Geometric and Semantic Coupling for Interaction Understanding in 3D Scenes

Hanyang Kong[1], **Xingyi Yang**[2]

[1]National University of Singapore, [2]The Hong Kong Polytechnic University

Abstract

Interaction understanding in 3D scenes requires a joint description of movable parts, their motion, and the regions through which they can be operated. We present SEGMENT–SNAP, which connects these outputs through the physical relationship between parts and handles. Learned predictors identify broad part surfaces and small handles. A geometric decoder uses planar and upright priors to constrain motion, then selects hinge lines using predicted handle locations, without training a motion regressor. Conversely, a joint part-and-handle predictor supplies additional handle candidates, whose motion classes are refined using containing parts. Each information transfer is applied once, without iterative feedback. On Articulate3D validation, handle guidance raises motion-gated AP from 13.74% to 40.98% at fixed masks and axes. Additional handle candidates raise handle AP from 24.63% to 29.65%; part-based class correction adds 0.98 points, and full context reaches 30.99%. Repeated training, learned-decoder controls and paired visualizations establish the benefits and limitations of combining geometric and semantic evidence for interaction understanding.

Correspondence: Xingyi Yang (xingyi.yang@polyu.edu.hk)
Author email: Hanyang Kong (hanyang.k@u.nus.edu)
Date: September 9, 2026
Keywords: 3D interaction understanding, part-handle coupling, part segmentation, motion estimation
Project Page: https://hyokong.github.io/segment-snap-page/

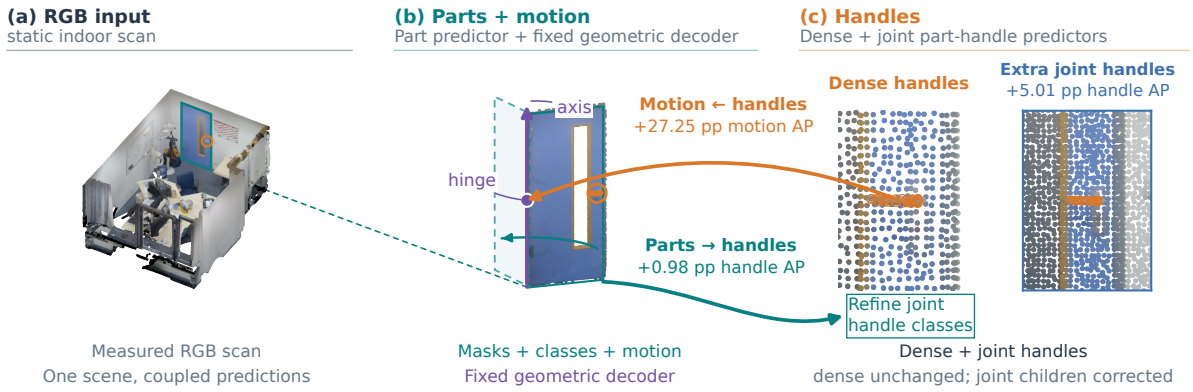

Figure 1. Coupled part and handle understanding. From a static RGB scene **(a)**, we recover movable parts and their motion **(b)** together with handles **(c)**. **Handles guide motion** (orange arrow): handle locations select the opposite hinge side, improving motion AP by **27.25 points**. **Parts refine handles** (teal arrow): part motion classes correct the joint predictor's handle labels, improving handle AP by **0.98 points** with part-only context. The joint predictor also uses its own internal part features to propose extra handles (blue, **+5.01 points**). Full class correction additionally consults dense-handle labels, gaining 1.34 points instead of 0.98. Only the added handles' classes change; dense detections remain unchanged. Each transfer runs once, with no feedback from final handles to motion. Gains are validation-wide (Table 2); the dashed door indicates rotation.

1 Introduction

Understanding a 3D scene for interaction requires more than recognizing its objects. For a cabinet, we need to identify which surfaces are movable, where they can be operated, and how they move: does a panel rotate about a hinge or translate like a drawer? Recent benchmarks formulate these questions as the prediction of movable parts, their motion parameters, and small interactable regions from static scene observations [3, 5]. We call these interactable regions *handles*, including annotated operating regions that need not have a conventional handle shape. Our goal is to recover this interaction structure from an RGB point cloud, without observing motion or executing an action.

Two ambiguities make this task difficult. The first concerns scale and context: a door occupies a broad surface, whereas its handle may contain only a few observed points. Grouping points into large regions helps delineate the door but can absorb its handle into the surrounding surface. Conversely, a small handle crop may reveal where to interact without revealing whether the associated part rotates or translates. The second ambiguity concerns motion geometry. Even a well-segmented, closed door may be compatible with a hinge on either side. Its surface constrains the possible motion but does not, by itself, identify the attachment side. Better part masks alone therefore need not yield better motion estimates, just as accurate handle locations alone do not determine their motion class.

Existing methods address these issues through different forms of scene representation. USDNet jointly predicts movable parts, interactable regions and motion, combining part-instance features with dense pointwise motion predictions [5]. However, its motion decoder does not explicitly use the detected handle locations to choose a hinge. Functional scene graphs represent which interactive element controls which object [15], but this semantic association does not specify a metric motion axis or origin. Object-reconstruction methods infer articulation from completed geometry and geometric rules [7, 16], under different input and reconstruction assumptions. We investigate a complementary route: using the predicted relationship between movable parts and handles to resolve ambiguities in both outputs.

Movable parts and handles are physically linked: a handle is attached to a part and moves with it. On many upright doors, the handle lies away from the hinge, so its position can distinguish two otherwise plausible attachment sides. In the other direction, the surrounding part gives meaning to the handle: a region on a hinged door is associated with rotation, whereas one on a drawer is associated with translation. The full part also offers a larger visual context from which to predict its small operating region. These observations motivate two distinct uses of the relationship: *handle locations guide part motion*, while *part features and motion classes guide handle prediction*.

We introduce Segment–Snap, with three independently trained predictors that receive the same scene point cloud (Figures 1 and 2). The *part predictor* segments movable surfaces and classifies each as rotating or translating; it supplies the final parts, but does not estimate their axes or hinge locations. The *dense handle predictor* classifies individual points and groups foreground points into handle instances, providing an initial set of handles. The *joint part-handle predictor* offers a complementary way to find handles: it detects its own parts, then uses each part's feature to predict an associated handle mask. Its parts serve as internal context; only its handle predictions enter the final outputs. The two handle sources thus localize operating regions in complementary ways: directly from pointwise predictions or conditioned on a predicted parent part.

We combine these predictions to recover a joint description of interaction. For *part motion*, the part predictor supplies surface geometry and a rotation/translation class, while the dense handle predictor supplies handle locations. A fixed geometric decoder fits the part surface, constructs candidate hinge lines under an upright-door prior, and uses a nearby handle to choose the opposite attachment side. For a translating part, the fitted surface's thin direction supplies the motion axis. This is why motion decoding is training-free: axes and origins are computed from predicted masks and geometric rules, not produced by a trained motion-regression head.

For *handle detection*, we retain the dense detections and add the joint predictor's handles. Each added handle initially inherits its joint-model parent's rotation/translation class. A confident containing part from the separate part predictor can correct this label, with dense-handle labels providing a fallback (Section 4.5). This changes only the added handles' classes, leaving their masks and scores, and all dense detections, unchanged. Final parts therefore come from the part predictor and geometric decoder; final handles combine both handle sources. These handles are not fed back into motion decoding.

On Articulate3D's public validation split, handle-guided hinge selection raises motion-gated AP from

13.74 to 40.98 (+27.25 percentage points) while keeping part masks, scores, classes and axes fixed. Adding the part-associated handle candidates raises handle AP from 24.63 to 29.65. Correcting their classes using parts alone adds 0.98 points; the full rule, including the dense-handle fallback, reaches 30.99. Repeated training runs support the hinge-guidance and added-proposal gains, while showing greater variability in the benefit of class correction. Fixed-input ablations, learned alternatives and qualitative comparisons examine when each source of evidence is useful for the joint scene description.

Our contributions are:

1. A part-handle coupling strategy for 3D interaction understanding: handle locations help recover part motion, while part-associated predictions and part motion classes improve handle detection.

2. A geometric motion decoder that combines predicted part support, explicit axis priors and handle-guided hinge selection, without training a motion regressor.

3. A controlled study of both information transfers, with fixed-input ablations, learned-decoder comparisons, repeated training runs and qualitative failure analysis that establish where the coupling helps and where its assumptions break down.

2 Related work

Interaction-oriented scene understanding. SceneFun3D studies fine-grained functional elements, affordance grounding and motion estimation in real scenes [3]. Articulate3D extends interaction-oriented annotation to ScanNet++ reconstructions, including movable and interactable parts and motion parameters [5, 13]. Its USDNet baseline extends query-based instance segmentation [11] with dense motion prediction. Our work shares the goal of recovering these outputs, but changes how evidence reaches the motion decoder: predicted handle instances explicitly determine hinge selection. SceneFun3D uses ARKitScenes assets and a different annotation domain; its results are not interchangeable with Articulate3D validation scores.

Articulation from geometry and reconstruction. Real2Code reconstructs and completes articulated-object parts from multiview observations, then generates executable joint code from geometric representations [16]. S2O augments static container meshes with openable parts, articulation and interior geometry [7]. Its released door operator already uses box edges opposite a geometry-derived handle proxy [9]. We build on the same physical intuition, rather than claiming the opposite-handle primitive as new: the distinctive input here is an independently detected scene handle, and the reverse part-to-handle transfer is part of the same inference system. REACT3D addresses the broader scene-to-simulation problem, combining perception, articulation recovery and completion [6]. These systems motivate geometric structure, but their reconstruction requirements differ from our fixed-scan prediction setting. Our learned-decoder experiments isolate an internal design choice on frozen features; they are not end-to-end comparisons against these pipelines.

Instance scale and functional relations. Mask3D and SPFormer provide query-based instance grouping, with SPFormer decoding over superpoint features [11, 12]. We retain this structure for broad movable surfaces and use dense point/voxel support for small interaction regions. This is a scale-specific use of existing representations, not a new backbone. OpenFunGraph predicts directed functional relations between interactive elements and objects from posed RGB-D observations [15]. Its graph makes the interaction relation explicit at a semantic level; our relation additionally determines a metric hinge and transfers a part's motion class to a handle. Metric articulation could complement such graphs as input to downstream reasoning and interaction systems, although we do not evaluate that integration here.

