixed after MPS; inactive slots are zero padded and masked.
Group context	$M \times 7$	Nearest-SQ distance, group-size ratio, group diagonal, group-center radius, group support, candidate degree, and nearest-group AABB gap.	Recomputed after each merge from all members of the current group.
Assignment statistics	$M \times 4$	Normalized assignment entropy, top-1 confidence, low-margin boundary ratio, and top-1–top-2 margin.	Recomputed from soft point-to-SQ assignments.
Shape and fit	$M \times 6$	PCA linearity, planarity, and scattering; SQ–PCA axis alignment; mean and 90th-percentile surface residuals.	Recomputed from assigned points; residuals are divided by scene radius.
Global progress	4	Step progress, primitive-slot occupancy, group-slot occupancy, and group count relative to initialization.	Recomputed after each action from normalized counts.
Global group state	4	Singleton fraction, mean group-size ratio, valid-merge density, and mean candidate degree.	Recomputed from the current partition and action mask.
Pair compatibility	$C \times 6$	Validity, representative-center distance, representative AABB gap, scale similarity, volume ratio, and orientation similarity.	Computed for the current κ-nearest candidate pairs.
Pair union geometry	$C \times 8$	Combined group size, group-center distance, AABB IoU, support-ratio similarity, diagonal ratio, group AABB gap, union-volume expansion, and diagonal growth.	Uses full-group bounds and support; distance terms use scene radius.
Contact and orientation	$C \times 3$	AABB contact ratio, group PCA-axis alignment, and mean-normal alignment.	Recomputed for each current candidate pair.

Table B.7: Label-derived quantities available only during training. They are excluded from the deployed actor observation.

State field	Dim.	Components	Use
`privileged_prim_stats`	$M \times 6$	Point-support ratio, label purity, second-label share, unique-label ratio, normalized label entropy, and fragmentation $(1 - \text{purity})$.	Critic only; recomputed after each merge.
`privileged_global_features`	4	Target part-count ratio, signed and absolute group-count errors, and exact-count indicator.	Critic only; recomputed after each merge.
`target_group_count_ratio`	1	Ground-truth part count divided by M.	Auxiliary target for the actor's geometry-based count predictor.

The critic receives a detached copy of the actor context and separately encodes the SQ, primitive/group, and privileged features in Table B.7. The actor predicts a target group-count ratio from geometry and provides that prediction to both action heads to judge when further merging is useful. The stored `target_group_count_ratio` is only its ground-truth auxiliary target and is never supplied as an actor observation. Thus, the deployed policy remains geometry-only.

B.2 Grouped Feature Ablation

We group related feature dimensions and remove one group at a time by setting its values to zero while preserving the network architecture and all other training settings. This avoids changing model capacity across variants. The study uses 100 PartNet furniture shapes and trains each policy for 100 epochs, with no behavior-cloning initialization. Primitive-level features are divided into group context, assignment confidence, and shape-fit statistics. Global features are divided into episode progress and current group state. Candidate-pair features are divided into pair compatibility, full-group union geometry, and contact-orientation cues.

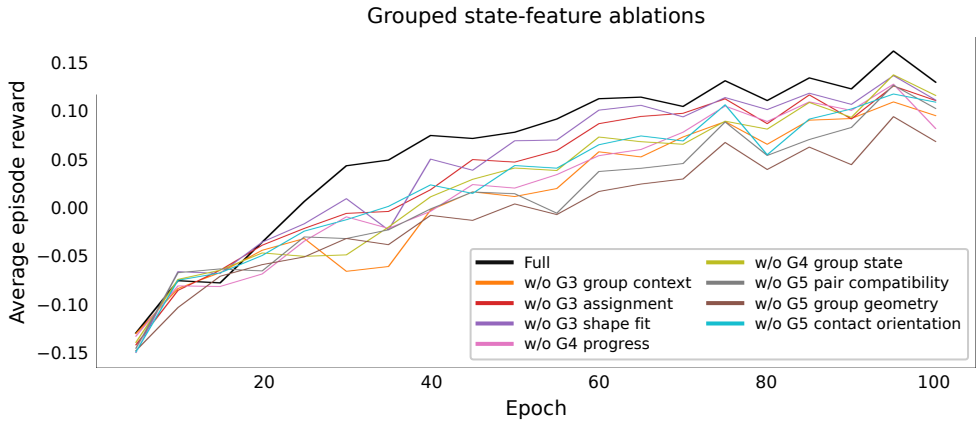

Figure B.3: Grouped feature ablations on 100 PartNet furniture shapes. Curves show PPO reward averaged over every five epochs during the 100-epoch training runs. Each *w/o* variant zeros only the indicated feature group.

