ACFUNSG: ACTIVE EVIDENCE SEEKING FOR OPEN-VOCABULARY FUNCTIONAL 3D SCENE GRAPH GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Functional 3D scene understanding goes beyond scene geometry and object-level semantics to model part-level affordances for embodied interaction. However, incomplete or ambiguous observations can introduce errors in cross-view association and functional reasoning that propagate into the scene graph. Existing methods typically construct graphs from a fixed observation set, with limited support for subsequent evidence acquisition and graph revision. To address this limitation, we propose AcFunSG, a training-free, closed-loop framework for open-vocabulary functional 3D scene graph generation that progressively completes and refines an initial graph through uncertainty-guided evidence acquisition. Given posed RGB-D observations, it builds an initial graph through cross-view association and relation-specific global reasoning while retaining entities and relations unresolved by the current evidence. These uncertainties guide further evidence acquisition. Under a limited budget, Ac-Agents selects informative views and queries, integrates the resulting evidence, and revises affected predictions. A deterministic verifier enforces the same consistency constraints used during construction and accepts only updates that pass structural and evidential checks. The graph is progressively completed and corrected, while indeterminate elements remain unresolved. Experiments on SceneFun3D, FunGraph3D, and FunThor show state-of-the-art overall instance recall, with a functional part mapping Recall@3 of 85.1% and relation F1 of 42.0% on FunThor.

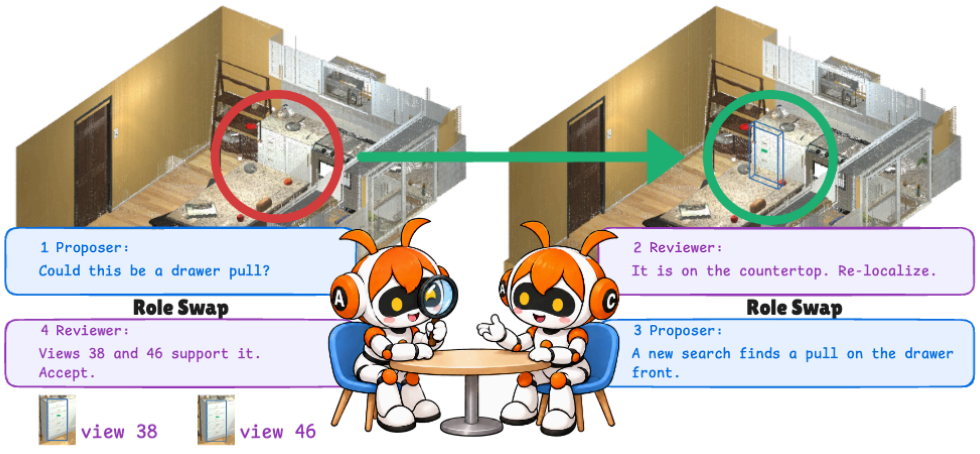

Figure 1: **Active completion of functional scene graphs with AcFunSG.** Two agents iteratively propose candidates, review evidence, and guide visual search. An incorrect drawer pull localization is corrected through multiview verification, recovering the missing node and relation.

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

1 INTRODUCTION

3D scene understanding reconstructs scene geometry, object semantics, and their relationships from multi-view observations, and is a key problem in computer vision (Dai et al., 2017; Chang et al., 2017; Armeni et al., 2019). Compared with 2D visual representations (Johnson et al., 2015; Krishna et al., 2017), it establishes cross-view correspondences in a unified coordinate system, providing structured scene representations for contextual reasoning, navigation, object manipulation, and task planning (Ma et al., 2022; Zhu et al., 2025; Werby et al., 2024; Huang et al., 2023; Rana et al., 2023).

Existing work has expanded the range of entities and relations that scene graphs can represent through open-vocabulary model (Koch et al., 2024; Gu et al., 2024). Other studies have incorporated interactive parts and their functional connections into scene graphs, while also modeling dependencies among functional relations (Zhang et al., 2025; Fu et al., 2026). Despite this progress, most methods first identify object and part nodes from multi-view observations and then infer relations over this fixed set of entities. This design is particularly fragile for functional parts, which are often small, occluded, or clearly visible in only a few views. For example, if a drawer handle is not created as an independent node during initial graph construction, the model cannot represent that pulling the handle opens the drawer. The limitation therefore lies not only in relation prediction, but also in fixing the entities available for reasoning too early. Functional graph construction should instead move beyond one-shot feed-forward prediction toward active completion. The current graph can guide the search for potentially missing functional parts, trigger targeted multi-view evidence collection, and be revised as new evidence becomes available.

Motivated by this observation, we propose AcFunSG, a training-free method for active open vocabulary functional 3D scene graph construction. Our method first establishes cross-view consistent representations of objects and parts from multi-view RGB-D observations. It then performs scene-level joint inference to construct the initial functional relations. Based on the current graph, the system hypothesizes potentially missing functional parts and actively inspects relevant views to acquire new visual evidence. Observations from different views are associated into physical candidates. Before being added to the graph, these candidates are verified against visual evidence and structural constraints. As illustrated in Fig. 1, when a candidate is incorrectly localized, verification feedback triggers a new search and additional multi-view evidence collection before the graph is updated. In this way, functional 3D scene graph construction no longer ends with a one-pass feed-forward prediction. Instead, it forms a closed loop of initial graph construction, missing-part discovery, and verified graph updates. Our main contributions are summarized as follows.

- We construct functional scene graphs from multi-view observations by combining Hierarchical Consistency (HC) for cross-view identity association with Relation Reasoning Calibration (RRC). Together, they maintain consistent object and part identities and jointly reason over functional relations and structural constraints, providing an initial graph for active completion.
- We propose **Ac-Agents**, an active open-vocabulary completion framework in which two agents alternate between proposing graph edits and reviewing their supporting evidence. Review feedback actively redirects subsequent multi-view search, enabling the system to revise or rediscover uncertain candidates before committing only visually and structurally supported edits.
- We conduct extensive experiments on SceneFun3D, FunGraph3D, and FunThor to evaluate AcFunSG in instance reconstruction and functional triplet prediction. Our method achieves state-of-the-art overall instance recall across all three benchmarks, with overall functional triplet Recall@10 reaching 67.4% on FunGraph3D and 65.1% on SceneFun3D.

2 RELATED WORK

Functional 3D scene graphs. 3D scene graphs represent environments through entities and their relations (Armeni et al., 2019; Wald et al., 2020). SceneGraphFusion and Hydra (Wu et al., 2021; Hughes et al., 2022) support incremental graph construction and online spatial mapping. With advances in open vocabulary perception, OpenScene, ConceptFusion, and OpenMask3D (Peng et al., 2023; Jatavallabhula et al., 2023; Takmaz et al., 2023) aggregate visual features across views to support semantic queries over 3D scenes and instances. Open3DSG and ConceptGraphs (Koch et al., 2024; Gu et al., 2024) further incorporate open vocabulary semantics into explicit graph representations. Functional scene understanding focuses on the parts involved in interaction and their

relations to other entities. SceneFun3D (Delitzas et al., 2024) provides detailed annotations of functional parts and their affordances. FunGraph (Rotondi et al., 2025) models functional elements and relations within objects, while OpenFunGraph (Zhang et al., 2025) constructs open vocabulary functional graphs from multiview RGB-D observations. Hu et al. (2026) combine visual information, entity association across frames, and temporal graph optimization. Their method produces hierarchical representations of objects and parts. FunFact (Fu et al., 2026) uses factor graphs to jointly infer candidate functional relations. Other approaches explore physical interaction and state changes. SpotLight (Engelbracht et al., 2024) discovers functional relations through physical interaction, while MomaGraph (Ju et al., 2026) maintains spatial and functional relations as object states change. Despite these advances, functional scene graphs may still contain omissions and errors. Building on entity association across views and joint functional relation inference, we introduce a repair mechanism with two agents that alternate roles. Review feedback guides active visual evidence acquisition, and the graph is completed and revised only after consistency checks.

