TASK-GROUNDED OPEN-VOCABULARY 3D SCENE GRAPH INCREMENTAL GENERATION FOR TASK PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

For a robot operating in an indoor 3D scene, a planning task is inherently local: it is defined over a small subset of entities, relations, and states that the goal depends on, rather than over the scene as a whole. Robotic task planning therefore demands task-grounded, on-demand scene understanding—yet existing 3D scene graph pipelines do the opposite, consuming all frames and exhaustively enumerating objects and relations before any task is known, on a closed scene graph with predefined categories and structure. This global, task-agnostic construction is mismatched to planning: in the 2D-to-3D reconstruction process, more frames accumulate larger back-projection and registration errors, which instead amplify object recognition bias; pose drift corrupts the association of small interaction elements; and computation is wasted on task-irrelevant regions. We propose Task-Grounded Open-Vocabulary 3D Scene Graph Incremental Generation (TG-OSG), which reframes the scene graph into an open representation generated on demand. TG-OSG (1) builds a coarse open scene graph from a few dominant-viewpoint frames, (2) incrementally refines it by acquiring additional frames only for task-relevant regions and discovering novel entities and relations, and (3) terminates once the task is determined to be executable—yielding an execution sequence—or infeasible. We evaluate on three public 3D scene graph datasets together with a self-built simulation for task-grounded scene graph generation, and demonstrate significant improvements and advantages over existing closed and open SGG methods on both open scene graph generation and the corresponding task planning[1].

1 INTRODUCTION

Robots operating in complex 3D environments do not need to understand everything around them to accomplish a task. Instead, they need to identify the entities, relations, and states that are relevant to the current task and ground them to executable actions Shridhar et al. (2020); Ahn et al. (2022); Huang et al. (2023); Singh et al. (2023). For example, placing a cup into a cabinet requires the cup, the cabinet, their spatial relation, and the free space needed for manipulation, while most other objects are irrelevant. This task-dependent nature of embodied interaction suggests that robotic scene understanding should focus on acquiring the information necessary for the current task, rather than exhaustively reconstructing the entire environment.

Existing 3D scene graph generation (SGG) methods largely follow a task-agnostic paradigm: given RGB-D or multi-view observations, they construct a global scene graph by detecting entities and inferring their spatial and semantic relations (Wald et al., 2020; Wu et al., 2021; 2023). While effective for building comprehensive scene representations, this paradigm treats scene reconstruction as the primary objective, regardless of the downstream task. As illustrated in Figure 1, processing all available observations can introduce unnecessary computation and accumulate registration and pose errors, while producing substantial task-irrelevant information. More fundamentally, **scene completeness does not imply decision sufficiency**: a globally complete graph may contain information

[1] Our code is available at https://anonymous.4open.science/r/TASK-GROUNDED-OPEN-VOCABULARY-3D-SCENE-GRAPH-INCREMENTAL-GENERATION-FOR-TASK-PLANNING-130B/README.md.

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

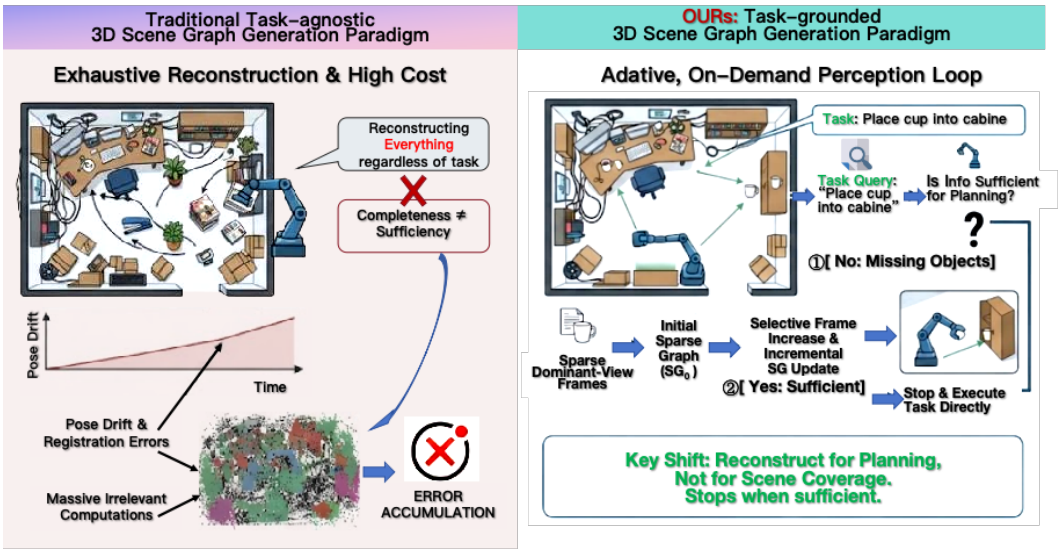

Figure 1: **Left:** Traditional task-agnostic methods perform exhaustive global reconstruction, suffering from error accumulation and the mismatch between scene completeness and decision sufficiency. **Right:** We propose the task-grounded incremental paradigm that adaptively acquires observations and stops once sufficient for planning.

irrelevant to the current task, whereas a partial graph may already provide sufficient evidence for planning.

