GEOMETRIC AND SEMANTIC COUPLING FOR INTERACTION UNDERSTANDING IN 3D SCENES

Anonymous authors
Paper under double-blind review

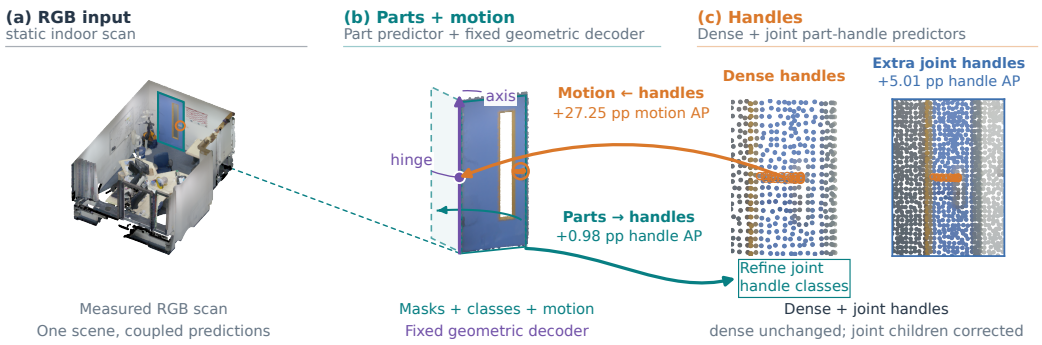


Figure 1: **Coupled interaction understanding from a static RGB point cloud.** Handles guide a training-free geometric decoder to select a part's hinge side (+27.25 percentage points in motion AP). Conversely, a joint part-handle predictor adds handles (+5.01 points in handle AP), while part classes refine their rotation/translation labels (+0.98 points). Orange and teal arrows show one-pass transfers without iterative feedback. Gains are from controlled validation comparisons (Tab. 4).

ABSTRACT

Understanding interaction in a 3D scene requires recovering movable parts, their motion, and where they can be operated. These quantities are related, and their predictions can inform one another. A closed cabinet door, for instance, reveals a movable surface but may leave the hinge side ambiguous; its handle helps resolve this ambiguity, while the part provides context for localizing and interpreting the small handle. Building on this observation, we present SEGMENT–SNAP, which combines geometric and semantic evidence through part-handle coupling. Three independently trained predictors recover movable parts, dense handles, and part-associated handle proposals. We couple their outputs in two directions. For part motion, a training-free geometric decoder fits predicted part surfaces under explicit physical priors and uses detected handles to select candidate hinge lines. For handle prediction, a part-conditioned branch proposes additional handles, while standalone part classes refine their rotation/translation labels, with dense-handle labels as a fallback. Each transfer is applied once, without iterative feedback. On the Articulate3D validation set, handle guidance raises motion-gated AP from 13.74 to 40.98 under fixed masks and axes. Additional handle proposals raise handle AP from 24.63 to 29.65, and full contextual class correction raises it to 30.99 in the reference configuration. Fixed-input controls, retraining ablations, learned-decoder comparisons, and paired visualizations together characterize the benefits and limits of this coupling.

1 INTRODUCTION

A 3D reconstruction can tell us that a cabinet is present without explaining how to open it. Progress in annotated indoor reconstruction and open-vocabulary scene representations makes objects increasingly accessible to semantic queries (Dai et al., 2017; Yeshwanth et al., 2023; Takmaz et al.,

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

2023; Gu et al., 2024). Interaction requires a richer description: which surfaces are movable, where they can be operated, and whether they rotate about a hinge or translate along a direction. Recovering this structure connects scene perception to the geometry needed for subsequent interaction. Recent benchmarks make these questions concrete by annotating movable parts, motion parameters, and small operating regions in real indoor scans (Halacheva et al., 2025; Delitzas et al., 2024). We call the operating regions *handles*, including regions without a conventional handle shape. Our task is to infer this joint description from a static RGB point cloud, without observing an object move or executing an action.

Two coupled ambiguities make this task more demanding than object recognition. The first is *scale and context*. A door occupies a broad surface, whereas its handle may comprise only a few observed points. Grouping points into coherent regions helps delineate the door but can absorb a small handle into its surroundings. Conversely, preserving fine spatial detail can reveal a handle while leaving its function ambiguous: a locally similar region may operate either a hinged door or a sliding drawer. Accurate localization therefore does not automatically give the correct motion class.

The second ambiguity is *motion geometry*. A closed, approximately planar panel constrains the orientation and extent of a movable part, but not necessarily its attachment side. Even a perfect door mask can admit a vertical hinge on either edge. Observations across articulation states can expose that joint (Jiang et al., 2022b; Liu et al., 2023a), but this evidence is absent from a single static scene. A motion estimate must distinguish the alternatives using other cues. This matters because an instance may be well segmented yet fail a motion-sensitive metric, while a small but correctly located handle can provide the missing geometric evidence.

Existing approaches address these ambiguities through different sources of evidence. USDNet combines instance segmentation and dense motion prediction to recover movable parts, handles, and articulation, without explicitly using detected handle locations to select hinges (Halacheva et al., 2025). Functional scene graphs relate interactive elements to objects but do not directly specify metric motion axes or origins (Zhang et al., 2025). Reconstruction approaches infer joints from completed geometry or scene-to-simulation pipelines (Zhao et al., 2025; Iliash et al., 2026; Huang et al., 2026). Our question is complementary: how can the part and handle predictions already available in a static scan provide evidence for one another?

We exploit the physical attachment between a handle and its movable part. On many upright doors, a handle lies away from the hinge, making its position informative about the attachment side. In the reverse direction, the surrounding part supplies two kinds of context: a larger region from which to propose a tiny handle, and a motion class with which to interpret that handle. The first concerns detection; the second concerns the label of an existing detection. Thus, *handle locations guide part motion, while part-associated proposals and part classes improve handle prediction*. This relation connects the outputs without requiring the same spatial representation or inference rule for both.

We introduce SEGMENT–SNAP, which separates scale-appropriate perception from explicit evidence transfer (Figs. 1 and 2). Three independently trained predictors have distinct responsibilities. A standalone *part predictor* supplies the final movable surfaces and labels each as rotating or translating. A *dense handle predictor* classifies individual points and groups them into initial handle instances. A *joint part-handle predictor* supplies a second handle source: a learned feature representing each of its own parts predicts both that part and an associated handle. These internal parts organize the additional proposals; they do not replace the standalone parts. The design therefore preserves the standalone part and dense-handle predictions while allowing part-associated detections to supplement the handle set.

For motion, we fit the standalone part surfaces and construct candidate hinge lines under explicit planar and upright priors. An associated dense handle selects the candidate farthest from its location. Translating parts instead use the thin direction of their fitted geometry. Axes and origins are computed by a fixed geometric decoder rather than a trained motion-regression head. Only this decoder is training-free; the perception networks supplying its inputs are learned.

For handles, the joint predictor provides additional candidates, initially labelled by their internal parent parts. Confident standalone parts that contain these candidates can correct their rotation/translation labels; overlapping dense handles provide a fallback when no part qualifies. We append the resulting candidates without changing the original dense detections. This reverse transfer uses part context for both proposals and labels, but the two operations draw on different part predictors. Each transfer is applied once, without feeding the final handles back to recompute hinges.

On Articulate3D validation, handle guidance raises motion-gated AP from 13.74 to 40.98 while keeping part masks, classes, scores, and axes fixed. Appending associated handle proposals raises handle AP from 24.63 to 29.65. Contextual class correction of these added handles then raises AP to 30.99 in the reference configuration, with a smaller and more variable gain across repeated training. Our controls distinguish the benefit of an associated proposal source from that of query conditioning alone, and separate label correction from changes in localization.

Our contributions are:

- A geometric and semantic coupling strategy for interaction understanding: dense handles guide part motion, while part-associated proposals and standalone part classes improve handle detection through distinct, one-pass transfers.
- A training-free motion decoder that separates support fitting, physical axis priors, and handle-guided hinge selection, using independently predicted scene handles to resolve hinge-side ambiguity.
- An evaluation combining fixed-input interventions, learned-decoder controls, repeated training, and scene-level comparisons, separating the effects of coupling from differences in inputs, supervision, and inference cost.

2 Related work

From semantic to functional scene descriptions. ScanNet and ScanNet++ support semantic and instance-level indoor scene understanding (Dai et al., 2017; Yeshwanth et al., 2023); 3DSSG and ConceptGraphs add semantic and open-vocabulary object relations (Wald et al., 2020; Gu et al., 2024). SceneFun3D annotates functionality and affordances (Delitzas et al., 2024), while Articulate3D combines movable parts, interaction regions, and motion parameters, jointly predicted by USDNet (Halacheva et al., 2025). OpenFunGraph relates interactive elements to objects (Zhang et al., 2025). We use the part-handle relation to inform predictions: associated handles select metric hinge hypotheses, while part evidence supplements handle proposals and labels.

Point-cloud representations and instance prediction. PointNet++ and Point Transformer V3 learn point-cloud features through hierarchical neighborhoods and serialized attention, respectively (Qi et al., 2017; Wu et al., 2024); superpoint methods organize context around coherent regions (Landrieu & Simonovsky, 2018; Damien et al., 2023). PointGroup and SoftGroup group pointwise evidence into instances (Jiang et al., 2020; Vu et al., 2022), whereas Mask3D, SPFormer, and OneFormer3D decode masks with queries (Schult et al., 2023; Sun et al., 2023; Kolodiazhnyi et al., 2024). OpenMask3D adds open-vocabulary features, and Open3DIS uses multi-view 2D mask guidance (Takmaz et al., 2023; Nguyen et al., 2024). We adopt established perception components with broad region support for parts and fine point/voxel support for handles; our contribution is their coupling.

Parts, affordances, and interaction. PartNet supplies hierarchical part annotations, SAPIEN provides articulated simulation assets, and PartSLIP transfers image-language knowledge to low-shot part segmentation (Mo et al., 2019; Xiang et al., 2020; Liu et al., 2023b). Where2Act predicts push/pull actionability; GAPartNet connects actionable part categories to cross-category perception and manipulation (Mo et al., 2021; Geng et al., 2023). We predict an interaction description, not an action policy. A handle's rotation/translation label describes the part it operates, motivating contextual correction when local handle appearance is ambiguous.

Articulation from static and changing observations. Shape2Motion estimates movable parts and motion from static 3D shapes; ANCSH recovers articulated pose and joints from a depth point cloud (Wang et al., 2019; Li et al., 2020). OPD and OPDMulti predict openable parts and motion from single images (Jiang et al., 2022a; Sun et al., 2024). Ditto instead uses observations before and after interaction, while PARIS uses two articulation states (Jiang et al., 2022b; Liu et al., 2023a). We infer motion from a static scene using learned part-handle evidence and explicit planar/upright priors, without a separate motion regressor or observed state changes.