Multimodal agents and collaborative refinement. Agents built on large models combine tool use with environmental feedback to support iterative problem solving. ReAct (Yao et al., 2023) interleaves reasoning and action. This enables models to gather external information and adjust their plans and subsequent decisions. Reflexion (Shinn et al., 2023) converts task feedback into reflective memory, allowing past experience to guide subsequent actions. For visual tasks, ViperGPT (Surís et al., 2023) decomposes tasks into executable module calls and logical operations, enabling complex visual reasoning without additional training. MM-ReAct (Yang et al., 2023) uses a language model to coordinate vision experts for tasks such as understanding multiple images and spatial reasoning. V* (Wu & Xie, 2024) uses language to guide visual search toward relevant regions, improving the recognition of local details in high resolution images. For feedback and collaboration, CRITIC (Gou et al., 2024) uses external tools to check model outputs and correct errors based on verification feedback. PairCoder (Chen et al., 2026) assigns planning and implementation to Navigator and Driver agents, respectively. Execution feedback guides plan revision and code refinement to improve performance on complex programming tasks. Our work combines visual evidence acquisition and collaborative refinement to repair functional 3D scene graphs. Two agents alternate between proposing candidates and reviewing evidence. Review feedback guides further multiview evidence acquisition, while a deterministic verifier constrains updates to objects and functional relations.

3 METHOD

3.1 OVERVIEW

As shown in Fig. 2, AcFunSG separates functional scene graph construction into two complementary stages: reliable initialization and evidence-driven completion. For initialization, we introduce **HC** to reduce cross-view identity errors and **RRC** to resolve ambiguous functional correspondences, yielding a structurally coherent initial graph G_0. For completion, we introduce **Ac-Agents**, where two agents collaboratively identify missing graph elements and actively seek visual evidence to verify them. Graph hypotheses determine what evidence should be acquired next, while review feedback revises subsequent hypotheses and search. This feedback loop expands the graph conservatively, committing an edit only when visual support and structural consistency agree.

3.2 PROBLEM DEFINITION

Given multiview RGB-D observations with known camera poses, $\mathcal{O} = \{(I_k, D_k, K_k, T_k)\}_{k=1}^{N}$, our goal is to construct a functional 3D scene graph $G = (V, E)$. For view k, I_k, D_k, and K_k denote the RGB image, depth map, and camera intrinsics, respectively. The transformation $T_k \in SE(3)$ maps camera coordinates to world coordinates. Each node represents an object or functional part instance and stores its name, 3D geometry, and instance identity. Each directed edge $(u, \varphi, v) \in E$ describes a functional relation through a natural language phrase φ. Both node names and relation descriptions are drawn from an open vocabulary.

However, the graph inferred from $\mathcal{O}$ is not necessarily complete, since functional parts may be occluded, only weakly observed, or overlooked during initial construction. Our problem is therefore not only to infer an initial graph G_0, but also to actively complete it under a limited evidence budget. At repair iteration t, graph G_t guides which observations should be revisited and what evidence

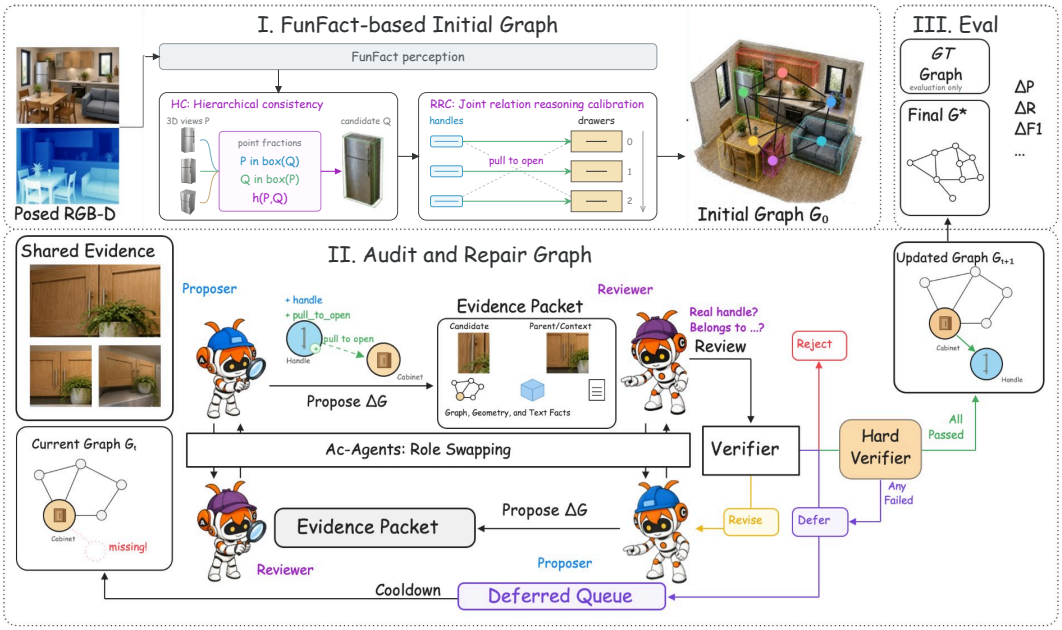

Figure 2: **Overview of AcFunSG. (I)** Hierarchical Consistency (HC) associates the multi-view identity of objects and functional parts across views, and Relation Reasoning Calibration (RRC) jointly infers functional relations to construct the initial graph G_0 from posed RGB-D observations. **(II)** AC-Agents iteratively audit and repair the graph through role-swapping proposal and review. Candidate edits are supported by targeted multi-view evidence, tested on a trial graph, and committed only after visual and structural verification; otherwise, they are revised, rejected, or deferred. **(III)** The final graph is evaluated against the ground truth.

should be collected. The resulting evidence may induce a candidate graph edit ΔG_t, consisting of newly discovered parts and functional relations. We accept ΔG_t only when it is supported by multi-view evidence and satisfies graph consistency constraints, producing G_{t+1}.

3.3 Initial Functional Scene Graph Construction

We build the initial functional scene graph G_0 on the perception and relation reasoning pipeline of FunFact (Fu et al., 2026). While this pipeline provides open-vocabulary object and functional-part observations, its initial graph can still suffer from incorrect cross-view identity associations and ambiguous functional correspondences. We address these two sources of error with two complementary extensions: **(i)** HC that strengthens cross-view instance association through bidirectional geometric consistency, and **(ii)** RRC that introduces relative-order constraints into joint functional relation inference. Together, they provide a more reliable G_0 for subsequent active completion.

FunFact-based instance perception. We follow FunFact (Fu et al., 2026) to obtain open-vocabulary object and functional-part observations in each view. A vision-language model generates textual descriptions that are grounded by Grounding DINO (Liu et al., 2024) and segmented by SAM 2.1 (Ravi et al., 2024), with part detection restricted to the corresponding object region. Each mask is back-projected into a world-coordinate point set using depth and camera poses, while average-pooled DINOv2 (Oquab et al., 2023) features support cross-view association.

Hierarchical consistency (HC) for multi-view identity association. FunFact associates repeated observations across views using appearance and geometric cues. However, partial visibility and segmentation errors can cause observations of the same instance to exhibit substantially different geometric extents, while nearby objects or functional parts may be spatially nested. Consequently, appearance similarity or strong containment in only one direction can lead to incorrect merges. We introduce HC to make cross-view identity association more conservative for both object and functional-part instances. Let P_i denote an observation point set and Q_j an accumulated instance

point set. We measure how well their geometric extents support each other in both directions. With $B_\delta(\cdot)$ denoting an axis-aligned bounding box expanded outward by δ, we define,

$$c(P_i \to Q_j) = \frac{1}{|P_i|} \sum_{\mathbf{p} \in P_i} \mathbb{1}[\mathbf{p} \in B_\delta(Q_j)], \quad h(P_i, Q_j) = \sqrt{c(P_i \to Q_j) c(Q_j \to P_i)}. \tag{1}$$

The geometric mean suppresses matches supported by strong local containment in only one direction. We retain the spatial and appearance filtering used by FunFact and additionally require, $h(P_i, Q_j) \geq \tau_h$. For the surviving candidates $\mathcal{M}_i$, we select,

$$j_i^\star = \arg\max_{j \in \mathcal{M}_i} \left[(1-\lambda) s_{ij}^v + \lambda h(P_i, Q_j) \right], \tag{2}$$

where s_{ij}^v is the cosine similarity between appearance descriptors and λ balances appearance and geometric consistency. A matched observation is fused into $Q_{j_i^\star}$; otherwise, it initializes a new instance. Applying this consistency criterion to both object and part observations reduces erroneous cross-view merges and provides more stable entity identities for subsequent relation reasoning.