Recent work has progressively extended 3D scene representations toward open-vocabulary semantics and task-oriented reasoning. Open-vocabulary methods align 3D entities with vision-language representations, enabling free-form semantic queries beyond predefined categories (Peng et al., 2023; Jatavallabhula et al., 2023; Chang et al., 2023; Gu et al., 2024; Koch et al., 2024), while recent approaches further incorporate functional semantics, open-world discovery, and robotic interaction (Koch et al., 2024; Linok et al., 2025; Fei et al., 2026). In parallel, task-oriented methods demonstrate that reasoning over task-relevant scene structures can facilitate robotic planning (Agia et al., 2022; Amiri et al., 2022; Rana et al., 2023). Despite these advances, existing approaches primarily improve *what can be represented* or *what can be reasoned over* from the available observations. They do not explicitly address *what should be observed next* when the current scene graph lacks information required by the task, nor *when observation should stop* once sufficient information has been acquired. This motivates a task-grounded formulation in which the task itself guides both incremental scene graph construction and the acquisition of additional observations.

We formulate this problem as a **task-grounded adaptive information acquisition problem**, in which a robot determines whether its current scene graph contains sufficient information for the given task and acquires additional observations only when necessary. Based on this formulation, we propose **Task-Grounded Open-Vocabulary 3D Scene Graph Incremental Generation (TG-OSG)**, which incrementally refines an open-vocabulary 3D scene graph by incorporating task-relevant entities and relations from selectively acquired observations, and terminates when the graph supports an executable plan or provides sufficient evidence of task infeasibility. We evaluate TG-OSG on three public 3D scene graph datasets and a task-oriented simulation benchmark, assessing both scene graph quality and downstream task planning under progressively increasing observation budgets. The results demonstrate that TG-OSG can acquire planning-sufficient scene representations while avoiding unnecessary global reconstruction.

Our contributions are threefold: (1) We formulate **task-grounded open-vocabulary 3D scene graph generation** as an adaptive information acquisition problem for robotic task planning, explicitly distinguishing scene completeness from task-level decision sufficiency. (2) We develop **TG-OSG**, an incremental framework that selectively acquires observations and updates the scene graph according to task requirements. (3) We establish a task-oriented evaluation protocol across three

public 3D scene graph datasets and a simulation benchmark, demonstrating the effectiveness of TG-OSG in generating task-sufficient scene representations for downstream planning.

2 Related Work

2.1 3D Scene Graph Generation

3D scene graphs provide structured representations of environments by modeling objects and their spatial or semantic relations. 3DSSG established a learning-based formulation and benchmark for 3D scene graph generation, while SceneGraphFusion and MonoSSG extended graph construction to RGB-D and monocular sequences (Wald et al., 2020; Wu et al., 2021; 2023). VL-SAT further introduced vision-language supervision for relation prediction (Wang et al., 2023). These methods primarily aim to construct consistent and comprehensive scene representations from available observations. Although some update graphs incrementally, the observation process remains largely driven by the incoming sensor stream rather than by the requirements of a downstream task. TG-OSG instead treats observation acquisition as part of the scene graph generation process and makes it dependent on task-level sufficiency.

2.2 Open-Vocabulary 3D Scene Graphs

Open-vocabulary 3D understanding relaxes the fixed-category assumption by transferring vision-language knowledge to 3D representations. OpenScene, ConceptFusion, and OpenMask3D enable open-vocabulary 3D querying or instance segmentation, while OVSG, ConceptGraphs, and Open3DSG further incorporate open semantics into object-centric scene graphs (Peng et al., 2023; Jatavallabhula et al., 2023; Takmaz et al., 2023; Chang et al., 2023; Gu et al., 2024; Koch et al., 2024). Recent work has extended these representations toward functional semantics, dynamic open-vocabulary graphs, and open-world reasoning for robotic interaction and planning (Chang et al., 2023; Gu et al., 2024; Koch et al., 2024; Fei et al., 2026). These methods substantially broaden *what* can be represented and queried, but generally operate on observations that have already been acquired. TG-OSG addresses the complementary problem of deciding *what additional information should be acquired* when the current open-vocabulary graph is insufficient for a task.