As shown in Figure B.3, the complete representation achieves the highest overall reward. Removing full-group union geometry causes the clearest degradation, because a representative primitive alone cannot describe how two multi-primitive groups would expand after merging. Pair-compatibility features are also important for rejecting nearby but structurally inconsistent candidates. Group context, assignment confidence, fitting statistics, and global progress cues have smaller individual effects, but their removals consistently remain below the complete model. These results indicate that the retained feature blocks are complementary: SQ primitives describe individual primitives, while group-level and episode-level evidence helps the policy evaluate their longer-term effect on the partition.

B.3 Asymmetric Training Signals

The standard asymmetric model uses label-derived information only to improve training. The per-primitive privileged statistics describe label purity, ambiguity, and fragmentation, while the privileged global features summarize target-count mismatch and terminal grouping quality. The critic combines these signals with a detached actor representation. Detachment prevents critic-side supervision from directly changing the actor encoder, while the improved value estimate can still reduce noise in PPO advantages. Separately, the ground-truth group-count ratio supervises the actor's geometry-based count predictor; at inference, the predictor operates without labels.

For the symmetric variant, we remove `privileged_prim_stats`, `privileged_global_features`, and `target_group_count_ratio`, disable the target-count auxiliary loss, and

let the critic use only the detached actor representation. Both variants retain the same geometry-only actor, action space, and reward definition, and ground-truth labels remain available only for computing the training reward. This comparison therefore isolates the contribution of asymmetric training rather than changing the inference inputs.

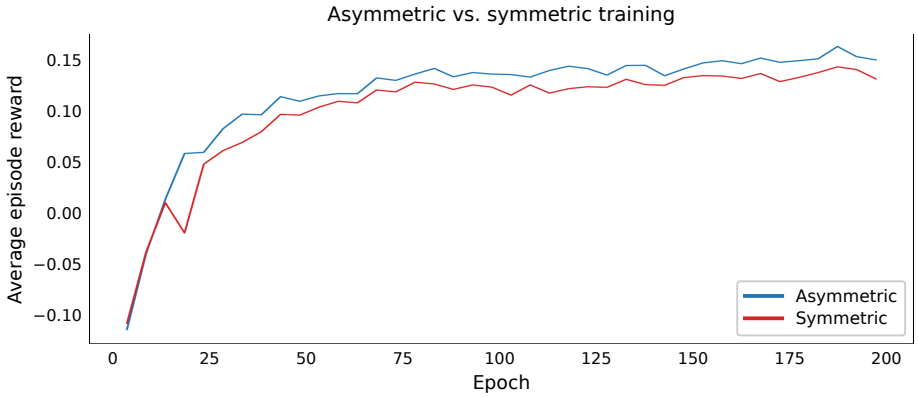

Figure B.4: Asymmetric and symmetric training on 300 PartNet furniture shapes for 200 epochs. Curves show PPO reward averaged over every five epochs. The asymmetric variant uses label-derived critic features and target-count supervision only during training; both actors use geometry-only inputs.

Figure B.4 shows that the asymmetric model maintains a higher reward through most of training and finishes above the symmetric variant. Privileged purity and fragmentation statistics help the critic distinguish states that can appear geometrically similar but differ in partition quality, producing more informative advantage estimates for earlier merge decisions. Count supervision also provides a direct training signal for stopping granularity. The gain is moderate rather than dominant, which agrees with the main design: geometric state features drive the policy, while asymmetric information makes their optimization more effective without introducing label dependence at inference.