Joint relation reasoning calibration (RRC). Given associated instances, FunFact infers functional relations from semantic and spatial evidence within a factor-graph formulation. Such evidence can be ambiguous for repeated structures: when similar functional parts lie near multiple same-type targets, candidate relations may receive comparable scores. For vertically arranged handles and drawers, absolute distance may not reliably resolve correspondences, whereas relative ordering provides an additional structural cue. We therefore jointly calibrate candidate relations by incorporating relative-rank consistency into the original relation reasoning framework.

For a relation group g expected to follow one-to-one correspondences, we determine a ranking direction $\ell_g^\star$ from the spatial arrangement of its targets. The K targets are assigned ranks $r_j \in \{0, \dots, K-1\}$ in ascending coordinate order. Using the target coordinate range $[a_{\min}, a_{\max}]$, we map the source center coordinate $\mu_{i,\ell_g^\star}$ into the same rank space,

$$\widehat{r}_i = (K-1)\, \mathrm{clip}_{[0,1]} \left(\frac{\mu_{i,\ell_g^\star} - a_{\min}}{a_{\max} - a_{\min}} \right), \quad o_{ij} = \mathrm{clip}_{[0.1,0.9]} \left(\exp\left[-\frac{|r_j - \widehat{r}_i|}{\sigma_o} \right] \right). \tag{3}$$

A smaller rank discrepancy therefore yields a stronger preference for the corresponding candidate relation. Let $x_{ij} \in \{0,1\}$ indicate whether candidate relation (i,j) holds, and define the rank potential $\psi_{ij}^o(x) = o_{ij}^x (1-o_{ij})^{1-x}$. Rather than calibrating each candidate independently, we incorporate this potential into the factor graph of FunFact,

$$p_g(\mathbf{x}) \propto \prod_{(i,j) \in \mathcal{E}_g} \psi_{ij}^p(x_{ij}) \prod_{(i,j) \in \mathcal{C}_g} \psi_{ij}^o(x_{ij}) \prod_{i \in \mathcal{S}_g} \psi^c \left(\sum_j x_{ij} \right) \prod_{j \in \mathcal{T}_g} \psi^c \left(\sum_i x_{ij} \right), \tag{4}$$

where ψ^p and ψ^c are the spatial and correspondence-count potentials inherited from FunFact, respectively, while ψ^o introduces our relative-rank calibration. The count potentials couple candidates sharing the same source or target and favor the expected one-to-one correspondence. We omit the rank potential when the targets do not admit a reliable ordering, while non-one-to-one relations retain the original FunFact scoring. Belief propagation yields the calibrated marginal probabilities of candidate relations. Together with the identities established by HC, these relations form the initial graph G_0 used for subsequent active completion.

3.4 Active Evidence Acquisition and Collaborative Repair Agents (Ac-Agents)

The initial graph may still miss functional parts due to insufficient observations. We propose Ac-Agents, which generate hypotheses from the current graph G_t and verify them through targeted visual investigation. Ac-Agents use two multimodal model sessions as the **Proposer** and **Reviewer**. They share the graph, candidate evidence, and explicit feedback, while maintaining separate conversation histories. Candidates are progressively refined through multi-view evidence acquisition and review. Review feedback guides subsequent investigation, and proposed updates are incorporated into the graph only when supported by sufficient evidence and validated by consistency checks.

Hypothesis-driven multi-view evidence acquisition. Because missing parts have not yet formed graph nodes, Ac-Agents search for further evidence from existing objects and their observations. The proposer selects an object p to inspect based on the instances, functional relations, and related images in the current graph G_t. It then considers the existing parts of p and previous attempts to formulate an open-vocabulary hypothesis $h = (p, \ell, \varphi, \mathcal{Q})$, where ℓ denotes the candidate part name, φ denotes its functional relation to object p, and $\mathcal{Q}$ is the set of text queries used to localize the part. To verify the hypothesis, the proposer selects the query, view, and observation scale, and invokes the perception module to obtain 3D fragments from the original RGB-D observations. Fragments from different views are grouped into the same physical candidate only if they are pairwise geometrically consistent. The resulting candidates are then deduplicated against existing nodes while preserving their observation evidence. Deduplication is based on instance identity and geometric location, allowing spatially distinct parts with the same name to remain separate.

Review feedback and role alternation. Candidates sharing a hypothesis, parent object, and functional relation are grouped in the r-th revision state ξ_r, together with the semantic description and supporting evidence. The two agents maintain separate interaction histories and serve as the proposer (P_r) and reviewer (R_r) in each round. The reviewer checks whether evidence supports each candidate's localization, semantic name, and functional association, and provides feedback f_r. If revision is required and executable actions and sufficient budget remain, the agents swap roles:

$$\begin{aligned}(P_{r+1}, R_{r+1}) &= (R_r, P_r),\\ \xi_{r+1} &= P_{r+1}\big(G_t, \xi_r, \mathcal{E}_{r+1}, f_r; \mathcal{M}^{P_{r+1}}\big)\,.\end{aligned} \tag{5}$$

where $\mathcal{M}^{P_{r+1}}$ denotes the new proposer's retained interaction history and $\mathcal{E}_{r+1}$ denotes the evidence available for revision. Guided by f_r, the new proposer may revise the semantic description, remove incorrect candidates, or acquire additional evidence. If all candidates are rejected but the part hypothesis remains plausible, the proposer may reformulate the query and restart the search.

Verification-constrained graph update. To support the review process above, the system organizes the new nodes and their functional edges in a candidate revision ξ_r into an update Δ_r. The update is first applied to a copy of the current graph, producing a trial graph $\widetilde{G}_r = G_t \oplus \Delta_r$. This trial execution is used to examine how the candidate update affects the graph structure, and its result is considered together with the visual evidence when deciding whether to accept the revision. Specifically, $S_r = 1$ indicates that all retained candidates receive explicit semantic support from the reviewer, while $V_r = 1$ indicates that all checks on multi-view support, instance identity, relation endpoints, and structural consistency are satisfied. Each newly added part must also be supported by at least two distinct views. The grouped node and functional-edge updates are committed to the working graph only when both conditions hold:

$$G_{t+1} = \begin{cases}\widetilde{G}_r, & S_r = 1 \wedge V_r = 1,\\ G_t, & \text{otherwise.}\end{cases} \tag{6}$$

Rejected updates leave the current graph unchanged. If executable revisions remain and the budget allows, the verification feedback is passed back to the role-alternation process to guide the next revision ξ_{r+1}. Otherwise, the current hypothesis is terminated. Evidence acquisition, review, and retry actions all consume the shared cumulative budget. The final output is the functional 3D scene graph containing all accepted updates.

4 EXPERIMENTS

We evaluate our method in terms of object detection and functional relation prediction, and use baseline comparisons and ablation studies to analyze the contribution of each module. Qualitative results further demonstrate the model's ability to capture fine-grained functional parts and their functional relations. These include within-object relations, such as a drawer handle used to open a drawer, and cross-object relations, such as a remote control used to control a television. Through active evidence acquisition and evidence-driven repair, the model can further discover functional parts missed in the initial observations and refine their corresponding functional relations.

Table 1: Instance reconstruction results (%) on three benchmarks. Recall@10 is reported on FunGraph3D and SceneFun3D, while the original instance mapping metrics are used on FunThor. Obj, Int, and Ovr denote objects, functional parts, and all instances, respectively.

	FunGraph3D			SceneFun3D			FunThor		
Method	Obj@10	Int@10	Ovr@10	Obj@10	Int@10	Ovr@10	MapObj	MapInt	MapOvr
Open3DSG	58.1	33.9	43.6	70.7	61.8	64.7	–	–	–
ConceptGraph	66.3	4.1	25.2	77.1	8.6	31.4	–	–	–
ConceptGraph + IED	66.3	33.4	45.0	77.1	59.5	66.0	–	–	–
OpenFunGraph	79.1	57.6	65.8	87.8	**79.5**	82.8	54.6	41.1	51.2
FunFact	**96.6**	78.7	86.2	93.3	78.8	83.6	**68.2**	69.5	68.5
Ours	95.6	**80.5**	**87.4**	**96.4**	79.0	**85.2**	67.7	**85.1**	**72.1**

4.1 Experimental Setup

Benchmarks. We evaluate our method on FunThor (Fu et al., 2026), FunGraph3D (Zhang et al., 2025), and SceneFun3D (Delitzas et al., 2024) to assess object detection and functional relation prediction in both synthetic and real-world scenes. **FunThor** is a synthetic benchmark built on AI2-THOR (Kolve et al., 2017). It contains 12 indoor scenes with 621 object/part nodes, including 92 functional parts and 164 functional relations. Each scene provides 60 posed RGB-D frames. **FunGraph3D** contains 14 real-world indoor scenes with 146 target objects, 201 functional parts, and 228 open-vocabulary functional relations. **SceneFun3D** contains 710 real-world indoor scenes with more than 14.8K interaction annotations. Here, we follow the OpenFunGraph setting and evaluate on 20 scenes with functional relation annotations, including 8 validation scenes and 12 test scenes. These scenes contain 105 target objects, 212 functional parts, and 195 functional relations.