3 Problem formulation

We study *interaction understanding in 3D scenes*: recovering what can move, how it moves, and where it can be operated from a static observation. Given an RGB point cloud $X = \{(\boldsymbol{x}_i, \boldsymbol{c}_i, \boldsymbol{n}_i)\}_{i=1}^{N}$, with position, color and normal at every point, we seek a joint description of movable parts, their motion and their handles. These are related components of one scene-understanding task.

A movable-part prediction comprises an instance mask, confidence, rotation/translation class, unit motion axis $\hat{\boldsymbol{a}}$ and representative origin $\hat{\boldsymbol{o}}$. A handle prediction comprises an instance mask, confidence and the motion class of the associated part. Here a *handle* means the local region through which a part is operated, not necessarily a conventional mechanical handle. A rotational axis and origin specify a hinge

line; the origin is a representative point on that line, not a unique physical pivot. Translation is described by its direction. The outputs characterize possible interaction from observation; they do not include an executed action, a motion trajectory or a completed simulation asset.

The part-handle relationship supplies information in both directions. A handle's position can disambiguate a part's motion geometry, while the part's extent and motion class provide context for locating and interpreting a handle. Our method uses these complementary cues within the joint prediction problem. Dataset-specific annotation and scoring conventions are introduced with the experimental protocol in Section 5.1.

4 Method

SEGMENT–SNAP recovers interaction structure by exchanging explicit geometric and semantic evidence between movable parts and handles. We first define the prediction interface, then describe part and handle perception, handle-guided motion decoding, and the reverse transfer to handle proposals and labels. Figure 2 illustrates the operations on concrete scene geometry.

4.1 Prediction interface and information flow

Our pipeline first predicts movable parts and handles from the scene, then combines their geometric and semantic evidence using fixed rules (Figure 2). Three independently trained networks take the same point cloud X from Section 3:

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

Standalone part and handle predictions. The *part predictor* returns $\mathcal{P} = \{(M_i, s_i, y_i)\}$, where M_i is a movable-part mask, s_i its confidence, and $y_i \in \{\text{rot}, \text{trans}\}$ its motion class. These predictions supply all final movable-part instances. The *dense handle predictor* groups pointwise labels into handle instances $\mathcal{H}_d = \{(H_k^d, s_k^d, y_k^d)\}$. Their locations guide motion decoding, and their masks, scores and classes are retained in the final handle output.

Joint part-handle predictions. The *joint part-handle predictor* provides a complementary source of handles by modelling them together with their supporting parts. It processes the scene independently and decodes its own movable parts $\mathcal{Q}$ and associated handles $\mathcal{H}_c$ from shared features. The part predictions determine which handle candidates are retained and provide their initial motion labels. These parts remain internal to the joint model; the standalone part predictor supplies the final movable-part instances. The additional handles complement the dense detections, with their labels subsequently refined using standalone part context. Section 4.4 describes the architecture, training and candidate selection.

Preparing part geometry and confidence. Before coupling, we identify the largest spatially connected component $S_i \subseteq M_i$ for geometric fitting and recalibrate s_i to $\bar{s}_i$ using the mask's connectivity (Section 4.3 and Equation (4)). The fitting subset limits the influence of stray points; the confidence adjustment downweights fragmented predictions. Neither operation replaces the output mask M_i or changes its class y_i. We collect these quantities as $\overline{\mathcal{P}} = \{(M_i, S_i, \bar{s}_i, y_i)\}$. These deterministic operations prepare the part geometry and confidence for the reasoning stage in Figure 2.

Final part motion and handle instances. The training-free decoder D fits geometry to S_i using the coordinates in X, applies the motion-class axis prior, and uses dense-handle locations to select a rotational hinge line (Section 4.3). It augments each prepared part with an axis $\hat{\boldsymbol{a}}_i$ and origin $\hat{\boldsymbol{o}}_i$:

$$\mathcal{M} = D(X, \overline{\mathcal{P}}, \mathcal{H}_d) = \{(M_i, \bar{s}_i, y_i, \hat{\boldsymbol{a}}_i, \hat{\boldsymbol{o}}_i)\}. \tag{2}$$

The final part records therefore retain the original masks and classes, use calibrated scores, and add geometric motion parameters. The fitting subsets S_i are internal; they are not substituted for the output masks.

For handles, a separate fixed rule C corrects only the joint children's motion classes. It consults the masks, calibrated confidence and classes in $\overline{\mathcal{P}}$, with dense-handle labels as a fallback (Section 4.5); it does not use S_i or the decoded axes and origins. Writing $\widetilde{\mathcal{H}}_c$ for these class-corrected children, the final handle set is

$$\widetilde{\mathcal{H}}_c = C(\mathcal{H}_c; \overline{\mathcal{P}}, \mathcal{H}_d), \qquad \mathcal{H} = \mathcal{H}_d \cup \widetilde{\mathcal{H}}_c. \tag{3}$$

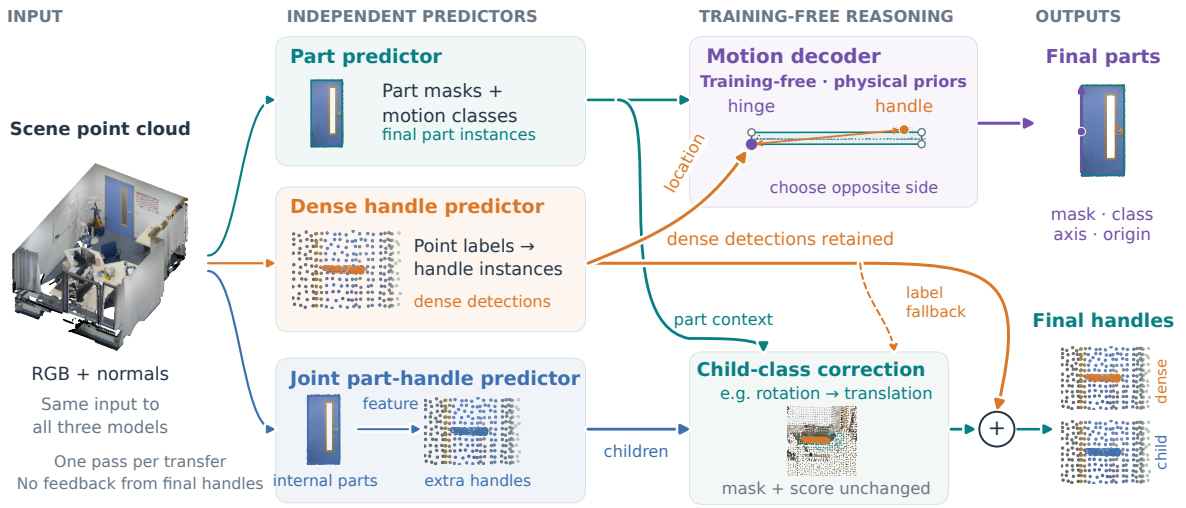

Figure 2. From scene perception to coupled interaction understanding. The same RGB point cloud and normals feed three independently trained predictors. The *part predictor* produces movable-part masks and rotation/translation classes; the *dense handle predictor* groups pointwise labels into handle instances. Their outputs enter a **training-free motion decoder**: it fits the part surface and applies physical priors, using an upright axis and the hinge line farthest from a nearby handle for rotation, or the fitted surface's thin direction for translation. No motion-regression head is trained. The top view illustrates handle-guided hinge selection; purple marks the final axis and origin. In parallel, the *joint part-handle predictor* uses its own internal part features to propose extra handles. The standalone part predictions provide class context (teal), complemented by dense-label fallback (dashed orange), to correct only these added handles' rotation/translation classes; their masks and scores stay fixed. Corrected children are concatenated with unchanged dense detections (+). Each information transfer is applied once; final handles do not feed back into motion decoding.

Correction leaves child masks and scores fixed, and the union concatenates detections rather than merging masks. Dense detections remain unchanged throughout.

These equations distinguish the two information transfers: handle locations guide hinge placement, whereas part context refines handle classes. Each transfer is applied once. The final handle set $\mathcal{H}$ is not fed back into D, and changing a decoded hinge alone cannot change handle labels. A second motion pass is evaluated only as a separate variant in Section 5.

4.2 Part and handle perception

All three components use a Volt-B voxel Transformer initialized from ScanNet++ pretraining [14]. Input features comprise measured RGB and surface normals on a 2 cm voxel grid, with $5\times5\times5$ voxel patches. Coordinates preserve the scene's upright reference frame for downstream motion decoding. We choose the prediction support according to the target scale: superpoint grouping for broad movable surfaces and fine point/voxel fields for small interaction regions.

Movable-part instances. A SPFormer decoder attends to features pooled over graph-partitioned superpoints [4, 12]. Its 200 queries predict instance masks, rotation/translation classes and confidence. Parent supervision uses Hungarian assignment with classification, mask binary cross-entropy (BCE), Dice and score terms. At inference each standalone query emits its highest-scoring class, followed by the predictor's instance filtering. This per-query selection avoids treating two class hypotheses from one query as two distinct parts. It is held fixed in motion-decoder comparisons, since changing query selection changes the mask set.

Dense handle instances. A separate network predicts pointwise probabilities for background, rotation handles and translation handles. Class-weighted cross-entropy, label smoothing and multiclass Lovász loss supervise this field [2]. We assign each point its maximum-probability class and form class-wise connected components on a radius graph with radius 2.5 cm. Components with at least three points become instances; their confidence is the mean probability of the assigned class over the component. This path does not pool handle targets by superpoint majority, and its small component threshold preserves tiny candidates. The resulting detections are both the initial handle output and the spatial evidence supplied to D.

4.3 Handles to parts: geometric motion decoding

A static planar surface constrains motion without uniquely determining it. We separate this problem into support estimation, an axis prior, and handle-guided selection of a hinge line. The decoder is training-free; the masks and handle locations on which it operates are learned predictions.

Stable geometric support. For a predicted mask M_i, we retain its largest connected component under a 5 cm radius graph as support S_i. The original M_i remains the output mask. From the support covariance we obtain a PCA plane normal, project the points into that plane, and fit a minimum-area rectangle over their convex hull. The resulting box has orthonormal directions $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$, ordered by decreasing extent $\ell_1 \geq \ell_2 \geq \ell_3$, and reference centre $\boldsymbol{c}_i$, the support-point mean. This is a planar rectangle fit, not a general minimum-volume box.