Reconstruction and geometric articulation priors. Real2Code reconstructs articulated objects and generates joint code; Articulate-Anything uses a vision-language model to construct articulated assets (Zhao et al., 2025; Le et al., 2025). S2O adds openable parts and interiors to static container

meshes (Iliash et al., 2026); its released rule already selects box edges opposite a geometry-derived handle proxy. We do not claim this primitive as new. REACT3D targets scene-to-simulation recovery with articulation and completion (Huang et al., 2026). Independently detected scene handles guide our motion decoding, while part-associated proposals and standalone part classes improve the handle output. Comparisons retain the differences in inputs and training conditions.

3 METHOD

SEGMENT–SNAP separates learning to locate parts and handles from reasoning about their physical relationship. Three independently trained predictors provide final movable-part instances, dense handle detections, and additional part-associated handle proposals. Their outputs meet in two directed transfers: dense-handle locations guide a training-free decoder of part motion, while joint-model proposals augment handle coverage and standalone part classes refine their labels. The decoder has no learned motion head; only the perception networks are trained. We first define these prediction interfaces, then describe their learning and the two transfers. Fig. 2 shows the complete workflow.

3.1 PROBLEM FORMULATION AND PREDICTION INTERFACE

The input is a static cloud of N points, $X = \{(\boldsymbol{x}_n, \boldsymbol{c}_n, \boldsymbol{n}_n)\}_{n=1}^{N}$, where $\boldsymbol{x}_n$, $\boldsymbol{c}_n$, and $\boldsymbol{n}_n$ are the 3D position, measured RGB, and surface normal of point n. Coordinates are expressed in an upright scene frame. Each output movable part has an instance mask, confidence, rotation/translation class, unit motion axis, and representative origin. Each handle has a mask, confidence, and the motion class of the part it operates. The objective is to predict these quantities from the observed scan, without motion sequences, action execution, or completed object geometry.

Three networks process the same input using the same backbone architecture but separate parameters and independent training. Their outputs are coupled only at inference:

$$f_{\text{part}}(X) = \mathcal{P}, \qquad f_{\text{dense}}(X) = \mathcal{H}_d, \qquad f_{\text{joint}}(X) = (\mathcal{Q}, \mathcal{H}_c). \tag{1}$$

The standalone parts $\mathcal{P} = \{(M_i, s_i, y_i)\}$ supply *all final movable-part masks and classes*: M_i is a set of input-point indices, $s_i \in [0, 1]$ its confidence, and $y_i \in \{\text{rotation}, \text{translation}\}$ its motion class. They also provide the surfaces for geometric fitting and the context for handle-label correction. Dense handles $\mathcal{H}_d$ are both initial handle detections and spatial cues for motion.

The joint predictor supplies additional handles $\mathcal{H}_c$ associated with its *own* parts $\mathcal{Q}$. We call each such part and its handle a parent-child pair. Both are predicted from a common part-query feature within this network; the child is not predicted from a thresholded parent mask. Each parent or handle prediction carries a mask, confidence, and class. Although $\mathcal{P}$ and $\mathcal{Q}$ describe the same kind of movable part, their roles differ: $\mathcal{P}$ determines final part outputs and contextual labels, whereas $\mathcal{Q}$ associates and initializes the additional handle proposals. The joint model neither takes $\mathcal{P}$ as input nor replaces it. No matching between the two part sets is required; standalone parts influence children through spatial containment.

3.2 LEARNING BROAD PARTS AND FINE INTERACTION REGIONS

A movable surface and its operating region require different spatial support. Superpoint grouping offers a compact representation of a broad door, but a handle occupying only a few points can disappear under majority pooling. We therefore use instance queries for parts and fine point/voxel masks for handles. Each network has its own Volt-B voxel Transformer initialized from ScanNet++ pretraining (Yilmaz et al., 2026), with 2 cm voxels and RGB/normal features. This common backbone supports different prediction heads; the contribution lies in combining their outputs, not in a new backbone.

Standalone part instances. An SPFormer decoder (Sun et al., 2023) attends to features pooled over graph-partitioned superpoints (Felzenszwalb & Huttenlocher, 2004). Its 200 queries are learned instance features, each predicting a part mask, rotation/translation class, and confidence score. Hungarian assignment supervises the queries using classification, mask binary cross-entropy (BCE), Dice, and score terms. At inference, each standalone query emits its highest-scoring class before instance filtering. This yields $\mathcal{P}$ without treating different class hypotheses from one query as different parts. The network does not regress motion axes or origins.

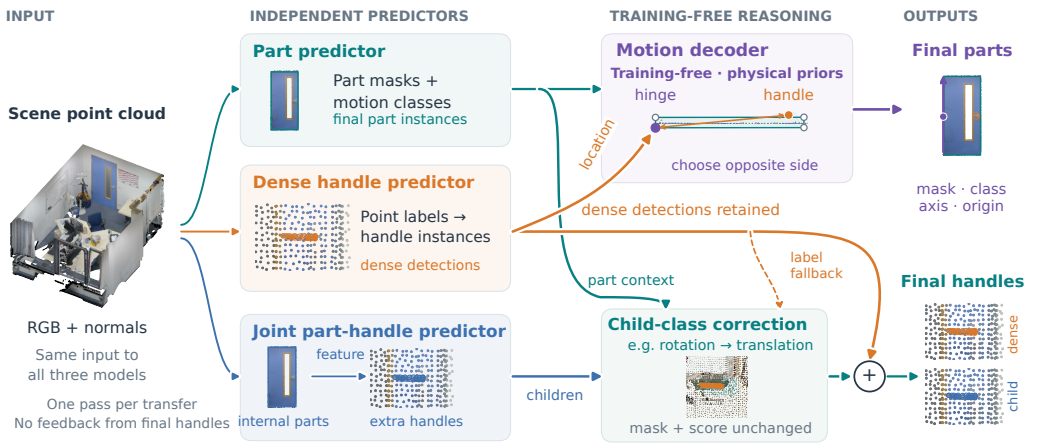


Figure 2: **Three predictors, two directed transfers.** The same RGB point cloud and normals enter independently trained part, dense-handle, and joint part-handle predictors. Standalone parts supply all final part masks and classes. Dense handles provide locations to a training-free geometric decoder, which fits part surfaces, applies axis priors, and selects hinges opposite the handles. The joint predictor uses its own part features to propose additional handles. Standalone parts correct child classes, with dense labels as fallback, before the children are appended to unchanged dense detections. The final handles do not feed back into motion decoding.

Dense handle instances. A separate network predicts background, rotation-handle, and translation-handle probabilities at each point. Class-weighted cross-entropy with label smoothing and multiclass Lovász loss supervise the field (Berman et al., 2018). Maximum-probability labels are grouped by class using a 2.5 cm-radius graph. Components with at least three points become instances, scored by their mean assigned-class probability. The small minimum size and fine support retain candidates that a part-scale grouping could suppress. These detections remain unchanged in the final handle set.

Joint parent-child prediction. The joint network first decodes its own part-query features from its backbone. Each query $\boldsymbol{q}_j$ then feeds a parent-part head and a child-handle head in parallel. The heads share this instance feature within the joint network; the child head does not wait for, crop by, or consume a thresholded parent mask. Instead, it compares the part-query feature with fine-resolution voxel features. For voxel v with feature $\boldsymbol{f}_v$, the child-head output p_{jv} is the probability of belonging to query j's handle:

$$p_{jv} = \mathrm{sigmoid}\big(g_q(\boldsymbol{q}_j)^\top g_v(\boldsymbol{f}_v)\big)\,, \qquad \mathcal{L}_{\mathrm{child}} = \mathcal{L}_{\mathrm{BCE}} + \mathcal{L}_{\mathrm{Dice}} + 2\mathcal{L}_{\mathrm{Tv}}. \tag{2}$$

Here g_q, g_v project query and voxel features to a common dimension. The child loss combines BCE, Dice, and Tversky ($\mathcal{L}_{\mathrm{Tv}}$) terms. Parent-only Hungarian assignment determines the ground-truth part for each query; the child target is its associated interactable region. A voxel is positive if it contains any target child point. Tversky false-positive and false-negative weights are 0.7 and 0.3 (Salehi et al., 2017), penalizing the spread of a small handle over its parent surface. Parent and child heads are trained together within this third network, independently of the standalone networks. Training and selection details are in Supp. Sec. A.

3.3 Handles to Parts: Geometric Motion Decoding

A part surface constrains plausible motion but may leave the attachment side ambiguous. We separate this inference into reliable support fitting, a physical axis prior, and handle-guided hinge selection. In particular, a door's extent supplies candidate hinge locations, while an associated handle distinguishes the two sides. The decoder assumes approximately planar surfaces and upright rotational axes; it applies fixed geometric rules, not a learned model of arbitrary articulation.

Reliable support and a local frame. Detached points can distort a box fit even when the predicted mask remains useful. For each M_i, we fit geometry only to its largest connected component S_i under a 5 cm-radius graph. The output mask and class remain unchanged. We also recalibrate confidence

to $\bar{s}_i = s_i|S_i|/|M_i|$, where $|\cdot|$ counts points, downweighting fragmented predictions. Fitting affects geometry; rescoring affects ranking.

We estimate a plane from S_i by principal component analysis (PCA) and fit a minimum-area rectangle over the projected convex hull. This supplies orthonormal directions $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$ and extents $\ell_1 \geq \ell_2 \geq \ell_3$. We anchor the candidate construction at the mean $\boldsymbol{\mu}_i$ of the points used for fitting, not at the rectangle's bounding-box center. Fitting uses at most 20,000 deterministically sampled support points. Suppressing part indices on the local directions and extents, we assign the unit motion axis

$$\hat{\boldsymbol{a}}_i = \begin{cases} \boldsymbol{e}_z, & y_i = \text{rotation}, \\ \boldsymbol{b}_3, & y_i = \text{translation}. \end{cases} \tag{3}$$

Here $\boldsymbol{e}_z = (0, 0, 1)^\top$ is the scene's vertical unit vector. For an upright door, the hinge direction is fixed in that frame. For a drawer, the front's thin direction approximates translation normal to the surface; its representative origin is $\boldsymbol{\mu}_i$. The remaining handle-guided steps apply only to rotation-class parts. A handle cue selects a rotational origin; it cannot repair an incorrect axis prior.

Associating a handle and constructing candidates. For each dense-handle instance, we compute its point centroid. For part i, we select the single centroid $\boldsymbol{h}$ with the smallest distance to its support, where $d(\boldsymbol{h}, S_i) = \min_{n \in S_i} \|\boldsymbol{h} - \boldsymbol{x}_n\|_2$, and accept it only if $d(\boldsymbol{h}, S_i) < 0.5\,\text{m}$. Association uses handle location, not confidence or rotation/translation label: geometric usefulness and handle classification accuracy are distinct properties.

Let r, s index the two box directions least aligned with $\hat{\boldsymbol{a}}_i$ in absolute dot product. Four anchors define candidate lines parallel to that axis:

$$\boldsymbol{v}_{\alpha\beta} = \boldsymbol{\mu}_i + \tfrac{\alpha \ell_r}{2} \boldsymbol{b}_r + \tfrac{\beta \ell_s}{2} \boldsymbol{b}_s, \qquad \alpha, \beta \in \{-1, 1\}. \tag{4}$$

Index these anchors as $\boldsymbol{v}_k$, $k = 1, \ldots, 4$; candidate k is the line through $\boldsymbol{v}_k$ in direction $\hat{\boldsymbol{a}}_i$. For a thin door they represent two attachment sides, with a front/back pair on each. These mean-anchored, box-derived lines need not coincide with literal rectangle edges.