Evaluation metrics and protocols. We follow the evaluation protocol of FunFact (Fu et al., 2026). We first match predicted objects and functional parts to ground-truth instances using 3D spatial overlap and CLIP (Radford et al., 2021) semantic similarity, and report Recall@K for instance mapping. We then evaluate functional relation prediction. A relation is evaluated only when both of its endpoints are correctly matched. BERT (Devlin et al., 2019) semantic similarity is used to determine whether the predicted relation matches the ground-truth relation, from which relation-level Recall@K is computed. For FunThor, which provides more complete relation annotations, we additionally report Precision, Recall, and F1. For FunGraph3D and SceneFun3D, we follow their original settings and report recall-based metrics only.

Implementation details. We follow the setup of FunFact (Fu et al., 2026) and use GPT-4.1 for initial scene analysis and relation proposal. During active repair, both the Proposer and the Reviewer use GPT-5.6 Luna. They maintain separate interaction histories and exchange roles when further revision is needed. Each candidate is allowed up to four rounds of proposal and review. The repair process stops when the candidate is verified, when no further valid evidence can be obtained, or when the round budget is exhausted. We set the HC consistency threshold to 0.3, the cycle-score weight to 0.5, and the RRC scaling parameter to 0.7. A candidate graph update is committed only if it is supported by multi-view evidence and passes consistency verification.

4.2 Comparison with State-of-the-Art Methods

We compare our method with Open3DSG, ConceptGraph, OpenFunGraph, and FunFact. We evaluate the final functional 3D scene graphs in terms of instance reconstruction and functional relation prediction, with a particular focus on whether fine-grained functional parts can be recovered and further composed into valid functional triplets.

As shown in Table 1, our method consistently improves fine-grained functional part reconstruction. The improvement is most evident on FunThor, where MapInt increases from 69.5% to 85.1%, while object mapping stays close to FunFact. Similar trends are observed on FunGraph3D and SceneFun3D. These results suggest that the gains mainly come from recovering functional parts that are easily missed during initial graph construction, rather than simply detecting more ordinary objects.

More complete instance reconstruction should also benefit functional relation reasoning. Table 2 shows corresponding gains in overall triplet recall. Compared with FunFact, Ovr@10 increases from

Table 2: Functional triplet prediction results (%) on FunGraph3D and SceneFun3D. NodeAsc, EdgePr, and Ovr denote node association, functional relation prediction, and overall triplet Recall@K, respectively.

	FunGraph3D						SceneFun3D					
Method	NodeAsc@5	NodeAsc@10	EdgePr@5	EdgePr@10	Ovr@5	Ovr@10	NodeAsc@5	NodeAsc@10	EdgePr@5	EdgePr@10	Ovr@5	Ovr@10
OpenFunGraph	45.8	49.3	65.1	**91.4**	29.8	45.0	68.3	73.0	**88.1**	**96.2**	**60.4**	**70.3**
FunFact	71.1	80.0	67.9	79.9	48.7	63.9	62.1	71.8	66.1	80.7	41.0	57.9
Ours	**75.2**	**82.2**	**71.7**	82.0	**53.9**	**67.4**	**71.8**	**75.9**	63.6	85.8	45.6	65.1

63.9% to 67.4% on FunGraph3D and from 57.9% to 65.1% on SceneFun3D. The improvement on SceneFun3D is accompanied by gains in both NodeAsc@10 and EdgePr@10. These results suggest that improved instance recovery supports functional relation reasoning by providing valid endpoints for more complete functional triplets.

4.3 Ablation Study

We conduct component-wise ablations on FunThor to examine the contributions of HC, RRC, Ac-Agents, and Role Swap. Following FunFact, we use ECE-All and ECE-Amb. to evaluate confidence calibration over all and ambiguous relations, respectively. As shown in Table 3, HC and RRC improve the initial graph, while Ac-Agents further recover missing functional structures. Adding Ac-Agents improves Int.@3 from 78.0% to 85.1% and further increases relation Recall, showing that active evidence acquisition extends the functional coverage of the initial graph. Fig. 4 further shows that both interactive-part Recall and triplet Recall increase consistently as the evidence-acquisition budget grows, indicating that additional evidence progressively contributes to recovering functional structures unresolved in the initial graph. Removing Role Swap degrades both instance mapping and relation prediction, supporting its role in reliable candidate search and verification.

The increased coverage from Ac-Agents comes with some loss in Precision. We find that some newly recovered predictions are consistently supported across multiple views and are functionally meaningful, yet are not annotated at the same granularity in the Ground Truth. For example, the cup handle and its *grasp to lift and carry* relation recovered in Fig. 3 are not independently annotated, despite providing useful part-level information for interaction. Such predictions are still counted as errors under the current protocol and may also affect confidence calibration. Nevertheless, our method achieves the best ECE-All, while the remaining gap on ECE-Amb. indicates that confidence calibration remains more challenging for highly ambiguous relations.

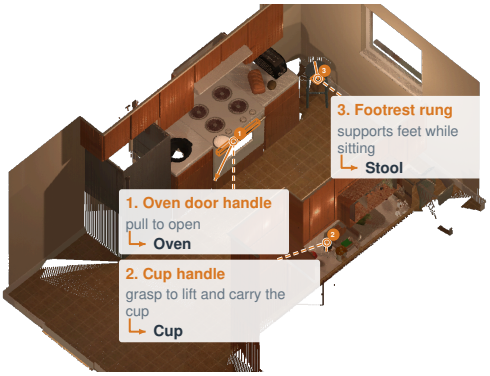

Figure 3: Visually supported functional structures beyond the Ground Truth. AcFunSG recovers fine-grained functional parts and relations supported by visual evidence but not independently annotated in the Ground Truth.

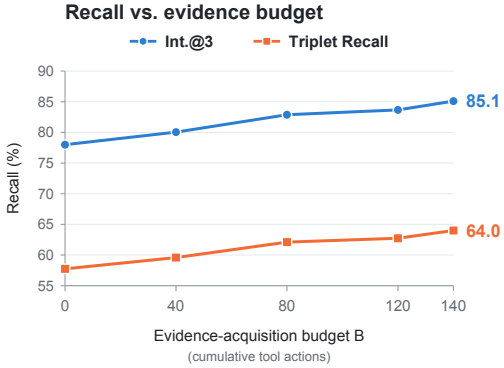

Figure 4: Effect of the evidence-acquisition budget. Interactive-part Recall and triplet Recall consistently improve as the evidence-acquisition budget increases, indicating that additional evidence helps recover missing functional structures.

4.4 Qualitative Results

As shown in Fig. 5, AcFunSG completes and corrects the functional scene graph constructed by FunFact. For the cabinet handle, the Proposer hypothesizes a potentially missing part, while the Reviewer identifies ambiguity in its localization and association with the cabinet and requests further

Table 3: Ablation results on FunThor. We report instance mapping quality, functional relation prediction performance, and confidence calibration error.