B.4 Future Work: LLM- and Agent-Based Policy Generation

The Composable Primitive Workspace separates semantic organization from dense surface prediction and is not tied to a particular policy architecture. This suggests exploring pretrained large language models (LLMs) or multimodal agents as alternatives to the current PPO-based policy in Reinforcement-Learning Part Reasoning. Compact primitives and explicit group relations could provide a more manageable basis for such models than direct reasoning over dense mesh elements. Related work on language-guided shape editing and agent-based articulation illustrates the potential of combining foundation models with geometric representations (Ganeshan et al., 2024). In our setting, an LLM or agent might guide merge-and-stop decisions while retaining the Composable Primitive Workspace and Boundary-Aware Surface Labeling, potentially avoiding task-specific RL training.

Such an alternative may draw on broader knowledge of object composition and reduce dependence on the limited category coverage of our training set, offering a possible route toward better generalization to unfamiliar structures. However, broader prior knowledge does not guarantee geometrically consistent grouping. Reliable stopping and label-free feedback remain open questions, since our current reward and privileged critic depend on ground-truth annotations unavailable at deployment. We have not implemented or evaluated this extension; its effectiveness and computational cost relative to PPO remain topics for future work.

C BOUNDARY-AWARE SURFACE LABELING

We ablate the two components of Boundary-Aware Surface Labeling that transfer terminal SQ groups back to mesh faces. All variants reuse exactly the same grouped-SQ predictions, raw meshes, SQ surface samples, and ground-truth face correspondences, so the comparison isolates Boundary-Aware Surface Labeling rather than Reinforcement-Learning Part Reasoning. We evaluate three increasingly structured variants on the daily-use, furniture, and container groups:

- **Facewise nearest SQ** assigns each face centroid independently to the group with the nearest sampled SQ surface point.
- **+ Curvature atoms** first over-segments the mesh along high-dihedral adjacencies. Valid proximity votes are aggregated within each atom, while rejected atoms are resolved by the common nearest-surface fallback. SQ volumetric rescue is disabled in this variant.
- **+ SQ volumetric evidence** is the complete Boundary-Aware Surface Labeling module. It reconsiders rejected atoms using the analytic signed-distance evidence of the grouped SQs before applying the same fallback to any remaining rejected atom.

For reproducibility, each SQ surface is sampled on a 72×36 parameter grid. The atom boundary threshold is the 90th percentile of adjacent-face dihedral angles, clipped to $[6^\circ, 12^\circ]$, followed by one label-smoothing iteration. We use uniform vote counts, a normalized face-distance threshold of 0.04, a maximum distant-face fraction of 0.6, and a minimum atom size of one face. For volumetric rescue, an atom is accepted when at least half of its centroids support one group with analytic SQ signed distance no greater than 0.005. This evidence is evaluated only for rejected atoms and therefore does not overwrite reliable proximity assignments. If both tests fail, an unrestricted nearest-surface majority vote assigns the whole atom. All partition metrics are computed after this common fallback over the same complete set of mesh faces with valid ground-truth labels; no unassigned prediction is excluded. Final coverage is therefore 100% for every variant. We additionally report the fraction of faces assigned by fallback as a diagnostic of how often the preceding stages remain unresolved.

Table C.8: Ablation of Boundary-Aware Surface Labeling on PartNet. Partition metrics are evaluated on all valid faces after the common fallback. Final coverage is 100% for every variant; FB faces denotes the percentage assigned only by fallback and does not apply to direct facewise nearest assignment. Bold denotes the preferred result.

Category	Projection	mIoU↑	SC↑	VoI↓	Final cov.	FB faces↓
Daily-use	Facewise nearest SQ	0.4500	0.5574	1.2427	1.0000	–
Daily-use	+ Curvature atoms	0.5351	0.6597	0.8608	1.0000	3.7864%
Daily-use	+ SQ volumetric evidence	**0.5432**	**0.6598**	**0.8607**	1.0000	**3.1309%**
Furniture	Facewise nearest SQ	0.3310	0.4645	1.8139	1.0000	–
Furniture	+ Curvature atoms	0.4203	**0.5536**	**1.3358**	1.0000	6.8709%
Furniture	+ SQ volumetric evidence	**0.4247**	0.5522	1.3373	1.0000	**0.9656%**
Container	Facewise nearest SQ	0.6139	0.7868	0.5791	1.0000	–
Container	+ Curvature atoms	0.6421	0.8280	**0.4256**	1.0000	2.9452%
Container	+ SQ volumetric evidence	**0.6456**	**0.8282**	0.4257	1.0000	**2.2130%**