Selecting the hinge. On many doors the handle lies away from the hinge. We therefore use $\boldsymbol{h}$ to select the candidate line farthest from the handle, then project the support mean onto that line to obtain a representative origin. Let $\Pi_i = I - \hat{\boldsymbol{a}}_i \hat{\boldsymbol{a}}_i^\top$ project onto the plane perpendicular to the axis, where I is the 3×3 identity. The selection and projection are

$$k^* = \arg\max_k \|\Pi_i(\boldsymbol{h} - \boldsymbol{v}_k)\|_2, \qquad \hat{\boldsymbol{o}}_i = \boldsymbol{v}_{k^*} + \hat{\boldsymbol{a}}_i \hat{\boldsymbol{a}}_i^\top (\boldsymbol{\mu}_i - \boldsymbol{v}_{k^*}). \tag{5}$$

The origin is a representative point on a hinge line, not a uniquely identified physical pivot. Without an accepted handle it falls back to $\boldsymbol{\mu}_i$. Translation evaluation has no origin clause. Opposite-handle reasoning has precedent in S2O (Iliash et al., 2026); here its evidence comes from an independently detected scene handle. All fitting and selection operations are fixed. Consequently, the *motion decoder* is training-free even though the perception networks supplying its inputs are learned.

3.4 Parts to handles: complementary proposals and labels

The reverse transfer has two distinct steps. The joint predictor uses its own part features to propose handles (Sec. 3.2); standalone part predictions then provide a separate source of evidence for their motion labels. We preserve the association that produced each proposal and correct only its class, leaving the original dense detections unchanged.

Preserving parent-child association. After filtering and reordering the joint model's parent detections, we retain each surviving parent's original query index and use it to retrieve the corresponding child field. Association therefore follows the shared query, not output position or a new spatial match. The child mask contains points whose voxel probability exceeds 0.30. It is neither split into components nor clipped to its parent, so an imperfect parent boundary need not truncate an otherwise localized handle. Its initial class and score come from that parent, with confidence scaled by 0.05. Children are appended as additional detections, not merged with dense masks; the original dense masks, classes, and scores are retained.

This union tests whether an additional, correctly associated source recovers missing handles. It does not by itself isolate query conditioning as the cause of the gain; the matched conditioning and association controls are separated in Supp. Sec. B. No monotonic-AP guarantee is assumed for arbitrary added proposals.

Table 1: Articulate3D validation results (%). All baseline results are taken from the USDNet benchmark paper (Halacheva et al., 2025). Origin and axis columns add the corresponding motion gates to part AP_{50}.

Method	Part AP_{50}	+Origin	+Axis	+Origin +Axis	Handle AP_{50}
SoftGroup	32.7	18.5	21.5	17.7	14.5
Mask3D	39.1	24.4	33.8	19.3	30.2
USDNet	41.8	31.4	34.6	25.0	**31.1**
SEGMENT–SNAP	**47.93**	**43.11**	**43.75**	**40.98**	30.99

Semantic correction from containing parts. A child can be well localized but inherit the wrong motion class from its joint-model parent. The standalone parts supply a separate contextual prediction. For child mask H, an eligible part satisfies

$$\frac{|H \cap M_i|}{|H|} \geq 0.9, \qquad \bar{s}_i \geq 0.3. \tag{6}$$

Here $\bar{s}_i$ is the component-rescaled confidence from Sec. 3.3. We first select the highest-confidence eligible part and assign its class to the child. Only if no part qualifies do we consult dense handles, using the one with the largest child containment and requiring at least 90% containment. If neither source qualifies, the child keeps its initial parent label. A qualifying part takes priority even when its class already agrees with the child; the fallback is not an equal vote between sources. Only child classes change; their geometry and confidence, and every dense detection, remain fixed.

Final outputs and inference schedule. The final part output is $\{(M_i, \bar{s}_i, y_i, \hat{\boldsymbol{a}}_i, \hat{\boldsymbol{o}}_i)\}$, and the final handle output concatenates $\mathcal{H}_d$ with the class-corrected children. Each information transfer is applied once. Dense handles guide hinges; standalone parts guide child labels. Decoded axes and origins are not inputs to class correction, and the final handle union is not fed back into motion decoding. This directed computation makes each source of evidence independently testable.

4 EXPERIMENTS

We compare against published baselines, evaluate articulated-part outputs with a common protocol, and isolate both transfers using fixed-input controls.

4.1 PROTOCOL AND PUBLISHED COMPARISON

Perception models are trained on Articulate3D's 195 training scenes and evaluated on its public 42-scene validation split (Halacheva et al., 2025). Model and hyperparameter selection consulted validation data, so these are development-set comparisons. We report macro AP at IoU 0.5 for parts and handles. Axis-gated part AP also requires sign-invariant axis error below 15°. Joint motion AP (AP_{50}^{AO}) additionally tests rotational origins: the predicted origin must lie within 0.25 m of the annotated axis line, and vice versa (Supp. Sec. A). Translations have no origin clause. The stricter Euclidean-origin variant is reported separately. Scores are percentages; changes are percentage points (pp).

Tab. 1 compares our reference with SoftGroup (Vu et al., 2022), Mask3D (Schult et al., 2023), and USDNet, using the baseline results reported in the USDNet benchmark paper. SEGMENT–SNAP reaches 47.93 part AP and 40.98 motion AP, compared with USDNet's 41.8 and 25.0, while handle AP is comparable (30.99 versus 31.1). These published scores provide context; direct comparisons and fixed-input controls follow. Protocol details are in Supp. Sec. B.

4.2 SAME-SCENE, COMMON-OUTPUT COMPARISON

We run REACT3D (Huang et al., 2026) on all 42 validation scenes and evaluate the same output: part instances on the evaluator's point cloud, each with a motion class, axis, and origin. Its predicted meshes are mapped to the cloud at a 2 cm nearest-neighbour tolerance, with axes normalized before evaluation. Three scenes with verified zero detections remain in the evaluation as empty prediction sets. REACT3D receives RGB-D keyframes, camera poses, and a mesh; our method receives an RGB point cloud. The training conditions also differ: our predictors use the Articulate3D training

Table 2: Articulated-part results on all 42 validation scenes (AP, %). REACT3D receives richer inputs, including RGB-D keyframes, camera poses, and a mesh, whereas SEGMENT–SNAP uses an RGB point cloud. Both are evaluated in the same point-cloud domain. The joint +Origin +Axis metric follows Tab. 1, with class-specific results also shown.

Method	Input	Part AP_{50}	+Origin +Axis All	+Origin +Axis Rotation	+Origin +Axis Translation
REACT3D	RGB-D, poses, mesh	14.46	8.09	12.29	3.89
SEGMENT–SNAP	RGB point cloud	47.93	40.98	56.37	25.59

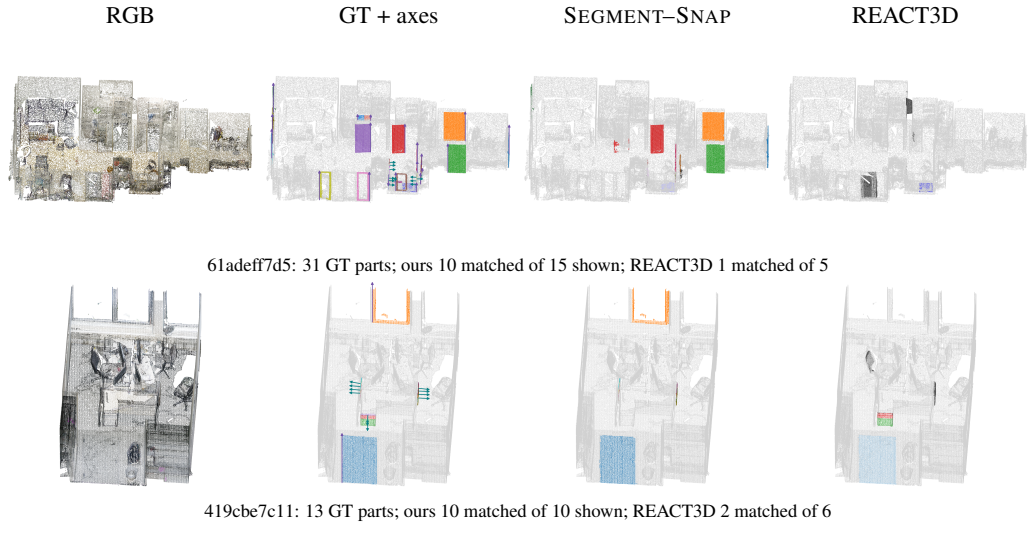


Figure 3: **Comparing part geometry and motion.** Columns show the RGB scene, ground-truth parts and axes, SEGMENT–SNAP, and REACT3D. Purple/teal axes denote rotation/translation. Saturated ground-truth colors mark correct masks and motion; pale colors pass IoU > 0.5 but fail a motion gate; gray predictions are unmatched. Missing colors reveal missed parts. These two favorable scenes are selected by the displayed true-positive advantage (Supp. Sec. E); our display threshold is $s \geq 0.50$, and all REACT3D predictions are shown. Supp. Fig. S9 reports every scene.

split, whereas REACT3D is not trained on it. Full protocol and development-scene overlap are documented in Supp. Sec. E.

Under this common output protocol, SEGMENT–SNAP achieves 40.98 motion AP versus 8.09 (Tab. 2); the paired scene-jackknife interval for our 32.89-pp difference is [18.49, 47.29]. This measures scene-sampling uncertainty, not training variability. Coverage explains much of REACT3D's translation deficit: 98 of 154 translating parts have no overlapping prediction, whereas all seven mask-matched translations pass the axis gate. Among 61 mask-matched rotations, 18 fail only the origin test. The comparison evaluates the systems' outputs under differing input and training conditions, not the superiority of one input modality.

Qualitative comparison. Fig. 3 compares part extents and motion in shared views. In the first scene, REACT3D assigns incorrect motion types and produces unmatched detections; the second also exposes extent and motion-parameter errors. Our displayed predictions recover more valid part-motion instances but still miss annotated parts. These favorable examples follow a disclosed ranking; Supp. Fig. S9 shows every scene. The paired galleries in Supp. Sec. D separately isolate hinge changes, child-only handle recovery, and localization failures that label correction cannot repair.

Inference cost. Tab. 3 measures one complete inference pass per scene for articulated-part output. Our path includes superpoint extraction, the standalone part and dense-handle networks, and motion decoding; the third predictor for additional handles is not required for this output. REACT3D's path starts from extracted keyframes, poses, and mesh and traverses its 13 stages. Median time is 33.78 s versus 3,638.1 s; peak host memory is 4.70 versus 22.94 GiB. Hardware, memory-logging coverage, inclusion rules, and the output-preserving memory refit are detailed in Supp. Sec. E.