	Mapping			Functional Graph			Calibration	
Methods	Obj.@3↑	Int.@3↑	Ovr.@3↑	Prec.↑	Recall↑	F1↑	ECE-All↓	ECE-Amb.↓
OpenFunGraph	54.6	41.1	51.2	23.4	12.2	16.0	0.43	0.51
FunFact	**68.2**	69.5	68.5	31.9	49.3	38.7	0.36	**0.07**
AcFunSG (Full)	67.9	**85.1**	**72.2**	31.3	**64.0**	42.0	**0.35**	0.12
HC + RRC only	67.7	78.0	70.3	**35.5**	57.8	**44.0**	**0.35**	0.11
Ac-Agents only	67.7	79.4	70.6	27.7	58.4	37.1	0.39	0.23
HC only	67.7	78.0	70.3	33.5	57.1	42.2	0.37	0.15
RRC only	67.5	73.8	69.0	31.0	49.7	38.2	0.39	0.20
HC + Ac-Agents	67.7	83.0	71.5	28.8	61.5	39.2	0.37	0.15
RRC + Ac-Agents	67.5	78.7	70.3	26.6	54.0	35.7	0.39	0.20
w/o Role Swap	67.9	80.1	71.0	29.2	61.5	39.6	**0.35**	0.13

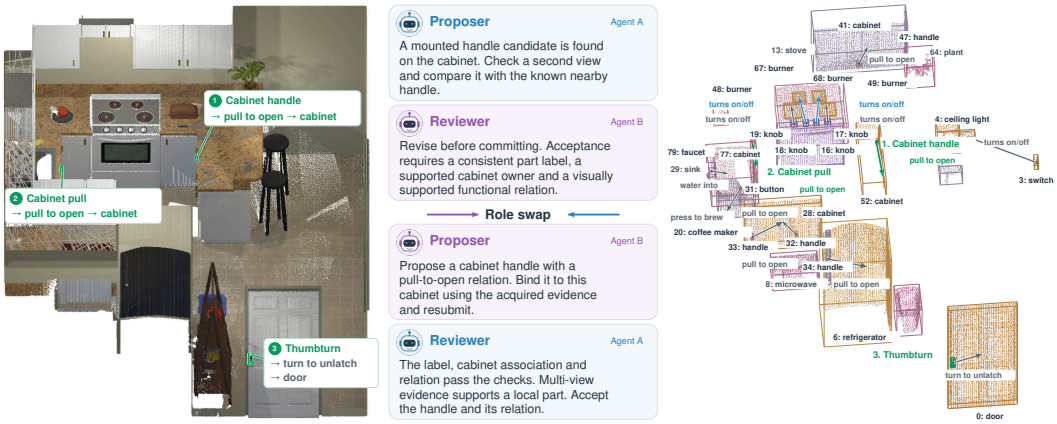

Figure 5: **Qualitative visualization of AcFunSG.** The left shows newly recovered functional structures in the 3D scene, the middle shows a representative Ac-Agents interaction with proposal, review, and Role Swap, and the right shows the repaired scene graph. Green marks newly added nodes or relations, blue marks corrected predictions, and gray marks structures retained from FunFact.

verification. After Role Swap, the two agents exchange proposal and review roles and revisit the candidate using additional views, ultimately recovering the part and its *pull to open* relation with the cabinet. The right side of the figure shows the repaired scene graph, where missing functional parts and relations are recovered and some existing relation predictions are corrected. This example illustrates how active evidence acquisition, role alternation, and graph revision form a closed loop that resolves omissions and ambiguities left by the initial construction.

5 Conclusion

We presented AcFunSG, a training-free framework that integrates initial inference and active repair for open-vocabulary functional 3D scene graph construction. Initial inference combines bidirectional geometric consistency for cross-view instance association with relative ordering cues for joint functional relation reasoning. Within Ac-Agents, two role-alternating agents propose hypotheses for missing parts, acquire multi-view evidence, and review updates. Review feedback guides investigation and revision, closing the loop between graph reasoning and evidence acquisition. Only updates supported by visual evidence and validated by consistency checks are incorporated into the graph. Experiments on FunGraph3D, SceneFun3D, and FunThor demonstrate state-of-the-art overall instance recall and improved functional triplet recall. These results highlight the value of active completion: recovering missing functional parts provides endpoints for relation inference, translating improved instance coverage into a more complete functional scene representation.

REFERENCES

Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5664–5673, 2019.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Junhao Chen, Xiang Li, Mingjin Chen, Boran Zhang, Henghaofan Zhang, Yibin Xu, Yuehan Cui, Fangsheng Weng, Fei Ma, Qi Tian, et al. Paircoder++: Pair programming as a universal paradigm for verified code-driven multimodal and structured-artifact generation. *arXiv preprint arXiv:2607.01883*, 2026.

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.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Tim Engelbracht, René Zurbrügg, Marc Pollefeys, Hermann Blum, and Zuria Bauer. Spotlight: Robotic scene understanding through interaction and affordance detection. *arXiv preprint arXiv:2409.11870*, 2024.

Zhengyu Fu, René Zurbrügg, Kaixian Qu, Marc Pollefeys, Marco Hutter, Hermann Blum, and Zuria Bauer. Funfact: building probabilistic functional 3d scene graphs via factor-graph reasoning. *arXiv preprint arXiv:2604.03696*, 2026.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024.

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.

Xinggang Hu, Chenyangguang Zhang, Alexandros Delitzas, Xiangkui Zhang, Marc Pollefeys, Francis Engelmann, and Xiangyang Ji. Hierarchical and holistic open-vocabulary functional 3d scene graphs for indoor spaces. *arXiv preprint arXiv:2605.15753*, 2026.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. *arXiv preprint arXiv:2201.13360*, 2022.

Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678, 2015.

Yuanchen Ju, Yongyuan Liang, Yen-Jen Wang, Gireesh Nandiraju, Yuanliang Ju, Seungjae Jay Lee, Qiao Gu, Elvis Hsieh, Furong Huang, and Koushil Sreenath. Momagraph: State-aware unified scene graphs with vision-language models for embodied task planning. In *International Conference on Learning Representations*, volume 2026, pp. 96361–96386, 2026.

Sebastian Koch, Narunas Vaskevicius, Mirco Colosi, Pedro Hermosilla, and Timo Ropinski. Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14183–14193. IEEE, 2024.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–824. IEEE, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. *arXiv preprint arXiv:2307.06135*, 2023.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Dennis Rotondi, Fabio Scaparro, Hermann Blum, and Kai O Arras. Fungraph: Functionality aware 3d scene graphs for language-prompted scene interaction. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4083–4090. IEEE, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11854–11864. IEEE, 2023.

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.

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.

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094. IEEE, 2024.

Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7511–7521. IEEE, 2021.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

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.

Haoyi Zhu, Honghui Yang, Yating Wang, Jiange Yang, Limin Wang, and Tong He. Spa: 3d spatial-awareness enables effective embodied representation. In *International Conference on Learning Representations*, volume 2025, pp. 26361–26391, 2025.

OVERVIEW OF APPENDICES

- Appendix A: Algorithm.
- Appendix B: Additional Mapping Evaluation.
- Appendix C: Additional Qualitative Results.
- Appendix D: Functional Generalization under Incomplete Annotations.
- Appendix E: Budget and Computational Cost.
- Appendix F: Limitations and Future Work.
- Appendix G: Prompt Templates.
- Appendix H: AI Use Statement

A ALGORITHM

Algorithm 1 summarizes the processing of a single hypothesis in Ac-Agents, corresponding to §3.4. Here, t indexes graph updates and r indexes revisions within the current hypothesis. Each revision state ξ_r contains the current candidates, their semantic descriptions, and supporting evidence. BUILDREVISION groups geometrically consistent observations across views and deduplicates the resulting candidates against existing graph nodes. BUILDUPDATE then converts the retained candidates into a grouped graph update Δ_r. The update is evaluated on a trial graph $\widetilde{G}_r = G_t \oplus \Delta_r$, while the working graph G_t remains unchanged until both reviewer support and deterministic verification succeed.

Algorithm 1 Ac-Agents evidence acquisition and verified graph update for one hypothesis.

Require: Current graph G_t, posed RGB-D observations $\mathcal{O}$, remaining budget B
Require: Initial roles P_0, R_0 with separate histories $\mathcal{M}^{P_0}, \mathcal{M}^{R_0}$
Ensure: Updated graph G_{t+1} and remaining budget B