Table C.8 evaluates the same complete face set used by the deployed output. The complete Boundary-Aware Surface Labeling module improves mIoU over facewise nearest assignment by 0.0932, 0.0937, and 0.0317 on daily-use, furniture, and container objects, respectively. The corresponding SC gains are 0.1024, 0.0877, and 0.0414, while VoI decreases by 0.3820, 0.4766, and 0.1534. These consistent final-output gains show that regional voting suppresses fragmented labels and better aligns group transitions with mesh structure.

Curvature atoms account for most of the metric improvement, while SQ volumetric evidence mainly resolves regions that surface voting leaves uncertain. It further raises mIoU from 0.5351 to 0.5432 on daily-use objects, from 0.4203 to 0.4247 on furniture, and from 0.6421 to 0.6456 on containers, while SC and VoI remain nearly unchanged. More importantly, it reduces the fallback-face rate from 3.7864% to 3.1309% for daily-use objects, from 6.8709% to 0.9656% for furniture, and from 2.9452% to 2.2130% for containers, corresponding to relative reductions of 17.31%, 85.95%, and

24.86%. Thus, volumetric evidence reduces reliance on the unrestricted fallback, while the shared fallback guarantees a complete final partition for every variant.

D Generalization to Unseen PartNet Categories

Our PartNet training set contains 14 categories grouped as containers (Bag, Bottle, Bowl, Mug, and Vase), furniture (Chair, Door, StorageFurniture, and Table), and daily-use objects (Earphone, Faucet, Knife, Lamp, and Scissors). The main PartNet evaluation uses held-out shapes from these category groups. To study transfer beyond the training categories, we additionally test on Bed, Display, Hat, Laptop, Refrigerator, and TrashCan, which are excluded from training but share some part-level structures with the seen categories. We randomly sample ten shapes from each unseen category and use the same evaluation protocol as in the main experiments.

Table D.9: Part segmentation on PartNet categories excluded from training. Results are averaged over the six unseen categories. Bold and underline denote the best and second-best results, respectively.

Method	mIoU↑	RI↑	VoI↓	SC↑
MPS	0.2103	0.4460	1.1156	0.4460
PartField	**0.4295**	**0.7416**	1.1886	<u>0.5875</u>
P3-SAM	0.3834	<u>0.7112</u>	<u>1.1090</u>	0.5851
SAMPart3D	0.2393	0.5380	1.3913	0.4574
PriMAny	0.2028	0.5793	2.7416	0.2379
RePart (Ours)	<u>0.3992</u>	0.7033	**0.9327**	**0.5934**

Table D.10: Category-level mIoU on PartNet categories excluded from training. Bold and underline denote the best and second-best results, respectively.

Category	MPS	PartField	P3-SAM	SAMPart3D	PriMAny	RePart
Bed	0.0789	**0.2908**	0.2313	0.1526	0.2094	<u>0.2661</u>
Display	0.1980	<u>0.4304</u>	0.4301	0.2888	0.2000	**0.4693**
Hat	0.4417	0.6135	**0.7073**	0.3813	0.1860	<u>0.6222</u>
Laptop	0.1662	<u>0.3291</u>	0.3184	0.2097	0.2385	**0.3672**
Refrigerator	0.1330	**0.3337**	0.1930	0.1464	0.1289	<u>0.2554</u>
TrashCan	0.2439	**0.5791**	<u>0.4206</u>	0.2530	0.2539	0.4153

As shown in Table D.9, RePart achieves the best VoI and SC and the second-best average mIoU, while remaining competitive in RI. Table D.10 further shows that it ranks first or second in mIoU on five of the six unseen categories, including the best results on Display and Laptop. The low VoI indicates that its predicted groups preserve coherent part structure rather than fragmenting objects excessively. These results and the examples in Figure D.5 show that the learned merge policy retains useful transfer to unseen categories when they contain local structures similar to those observed during training. This experiment supports generalization across related structures, rather than unrestricted open-category segmentation.