Table 3: Inference cost for articulated-part output, measured from each system's input through one complete pass per scene, including model loading. Timing boundaries, hardware, and memory-measurement coverage are specified in Supp. Sec. E.

Measurement	SEGMENT–SNAP	REACT3D
Median time (s/scene)	33.78	3,638.1
Peak host memory (GiB)	4.70	22.94
Logged GPU peak (MiB)	10,262	8,780
Learned models	2	5
Parameters (B)	0.201	5.398

Table 4: Two directed transfers on validation. Scores are AP (%); Δ is the change in percentage points, computed before rounding. Both class-correction rows start from the same proposal union, so their gains are not additive.

Controlled change	Before	After	Δ
Handles → parts: motion-gated AP			
Centroid → handle-guided origin	13.74	**40.98**	+27.25
Parts → handles: handle AP$_{50}$			
Dense field → append associated children	24.63	29.65	+5.01
Proposal union → parts-only class correction	29.65	30.63	+0.98
Proposal union → full contextual correction	29.65	**30.99**	+1.34

The origin intervention fixes masks, scores, classes, and axes (part $AP_{50} = 47.93$, axis-gated AP$= 43.75$). Class correction fixes child masks and scores and preserves all dense detections. The full rule adds a dense-label fallback to part context.

4.3 WHAT COUPLING CONTRIBUTES

Holding predicted part masks, labels, scores, and axes fixed, replacing centroid origins with handle-guided hinge lines raises motion AP from 13.74 to 40.98 (+27.25 pp). Part and axis-gated AP remain 47.93 and 43.75. Thus, the gain isolates hinge placement, not segmentation or axis prediction.

The reverse transfer adds complementary handle evidence. Appending associated children raises dense-handle AP from 24.63 to 29.65 (+5.01 pp). With masks and scores fixed, parts-only labels yield 30.63 (+0.98 pp), and the full rule including dense fallback reaches 30.99 (+1.34 pp). These two label gains overlap; they are not additive. Spatially permuting child locations gives no gain over 24.63, while corrupting part labels reduces corrected AP to 28.62 or 28.90. Thus, localization and class evidence both matter. The proposal gain persists across child-model repeats; contextual correction is smaller and more variable (Supp. Sec. B).

Source robustness and decoder controls. The same dense cue improves motion AP from 12.26 to 37.33 with joint-model parts and from 5.86 to 25.42 with a weaker random-initialization configuration. Four dense-model draws retain proposal and context gains (Supp. Tabs. S4 and S5). Matched learned motion heads also benefit from handle evidence. Their point estimates remain below the fixed rule, but the strongest selection heads' paired intervals include zero; we do not claim statistically resolved superiority over learned selection (Supp. Tabs. S8 and S9).

5 DISCUSSION AND CONCLUSION

SEGMENT–SNAP uses handle locations to disambiguate hinges, and part-associated proposals and classes to improve handle detection. The largest controlled gain comes from hinge-side evidence, not segmentation. Learned decoder controls show that this cue helps multiple decoder families; their overlapping uncertainty does not establish a universal advantage for fixed geometry.

The method remains limited by perception coverage and its planar, upright priors. In a first-failure census of 390 ground-truth parts, 139 lack a matching predicted mask, compared with 14 axis failures and 14 origin failures. Very small handles and non-vertical mechanisms remain difficult. We evaluate on one development dataset, and validation-informed model selection limits broader generalization claims. Within this scope, explicit geometric and semantic coupling provides an effective interaction description without training a motion regressor.

REPRODUCIBILITY STATEMENT

The supplementary appendix specifies the perception models, losses, training schedule, instance extraction, geometric decoder, contextual rule, metric variants, and selection protocol. Main-paper controls fix the inputs not under intervention. Supplementary tables distinguish variability across training runs from paired scene-level uncertainty. The qualitative galleries disclose their selection criteria and retain failure cases.

ETHICS STATEMENT

This work uses an existing research dataset of indoor scans. Interaction predictions can be wrong, especially for small handles or mechanisms outside the geometric priors, and should not be treated as safety guarantees for physical interaction. We do not evaluate autonomous execution on real objects. Use of the data and derived assets remains subject to the original dataset's terms.

STATEMENT OF AI ASSISTANCE

AI tools were used to aid and polish manuscript writing and presentation. The authors take full responsibility for the final content.

REFERENCES

Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4413–4421. IEEE, 2018.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2432–2443. IEEE, 2017.

Robert Damien, Raguet Hugo, et al. Efficient 3d semantic segmentation with superpoint transformer. In *ICCV*, pp. 17195–17204, 2023.

Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. Scenefun3d: Fine-grained functionality and affordance understanding in 3d scenes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14531–14542. IEEE, 2024.

Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

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.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024.

Anna-Maria Halacheva, Yang Miao, Jan-Nico Zaech, Xi Wang, Luc Van Gool, and Danda Pani Paudel. Articulate3d: Holistic understanding of 3d scenes as universal scene description. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5633–5644. IEEE, 2025.

Zhao Huang, Boyang Sun, Alexandros Delitzas, Jiaqi Chen, and Marc Pollefeys. React3d: Recovering articulations for interactive physical 3d scenes. *IEEE Robotics and Automation Letters*, 2026.

Denys Iliash, Hanxiao Jiang, Yiming Zhang, Manolis Savva, and Angel X Chang. S2o: Static to openable enhancement for articulated 3d objects. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6785–6795. IEEE, 2026.

Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X Chang. Opd: Single-view 3d openable part detection. In *European Conference on Computer Vision*, pp. 410–426. Springer, 2022a.

Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4866–4875. IEEE, 2020.

Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5606–5616. IEEE, 2022b.

Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20943–20953. IEEE, 2024.

Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567. IEEE, 2018.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In *International Conference on Learning Representations*, volume 2025, pp. 17578–17602, 2025.

Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3706–3715, 2020.

Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. Paris: Part-level reconstruction and motion analysis for articulated objects. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 352–363. 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.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

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.

Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6813–6823, 2021.

Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4018–4028. IEEE, 2024.

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

Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International workshop on machine learning in medical imaging*, pp. 379–387. Springer, 2017.

Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8216–8223. IEEE, 2023.

Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3d scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2393–2401, 2023.

Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel Chang. Opdmulti: Openable part detection for multiple objects. In *2024 International Conference on 3D Vision (3DV)*, pp. 169–178. IEEE, 2024.

Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023.

Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2698–2707. IEEE, 2022.

Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3960–3969. IEEE, 2020.

Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8868–8876. IEEE, 2019.

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

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11094–11104. IEEE, 2020.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12–22. IEEE, 2023.

Kadir Yilmaz, Adrian Kruse, Tristan Höfer, Daan de Geus, and Bastian Leibe. Volume transformer: Revisiting vanilla transformers for 3d scene understanding. *arXiv preprint arXiv:2604.19609*, 2026.

Chenyangguang Zhang, Alexandros Delitzas, Fangjinhua Wang, Ruida Zhang, Xiangyang Ji, Marc Pollefeys, and Francis Engelmann. Open-vocabulary functional 3d scene graphs for real-world indoor spaces. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19401–19413. IEEE, 2025.

Mandi Zhao, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation. In *International Conference on Learning Representations*, volume 2025, pp. 668–686, 2025.

APPENDIX: SUPPLEMENTARY MATERIAL

This appendix provides additional implementation details, quantitative controls, and qualitative results for SEGMENT–SNAP. Sec. A specifies the perception models, training, and evaluation. Sec. B presents repeated-training results, uncertainty estimates, association controls, and learned motion-decoder comparisons. Sec. C analyzes detection coverage and failure modes. Sec. D visualizes the two directions of coupling, including residual failures. Finally, Sec. E details the REACT3D comparison and reports additional scenes, inference costs, and motion accuracy on every validation scene. All supplementary experiments use the same 42-scene public validation split. References to main-paper figures and tables retain their original numbers; supplementary items use an S prefix.

A IMPLEMENTATION AND EVALUATION DETAILS

We specify the reference validation configuration and the implementation of the two coupling directions.

A.1 INDEPENDENT PERCEPTION MODELS

SEGMENT–SNAP uses three independently trained networks with the same RGB point cloud and surface normals as input. Each uses a Volt-B voxel Transformer initialized from ScanNet++ pretraining (Yilmaz et al., 2026), with 2 cm voxels and $5 \times 5 \times 5$ voxel patches. The *standalone part predictor* uses an SPFormer decoder (Sun et al., 2023) over graph-partitioned superpoints (Felzenszwalb & Huttenlocher, 2004). Its 200 queries predict a movable-part mask, rotation/translation class, and confidence; Hungarian matching uses classification, binary cross-entropy, Dice, and score terms. These predictions, denoted $\mathcal{P}$, provide every final movable-part instance.

The *dense handle predictor* emits a pointwise background/rotation-handle/translation-handle field. It is trained with class-weighted cross-entropy, label smoothing, and multiclass Lovász loss (Berman et al., 2018). At inference, class-wise radius-graph components (2.5 cm radius, at least three points) become handle instances, scored by mean assigned-class probability. These dense instances $\mathcal{H}_d$ both remain in the final handle set and provide the geometric cue for motion decoding.

The third, *joint part-handle predictor*, decodes its own parent parts $\mathcal{Q}$ and a fine child-handle mask for each retained parent query. It does not replace $\mathcal{P}$. A parent head and a child head read the same query features; the child head does not consume the thresholded parent mask. For query $\boldsymbol{q}_j$ and voxel feature $\boldsymbol{f}_v$, the probability that voxel v belongs to its child is

$$p_{jv} = \mathrm{sigmoid}\big(g_q(\boldsymbol{q}_j)^\mathsf{T} g_v(\boldsymbol{f}_v)\big)\,, \tag{7}$$

where g_q and g_v are learned projections to a common feature dimension. Parents alone receive Hungarian assignment. Child supervision uses binary cross-entropy (BCE), Dice, and twice a Tversky loss with false-positive and false-negative weights 0.7 and 0.3, respectively (Salehi et al., 2017); a voxel is positive when it contains any target child point. This preserves small supports that majority pooling could erase.

A.2 DETERMINISTIC COUPLING

For each standalone part mask M_i, we retain its largest 5 cm-radius connected component S_i only for fitting, and rescale its confidence s_i to $\bar{s}_i = s_i|S_i|/|M_i|$, where $|\cdot|$ counts points. The output mask and class are unchanged. Geometry fitting uses all support points up to a cap of 20,000, with deterministic subsampling above that cap. Their mean is $\boldsymbol{\mu}_i$. A PCA plane and an in-plane minimum-area rectangle supply orientation and extents; the anchor remains $\boldsymbol{\mu}_i$, not the rectangle's bounding-box center. A rotational part uses the vertical unit vector $\boldsymbol{e}_z = (0, 0, 1)^\top$; a translating part uses the box's thinnest direction. The decoder contains no learned motion regressor, although its masks and handle locations are learned.