$r \leftarrow 0$
P_0 selects an object p and proposes $h = (p, \ell, \varphi, \mathcal{Q})$
P_0 selects queries, views, and observation scales and acquires evidence $\mathcal{E}_0$ from $\mathcal{O}$
$\xi_0 \leftarrow$ BUILDREVISION$(G_t, h, \mathcal{E}_0)$
while the current hypothesis remains executable under B **do**
  $\Delta_r \leftarrow$ BUILDUPDATE(ξ_r)
  if $\Delta_r = \varnothing$ **then**
    if the revision limit is reached **or** no further search is executable under B **then**
      return (G_t, B)
    end if
  else
    $\widetilde{G}_r \leftarrow G_t \oplus \Delta_r$ {apply the update to a copy of G_t}
    $(S_r, f_r^{\text{rev}}) \leftarrow R_r(G_t, \xi_r, \mathcal{E}_r, \widetilde{G}_r; \mathcal{M}^{R_r})$
    $(V_r, f_r^{\text{ver}}) \leftarrow$ VERIFY$(G_t, \widetilde{G}_r, \xi_r, \mathcal{E}_r)$
    $f_r \leftarrow f_r^{\text{rev}} \cup f_r^{\text{ver}}$
    if $S_r = 1$ **and** $V_r = 1$ **then**
      return $(\widetilde{G}_r, B)$ {commit the grouped update}
    end if
    if the revision limit is reached **or** no executable revision remains under B **then**
      return (G_t, B)
    end if
  end if
  $(P_{r+1}, R_{r+1}) \leftarrow (R_r, P_r)$ {role alternation}
  P_{r+1} uses ξ_r and the available feedback to revise the search and acquire additional evidence $\mathcal{E}_{r+1}$ while retaining previous evidence
  $h_{r+1} \leftarrow P_{r+1}(G_t, \xi_r, \mathcal{E}_{r+1}, f_r; \mathcal{M}^{P_{r+1}})$
  $\xi_{r+1} \leftarrow$ BUILDREVISION$(G_t, h_{r+1}, \mathcal{E}_{r+1})$
  $r \leftarrow r + 1$
end while
return (G_t, B)

Table 4: **Mapping Quality under Exclusive Matching.** We report Recall@K (R@3/R@10; higher is better) for Objects, Functional Parts, and Overall Nodes on SceneFun3D, FunGraph3D, and FunThor. Under exclusive matching, each predicted node can be matched to at most one Ground Truth instance.

Methods	SceneFun3D						FunGraph3D						FunThor					
	Objects (↑)		Inter. Elements (↑)		Overall Nodes (↑)		Objects (↑)		Inter. Elements (↑)		Overall Nodes (↑)		Objects (↑)		Inter. Elements (↑)		Overall Nodes (↑)	
	R@3	R@10	R@3	R@10	R@3	R@10	R@3	R@10	R@3	R@10	R@3	R@10	R@3	R@10	R@3	R@10	R@3	R@10
OpenFunGraph	78.1	79.0	64.4	**78.3**	69.1	78.5	61.1	68.5	42.1	52.5	50.3	59.2	43.7	52.3	34.8	40.4	41.5	49.3
FunFact	90.5	90.5	64.4	75.5	73.2	80.4	84.9	89.0	62.9	75.2	72.1	81.0	**57.2**	**61.0**	64.5	68.8	59.1	63.0
AcFunSG (Ours)	**92.0**	**95.5**	**68.8**	76.6	**77.0**	**83.3**	**88.0**	**92.4**	**70.5**	**77.9**	**78.4**	**84.5**	56.5	60.6	**81.6**	**83.0**	**62.8**	**66.2**

All evidence acquisition, model calls, perception operations, review steps, and verification operations consume the shared cumulative budget B. For clarity, budget updates performed by these primitive operations are omitted from the pseudocode. Before each operation, the system checks whether sufficient budget remains; if not, the current hypothesis terminates and G_t remains unchanged. The two agents retain their separate interaction histories across role alternation.

VERIFY checks multi-view support, instance identity, relation endpoints, and structural consistency. Each newly added part must be supported by at least two distinct views. An empty candidate set cannot produce a graph update, but may trigger a new search when the hypothesis remains plausible and sufficient budget remains. The same procedure is applied to subsequent hypotheses until no executable hypothesis remains or the global budget is exhausted.

B ADDITIONAL MAPPING EVALUATION

Following FunFact (Fu et al., 2026), we further evaluate instance mapping under exclusive matching. In this setting, each predicted node can be matched to at most one Ground Truth instance, providing a stricter evaluation of instance recovery. Table 4 reports Recall@3 and Recall@10 for objects, functional parts, and overall instances on SceneFun3D, FunGraph3D, and FunThor.

Under exclusive matching, AcFunSG achieves the best overall performance across all three benchmarks. It obtains the highest overall Recall@3 and Recall@10 on SceneFun3D, FunGraph3D, and FunThor, demonstrating stronger recovery of complete scene structure under stricter instance correspondence. The improvement is particularly pronounced for functional parts. On FunThor, Recall@3 increases from 64.5% to 81.6%, while Recall@10 increases from 68.8% to 83.0%. These results show that AcFunSG effectively recovers functional parts missed during initial graph construction, leading to more complete functional scene representations.

C ADDITIONAL QUALITATIVE RESULTS

We provide additional qualitative results on SceneFun3D, FunGraph3D, and FunThor to complement the analysis in the main paper. Green marks newly recovered structures matched by the Ground Truth, blue marks corrected predictions, gray marks structures retained from the initial graph, and orange marks visually supported functional structures that are not annotated in the Ground Truth. The four examples cover different evidence conditions and illustrate how Ac-Agents varies the repair process from direct verification to iterative evidence acquisition and revision.

As shown in Fig. 6, the initial search produces a handle candidate whose localization and association remain ambiguous, prompting the Reviewer to request additional evidence from other views. After Role Swap, the new Proposer revises the search according to the review feedback and uses the newly acquired multi-view evidence to localize the refrigerator handle. Once verified, the handle and its *pull to open* relation are incorporated into the graph. This example shows how review feedback redirects subsequent evidence acquisition when the initial candidate remains uncertain.

In contrast, Fig. 7 shows a case that can be completed within a single review round. The Proposer identifies a missing cabinet pull with consistent localization and functional evidence across multiple views. The Reviewer confirms the candidate and its *pull to open* relation in the first round, allowing the update to be committed without further search or Role Swap. Together with the previous example, this shows that the repair process adapts to the available evidence rather than following a fixed number of revisions.

Fig. 8 further illustrates how Ac-Agents refines the localization and granularity of a functional-part candidate. The initial prediction covers a broader valve assembly rather than the operable valve knob. Reviewer feedback triggers a more localized search after Role Swap, and the new Proposer uses additional views to isolate the valve knob and recover its functional relation to the valve. Active evidence acquisition therefore supports not only the recovery of missing parts, but also the refinement of coarse candidates into more precise functional structures.

More challenging cases may require multiple rounds of revision. As shown in Fig. 9, the initial rack end cap candidate remains confused with competing rack regions. Successive Role Swaps use feedback from preceding rounds to redirect evidence acquisition and progressively refine the candidate until its localization and ownership are resolved. The final graph recovers its functional association with the towel rack. The orange structure is visually supported in the observed scene but is not annotated in the Ground Truth. Across these examples, Ac-Agents directly commits well-supported candidates while continuing evidence acquisition and revision when ambiguity remains.



Figure 6: **Qualitative result on FunGraph3D.** The initial handle hypothesis remains ambiguous after the first search, and the Reviewer requests additional evidence before the graph is updated. After Role Swap, the new Proposer revises the search across additional views and localizes the refrigerator handle more accurately. The verified update recovers the handle and its *pull to open* relation, while the final graph also contains other newly recovered functional structures.

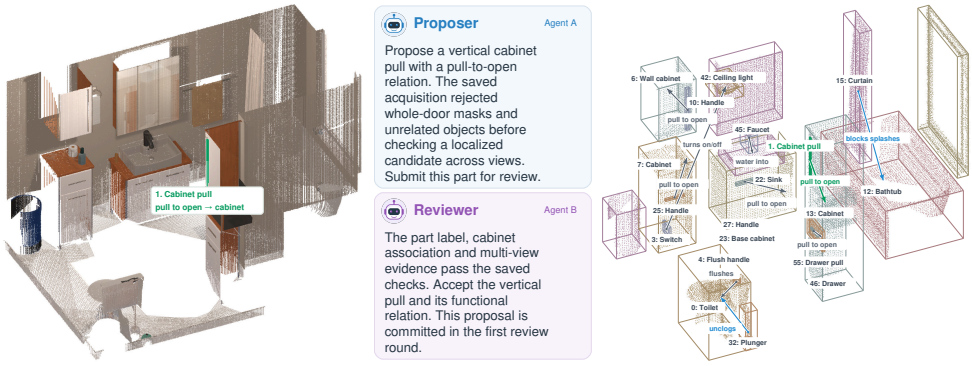

Figure 7: **Single-round repair on FunThor.** The Proposer identifies a missing cabinet pull whose localization and association are consistently supported across multiple views. The Reviewer confirms the candidate and its *pull to open* relation in the first review round, allowing the update to be committed without further revision. This example illustrates the direct verification path when the available evidence is already sufficient.