Support cleanup and confidence calibration are distinct. We use the same connected component to rescale confidence,

$$\bar{s}_i = s_i \left(\frac{|S_i|}{|M_i|} \right)^{\gamma}, \qquad \gamma = 1, \tag{4}$$

which downweights fragmented predictions without changing their extent. Cleanup affects the fitted geometry; rescoring affects ranking. Their separate and combined effects are measured in the component controls.

Motion-axis prior. The axis depends on the predicted motion class:

$$\hat{\boldsymbol{a}}_i = \begin{cases} \boldsymbol{e}_z, & y_i = \text{rot}, \\ \boldsymbol{b}_3, & y_i = \text{trans}. \end{cases} \tag{5}$$

Rotational axes therefore use a constant world-vertical prior, appropriate to many upright doors. Translation axes use the fitted thinnest direction, appropriate to drawers moving approximately normal to their front. In particular, the rotational axis is not selected from the box and is not learned. Horizontal hinges and tilted mechanisms are limitations of this prior, rather than cases resolved by the handle cue.

Associating a handle. For every dense handle instance, let $\boldsymbol{h}_k$ be its point centroid. We choose the centroid nearest the part's geometric support:

$$k^* = \arg\min_k \min_{j \in S_i} \|\boldsymbol{h}_k - \boldsymbol{x}_j\|_2. \tag{6}$$

The association is accepted only if this distance is below 0.5 m. It uses the handle's location, not its motion label. Thus the handle model's classification quality and its utility as a hinge cue need not vary together. If no handle passes the gate, the origin falls back to $\boldsymbol{c}_i$.

Selecting the hinge side. For a rotational part with an associated handle, identify the box direction most parallel to $\hat{\boldsymbol{a}}_i$. Let u, v index the other two box directions. Four anchors define axis-parallel candidate lines:

$$\boldsymbol{b}_{\sigma\tau} = \boldsymbol{c}_i + \frac{\sigma \ell_u}{2} \boldsymbol{b}_u + \frac{\tau \ell_v}{2} \boldsymbol{b}_v, \qquad L_{\sigma\tau} = \{\boldsymbol{b}_{\sigma\tau} + t\hat{\boldsymbol{a}}_i : t \in \mathbb{R}\}, \quad \sigma, \tau \in \{-1, 1\}. \tag{7}$$

These are box-derived lines, not necessarily literal box edges when the vertical axis and the box orientation differ. For a thin door, the four lines form two physical sides, with a front/back pair on each side. The handle primarily disambiguates these attachment sides.

Writing $Q_i = I - \hat{\boldsymbol{a}}_i \hat{\boldsymbol{a}}_i^{\mathsf{T}}$ for perpendicular projection, we select the candidate farthest from the handle and project the support centroid onto that line:

$$L^* = \arg\max_{L_{\sigma\tau}} \|Q_i(\boldsymbol{h}_{k^*} - \boldsymbol{b}_{\sigma\tau})\|_2, \qquad \hat{\boldsymbol{o}}_i = \boldsymbol{b}^* + \hat{\boldsymbol{a}}_i \hat{\boldsymbol{a}}_i^{\mathsf{T}} (\boldsymbol{c}_i - \boldsymbol{b}^*). \tag{8}$$

Here $\boldsymbol{b}^*$ is the selected anchor. The origin is a representative point on a hinge *line*; it is not claimed to be a unique physical pivot. Translation origins are emitted as the support centroid but are not scored by the motion-origin metric. Opposite-handle geometric reasoning has precedent in S2O [7, 9]; its role here is to transfer independently predicted scene-handle evidence into the part's metric motion estimate.

4.4 Parts to handles: complementary proposals

Dense localization and part-associated prediction expose different failure modes. The dense path identifies foreground locally, while a part query represents a larger movable surface to which a small interaction region belongs. We retain both proposal sources and evaluate their complementarity, rather than assuming one representation subsumes the other.

Parent-associated child masks. The joint model attaches a voxel-level child head to each parent query. For query $\boldsymbol{q}_j$ and voxel feature $\boldsymbol{f}_v$, learned projections g_q, g_v produce

$$p_{jv} = \sigma\big(g_q(\boldsymbol{q}_j)^{\mathsf{T}} g_v(\boldsymbol{f}_v)\big)\,. \tag{9}$$

The query carries parent context while the voxel field preserves fine spatial support. Parent-only Hungarian assignment associates a predicted query with its ground-truth movable instance; the child term does not change that assignment. A child's target consists of interactable points belonging to the assigned parent. Boolean max pooling marks a voxel positive if it contains any target child point, avoiding the disappearance of a small target under majority pooling.

For each matched query, the child loss is

$$\mathcal{L}_{\text{child}} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{Dice}} + 2\mathcal{L}_{\text{Tv}}, \qquad \mathcal{L}_{\text{Tv}} = 1 - \frac{\text{TP} + \epsilon}{\text{TP} + 0.7\,\text{FP} + 0.3\,\text{FN} + \epsilon}, \tag{10}$$

where TP, FP and FN are soft sums computed from p_{jv} and the binary target [10]. The larger false-positive weight penalizes spreading a small handle mask over its parent. This is a design motivation; the separate loss contribution is not inferred from the union gain.

Selection, association and union. The joint parent decoder uses its own top-k class-query selection and non-maximum suppression. For each retained parent we retrieve the child field by the *originating query index*, preserving that identity through reordering. The child mask contains all points whose voxel probability exceeds 0.30; it is neither split into connected components nor clipped to the predicted parent. This keeps an imperfect parent boundary from truncating a localized child. Its initial class and confidence come from that parent, with confidence scaled by 0.05 before concatenation with dense detections.

The dense masks, labels and scores are retained unchanged. In the reference outputs, appended children also rank below every dense incumbent globally within each class; the matching and ranking-prefix checks are reported in Section 5. This is a verified property of those outputs, not a general monotonic-AP guarantee for arbitrary appended proposals. The matched unconditioned experiment separately tests whether query conditioning, as opposed to the additional source and its association, explains the gain.

4.5 Parts to handles: contextual motion labels

The child inherits a motion class from its own joint-model parent, but another part prediction may provide better evidence. For a child mask H and a candidate containing mask A, define

$$r(H, A) = \frac{|H \cap A|}{|H|}. \tag{11}$$

Among standalone part predictions with $r(H, M_i) \geq 0.9$ and $\bar{s}_i \geq 0.3$, select the highest-confidence part and denote its class by y_P. If it differs from the child's initial label y_0, adopt it. Otherwise—including when no qualifying part exists—the dense incumbent with greatest containment supplies a fallback label y_D if its containment is at least 0.9. With $\varnothing$ denoting an unavailable label,

$$\tilde{y}(H) = \begin{cases} y_P, & y_P \neq \varnothing \text{ and } y_P \neq y_0, \\ y_D, & y_D \neq \varnothing, \\ y_0, & \text{otherwise.} \end{cases} \tag{12}$$

The second case is reached only when the first does not apply: agreement with a part can still permit an incumbent override. The fallback uses the original dense incumbents, not previously corrected children.

This operation changes only child labels. It does not move a handle, alter its confidence, or correct a dense detection. The part-based case assigns the child's motion class from a confident containing part. Parts-only and dense-only ablations distinguish this part-to-handle transfer from the handle-based fallback; their improvements are overlapping effects, not additive shares of the full rule.

4.6 Training and inference protocol

The three perception components are optimized independently with AdamW [8]; joint parent-child supervision occurs only within the third component. All use a 400-epoch schedule, peak learning rate 3×10^{-4}, cosine one-cycle scheduling, a batch of two with eight-step gradient accumulation, and exponential-moving-average weights with decay 0.999. Weight decay is 0.1 for the part and joint models and 0.05 for

dense handles. Training uses spatial crops, rotations, scaling and color augmentation; the dense model additionally uses flips, jitter and elastic distortion.

For dense-handle training, foreground targets are dilated by 0.10 m during the first half of training, 0.04 m during the next 30%, and zero during the final 20%. This schedule specifies the training targets; inference uses the pointwise predictions described above.

The reference part predictor is selected by validation motion-gated AP at 70% of its schedule. The dense branch uses its segmentation-selected checkpoint, while the joint child branch uses its final weights. These are different selection rules, not three instances of the same metric-based choice. Validation-based model and hyperparameter selection is disclosed in Section 5. No motion-regression loss trains the geometric decoder.

At inference, we compute standalone parts, dense handles and joint-model children, construct stable part supports, then evaluate the two outputs in Equations (2) and (3). Handles guide part motion once, and part context guides child labels once. The final handle set is an output, not a new decoder input. These explicit boundaries let the ablations change one information source while preserving the other predictions.

5 Experiments

We evaluate three questions: can predicted handles resolve part motion, can part-associated proposals recover additional handles, and can part semantics improve their motion labels? We then examine the fixed geometric decoder, training variability and the failure modes of the resulting interaction description.

5.1 Dataset and evaluation protocol

We use Articulate3D [5], whose annotations cover movable parts, motion parameters and interactable regions. All perception models are trained on its 195 training scenes, and all component studies use the 42 public validation scenes. These are development-set comparisons: model and hyperparameter choices have consulted validation data, and the results are not an untouched generalization estimate. Test annotations are withheld; only final public leaderboard metrics are reported for that split.

Metrics. For movable-part and handle instances, we report macro-averaged AP_{50} over rotation and translation, with IoU greater than 0.5 required for an instance match. For movable parts, we additionally use motion-gated average precision. A full motion match must satisfy the mask criterion and a sign-invariant axis error below 15°. For rotations, the displacement between the predicted and annotated representative origins must also be below 0.25 m when projected perpendicular to each of the predicted and annotated axes. Translation origins are not evaluated. We denote mask AP by AP_{50}, axis-gated AP by $\mathrm{AP}^{\mathrm{A}}_{50}$, the origin-only diagnostic by $\mathrm{AP}^{\mathrm{O}}_{50}$, and jointly gated AP by $\mathrm{AP}^{\mathrm{AO}}_{50}$. The latter corresponds to the benchmark's MAO-ST score; its rotational origin check uses both axes, unlike the one-sided origin-only diagnostic. Scores are percentages; differences are reported in percentage points (pp) and computed before rounding displayed scores.