For a rotational part, the nearest dense-handle centroid is accepted when it lies within 0.5 m of S_i. Four candidate lines run parallel to the predicted axis through anchors obtained by adding signed half-extents to $\boldsymbol{\mu}_i$ along the two least-aligned box directions. The selected line is farthest from the handle in perpendicular distance; the origin is the projection of $\boldsymbol{\mu}_i$ onto that line. Without an

accepted handle, the origin is $\boldsymbol{\mu}_i$; translations use the same mean. For numerical stability in the evaluator, emitted axes receive a deterministic 10^{-6}-radian perturbation and are renormalized after geometry decoding. The construction assumes approximately planar surfaces and upright rotational axes; it does not model arbitrary mechanisms.

The joint branch retains child voxels above probability 0.30 and retrieves each child by its source-query index after parent filtering. Children are appended—not mask-merged—to dense handles with their parent class and a score equal to 0.05 times the parent confidence. Only an appended child's class can then change. Eligible standalone parts contain at least 90% of the child and have calibrated confidence at least 0.3. The highest-confidence eligible part assigns its class to the child, even if the existing label already agrees. Only when no part qualifies does the dense handle with largest containment, at least 90%, supply a fallback class. Absent both cues, the initial child label remains. Masks and scores remain fixed, and dense detections are never relabelled. Dense-handle centroids guide part origins once; joint-model parents supply child proposals, and standalone parts supply child-label context. Final handles are not fed back into motion decoding.

A.3 Optimization, selection, and metrics

All three networks are optimized separately with AdamW (Loshchilov et al., 2017) for 400 epochs, peak learning rate 3×10^{-4}, cosine one-cycle scheduling, batch size two with eight-step gradient accumulation, and EMA decay 0.999. Weight decay is 0.1 for part and joint models and 0.05 for dense handles. We use spatial crops, rotations, scaling, and color augmentation; the dense model additionally uses flips, jitter, and elastic distortion. Its foreground targets are dilated by 0.10 m for the first 50% of training, 0.04 m for the next 30%, and not dilated for the final 20%. This is a configuration choice, not a claimed contribution.

The part model is selected by validation motion-gated AP at 70% of training; the dense branch uses a segmentation-selected checkpoint and the joint branch its final weights. Consequently, validation results are development-set evidence rather than an untouched generalization estimate. We report macro AP at IoU 0.5. Axis-gated AP additionally requires sign-invariant axis error below 15°. For the primary joint metric ($\mathrm{AP}_{50}^{\mathrm{AO}}$), a rotation must pass two origin checks: the predicted origin's distance to the annotated axis line and the annotated origin's distance to the predicted axis line must both be below 0.25 m. Translations have no origin clause. All axes are normalized before these projections. The evaluator also exposes a stricter Euclidean-origin variant, which checks full representative-origin displacement. These two conventions are reported distinctly and are not interchangeable.

B Controls and Uncertainty

Published comparison versus released models. All baseline values in main-paper Tab. 1 are quoted from the USDNet benchmark paper, including its implementations of SoftGroup and Mask3D. Its released validation list matches our 42 scenes; these remain published evaluation runs rather than predictions re-scored by us, and training resources are not matched. The publication does not disambiguate the origin-distance implementations: our joint AP is 40.98 with perpendicular line distance and 38.99 with the stricter Euclidean-origin gate. We do not rerun the released USDNet checkpoints or include them in qualitative comparisons because the public test-submission workflow uses combined training and validation data.[1] This workflow observation is not a claim about the training protocol behind the published benchmark numbers. The uncertainty analyses below concern our own fixed-input validation comparisons.

Tab. S1 separates repeated-training variation from paired scene uncertainty. The jackknife brackets are lower and upper bounds for the AP *difference* in percentage points, not intervals for an absolute score or ranges across seeds. The large handle-to-origin intervention and proposal-union change remain positive in their reference scene intervals. The smaller contextual-vote effect is positive for the reference configuration but varies appreciably across child draws; it should not be read as a universal effect size.

[1] Official USDNet release, pinned training and inference configuration.

Table S1: Observed variation of the directed effects on validation. Reference gains and brackets are AP changes in pp. "Seed span" is the maximum minus minimum paired gain across training draws, not a confidence interval.

Mechanism	Reference gain	Seed span	Paired 95% scene CI
Dense handles → hinge origin	$+27.25$	2.11 (n=4)	$[+21.86, +32.63]$
Children appended to dense handles	$+5.01$	1.08 (n=3)	$[+1.47, +8.56]$
Contextual child-label vote	$+1.34$	1.15 (n=3)	$[+0.37, +2.31]$

The child draws give append gains of 5.01/4.89/3.94 pp and vote gains of 1.34/0.19/0.38 pp. The proposal interval compares the reference union with dense detections alone; the vote interval compares the contextual output with that union.

Table S2: Controls for complementary handle proposals and semantic correction (handle AP_{50}, %). Dense detections are fixed in all rows.

Configuration	Handle AP_{50}
Proposal localization; no class correction	
Dense detections only	24.63
Append spatially permuted child masks	24.63
Append random size-matched child masks	24.63
Append query-associated child masks	29.65
Class evidence; the real proposal union is fixed	
No class correction	29.65
Parts-only correction	30.63
Dense-only correction	30.66
Full contextual correction	**30.99**
Part classes permuted within scene	28.62
Part classes randomized	28.90

Spatial controls preserve child sizes and scores but disrupt localization; semantic controls fix child masks and scores but corrupt part labels. Parts-only and dense-only gains overlap and are not additive. The real union retains all 140 dense-incumbent GT matches and adds 53 matches, preserving the incumbent TP/FP ranking prefix.

The spatial controls in Tab. S2 preserve the number, size, and confidence of child candidates but remove their localization; they recover no AP beyond dense handles. With child masks and scores fixed, corrupting part classes lowers the contextual result below the uncorrected union. These are localization and semantic-evidence controls, respectively.

B.1 Conditioning versus association

Table S3: Conditioning and association are separate comparisons. Scores are handle AP (%). Panel (a) holds the shared-field readout fixed when changing conditioning. Panel (b) holds a different dense prediction field fixed and changes child association; its absolute scores are not the reference chain.

Predictor / readout	Proposal union	After correction
(a) Parent conditioning with matched shared-field readout		
Unconditioned child head	28.07	29.78
Conditioned child head	28.41	29.26
(b) Parent-child association on an alternate dense-model basis		
Output-position association	27.72	29.18
Query-index association	28.96	30.37

In (a), conditioned minus unconditioned is +0.34 pp before correction (95% CI $[-0.99, +1.66]$) and -0.52 pp after ($[-2.66, +1.62]$). In (b), all 443 dense instances are identical. Query and position association emit 4,129 and 3,317 child instances; among 1,691 shared distinct child masks, 1,688 receive different scores. The 1.24-pp union gain reflects association and ranking, not newly learned mask geometry.

Tab. S3 does not identify parent conditioning alone as the cause of the proposal gain: the matched shared-field contrast changes sign after class correction, and both paired intervals include zero. The separate association control improves AP while fixing the dense instances. A shared child keeps its geometry but can inherit a different parent's confidence; association also changes which children survive parent filtering. The result therefore concerns emission and ranking, not re-detection. The alternate dense-model basis is kept separate from the reference 29.65/30.99 chain.

B.2 Source strength and repeated training

Table S4: Coupling under different perception configurations. Scores are validation AP (%); the complementary cue is fixed within each block. The random-initialization configurations also use longer training schedules, so these rows do not isolate pretraining.

Prediction source	Without cue	With cue
Motion AP: fixed dense handles, varying part predictor		
Standalone reference parts	13.74	40.98
Joint-model part head	12.26	37.33
Random-initialization part configuration	5.86	25.42
Handle AP: fixed child proposals, varying dense predictor		
Reference dense model	24.63	29.65
Random-initialization dense configuration	1.75	13.63

Table S5: Repeated dense-handle training with part and child predictions fixed. All scores are AP (%). The first three columns measure handles; the last measures part motion using that row's dense field as the geometric cue.

Dense-model draw	Dense handles	+ Children	+ Context	Part motion
Reference	24.63	29.65	30.99	40.98
Repeat 1	23.76	27.68	28.81	40.66
Repeat 2	22.83	27.67	28.90	40.46
Repeat 3	25.16	29.70	30.82	39.51

Tab. S4 shows the geometric cue with three part sources and the complementary children with two dense sources. Guidance improves each listed configuration, including weaker perception. The random-initialization rows also change training duration, so they do not estimate an isolated pretraining effect. Tab. S5 instead holds the part and child predictors fixed across four dense-model training draws. Adding children and applying context improve handle AP in every row. The highest dense-handle AP does not give the highest motion AP: hinge selection depends on local handle position rather than the entire mask-and-class ranking. These repeats complement, rather than pool with, the child-seed variation in Tab. S1.

B.3 Geometric components and learned alternatives

Tab. S6 tests components both when added to a naive configuration and when removed from the full decoder. Naive decoding uses the stock query selection, unfiltered geometric support, centroid origins, and uncalibrated confidence. The component effects do not add: cleanup is slightly negative alone but costs 2.95 pp when removed from the full system, since it changes the box from which handle-guided candidates are constructed. The full-minus-naive gain is 28.74 pp. Among 183 rotations jointly covered by the geometric variants, cleanup leaves origin error unchanged for 120 and changes it by more than 0.25 m for 25; the effect is concentrated rather than uniform. Fig. S3 visualizes the largest and median changes.

Learned-decoder protocol. The heads use frozen part-query features, the fitted box frame, and gravity-alignment features; handle-aware variants additionally receive the same predicted-handle offset used by the rule. They train on predictions from the 195 training scenes and are evaluated on the 42 validation scenes without changing masks, classes, or scores. Three seeds vary initialization

Table S6: Geometric components measured from both ends of the pipeline. Values are motion-AP changes in pp. "Add" inserts one component into the naive configuration; "Remove" is the full decoder minus the result after removing it. Their difference diagnoses non-additivity, not a unique causal interaction.

Component	Add gain	Removal loss	Difference
Per-query class selection	+0.13	+0.19	+0.06
Largest-component fitting support	−0.11	+2.95	+3.07
Dense-handle origin cue	+23.79	+27.25	+3.45
Connectivity-based confidence	+1.49	+1.60	+0.11
Full versus naive	+28.74; 95% CI $[+22.12, +35.37]$		

Naive and full motion AP are 12.24 and 40.98. Differences use unrounded scores. The bracket is a paired scene-jackknife interval for the full-minus-naive AP difference, not an absolute-score interval. Support fitting and handle-origin changes preserve the output masks.

Table S7: Fixed-input origin control (AP, %). Part masks, classes, scores, and axes are identical; only the representative-origin rule changes.

Origin rule	Part AP_{50}	Axis-gated	Origin-gated	AP_{50}^{AO}
Centroid (no handle cue)	47.93	43.75	14.37	13.74
Handle-guided hinge	47.93	43.75	43.11	**40.98**

Table S8: Matched learned heads on two frozen part-model draws (joint line-distance AP, %). Brackets are paired 95% scene-jackknife intervals for learned-head minus training-free-rule AP (pp); each includes zero.