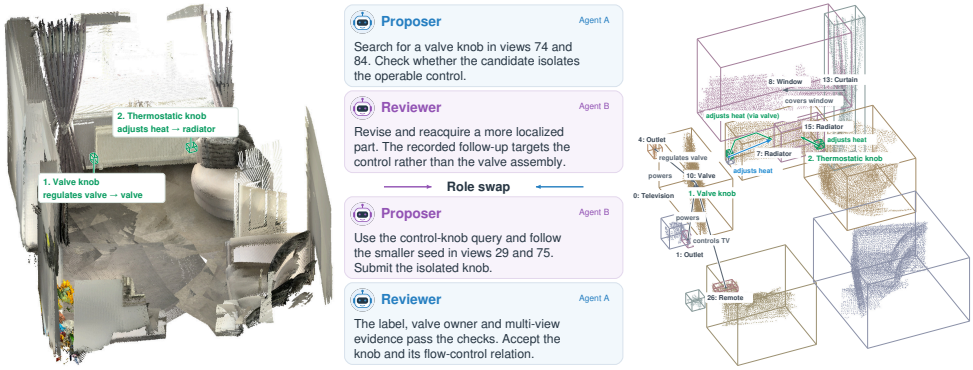

Figure 8: **Qualitative result on SceneFun3D.** The initial candidate covers a broader valve assembly instead of the operable control itself. Reviewer feedback triggers a more localized search after Role Swap, and additional views allow the new Proposer to isolate the valve knob. The resulting update recovers the finer-grained functional part together with its functional relation to the valve.

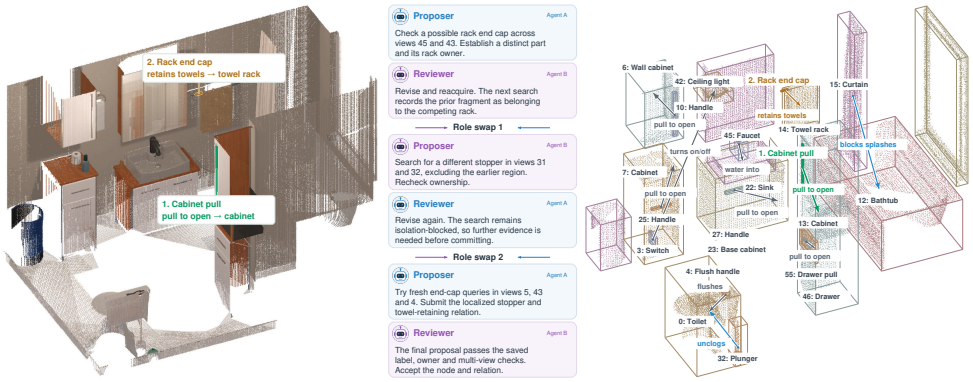

Figure 9: **Multi-round repair on FunThor.** The initial rack end cap hypothesis remains ambiguous because the candidate overlaps with competing rack regions. Review feedback drives successive Role Swaps and additional evidence acquisition, progressively refining the candidate until its localization and ownership are resolved. The final graph contains the recovered rack end cap and its functional association with the towel rack. The orange structure denotes a visually supported functional prediction that is not annotated in the Ground Truth.

D FUNCTIONAL GENERALIZATION UNDER INCOMPLETE ANNOTATIONS

Existing benchmark annotations do not exhaust all interaction-relevant parts and functional relations present in a scene. We therefore further examine the functional structures predicted by AcFunSG beyond the available Ground Truth annotations. These predictions correspond to visually identifiable object parts and describe how they naturally participate in object manipulation and everyday interaction.

As shown in Fig. 10, AcFunSG identifies a lid knob that can be lifted to remove a pot lid and a loop carrying handle that can be grasped to lift and carry a laundry basket in FunGraph3D. Both predictions associate visible, operable parts with their corresponding interaction functions, extending the scene graph with functional knowledge beyond the annotated structures.

A similar pattern appears in FunThor. Fig. 11 shows that AcFunSG recovers a touchpad used to control a laptop pointer, a toggle used to turn a light on or off, and a drawer pull used to open a cabinet drawer. These predictions go beyond identifying the parts themselves by explicitly describing how each part participates in a concrete interaction, providing finer-grained functional semantics for the scene.

Fig. 12 further shows examples from SceneFun3D. AcFunSG identifies a door hinge that enables the door to pivot open and closed and a remote-control button that can be pressed to issue a command. Although these functional structures do not have corresponding annotations in the Ground Truth, they are supported by visible scene structure and are consistent with common human–object interaction knowledge.

Taken together, these examples show that the functional predictions of AcFunSG are not limited to structures explicitly covered by benchmark annotations. The method can further recover fine-grained, visually grounded functional knowledge that captures meaningful ways in which object parts support interaction, demonstrating functional generalization beyond the available annotation coverage.



Figure 10: **Functional predictions beyond the annotated structures on FunGraph3D.** AcFunSG recovers a lid knob whose function is to lift and remove the pot lid and a loop carrying handle used to grip, lift, and carry a laundry basket. Both examples associate visible functional parts with natural object manipulation patterns despite the absence of corresponding functional Ground Truth annotations.

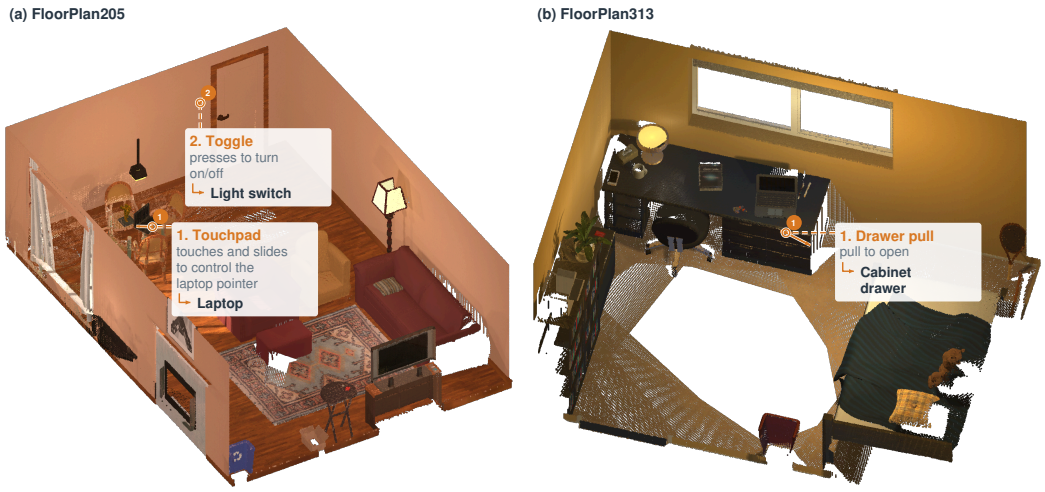

Figure 11: **Functional predictions beyond the annotated structures on FunThor.** AcFunSG identifies interaction-relevant parts including a laptop touchpad, a light-switch toggle, and a cabinet drawer pull, together with their corresponding control, switching, and opening functions. These predictions capture fine-grained ways in which visible parts participate in everyday object interaction.



Figure 12: **Functional predictions beyond the annotated structures on SceneFun3D.** AcFunSG recovers a door hinge that supports the opening and closing motion of a door and a remote-control button used to issue a command. The recovered relations are grounded in visible object structure and align with common human–object interaction patterns.

E BUDGET AND COMPUTATIONAL COST.

To consistently control the inference cost of active repair, we use the same scene-level global budget across all datasets, allowing at most 500 logical model calls per scene. For each hypothesis, Ac-Agents performs at most four rounds of proposal and review. If a candidate requires further verification, subsequent evidence acquisition, renewed search, and candidate refinement continue to consume the remaining global budget of the scene. The current hypothesis terminates once the proposed update is verified, no executable revision remains, or the scene-level budget is exhausted. Therefore, the 500-call budget serves as a uniform upper bound for the complete active repair process rather than a fixed computational allocation for each hypothesis.

Under this unified budget, active repair requires an average of 194.5 logical model calls per scene on FunThor and 332.6 on FunGraph3D, while the average on the SceneFun3D test split is 403.5. Logical calls are counted according to model requests issued by the agents. Requests resolved through cache hits without actual model execution and requests that trigger actual model execution are recorded separately. We further report the proportion of scenes that reach the global budget limit to examine whether the budget constrains the repair process in practice across different datasets.