Because the translation-origin clause is omitted, the primary motion score decomposes as

$$\mathrm{AP}^{\mathrm{AO}}_{50} = \tfrac{1}{2}\left(\mathrm{AP}^{\mathrm{AO}}_{50,\mathrm{rot}} + \mathrm{AP}^{\mathrm{A}}_{50,\mathrm{trans}}\right). \tag{13}$$

An origin-only intervention therefore changes only the rotation term. We retain the macro score for the benchmark comparison and report the class breakdown to show which population benefits. Where available, class-specific intervals are computed separately; uncertainty estimated for the macro score is not transferred to a class-specific claim.

For the published-number comparison only, we additionally report the evaluator's Euclidean-origin variant, which thresholds the full displacement between representative origins instead of perpendicular line distances and also applies an origin check to translations. It is not interchangeable with $\mathrm{AP}^{\mathrm{AO}}_{50}$. Reporting both conventions avoids assigning an unspecified implementation to a published baseline; all internal motion comparisons use the line-distance convention above.

Uncertainty notation. Brackets $[L, U]$ give the lower and upper bounds of an approximate 95% confidence interval for the stated *difference* in AP, in percentage points. We use a paired scene jackknife: omit one validation scene at a time and recompute both configurations on the same remaining scenes. The interval is the full-validation difference plus or minus 1.96 jackknife standard errors, not the minimum and maximum omitted-scene results. An interval entirely above zero supports a positive change under

this analysis; an interval containing zero leaves its sign unresolved, rather than demonstrating no effect. Model predictions are fixed, so these intervals describe scene-sampling variability, not retraining variability. Training-seed ranges are reported separately as descriptive statistics.

5.2 Reference results

Table 1. Published benchmark results and our validation reference. USDNet values are transcribed from Tables 2–4 of Halacheva et al. [5]; our values use the public challenge validation split. Both evaluations contain 42 Articulate3D scenes, but their scene-list identity has not been verified. This is a contextual comparison, not a matched-system experiment. All scores are percentages.

Method	Movable parts				Handles
	AP_{50}	+Axis†	+Origin†	Joint motion†	AP_{50}
USDNet (published)	41.8	34.6	31.4	25.0	31.1
Ours (public validation)	47.93	43.75	43.11	38.99 / 40.98	30.99

† Gated columns retain each source's reported convention; equivalent origin-diagnostic implementations are not verified. The published protocol specifies IoU > 0.5, a 15° axis threshold and a 0.25 m origin threshold, but does not disambiguate the evaluator's two joint-origin implementations. For ours, the joint entries report Euclidean-origin AP (38.99) and the line-distance MAO-ST convention used throughout our ablations (40.98), respectively. Training resources and complete split equivalence are not controlled. Handle AP is numerically similar (31.1 versus 30.99); the challenge test score is not substituted for our validation result.

The reference configuration combines the three independently trained predictors with geometric motion decoding and contextual child labels. It achieves 47.93 movable-part AP, 40.98 motion-gated AP and 30.99 handle AP on public validation (Table 1). Its published-number comparison with USDNet provides benchmark context: part scores are numerically higher and handle AP is similar, but the scene-list identity, training resources and complete evaluator implementations are not matched. The following controlled comparisons establish which information transfers improve our own system.

5.3 Contributions of part-handle coupling

Table 2. Three measured mechanisms in coupled part-handle inference. All results use the public validation split. The mechanisms act once in an acyclic graph and address different outputs; their gains must not be added. Scores and parenthesized differences are percentages and percentage points, respectively. Differences are computed before rounding the displayed endpoints.

Direction	Measured intervention	Metric	Result
Handles→parts	Dense handle selects a box-derived hinge line	AP_{50}^{AO}	13.74 → **40.98** (+27.25)
Parts→handles	Append query-associated child proposals to dense handles	AP_{50}	24.63 → **29.65** (+5.01)
Parts→handles	Part context only for child labels	AP_{50}	29.65 → **30.63** (+0.98)
Parts + dense→handles	Reference contextual rule for child labels	AP_{50}	29.65 → **30.99** (+1.34)

The first row fixes movable masks, classes, scores, and axes. The child-union gain is evidence for a complementary proposal source, not a causal estimate of parent conditioning: the matched conditioning control is reported separately (Table 15). The parts-only and full-vote gains overlap, so the +0.98 is not an additive "share" of +1.34. Dense incumbents are never relabelled.

Table 3. Per-class localization of the directed effects. The evaluator macro-averages rotation and translation classes. Absolute rows are scores (%); difference rows are pp. Each bracket gives the lower and upper bounds of the paired 95% scene-jackknife confidence interval (CI) for the difference immediately above it, in the same class or macro column. Interval endpoints are pp, not absolute AP scores.

Configuration or difference	Rotation	Translation	Macro
Part-motion centroid origin	1.88	25.59	13.74
Part-motion dense-handle origin	56.37	25.59	40.98
Dense-handle minus centroid	+54.49	+0.00	+27.25
95% paired CI	[+43.71, +65.27]	[0.00, 0.00]	[+21.86, +32.63]
Handle child union	48.56	10.74	29.65
Handle contextual vote	48.54	13.45	30.99
Vote minus union	−0.02	+2.71	+1.34
95% paired CI	[−0.15, +0.10]	[+0.80, +4.62]	[+0.37, +2.31]

Origin is not evaluated for translations, so the handle-origin change acts only on rotations; the zero translation interval reflects identical outputs in every paired scene omission. Conversely, the contextual vote's macro improvement is concentrated on translation handles. The class-resolved intervals describe these populations and do not substitute for a generalization claim beyond this split.

Handle locations resolve hinge ambiguity. Fixing part masks, classes, scores and axes isolates the origin rule. Replacing the centroid origin with a handle-guided hinge raises AP_{50}^{AO} from 13.74 to 40.98, a 27.25-pp gain (Tables 2 and 16). Mask AP and axis-gated AP remain unchanged. The class breakdown in Table 3 places this gain on rotational parts, whose motion AP rises from 1.88 to 56.37. Translation scores are unchanged because their origins are not evaluated.

Parts supply complementary handles and motion context. Appending query-associated children raises handle AP from 24.63 to 29.65. Holding the resulting masks and scores fixed, parts-only class correction adds 0.98 pp; the full rule with dense-handle fallback adds 1.34 pp (Table 2). The two label gains overlap and must not be summed. The class-resolved result shows a +2.71-pp translation-handle gain with a positive paired interval, while the −0.02-pp rotation-handle change is unresolved (Table 3). Thus, the net label benefit is concentrated on translation handles, although the rule can relabel either class.

Two directed transfers in one inference pass. Table 14 enables handle guidance and contextual labels separately and together, with the child union fixed. Each retains its own benefit when both are enabled: handles change part origins, whereas parts change child classes. These distinct outputs yield zero interaction in this control, consistent with the acyclic one-pass dependency structure.

5.4 Training-free motion decoding

Table 4. Matched learned motion decoders on frozen part inputs. Learned rows report minimum–maximum AP_{50}^{AO} scores (%) over three training seeds and receive the same predicted handle evidence when marked. All rows retain $AP_{50} = 47.93$ because masks are fixed. The bracket in the note gives lower and upper bounds of a paired 95% scene-jackknife confidence interval for the closest learned head minus the rule (pp); it is distinct from the three-seed score ranges.

Motion decoder	Handle cue	AP_{50}^{AO} (three-seed range)
Reference training-free rule	✓	**40.98**
Box-axis rule with recomputed origin	✓	41.20
Continuous regression	–	17.24–18.24
Continuous regression	✓	29.61–31.30
Learned candidate selection	–	25.38–28.32
Learned candidate selection	✓	37.27–38.63
Learned selection, box axes only	–	25.07–27.84
Learned selection, box axes only	✓	37.11–38.78
Rule axis + learned edge	–	25.36–28.75
Rule axis + learned edge	✓	37.69–39.36

These heads consume frozen query features and geometry, not jointly trained backbones. Discrete learned selection exceeds continuous regression with identical handle evidence, but neither establishes an improvement over the training-free rule. The box-axis variant changes the axis and recomputes the origin; it is not an axis-only control. The closest learned row has paired jackknife 95% interval $[-4.31, +1.05]$ pp relative to the rule.

The decoder combines a world-Z rotational-axis prior, a thin-box translation direction and handle-guided hinge selection. On frozen part inputs, discrete learned selection has higher point estimates than continuous regression at the same handle-cue availability (Table 4). The training-free rule reaches 40.98, compared with at most 39.36 for the displayed learned heads on the reference part model. The two-part-model comparison in Table 17 retains this ordering, but its paired intervals include zero. These results support an effective decoder without motion-regression training, not a statistically resolved or general advantage over learned motion estimation. The separate box-axis rule scores 41.20 while also recomputing origins; it is not an axis-only comparison.

Table 5. Component changes measured from both ends of the part-motion pipeline. "Add gain" applies one component to the naive pipeline; "removal loss" is full decoder minus the result after removing that component. Their difference is a two-end non-additivity diagnostic, not a unique causal interaction. Values are changes in AP_{50}^{AO} (pp). The final row's bracket gives the lower and upper bounds of the paired 95% scene-jackknife confidence interval for the full-minus-naive change, also in pp.

Component	Add gain	Removal loss	Two-end difference
Per-query argmax selection	+0.13	+0.19	+0.06
Largest-CC support cleanup	−0.11	+2.95	+3.07
Dense-handle origins	+23.79	+27.25	+3.45
Connectivity rescoring	+1.49	+1.60	+0.11
Full versus naive pipeline		+28.74 **pp; 95% CI** $[+22.12, +35.37]$	

The naive and reference AP_{50}^{AO} values are 12.24 and 40.98. Cleanup and handle origins leave mask AP exactly unchanged, as required by their fixed-mask implementation. Cleanup is slightly negative alone but useful in the full decoder, demonstrating non-additivity rather than a contradictory result.

Table 5 measures each component both when added to the naive pipeline and when removed from the full decoder. The complete configuration improves motion AP from 12.24 to 40.98, with a paired gain interval of $[+22.12, +35.37]$ pp. Support cleanup is most useful when combined with handle-guided geometry: it changes the fitted box and candidate lines, costs 2.95 pp when removed from the full decoder, but gives a -0.11-pp change when added alone. This two-ended comparison shows why isolated component gains cannot simply be added.