Decoder	Reference part model	Second part-model draw
Training-free rule	40.98	39.45
Learned candidate selection	38.63 $[-5.35, +0.63]$	38.63 $[-4.54, +2.91]$
Learned selection, box axes only	38.78 $[-5.33, +0.92]$	39.39 $[-4.44, +4.31]$
Rule axis + learned edge	39.36 $[-4.31, +1.05]$	37.64 $[-4.66, +1.04]$

Each learned row is the best of three head seeds by 42-scene validation motion AP for its backbone, with predicted-handle evidence. Checkpoints within each seed are selected by held-out training-fold loss. This compares frozen-feature decoders, not jointly trained representations.

Table S9: Matched decoder ranges on frozen inputs (joint line-distance AP, %). Ranges span learned-head seeds; handle evidence is identical within each decoder family.

Decoder	No handles	Predicted handles
Learned regression	17.24–18.24	29.61–31.30
Learned candidate selection	25.38–28.32	37.27–38.63
Fixed axis + learned edge	25.36–28.75	37.69–39.36
Training-free geometric rule	13.74	**40.98**

The training-free no-handle entry is the centroid-origin control; all rows retain the same part masks ($AP_{50} = 47.93$). Paired intervals for the strongest selection-based heads versus the rule include zero (Tab. S8); the ranges alone do not establish a universal or statistically resolved ordering.

and a 20% held-out training-scene fold. Within each seed, the checkpoint minimizes loss on that fold. Best-seed summaries then report the highest validation motion AP among the three seeds; these are descriptive best-of-three comparisons, not independently selected estimates. Optimization uses AdamW for 1,500 full-batch epochs with cosine scheduling, learning rate 10^{-3}, and weight decay 10^{-4}. Continuous regression predicts axis and origin; selection chooses axis and edge candidates; the fixed-axis variant learns only edge choice. This tests decoding from a fixed representation, not end-to-end retraining.

Tab. S7 confirms that dense handles enter the primary motion comparison only through origins: mask and axis AP are exactly unchanged. Tab. S8 uses frozen decoder features and the same predicted-handle evidence within each family. Learned alternatives have lower observed point estimates on both part-model draws, but all displayed paired intervals include zero. The evidence therefore supports a useful training-free decoder here, not a statistically resolved or general advantage over learned heads.

C Coverage and Failure Accounting

Table S10: Coverage and first-failure diagnostics on validation. Candidate coverage and selection are not end-to-end AP.

Population / diagnostic	n	Coverage	Selection given coverage
Movable parts matched by predicted mask	390	64.4%	–
Matched rotations: passing hinge candidate	1,423	94.4%	88.3%
GT rotations: passing hinge candidate	236	95.8%	88.1%
GT handles matched by child union	388	49.7%	–

First movable-part failure	Rotation	Translation	Total
No mask with IoU > 0.5	50	89	139
Covered; axis gate fails	12	2	14
Covered; axis passes; origin fails	14	0	14
All motion gates pass	160	63	223
GT population	236	154	390

The handle-union coverage by ground-truth-size quartile is 22/97, 53/99, 63/96, and 55/96, from smallest to largest. Counts are descriptive and do not represent additive AP gains.

Candidate availability and candidate selection are separate diagnostics. The matched-rotation population may contain more than one prediction for a ground-truth instance, whereas the ground-truth-mask diagnostic uses one mask per annotated rotation. Thus, the high candidate-coverage fractions in Tab. S10 are not macro AP ceilings. The first-failure census instead shows where end-to-end instances are lost: missing movable-part masks dominate, especially for translations. On handles, the joint union covers 193 of 388 ground-truth handles; its smallest size quartile contains at most 11 points and has the lowest coverage. These are descriptive populations, not additive oracle gains.

The error magnitudes further distinguish these failures. Uncovered rotations have median best-mask IoU 0.30, versus 0.10 for translations. All 12 rotation-axis failures involve non-vertical hinges; 12 of the 14 origin failures lie between 0.26 and 0.41 m, near the 0.25 m gate. Of the 75 smallest-quartile handle misses, 38 have no overlapping prediction. Conversely, 37 of the 96 largest-quartile handles have an overlapping prediction at least twice their size. These diagnostics explain why improving hinge placement cannot by itself remove the coverage bottleneck.

D Additional Qualitative Results

The supplementary galleries complement the main-paper scene comparison with successful hinge selections, missing-handle and non-upright-axis failures, recovered child handles, and contextual class corrections. They are intended to make the coupling mechanisms inspectable, not to estimate their prevalence. The compact comparison in Fig. S1 fixes masks and cameras while changing the

origin rule, and contrasts a newly recovered handle with an unmatched child whose label correction cannot repair its extent. The larger galleries below state each selection population.

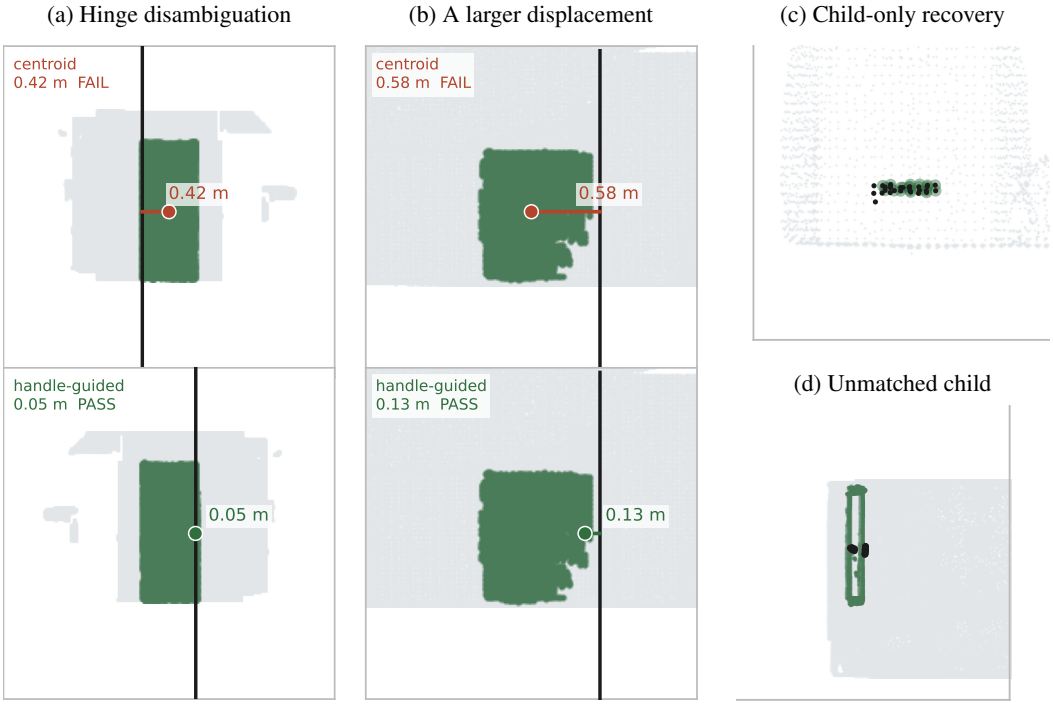


Figure S1: **What coupling changes.** (a,b) Top: centroid origins; bottom: handle-guided origins on the same predicted masks. Black lines are annotated hinges; colored segments show perpendicular origin error. The examples are median and 90th-percentile centroid errors among corrected rotations. (c) A child proposal recovers a handle missed by the dense field (black: ground truth; green: prediction). (d) A class-corrected child still has poor spatial overlap: class evidence cannot repair its mask. The complete galleries and selection populations appear in Sec. D.

Hinge-origin cases. Fig. S2 expands the main comparison to cases in which both rules pass, the vertical-axis prior fails, no handle is available, or the guided origin still has a large residual. These cases distinguish evidence failures from failures of the motion assumptions.

Support cleanup and the hinge cue. Fig. S3 connects the component ablations in Tab. S6 to their geometric action. Removing detached support can stabilize the fitted candidates for some parts, while many parts are unchanged. A good candidate frame alone still leaves the centroid-origin ambiguity.

Failure mechanisms. Fig. S4 visualizes the first-failure categories summarized in Sec. C. A missing mask cannot be repaired by motion decoding, and a non-vertical hinge cannot be corrected by choosing another vertical line. Small and over-extended handle predictions fail for different reasons despite both reducing handle recall.

Changing the part source. Fig. S5 illustrates the source-variation scores in Tab. S4. Each centroid-versus-guidance pair uses the same predicted mask and dense-handle field. Across rows, the part source changes, so the figure does not claim a fixed-mask comparison between networks.

Complementary handle predictions. Fig. S6 shows dense successes, child-only recoveries, and the limitation of class-only correction. The final unmatched child illustrates why label-correction accuracy on matched instances cannot be extended to every relabeled proposal.