F LIMITATIONS AND FUTURE WORK

AcFunSG is limited by the coverage and quality of the available posed RGB-D observations. Active search revisits existing views, while the requirement for support from at least two views may leave sparsely observed parts unresolved. Perception and geometric errors can also affect candidate verification. Future work could integrate embodied viewpoint planning and uncertainty-aware association to obtain more informative evidence.

Although deterministic verification enforces structural consistency, semantic acceptance still depends on multimodal model judgments. The Proposer and Reviewer share the same underlying model and may retain common biases despite role alternation. Moreover, visual evidence alone cannot fully establish actual functional effects or state-dependent affordances. Incorporating physical interaction feedback and complementary verification models could strengthen functional reliability.

Iterative acquisition and review introduce additional inference cost, motivating more efficient budget allocation and evidence reuse. Evaluation also reveals a precision–recall trade-off, which is complicated by incomplete functional annotations; unmatched predictions cannot all be assumed correct. Future work should assess scalability in larger scenes and combine more comprehensive annotations with downstream manipulation tasks to evaluate the practical utility of the completed graphs.

G PROMPT TEMPLATES

We provide prompt templates for hypothesis proposal, evidence acquisition, and candidate review in §3.4. The Proposer formulates a hypothesis and selects actions to investigate it, while the Reviewer assesses the resulting candidates and provides feedback for subsequent revisions. The two agents alternate roles across revisions.

The templates share instructions for component granularity and detection-query generation, which are inserted at the corresponding bracketed placeholders. Visual observations, candidate states, and previous feedback are supplied as task-specific context.

Hypothesis Proposal

You are the PROPOSER in an open-vocabulary functional 3D scene graph completion and repair process.

Given a selected parent object, its multi-view visual context, existing instances, and previous attempts, propose ONE testable hypothesis about a functional component that is not yet represented in the graph. Choose `DEFER` when the observations do not provide a concrete basis for further investigation.

Hypothesis Proposal (continued)

Ground the hypothesis in the supplied observations. Identify a physically distinguishable component and describe the functional effect it enables, controls, or mediates in relation to the selected parent. Category expectations, proximity, and containment alone are insufficient.

`[GRANULARITY_CONTRACT]`

Use `already_known_instances` to distinguish physical instances. A component at a different location may share an existing label. Do not propose an already represented instance under a synonym.

The raw label should name the component. The relation phrase should describe its specific functional association with the parent. Keep both at the level of detail supported by the observations.

`[DETECTION_QUERY_CONTRACT]`

Generate component names and relation phrases freely from the observations. Do not select them from dataset categories or predefined semantic slots. Refer only to supplied instance IDs. Do not introduce ground-truth information, confidence scores, or graph operations.

Return only the fields required by the supplied output schema.

Evidence Acquisition

You are the PROPOSER acquiring visual evidence for one open-vocabulary functional-part hypothesis.

Select exactly ONE action from `legal_actions`. Use the current acquisition state, previous tool results, and review feedback to identify the uncertainty that the action should resolve.

Distinguish three questions throughout the investigation.

1. Which physical region has been localized?
2. Do observations from different views correspond to the same physical instance?
3. Does that instance support the proposed component name and functional relation?

The hypothesis and detection queries specify what to investigate. They do not establish what is present in the observations. A `VERIFIED` tool status does not by itself establish semantic or functional correctness.

Prefer an action that addresses a specific unresolved issue, such as obtaining a complementary view, improving localization through an available crop or observation mode, or selecting a permitted query variant after a failed search.

Use previous attempts to avoid repeatedly confirming the wrong region. A new seed ID or different query wording does not necessarily identify a new physical instance. Repeat an action on unchanged inputs only when there is a concrete recovery need.

Respect the controller's success and stopping state. When no available action can provide useful evidence, select a permitted stop or defer action.

Use only the queries, views, modes, and identifiers supplied by the current action interface. Query reformulation must follow the permitted revision or rediscovery process. Do not fabricate detections, observations, confidence scores, or graph edits.

Return only the fields required by the supplied output schema.

Candidate Review

You are the REVIEWER assessing one grouped candidate revision in an open-vocabulary functional 3D scene graph completion and repair process.

Return exactly one `CandidateAssessment` for each active `candidate_id`, followed by one whole-patch verdict. Include every active candidate exactly once and use only the supplied identifiers.

Assess each candidate using its own visual evidence, the parent context, and the deterministic trial results. Keep semantic judgments separate from structural verification.

Label assessment

Identify what the localized region visibly supports before assessing the proposed name. Determine whether the region's appearance and spatial extent are consistent with that name. The detection query, the expected presence of a component, and a `VERIFIED` tool status are not sufficient evidence.

Agreement across views can establish that observations depict the same region while leaving its semantic identity unresolved. Report a clear mismatch when the evidence supports one. Otherwise, record uncertainty using the existing assessment fields.

`[GRANULARITY_CONTRACT]`

Relation assessment

Determine whether this particular candidate supports the proposed functional association with the specified parent. Consider the visible component, its context, and the available trial results. Proximity, containment, and a plausible relation phrase alone are insufficient. Physical attachment is not required for an external control.

Treat the parent name and localization as predictions that also require visual support. If the parent is unclear or mislocalized, report the relevant uncertainty or mismatch. Leave changes to labels, relations, and parent assignments to the Proposer.

Visual evidence assessment

Check whether the supplied views adequately show the candidate's location, extent, and identity. Evaluate whether observations from different views are consistent with the same physical instance and provide enough information for the semantic assessment. A small localized region is not automatically correct.

Candidate actions

`KEEP` when the evidence supports retaining the candidate.

`EXCLUDE_FROM_REVISION` when the candidate is a clearly incompatible entity or physical instance.

`REQUEST_EVIDENCE` when the intended candidate remains plausible and a specific additional observation could resolve the uncertainty.

For each assessment, briefly state the supporting observation, any remaining mismatch or uncertainty, and a concrete next step when one is available. Use the supplied schema's false or unknown values for unsupported conclusions. Do not invent observations or replacement identifiers.

Whole-patch verdict

`ACCEPT` when every active candidate is `KEEP`, all required semantic assessments are true, and the deterministic trial permits commitment.

`REVISE` when an executable candidate correction or evidence request remains. If all current candidates are excluded but the hypothesis remains plausible, allow the Proposer to initiate `REDISCOVER` through the supported action interface.

Candidate Review (continued)

`REJECT` only when `hypothesis_invalid=true` is supported by affirmative evidence against the hypothesis itself. Failed localization, insufficient visibility, or a failed trial does not establish that the hypothesized component is absent. Do not combine `REJECT` with `REQUEST_EVIDENCE` or a fully supported retained candidate.

`DEFER` when uncertainty remains and no executable correction or evidence-acquisition action is available.

A failed deterministic trial prevents `ACCEPT`. Preserve the semantic assessments and use the trial feedback to identify possible revisions. The controller checks the consistency of the candidate assessments and whole-patch verdict before any graph update is committed.

Return only the fields required by the supplied output schema.

Shared Granularity Instructions

Identify the functional component at the spatial and semantic granularity supported by the observations.

The proposed name, localized region, and functional relation must refer to the same physical entity. The candidate's observed extent should match the extent implied by its proposed name.

Do not substitute the parent object or a larger surrounding structure for a poorly localized component. Evidence for the surrounding object does not establish the presence or identity of the hypothesized subcomponent.

Assess a larger or articulated component according to its proposed function and observed extent. Its size or category name alone should not determine whether it is valid.

Use component names supported by the observations without restricting proposals to a predefined category list. If the component's extent or identity remains ambiguous, request a more informative observation or defer the hypothesis.

Shared Detection Query Instructions

Generate 3–5 short `query_variants` for GroundingDINO retrieval. All variants must refer to the same intended physical component.

Begin with a concise noun phrase naming the hypothesized component. Add semantically equivalent expressions or descriptions grounded in the available visual evidence, keeping all variants focused on the same physical target. If parent context is needed, include it in at most one query and place that query last.

Prefer component-focused queries because the image crop already supplies parent context. Query variants should change how the same component is described without shifting the target to another component or a larger surrounding structure.

Use noun phrases rather than complete sentences for the queries, and place the functional description in `relation_phrase`.

The query variants provide alternative retrieval expressions for one hypothesis; they do not introduce additional hypotheses.

H AI USE STATEMENT

Generative AI tools were used to assist with literature retrieval, research ideation and discussion, drafting selected portions of the manuscript, and improving writing clarity. All AI-assisted con-

tent, including research ideas, experimental interpretations, and manuscript text, was reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.