5.5 Complementary proposals and contextual labels

Table 6. Complementary handle localization. Each configuration appends a child source below the same dense-handle detections without label correction. The corrupted-mask controls preserve child count, mask size and confidence while disrupting localization. Scores are validation AP_{50} percentages.

Proposal source	Handle AP_{50}
Dense detections only	24.63
Dense + spatially permuted child masks	24.63
Dense + random size-matched child masks	24.63
Dense + query-associated child proposals	**29.65**

The real child union retains all 140 GT handles matched by dense incumbents and adds 53 matches, while preserving the incumbent TP/FP ranking prefix. Its gain demonstrates complementary localization, not an isolated effect of parent conditioning; Table 15 separates conditioning from the readout used to associate children.

The proposal controls in Table 6 test whether the extra source contributes localized handles rather than merely extending the ranked detection list. Spatially permuted and random size-matched children return exactly the dense-only AP. Real children preserve the dense detections' matching prefix and recover 53 additional GT handles, increasing AP by 5.01 pp. This demonstrates complementary localization. A separate conditioning/readout control in Table 15 does not isolate a reliable benefit of parent conditioning itself, so we attribute the gain to the implemented proposal source rather than conditioning alone.

Table 7. Contextual child-label controls. Child masks and scores are fixed; only labels of appended children can change. Scores are validation AP_{50} percentages.

Label rule	Part cue	Dense cue	AP_{50}	Δ
No correction	–	–	29.65	–
Parts only	✓	–	30.63	+0.98
Dense incumbents only	–	✓	30.66	+1.01
Full contextual rule	✓	✓	**30.99**	+1.34
Fallback only when no part speaks	✓	✓	30.99	+1.34
Part classes permuted within scene	corrupted	✓	28.62	−1.03
Part classes randomized	corrupted	✓	28.90	−0.75

The two cues are complementary but overlap: their individual gains must not be summed. Restricting fallback changes one child and is AP-identical on this split. In the last two rows, masks and scores are frozen and only the part-class channel is corrupted; dense labels remain intact. They relabel more children than the reference rule, but lower AP below the no-correction union, so relabel count alone is not evidence of useful context. Dense incumbents themselves are never relabelled.

Table 7 isolates the source of child-label evidence. Part context and dense-incumbent context each improve AP by about one point; their combination reaches 30.99. Corrupting only the part-class channel instead reduces AP below the uncorrected union, despite changing more labels. Useful semantic evidence, rather than the number of corrections, explains the benefit. The alternative fallback policy has identical AP on this split. All rows preserve child masks and scores and leave dense detections unchanged.

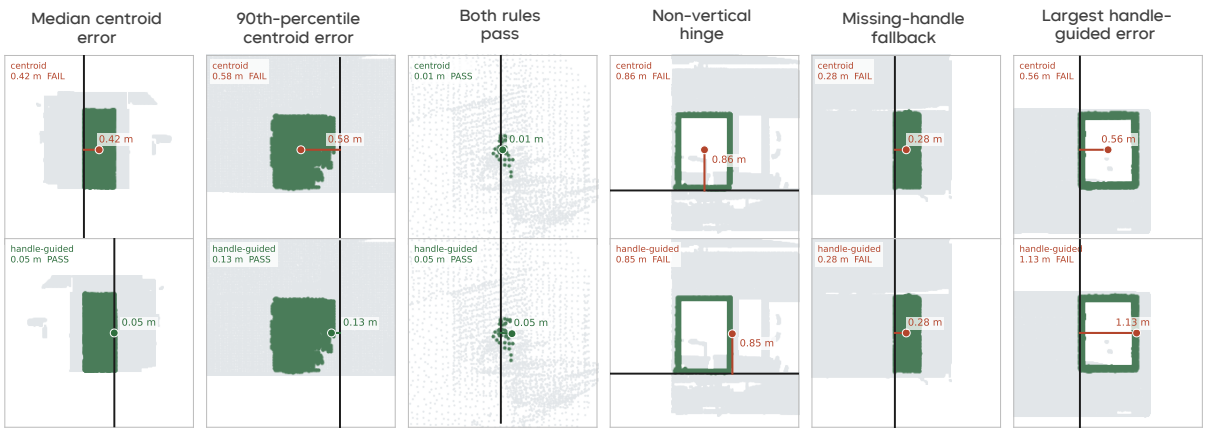

Figure 3. Paired hinge-origin cases on validation scenes. Each column fixes the predicted part mask and camera; only the origin rule changes. Black denotes the ground-truth hinge, and the colored segment depicts the perpendicular origin error. The first two columns are centroid-error ranks 61 and 109 among 121 rotations for which handle guidance passes but the centroid fails (median and 90th percentile). The remaining selections are the smallest centroid error among 37 both-pass cases, median axis error among 12 non-vertical hinges, median origin error among 19 missing-handle fallbacks, and the largest handle-guided error among 186 covered rotations. No case has a passing centroid but failing handle-guided origin in this population; the last column retains a large residual failure instead.

5.6 Stability and geometric coverage

Table 8. Observed variation of the directed effects. All values are AP changes in pp. "Seed span" is the maximum minus minimum paired gain across n training draws, not a confidence interval. Brackets give the lower and upper bounds of the paired 95% scene-jackknife confidence interval (CI) for the reference gain. A CI containing zero leaves the direction of change unresolved.

Mechanism	Reference gain	Seed span	95% scene CI
Dense handles → hinge origin	+27.25	2.11 ($n = 4$)	[+21.86, +32.63]
Child proposals 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 family varies the part-associated child predictor with dense incumbents fixed. Its append increments are 5.01/4.89/3.94 pp; its vote increments are 1.34/0.19/0.38 pp. The append interval compares the reference child union against dense detections alone; the gain remains positive in every leave-one-scene-out evaluation. This supports a complementary proposal source, not an isolated conditioning effect.

The three reference gains have positive paired scene intervals (Table 8). Handle guidance remains beneficial across four part-model draws. Appending children improves handle AP by 3.94–5.01 pp across three child-model draws, and its reference scene interval is [+1.47, +8.56] pp; the gain is positive under every single-scene omission. Contextual correction also improves all three draws, but varies more relative to its gain, from 0.19 to 1.34 pp (Table 13). We therefore distinguish the reference label gain from its smaller repeated-training gains.

The source-configuration comparisons in Table 11 show both transfers remain useful with weaker predictors. Table 12 additionally varies the dense-handle model while fixing the other predictors: proposal augmentation and label correction improve handle AP in all four draws. These comparisons test the listed configurations, not an isolated pretraining effect or a universal relationship between detector AP and hinge quality.

The candidate construction covers 94.4% of matched rotational predictions, with 88.3% correct selection among those covered (Table 18). This supports a compact physical search space for the represented mechanisms. The broader limitation is instance coverage: of 390 GT parts, 139 lack a matching mask, while 14 more fail the axis gate and another 14 fail the origin gate (Table 19). A hinge selector cannot recover an undetected part. The complete coverage analysis and population definitions are in Appendix A.4.

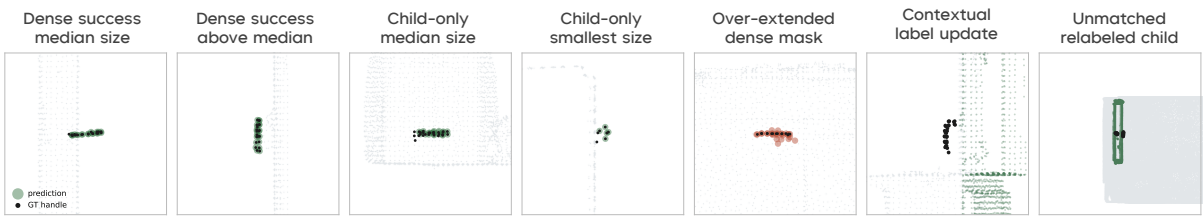

Figure 4. Dense and child handle complementarity on validation scenes. The selected strata show dense successes, handles recovered only after appending child proposals, an over-extended dense prediction, and contextual child-label updates. The final panel is deliberately a relabeled child without an IoU>0.5 ground-truth match, making explicit that label accuracy is only directly assessable for the matched subset. Specimens are deterministic quantile selections within their stated strata.

5.7 Qualitative comparisons

Figure 3 holds each predicted mask and camera fixed while comparing origins. The paired successes show how handle location resolves the hinge side; the non-vertical-axis, missing-handle and residual-error cases expose the decoder's physical limits. Figure 4 shows complementary child detections and contextual label changes. Its over-extended mask and unmatched relabelled child illustrate why correct class context cannot repair poor localization.

Additional galleries in Appendix B connect component ablations to changed geometric support, compare weaker prediction sources and illustrate the failure census. Cases are selected deterministically within stated strata; they explain the mechanisms rather than estimate their frequency.

5.8 Public test-set results

Our Articulate3D challenge submission (team *TnG*) achieved the highest published scores for movable-part motion and handle detection: 48.28% AP_{50}^{AO} and 34.46% AP_{50}, respectively. Tables 9 and 10 reproduce every public team entry and every metric exposed by the official board as accessed on September 6, 2026. These are challenge standings, not matched-resource comparisons. Test annotations are not public; all component studies above use public validation.

Table 9. Challenge leaderboard: movable-part motion result. All 11 public team entries and all four published metrics. Scores are percentages; ranking is by AP_{50}^{AO}.

Rank	Team	AP_{50}	AP_{50}^{A}	AP_{50}^{O}	AP_{50}^{AO}
1	**TnG (ours)**	**55.38**	**52.34**	**49.76**	**48.28**
2	coreMany	46.68	43.15	43.28	40.56
3	teamzb	47.20	44.99	38.55	37.11
4	core	44.44	41.05	39.55	36.89
5	ZZZ	47.72	44.81	37.77	35.72
6	linxii	43.52	41.35	35.47	33.60
7	coreN	43.70	40.60	35.94	33.38
8	ZeroR	39.53	37.25	30.14	28.61
9	wwwjh	43.66	37.98	31.54	26.34
10	ShinNam!	35.52	33.73	29.04	24.67
11	ttt	30.58	28.38	25.99	24.00

Source: official Articulate3D leaderboard [1], accessed September 6, 2026. Superscripts A, O, and AO denote axis, origin, and joint motion checks, respectively.