E Additional Qualitative Comparisons

Figures E.6 and E.7 extend the qualitative comparison in Figure 3.

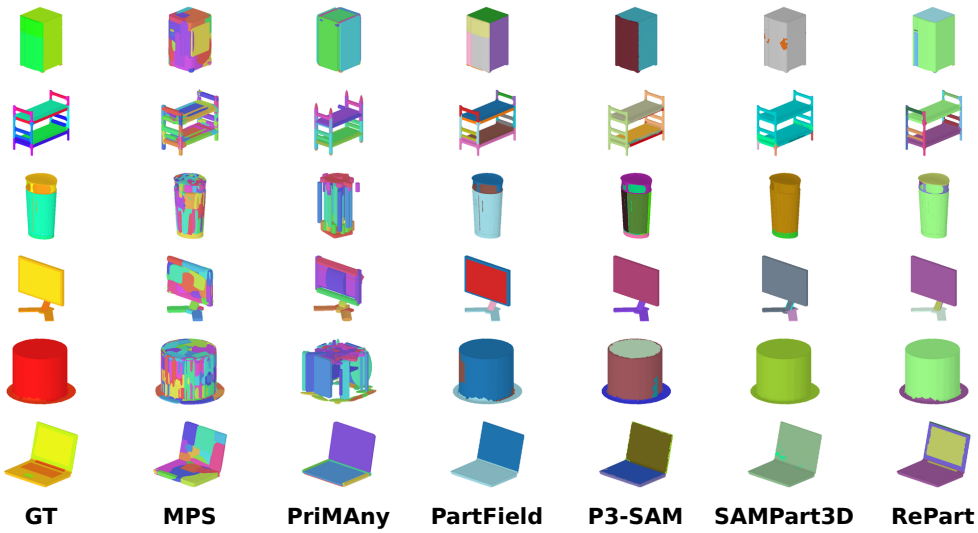

Figure D.5: Qualitative comparison on PartNet categories excluded from training.



Figure E.6: Additional qualitative comparisons on PartNet.