2.3 Task-Oriented Scene Understanding

Scene graphs have increasingly been used as intermediate representations for robotic reasoning and planning. Taskography extracts task-relevant structures from large scene graphs, SARP reasons over partial scene representations for object search, and SayPlan retrieves relevant scene graph substructures for long-horizon planning (Agia et al., 2022; Amiri et al., 2022; Rana et al., 2023). Recent open-world 3D scene graph work further demonstrates the use of open-vocabulary graphs for grounding, retrieval, and task planning (Fei et al., 2026). These approaches establish the importance of task-relevant scene representations, but primarily use the scene graph as an input to subsequent reasoning or planning. In contrast, TG-OSG closes the loop between task planning and perception: the task determines whether the current graph is sufficient, triggers additional observations when necessary, and provides the criterion for terminating scene graph expansion. This distinction turns incremental generation from observation-driven graph updating into **task-grounded adaptive information acquisition**.

3 Method

We consider a robot that observes an indoor 3D environment through a sequence of RGB-D frames and is required to execute a task specified in natural language. A key challenge is that the observations available at an early stage may be insufficient to construct the task-relevant portion of the scene graph. Conversely, processing the entire observation sequence before task reasoning is unnecessary when the task can already be solved from partial scene evidence.

We therefore formulate 3D scene graph construction as a *task-grounded incremental generation* problem. Instead of reconstructing a complete scene graph before planning, our method progressively acquires observations, incrementally updates a persistent open-vocabulary 3D scene graph,

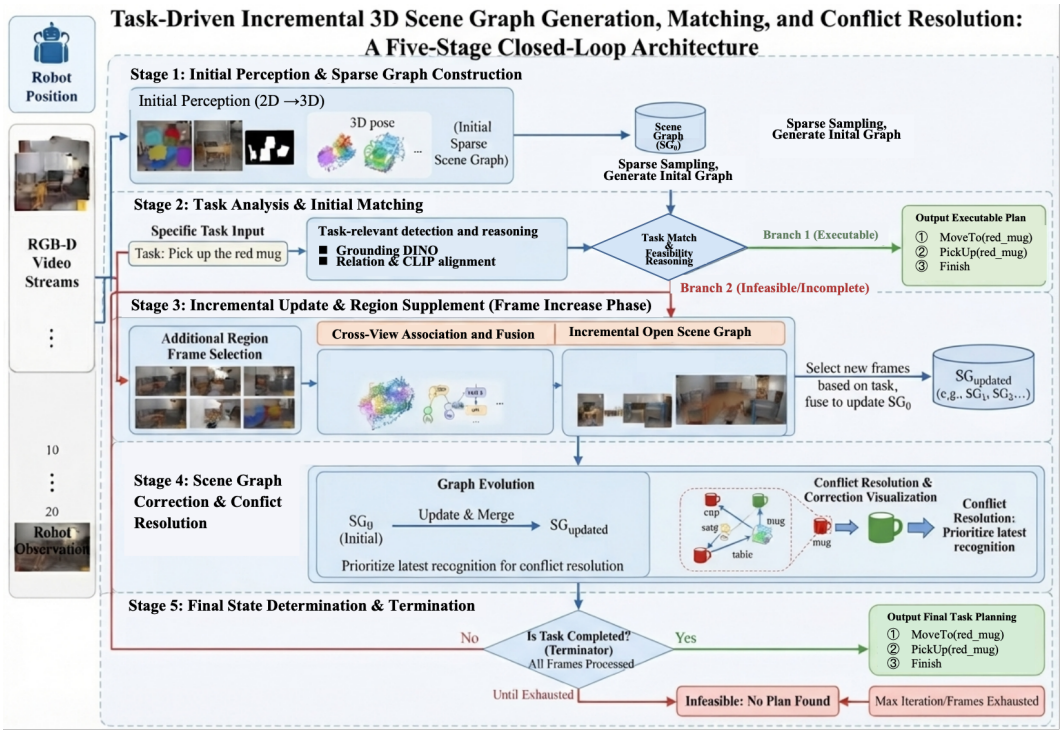

Figure 2: Overview of the framework, illustrating the pipeline from progressive observation and incremental open-vocabulary 3D scene graph generation, through persistent cross-stage graph updates, to task grounding and sufficient termination for task planning.

and invokes task planning to determine whether the current representation is sufficient for the task. The central principle is "***geometry determines what to observe, while the task determines when to stop.***" This formulation separates observation acquisition from task reasoning while coupling them through the sufficiency of the current scene graph. An overview of the proposed framework is illustrated in Figure 2.

3.1 Problem Formulation

Let $\mathcal{F} = \{F_i = (I_i, D_i, T_i)\}_{i=1}^{N}$ denote an RGB-D observation sequence with camera poses $T_i \in SE(3)$. Given a natural-language task τ, our goal is to construct an open-vocabulary 3D scene graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ that contains sufficient information to solve τ.