Table 10. Challenge leaderboard: handle-detection result. All eight public team entries. The board provides AP_{50} only; scores are percentages.

Rank	Team	AP_{50}
1	**TnG (ours)**	**34.46**
2	coreMany	32.86
3	core	32.60
4	teamzb	31.10
5	ZZZ	30.69
6	coreN	28.86
7	ZeroR	25.71
8	moon	24.80

Same source and access date as Table 9. The public movable-part-motion and handle-detection boards contain different participant sets.

6 Discussion and limitations

What coupling contributes. Handles and parts provide different forms of evidence: handle locations constrain a part's attachment side, while part-associated proposals and motion classes improve handle detection and interpretation. The fixed-input and label-source controls separate these roles (Tables 7 and 16). No iterative refinement is required because the two directions read and update different attributes. Within the geometric decoder, however, support estimation and origin selection remain interdependent, as the two-ended component ablation shows (Table 5).

Physical and perceptual limits. The decoder assumes approximately planar parts and upright rotational axes. Horizontal hinges, tilted or curved mechanisms, missing handles and fragmented support can violate these assumptions; proximity-based association can also select a neighboring handle (Figures 3 and 10). High hinge-candidate coverage does not resolve missing instances or incorrect motion classes. The coverage and failure census instead motivate better part detection and more general axis hypotheses alongside improved hinge selection (Tables 18 and 19).

Scope of the measured benefits. The proposal gain demonstrates complementary localization, not a separately established advantage of parent conditioning (Table 15). Label correction improves the reference configuration, but its magnitude changes across child-training draws (Table 13). Likewise, the frozen-feature decoder comparisons support the effectiveness of training-free geometry in this setting, without ruling out stronger jointly learned representations (Table 17). These distinctions define which aspects of the method are supported by the present controls.

7 Conclusion

We presented Segment–Snap for recovering movable parts, motion and handles as a joint description of interaction in static 3D scenes. Predicted handles guide training-free hinge selection; part-associated proposals expand handle coverage, and part context refines their motion labels. Fixed-input comparisons, repeated training and paired visualizations demonstrate the value of these directed information transfers. The results support geometric and semantic coupling as a useful strategy for 3D interaction understanding, with further progress requiring better instance coverage and motion models beyond upright planar mechanisms.

References

[1] Articulate3D Organizers. Articulate3D Challenge Public Leaderboard, 2026. URL https://art3d-challenge.mooo.com/web/challenges/challenge-page/1/leaderboard/. Accessed September 6, 2026; public test standings and metric columns.

[2] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Berman_The_LovaSz-Softmax_Loss_CVPR_2018_paper.html.

[3] Alexandros Delitzas, Ayça Takmaz, Federico Tombari, Robert W. Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-grained functionality and affordance understanding in 3d scenes. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Delitzas_SceneFun3D__Fine-Grained_Functionality_and_Affordance_Understanding_in_3D_Scenes_CVPR_2024_paper.html.

[4] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. doi: 10.1023/B:VISI.0000022288.19776.77. URL https://www.cs.cornell.edu/~dph/papers/seg-ijcv.pdf.

[5] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2412.01398.

[6] Zhao Huang, Boyang Sun, Alexandros Delitzas, Jiaqi Chen, and Marc Pollefeys. REACT3D: Recovering articulations for interactive physical 3d scenes. *IEEE Robotics and Automation Letters*, 11(5):5954–5961, 2026. doi: 10.1109/LRA.2026.3674028. URL https://doi.org/10.1109/LRA.2026.3674028.

[7] Denys Iliash, Hanxiao Jiang, Yiming Zhang, Manolis Savva, and Angel X. Chang. S2O: Static to openable enhancement for articulated 3d objects. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6785–6795, 2026. URL https://openaccess.thecvf.com/content/WACV2026/html/Iliash_S2O_Static_to_Openable_Enhancement_for_Articulated_3D_Objects_WACV_2026_paper.html.

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

[9] S2O Authors. S2O rule-based motion predictor, 2026. URL https://github.com/3dlg-hcvc/s2o/blob/35aa03f29b856f123c476c006638a38fc1e3ad4c/opmotion/engine/motion_predictor/rule_motion_predictor.py. Official implementation; accessed September 6, 2026.

[10] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *Machine Learning in Medical Imaging*, pages 379–387. Springer, 2017. doi: 10.1007/978-3-319-67389-9_44. URL https://arxiv.org/abs/1706.05721.

[11] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask transformer for 3d semantic instance segmentation. In *IEEE International Conference on Robotics and Automation*, pages 8216–8223, 2023. doi: 10.1109/ICRA48891.2023.10160590. URL https://doi.org/10.1109/ICRA48891.2023.10160590.

[12] 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, pages 2393–2401, 2023. doi: 10.1609/aaai.v37i2.25335. URL https://ojs.aaai.org/index.php/AAAI/article/view/25335.

[13] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://scannetpp.mlsg.cit.tum.de/scannetpp/.

[14] Kadir Yilmaz, Adrian Kruse, Tristan Höfer, Daan de Geus, and Bastian Leibe. Volume transformer: Revisiting vanilla transformers for 3d scene understanding. In *European Conference on Computer Vision*, 2026. URL https://github.com/YilmazKadir/Volt.

[15] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2503.19199.

[16] Mandi Zhao, Yijia Weng, Dominik Bauer, and Shuran Song. Real2Code: Reconstruct articulated objects via code generation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CAssIgPN4I.

A Additional quantitative analysis

These controls detail the prediction sources, motion decoder and geometric coverage underlying the main results. All use the public validation split. Alternative configurations are identified explicitly and do not replace the reference system.

A.1 Prediction sources and training variability

Table 11. Coupling under source-configuration perturbations. The complementary cue is held fixed within each row. Random-initialization rows also use a longer training schedule, so they are configuration comparisons rather than isolated pretraining effects. Scores are validation percentages.

Output and source configuration	Fixed complementary cue	Without cue	With cue
Part motion: reference part model	Dense handles	13.74	40.98
Part motion: joint part head	Dense handles	12.26	37.33
Part motion: random-init part model	Dense handles	5.86	25.42
Handle detection: reference dense model	Frozen child union	24.63	29.65
Handle detection: random-init dense model	Frozen child union	1.75	13.63

The upper block reports part-motion AP_{50}^{AO}; the lower block reports handle AP_{50}. These controls show that the measured transfers persist for the listed source configurations, but they do not establish a monotonic relation between source quality and coupling gain.

Table 11 changes the target predictor while fixing its complementary cue. Handle guidance improves the reference, joint-head and random-initialization part configurations; the same child source improves both reference and weak dense-handle configurations. Random-initialization runs also use a different training schedule, so these are source-configuration comparisons rather than isolated tests of pretraining.

Table 12. Dense-handle model repeats with fixed child and part sources. All columns are validation percentages. The first three columns are handle AP; the final column is part-motion AP_{50}^{AO} when that dense field supplies the handle cue.

Dense-model draw	Dense AP	Union AP	Full-context AP	Part motion AP
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

The same released child predictor, part predictions, and post-processing are fixed in every row. Appending children and applying contextual labels each increase handle AP in all four draws. Dense-handle AP and part-motion AP need not order the draws identically: the decoder uses handle position, not the detector's entire mask-and-class ranking.

Table 13. Child-training sensitivity of handle proposals and label correction. Dense incumbents and the part-voting input are fixed; only the child model's training draw changes. The append gain is relative to the frozen dense baseline; the vote gain is relative to that draw's own union. Entries are validation AP_{50} gains (pp).

Child training draw	Append child proposals	Contextual vote
Reference child	+5.01	+1.34
Child seed A	+4.89	+0.19
Child seed B	+3.94	+0.38

The union increment remains positive across the three child draws (range 1.08 pp). The vote increment ranges by 1.15 pp, including much smaller values for both repeats. This descriptive family does not replace a formal uncertainty analysis; the reference configuration's reported score remains 30.99.

Table 12 varies the dense predictor with both the part and child sources fixed. Child proposals and label correction each improve handle AP in all four rows. The ordering of dense AP differs from that of part-motion AP because a handle's local position and its complete detection quality are different quantities.

Table 13 instead varies only the child predictor, with dense incumbents and part context fixed. Proposal gains remain positive across all three draws; contextual gains are also positive but substantially smaller for the two repeats. These distinct training families describe sensitivity to different prediction sources and are not pooled into one uncertainty estimate.

A.2 Information flow and child association

Table 14. Orthogonal one-pass coupling controls. The child-proposal union is fixed in every cell; the factorial switches only dense-handle guidance for part origins and contextual child-label voting for handles. Scores are validation percentages. Brackets in the note give lower and upper bounds of paired 95% scene-jackknife confidence intervals for each signal's on-minus-off AP change, in pp.

	Switched signal		Ranking metric	
Configuration	**Handle-guided origin**	**Context vote**	**Part motion** AP_{50}^{AO}	**Handle** AP_{50}
Neither signal	–	–	13.74	29.65
Handle-guided hinge only	✓	–	**40.98**	29.65
Contextual vote only	–	✓	13.74	**30.99**
Both signals (reference)	✓	✓	**40.98**	**30.99**

Handles change part-motion AP by +27.25 pp (paired scene jackknife 95% interval $[+21.86, +32.63]$); voting changes handle AP by +1.34 pp ($[+0.37, +2.31]$). The measured interaction is exactly zero on both metrics: handle guidance changes origins only, while voting reads masks and classes. This is a factorial control of two directed transfers, not evidence of iterative refinement.

Table 15. Separating child conditioning from association. The shared-field rows differ in parent conditioning under a matched readout; the reference instead associates child masks by query index. Dense detections and contextual label rules are fixed. Scores are validation AP_{50} percentages. Brackets in the note are paired 95% scene-jackknife confidence intervals for conditioned minus unconditioned AP (pp).

Child predictor and readout	**Union**	**After label correction**
Unconditioned, shared-field readout	28.07	29.78
Conditioned, shared-field readout	28.41	29.26
Reference query-associated children	**29.65**	**30.99**

Within the shared-field pair, conditioning changes union AP by +0.34 pp (95% CI $[-0.99, +1.66]$) and corrected AP by −0.52 pp ($[-2.66, +1.62]$); both intervals include zero. Query-associated readout gives a separate +1.24-pp union difference from conditioned shared-field readout. These comparisons distinguish the implemented proposal source from a causal claim about conditioning alone.