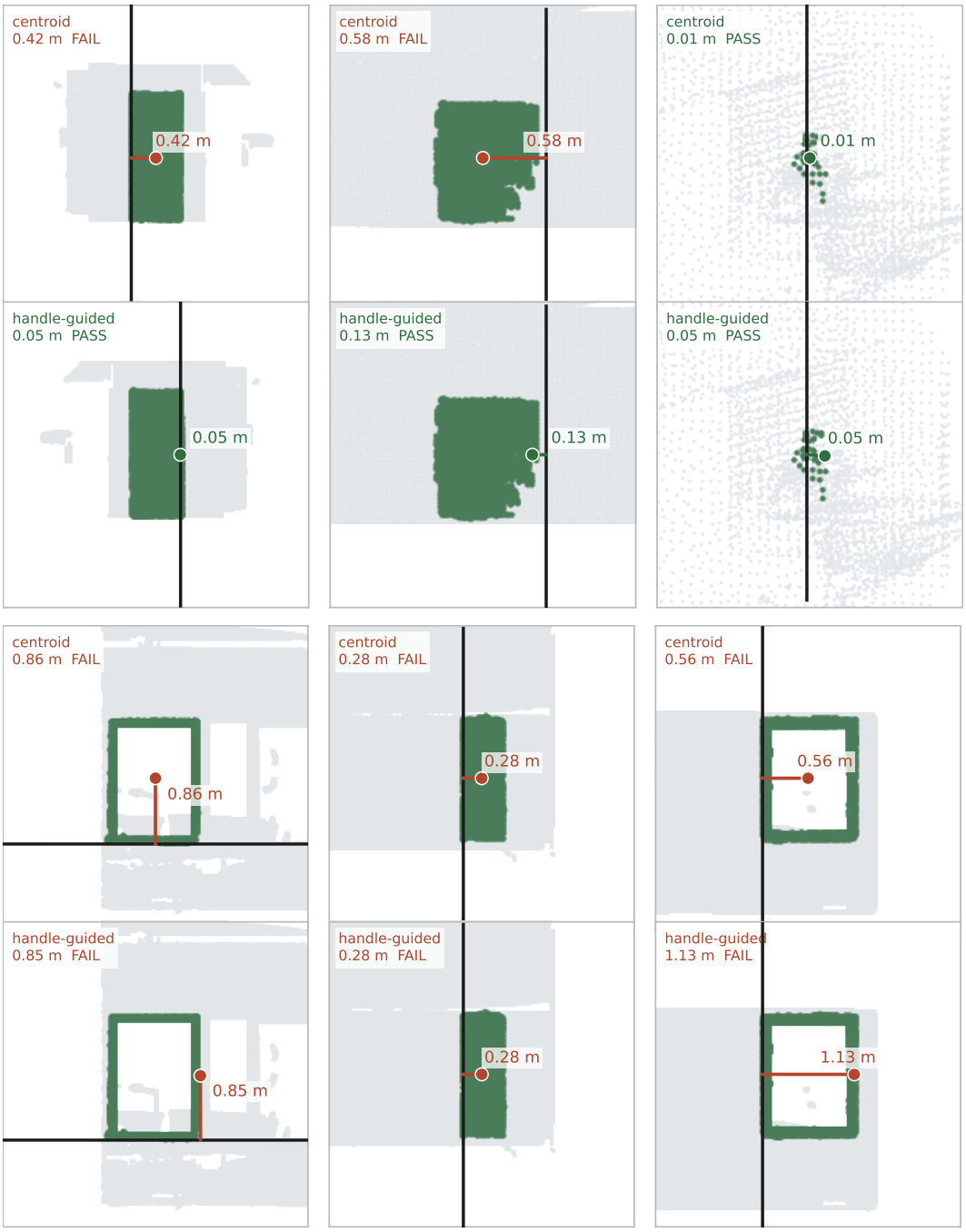


Figure S2: **Hinge successes and failures.** Within each pair, the top row uses a centroid origin and the bottom row uses handle guidance; masks and cameras are fixed. Upper group, left to right: centroid-error ranks 61 and 109 among 121 rotations that pass with guidance but fail with centroid origins, and the smallest centroid error among 37 both-pass cases. Lower group: median axis error among 12 non-vertical hinges, median origin error among 19 missing-handle fallbacks, and the largest guided-origin error among 186 covered rotations. Black lines denote annotated hinges. No case in this 186-instance population passes with the centroid but fails with guidance; the last column is a residual failure, not a reversal example.

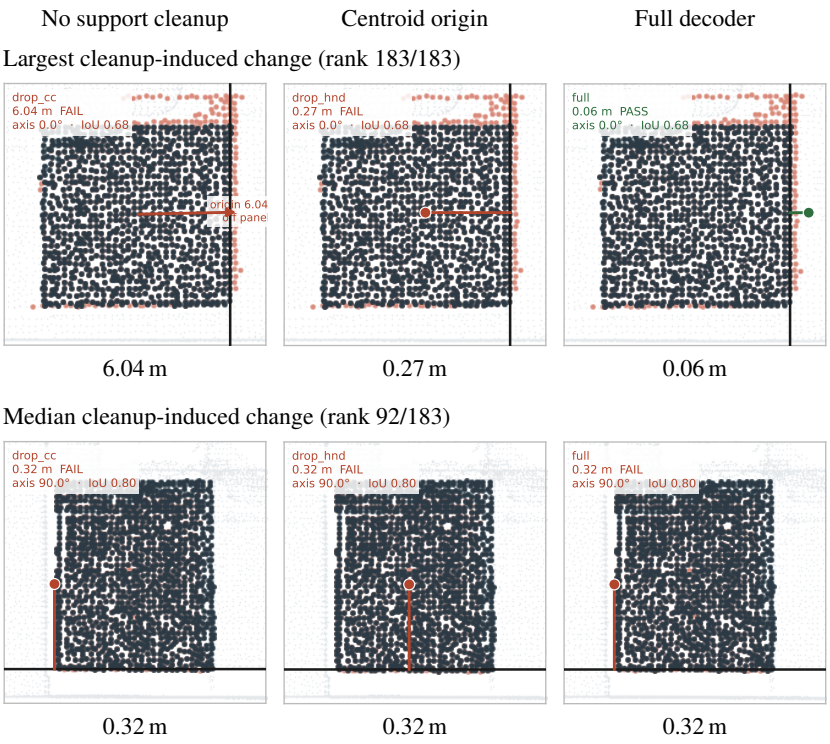


Figure S3: **Where geometric components matter.** Rows fix the part and camera. Dark points are ground truth, pale red the prediction, and black the annotated hinge; values are perpendicular origin errors. The upper case has the largest cleanup-induced change among 183 jointly covered rotations (its uncleaned origin is outside the view). The median-change case below is unchanged and has a horizontal hinge, violating the upright-axis prior.

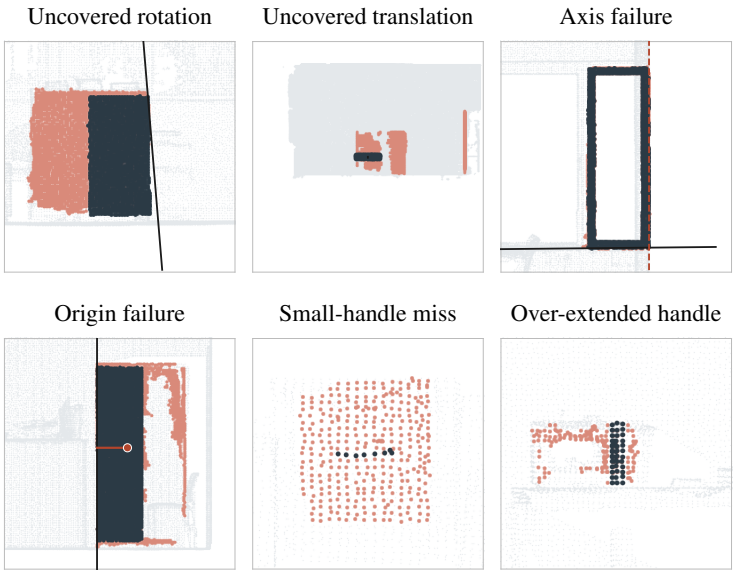


Figure S4: **Distinct sources of error.** Dark points are ground truth, pale red the prediction, and black the annotated hinge. Median-rank cases show uncovered rotations (25/50), uncovered translations (45/89), axis error (7/14), origin error (7/14), smallest-quartile handle misses (38/75), and over-extension (19/37). Frequencies come from the census, not this gallery.

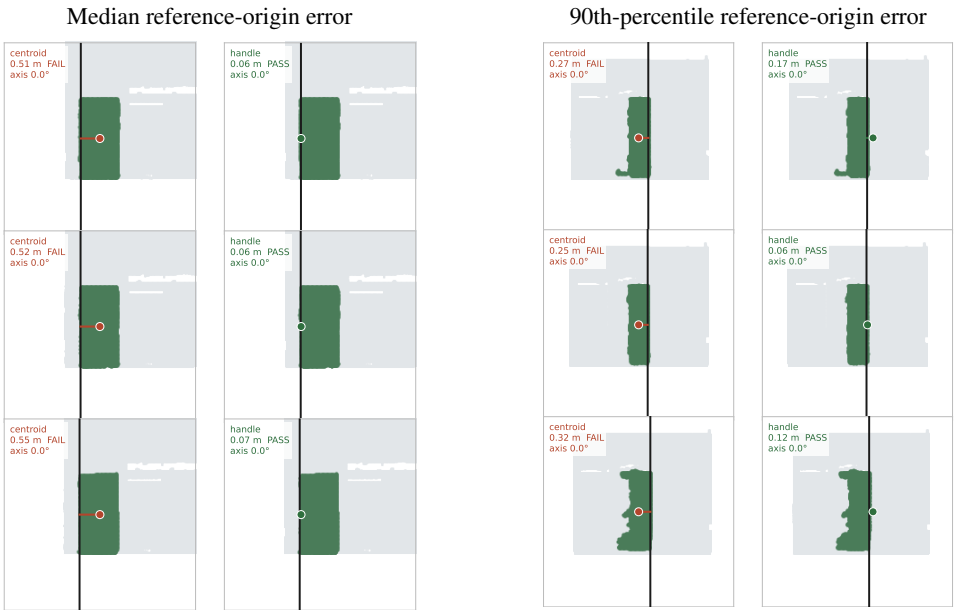


Figure S5: **Handle guidance with different part predictors.** Rows use reference, joint-model, and random-initialization parts. Each pair changes only the centroid origin (left) to dense-handle guidance (right); black marks the annotated hinge. Specimens have reference-origin-error ranks 61 and 110 among 122 rotations covered by all three sources. Guidance helps across sources. Random initialization also changes training duration and does not isolate pretraining.

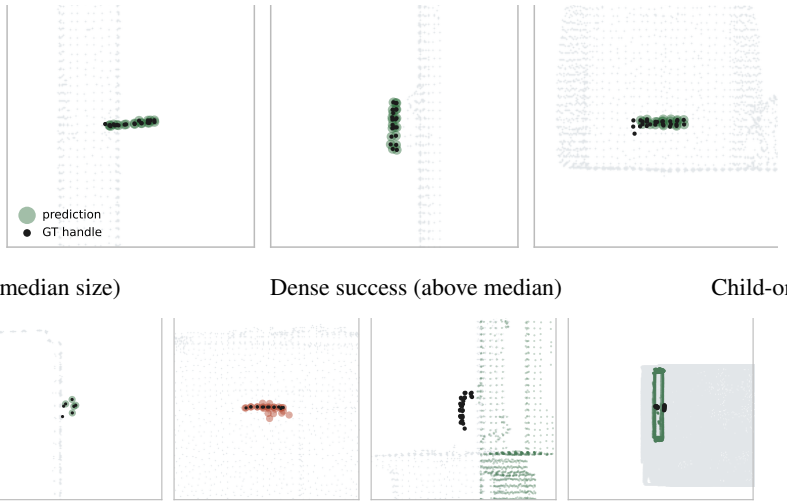


Figure S6: **Handle proposal and class behavior.** Black points are annotated handles; colors show predictions. The top row includes two dense successes and the median-size child-only recovery. Below: the smallest recovery, largest dense over-extension ratio, matched class correction, and an unmatched corrected child. Correction cases have median annotated size in their respective strata. Selections are deterministic, not random.

E EXTERNAL SYSTEM COMPARISON

We run REACT3D as an external baseline for articulated-part output on the same 42 validation scenes. Unlike the published-number comparison, REACT3D is evaluated by us with the same evaluator as SEGMENT–SNAP. Its pipeline consumes richer inputs—RGB-D keyframes, camera poses, and a scan mesh—instead of our RGB point cloud. Both outputs are expressed as part instances with a motion type, axis, and origin. Input and training conditions are not controlled comparison axes: REACT3D is not trained on Articulate3D, whereas our perception models use its training split. Nevertheless, 20 of the 42 validation scenes occur in REACT3D's authors' development lists and cannot be described as unseen during its development.

We use REACT3D's released processing stages with the compatibility changes required by the available hardware and data representation. A PyTorch API-compatibility patch to its MSDeformAttn dispatch preserves the kernel and numeric type. Data preparation follows the released image format and frame schedule. For evaluation, reconstructed part surfaces are transferred to the evaluation point cloud by a reverse nearest-neighbour rule: a cloud point belongs to a part when its nearest part vertex lies within 0.02 m. This expresses the predictions on the point-labelled evaluation domain. We normalize every predicted axis before scoring because raw REACT3D axis vectors are not unit length; without normalization, the evaluator's origin projection is invalid even though its axis-angle test is unchanged. These compatibility and conversion steps do not imply byte-identical execution; the released reference output was checked for compatible part, class, axis, and origin behavior.