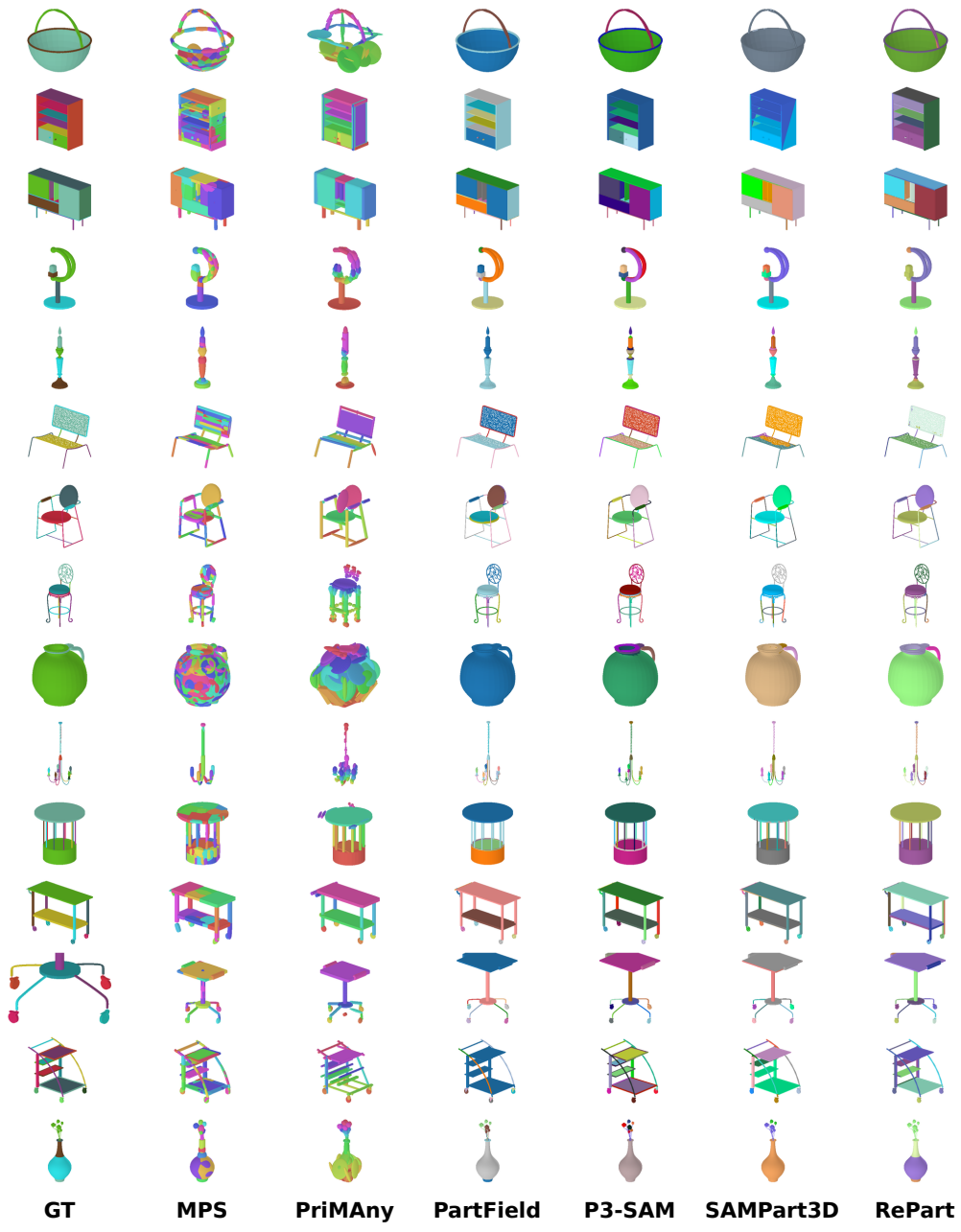

Figure E.7: Additional cross-dataset qualitative comparisons on 3DCoMPaT++.

F Superquadric Distance and Surface Evaluation

This section records the standard SQ functions used for point assignment and volumetric evidence in the main text. It does not introduce an additional fitting or inference stage. For $p_j = (\boldsymbol{e}_j, \boldsymbol{a}_j, \boldsymbol{\omega}_j, \boldsymbol{t}_j)$, let $\bar{\boldsymbol{x}}_j = R(\boldsymbol{\omega}_j)^\top(\boldsymbol{x} - \boldsymbol{t}_j)$, where R maps ZYX Euler angles to a rotation matrix. The inside–outside function is (Barr, 1981; Liu et al., 2023c)

$$F_j(\boldsymbol{x}) = \left[\left| \frac{\bar{x}_{j,x}}{a_{j,x}} \right|^{2/e_{j,2}} + \left| \frac{\bar{x}_{j,y}}{a_{j,y}} \right|^{2/e_{j,2}} \right]^{e_{j,2}/e_{j,1}} + \left| \frac{\bar{x}_{j,z}}{a_{j,z}} \right|^{2/e_{j,1}}. \tag{14}$$

The level set $F_j(\boldsymbol{x}) = 1$ defines the analytic surface sampled by $\mathcal{B}(p_j)$. Homogeneity along rays from the SQ center gives the radial signed-distance approximation (Solina & Bajcsy, 1990; Liu et al., 2023c)

$$d_{\mathrm{SQ}}(\boldsymbol{x}; p_j) = \|\bar{\boldsymbol{x}}_j\|_2 \left(1 - F_j(\boldsymbol{x})^{-e_{j,1}/2}\right). \tag{15}$$

It is positive outside the primitive and negative inside; it is a radial approximation, not an exact Euclidean signed distance. Its absolute value supplies the distance d_{ij} used in point-to-primitive assignment, while its sign supplies the volumetric evidence D_u used in Boundary-Aware Surface Labeling.