Table 14 verifies output ownership: dense-handle guidance affects part origins, and contextual labels affect handles. The child union is fixed throughout, so this is a control of the two exchanged signals, not all three proposal-and-correction mechanisms. Its zero interaction is consistent with the one-pass dependency structure.

Table 15 distinguishes the role of the additional predictor from parent conditioning and the readout used to associate children. The matched shared-field pair gives opposite-signed union and corrected-output differences, with both scene intervals including zero. Query-associated readout yields a separate 1.24-pp union difference. Together with the localization controls, this supports the implemented proposal source without identifying conditioning alone as the cause of its gain.

A.3 Motion-decoder controls

Table 16. Fixed-input handle-origin control. Part masks, classes, scores, and axes are identical in both rows; only the origin rule changes. Scores are validation percentages.

Origin rule	AP_{50}	AP^{A}_{50}	AP^{O}_{50}	AP^{AO}_{50}
Centroid origin (no handle cue)	47.93	43.75	14.37	13.74
Hinge guided by dense handles	47.93	43.75	43.11	**40.98**

Equal mask and axis columns are an implementation check: handles enter this comparison only through the representative origin.

Table 17. Matched learned heads on two part-model training draws. Each learned entry uses the best head seed for that backbone with predicted-handle evidence. A cell lists AP^{AO}_{50} (%) followed by the lower and upper bounds of the paired 95% scene-jackknife confidence interval for that head minus its column's training-free rule (pp). Thus, brackets describe uncertainty in the AP change, not the absolute score or a range across seeds.

Learned head	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 $[-4.66, +1.04]$

The displayed learned heads have lower point estimates than their own rule on both part-model draws, but every scene interval includes zero. These frozen-feature comparisons support the effectiveness of the training-free rule without establishing a statistically resolved advantage over the learned heads.

Table 16 isolates handle guidance by preserving masks, classes, scores and axes; equal mask and axis AP confirm that only the origin rule changes. Table 17 compares the training-free rule with matched learned heads on two fixed part-model draws, reporting the best head seed for each family. The rule has the higher point estimates, while the paired intervals leave the differences unresolved. The comparison concerns decoding from frozen features, not jointly trained representations.

A.4 Coverage and failure populations

Table 18. Coverage and selection diagnostics on validation. Coverage asks whether a valid candidate exists; selection asks whether the reference rule chooses it. These are different populations and are not end-to-end AP.

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

On predicted movable-part masks, 57.2% of ground-truth instances satisfy the joint motion criterion; 64.4% are reachable by mask overlap alone. For the four hinge candidates, absence (5.6% of matched rotations) and wrong edge selection (11.0%) are reported separately. Candidate coverage does not establish a learned selector's performance.

Table 19. Where validation instances are lost. The movable-part census assigns each of 390 GT instances to the first applicable bucket: missing mask, axis failure, origin failure, or all gates passed. Counts are not AP or additive oracle gains.

Movable-part outcome	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 passed	160	63	223
GT population	236	154	390

The union of dense and child handles covers 193 of 388 GT handles. By GT-size quartile, coverage is 22/97, 53/99, 63/96 and 55/96; the smallest quartile has at most 11 points and the largest more than 29. These descriptive bins use the validation population and were not used as training targets.

Table 18 separates mask reachability, hinge-candidate availability and selection. Matched-prediction statistics may include several predictions for one GT rotation; the GT-mask diagnostic instead uses one mask per annotated rotation. Candidate coverage is high on both populations, at 94.4% and 95.8%, respectively, with conditional selection rates of 88.3% and 88.1%. These instance fractions are not class-macro AP.

Table 19 assigns each GT part to its first failure: missing mask, axis, or origin. Missing masks account for the largest group, including 89 translations and 50 rotations. Handle coverage is lowest in the smallest size quartile, while the middle quartiles show that size alone does not determine detection success. Counts describe the available predictions, not additive or recoverable AP gains.

B Additional qualitative analysis

B.1 Coverage and geometric components

Figure 5 visualizes the census in Table 19: missing instances limit the gains available to motion decoding, and small handles remain difficult. Figure 6 compares the naive and full configurations on common GT instances, including a case where the naive origin is closer and a newly covered instance whose origin still fails. The predicted instance sets can differ between these configurations.

Figure 7 illustrates the component dependence measured in Table 5. Removing disconnected support can substantially change the fitted box and hinge, while a typical instance changes little. The two examples connect a population-level ablation to its geometric action without implying uniform improvement.

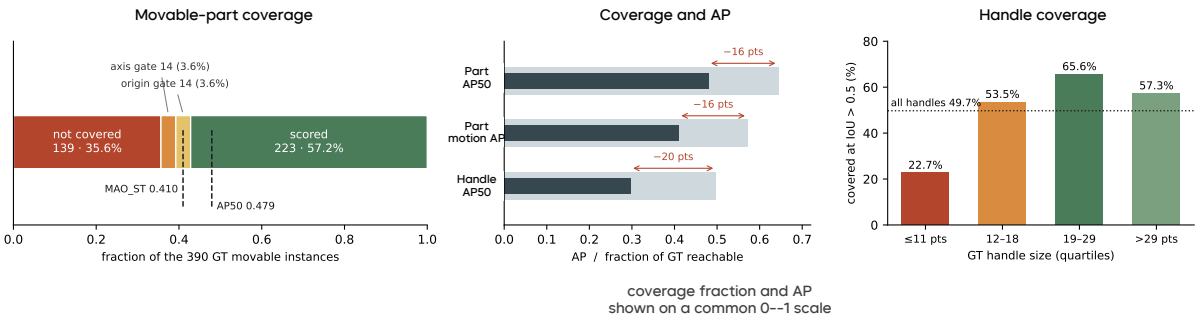

Figure 5. Coverage and selection limits on the public validation split. Left: the movable-part census assigns each ground-truth instance to coverage, axis, origin, or fully scored outcomes. Centre: each metric's macro AP is juxtaposed with a pooled instance-coverage fraction on a common 0–1 scale. These coverage fractions are diagnostic reachability statistics, not strict AP ceilings or quantities subtractable from AP. Right: handle coverage by ground-truth size quartile.

Figure 6. Naive versus full geometric decoding. The top and bottom rows compare the naive and full decoders against the same ground-truth part and camera; black denotes the annotated hinge. The first two columns are naive-error ranks 92 and 165 among 183 rotations covered by both configurations. The third is the median-size instance among three rotations covered only by the full configuration; its origin still fails. The last has the smallest naive-origin error among 38 both-pass cases, and the naive origin is closer. No translation is covered by only one configuration in the selection population. The gallery retains these exceptions rather than estimating their frequency.

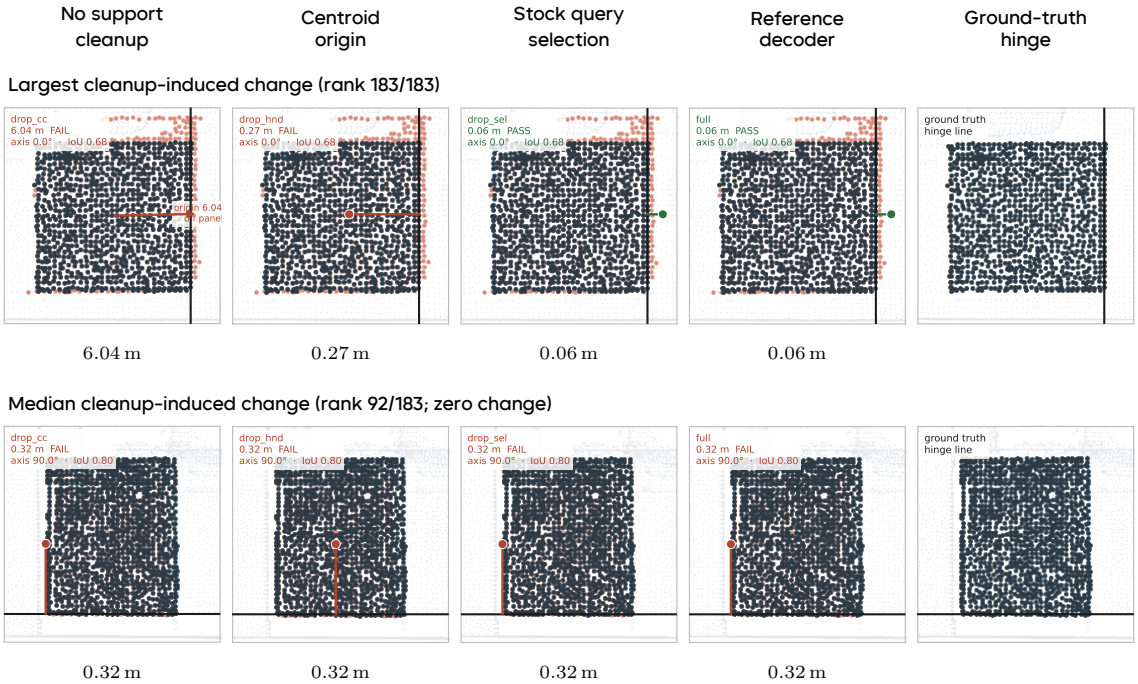

Figure 7. Geometric component changes on fixed validation specimens. Each row fixes the camera and ground-truth part; columns show four decoder variants and the annotated hinge. Dark points denote ground truth, pale red the matched prediction, and the colored segment the perpendicular origin error reported below each panel. The upper row has the largest cleanup-induced change among 183 jointly covered rotations; the lower row has the median change, which is zero. The large origin error in the first panel lies outside the view and is explicitly marked. Selection can change the matched mask; the other columns do not imply a fixed instance set across configurations.

B.2 Weaker prediction sources

Figure 8 fixes the predicted mask within each centroid-versus-handle comparison while varying the part model between rows. It illustrates the source-configuration results in Table 11. Figure 9 holds the scene

crop fixed and compares dense-handle fields from the reference and random-initialization models. It shows how the dense source fragments; the aggregate table, rather than this pre-union view, measures the benefit of adding children.