All 42 scenes were scored; none was omitted. Three scenes that completed the pipeline but produced no accepted part were certified from their stage logs and entered the evaluator as empty prediction sets. REACT3D produces 199 parts for 390 annotated parts. The diagnostic in Tab. S11 shows that its low translation result is principally a coverage issue: only 7 of 154 translation parts have a same-class mask match, while 98 are untouched and 48 are touched but below the IoU gate. All seven matched translations pass the axis gate. For rotations, 18 of 61 mask-matched parts fail only the origin gate, separating this residual motion error from the larger detection deficit. The paired scene comparison is descriptive of this fixed run: SEGMENT–SNAP has higher motion AP on 40 of the 41 scenes containing a ground-truth part and ties on one; one scene contains no ground-truth part.

Table S11: REACT3D coverage and motion diagnostics on all 42 validation scenes. The four mask-outcome rows partition the ground-truth parts in each motion class. Counts describe coverage, not AP contributions.

Outcome	Rotation	Translation
Ground-truth parts	236	154
Predicted parts	155	44
Same-class mask match (IoU > 0.5)	61	7
No overlapping prediction	158	98
Overlap below the mask gate	16	48
Above-gate overlap, wrong class	1	1
Mask-matched parts passing motion gates	39	7

The below-gate row combines weak overlap and near-miss categories. Among 61 mask-matched rotations, 18 fail only the origin gate and four fail both motion gates. All seven matched translations pass the axis gate. The prediction counts sum to 199; the annotation counts sum to 390.

Timing and memory accounting. The measurements in Tab. 3 use one RTX 5070 Ti and 30 GB of host RAM, one scene at a time, from each system's declared input through its articulated-part predictions. Time sums each stage's last successful attempt, including model loading; failed and superseded attempts are excluded. REACT3D's input starts at extracted keyframes, poses, and mesh, so video decoding and input-frame checks are excluded as dataset preparation. Our superpoint extraction is included and uses no additional worker parallelism. This measures a complete inference pass, not the elapsed duration of the experimental campaign.

REACT3D's 22.94 GiB as-shipped host-memory peak comes from its original point-assignment stage, used on 22 scenes. The other 20 use an output-identical memory refit, with a separately measured peak of 16.04 GiB. Its GPU logger began during the run: 8,780 MiB is the maximum over

nine logged scenes, whose peaks span 7,616–8,780 MiB; the other 33 were not GPU-logged. We do not present extrapolations to those scenes as measurements. Our 10,262 MiB nvidia-smi peak covers all 42 scenes at 0.2 s polling; its separately instrumented allocated-memory peak is 7.17 GiB. Our logged GPU peak is higher, but the unequal logging coverage precludes a matched GPU-efficiency conclusion.

The SEGMENT–SNAP timing path includes superpoints, dense handles, standalone parts, and motion decoding; it excludes the joint predictor needed only for additional handle outputs. Checkpoint-derived parameter counts cover these two models (200,610,183 parameters), versus five REACT3D models (5,397,612,242). The cost comparison therefore concerns articulated-part output, not the complete three-predictor interaction system.

The timed SEGMENT–SNAP path reproduces the reference validation masks, ordering, classes, axes, and origins on all 42 scenes. Scores differ by at most 9.39×10^{-6}, with no effect on any evaluated quantity; the source of this numerical residue is unresolved.

Figure selection and complete-scene reporting. The main showcase uses the first two scenes under the final, deterministic ranking: the advantage in matched true positives among displayed predictions, restricted to scenes with at least five annotated parts, with ties broken by motion-AP margin. Our display threshold is 0.50; all REACT3D parts are displayed. Fig. S7 adds the next two ranked scenes. This is a favorable-case selection, not a representative sample. Fig. S9 therefore reports every validation scene, including the three certified empty prediction sets and the scene without ground-truth parts. Fig. S8 summarizes the measured accuracy–time trade-off under the timing and memory scopes above.

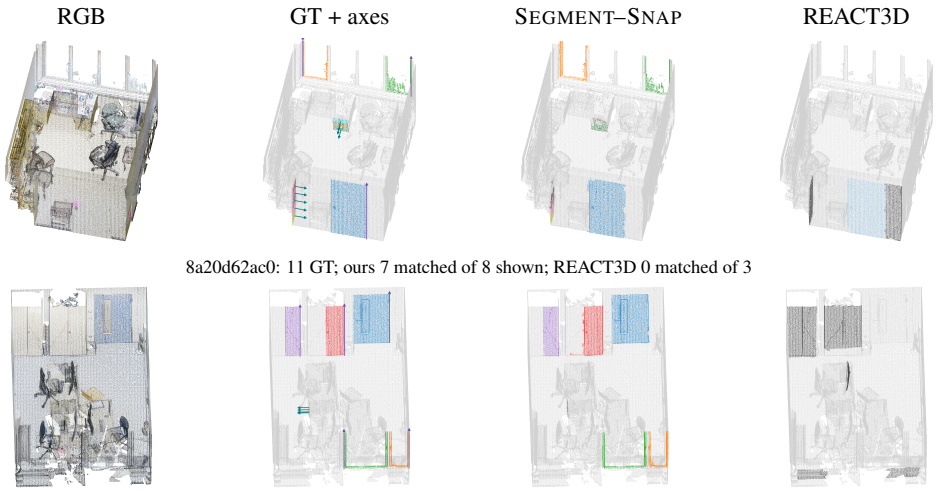


Figure S7: **Additional qualitative comparison.** The next two scenes under the ranking used in main-paper Fig. 3: at least five annotated parts, ordered by displayed true-positive advantage and then motion-AP margin. Our threshold is $s \geq 0.50$; all REACT3D predictions are shown. Saturated colors indicate correct masks and motion, pale colors indicate mask matches that fail a motion gate, and gray indicates unmatched detections. Purple/teal axes denote rotation/translation. The input and training conditions are given in Sec. E.

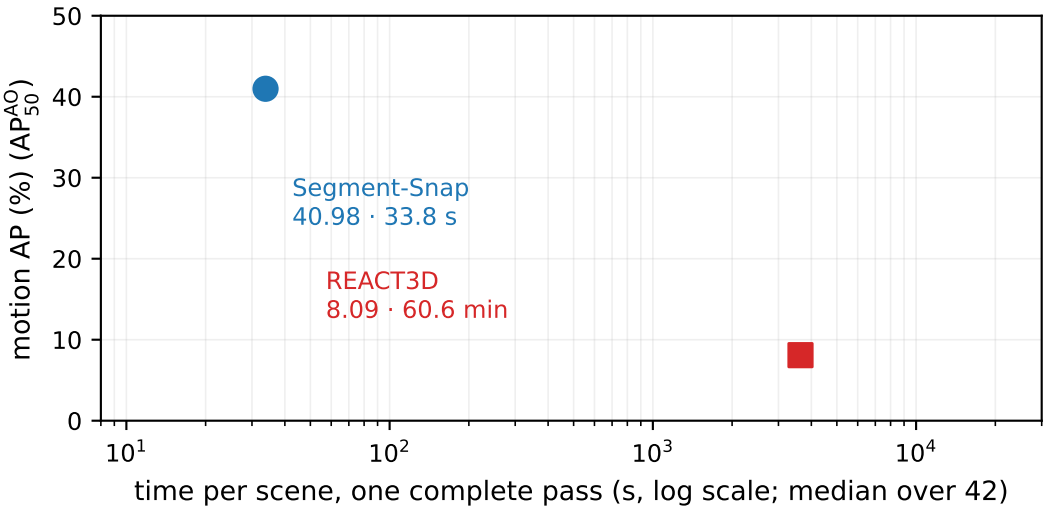


Figure S8: **Motion accuracy and inference time.** Full-validation motion AP versus median complete-pass time per scene (log scale), including model loading on the same RTX 5070 Ti. SEGMENT–SNAP's two-model part-motion path and REACT3D produce common outputs; their input, training, and timing scopes are specified in Sec. E.

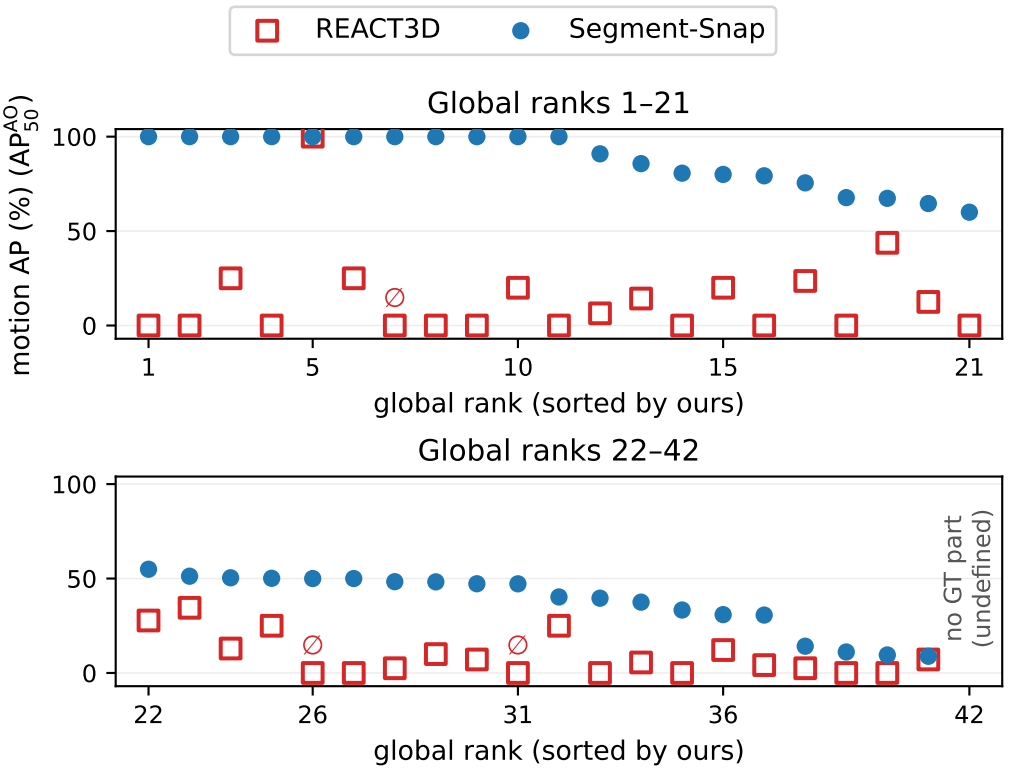


Figure S9: **Per-scene motion AP on all 42 validation scenes.** The 41 scenes with defined AP are sorted by SEGMENT–SNAP's score; ∅ marks three verified empty REACT3D outputs. The scene without an annotated part (a980334473) is appended at rank 42: its AP is undefined, not zero. SEGMENT–SNAP has higher AP on 40 scored scenes and ties on one; evaluation conditions are specified in Sec. E.