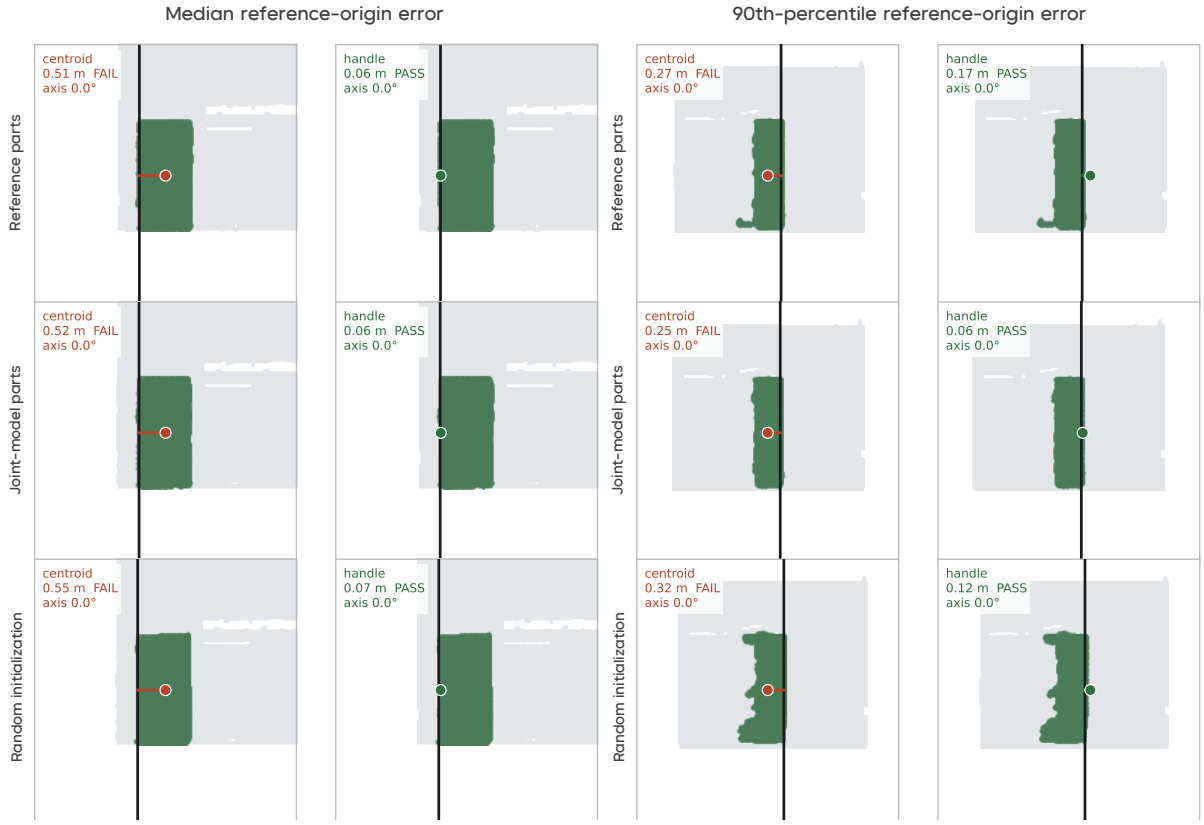

Figure 8. Handle guidance across part-source strength. Within each row, the same part prediction is decoded with centroid and handle-guided origins using the same dense handle field; across rows, the part source changes. Columns use reference-origin-error ranks 61 and 110 among 122 rotations covered by all three sources (median and 90th percentile). The comparison shows the handle cue on reference, joint and random-initialized part predictions, without treating masks from different rows as matched.

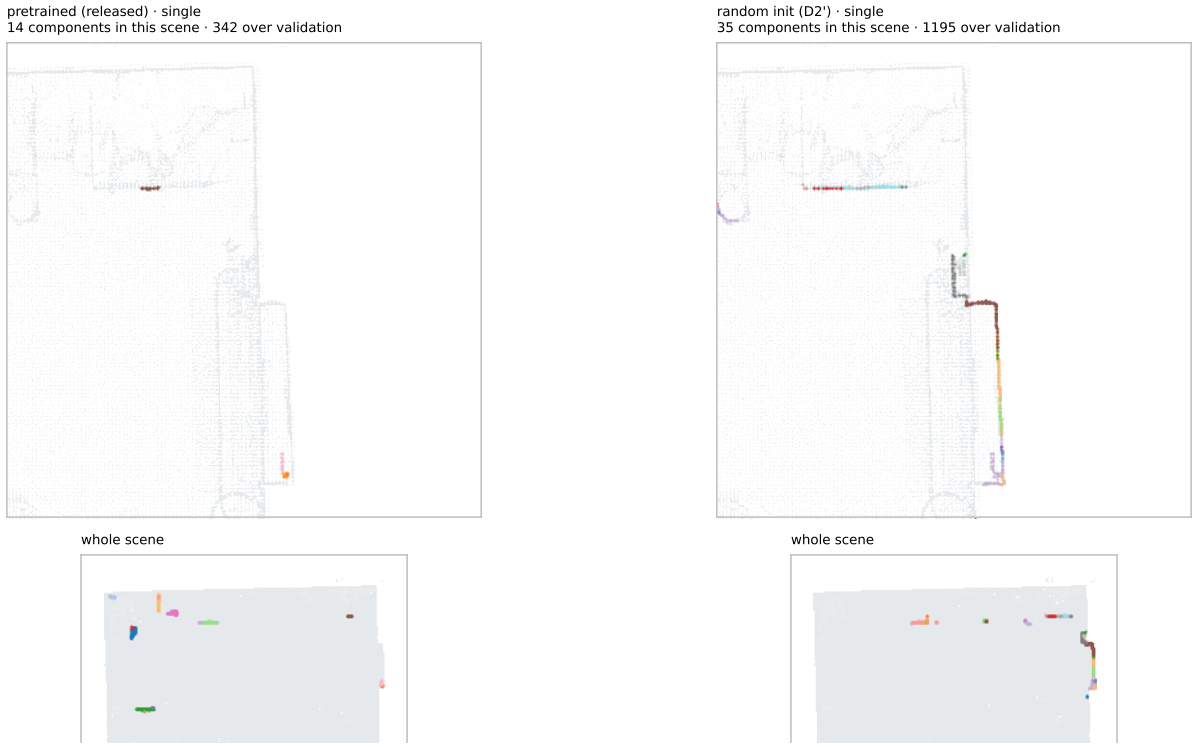

Figure 9. Dense-handle fields under source variation. The same validation crop and camera compare the reference and random-initialization configurations from Table 11. Their training schedules also differ, so the comparison does not isolate pretraining. The scene has the median ground-truth-handle count (21st of 42 scenes). The random-initialized field emits more fragmented components (35 versus 14 in this scene; 1,195 versus 342 over validation). These are dense predictions before child augmentation.

B.3 Representative failures

Figure 10 illustrates the categories in Table 19: missing parts, a violation of the upright-axis prior, an origin-selection error, and small or over-extended handle predictions. Each gallery uses deterministic cases from a stated population; the examples explain the error types, not their prevalence.

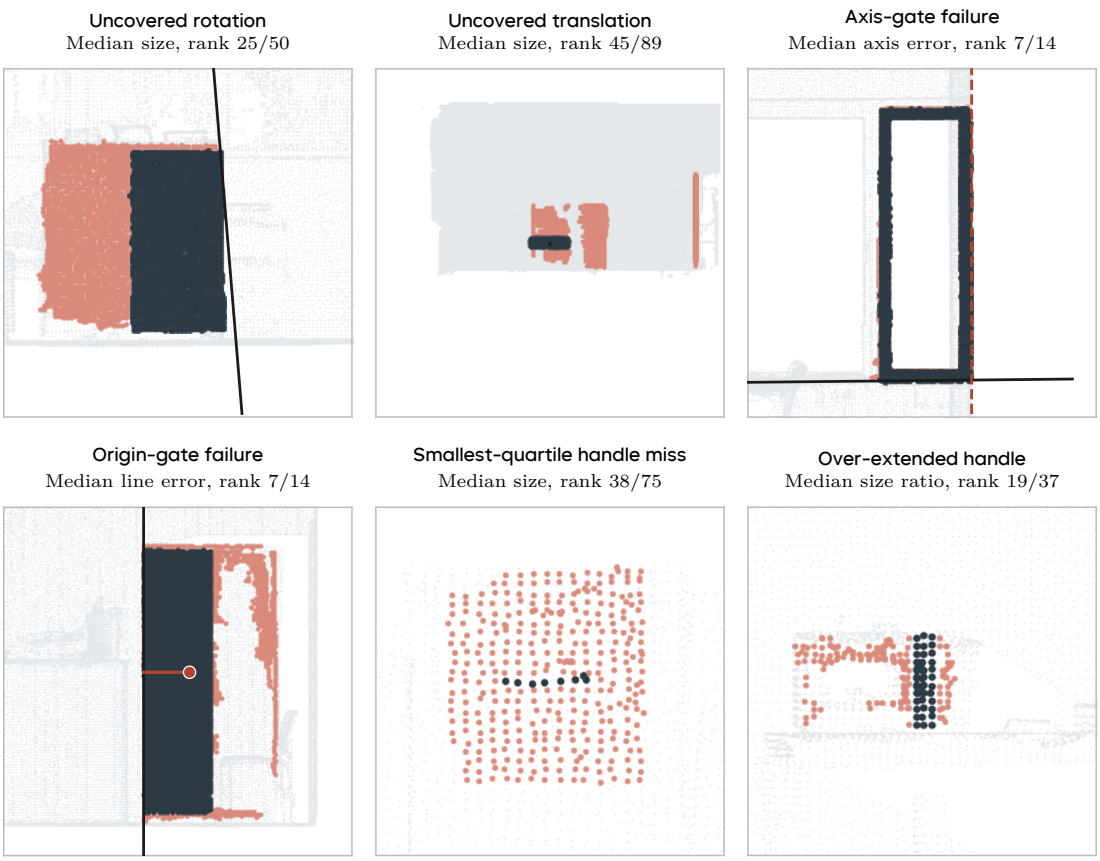

Figure 10. Different failure mechanisms on validation. Dark points show ground truth and pale red the displayed prediction; black denotes the annotated hinge where applicable. Each panel is a deterministic median-rank specimen of its named stratum. Missed rotations and translations have different overlap patterns; the horizontal hinge violates the rotational axis prior; a covered panel can still fail the origin gate. Small and over-extended handle failures illustrate distinct localization problems. The gallery explains mechanisms rather than estimating their frequency.