Each object node $v_i \in \mathcal{V}_t$ is represented as $v_i = (l_i, g_i, f_i, u_i)$, where l_i is an open-vocabulary semantic label, g_i is 3D geometry, f_i is a visual-semantic feature, and u_i is uncertainty. Edges $e_{ij} = (v_i, r_{ij}, v_j)$ capture spatial/semantic relations $r_{ij} \in \mathcal{R}_{\text{open}}$ (e.g., *on*, *inside*, *near*) without relying on a closed-set vocabulary.

We represent a task as $\tau = (\mathcal{C}_o, \mathcal{C}_r, \mathcal{C}_g)$ specifying object, relational, and goal constraints. A task planner produces a high-level action sequence $\mathcal{P}_t = \Psi(\tau, \mathcal{G}_t)$ (see Appendix C for examples).

3.2 Progressive Observation

To avoid redundant processing, observations are progressively introduced according to an ordering $\pi = (\pi_1, \ldots, \pi_N)$ derived from camera spatial distributions (see Appendix A.1 for the detailed distance-maximizing selection formulation).

At stage t, the available observations are $\mathcal{F}_t = \{F_{\pi(1)}, \ldots, F_{\pi(n_t)}\}$, with newly introduced frames denoted as $\Delta\mathcal{F}_t = \mathcal{F}_t \setminus \mathcal{F}_{t-1}$. This yields nested observation sets:

$$\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}_T. \tag{1}$$

3.3 Incremental Open-Vocabulary 3D Scene Graph Generation

At each stage, new frames $\Delta\mathcal{F}_t$ are converted into a stage-specific graph $\widehat{\mathcal{G}}_t$ via 3D object evidence extraction and geometry-grounded relation modeling (detailed pixel back-projection and association score formulations are deferred to Appendix A.2).

For an observation instance d and existing node v, association is determined by:

$$s_{\text{assoc}}(d, v) = (1 - \beta)s_{\text{geo}}(d, v) + \beta s_{\text{sem}}(d, v). \tag{2}$$

If $s_{\text{assoc}}(d, v) > \delta$, the observation is associated with the existing node; otherwise, a new node is initialized.

3.4 Persistent Cross-Stage Graph Update

Unlike independent stage-wise construction, all stages update a single persistent graph. Given new observations, object correspondence is established by minimizing matching costs over identity consistency measures (Appendix A.3). The persistent graph is updated via an integration operator Φ:

$$\mathcal{G}_t = \Phi(\mathcal{G}_{t-1}, \widehat{\mathcal{G}}_t). \tag{3}$$

Thus, $\widehat{\mathcal{G}}_t$ represents new stage evidence, whereas $\mathcal{G}_t$ represents the persistent accumulated graph. Subsequent stages refine rather than replace prior scene knowledge.

3.5 Task Grounding and Task-Sufficient Termination

Given the persistent graph $\mathcal{G}_t$, a task-relevant subgraph is extracted via $\mathcal{G}_t^\tau = \text{Ground}(\mathcal{G}_t, \tau)$. We define a task-sufficiency indicator $s_t \in \{0, 1\}$ based on whether a valid plan can be solved:

$$s_t = \mathbb{I}\left[\text{Solve}(\tau, \mathcal{G}_t) = 1\right]. \tag{4}$$

The observation process terminates at the earliest stage:

$$t^* = \min\left\{t : \text{Solve}(\tau, \mathcal{G}_t) = 1\right\}. \tag{5}$$

If $s_t = 1$, the loop terminates; otherwise, the next observation subset $\mathcal{F}_{t+1}$ is loaded. This separates observation acquisition from task planning through the sufficiency of the current graph.

4 Experiments

We evaluate TG-OSG on Replica (Straub et al., 2019), ScanNet (Dai et al., 2017), and 3RScan (Wald et al., 2019). Evaluation follows a dynamic closed loop after task arrival: the system first builds an initial scene graph from a small observation budget, attempts to solve the issued task on the current graph, and advances to the next observation stage only when entity, relation, or state evidence is insufficient. It then updates the persistent graph and attempts the task again. The primary evaluation asks whether this loop can ground tasks and produce reliable symbolic plans while terminating observation as early as possible. Scene-graph recall characterizes representation quality at each evidence stage and reveals how incremental graph updates support task solving.

4.1 Experimental Setup

Protocol and tasks. Each run begins by constructing the initial graph, after which the task is issued and the first solving attempt is made. If the current graph is insufficient, subsequent observation stages are activated in sequence, and execution stops immediately after the first success. Every stage uses a nested prefix of the same frozen Pose-FPS ordering; budgets and activation conditions are given in Appendix Table 2. Our cross-dataset task-planning benchmark spans multiple indoor scenes, five task types, and three relation-depth levels R1–R3. Its composition, action domain, and evaluation details are provided in Appendix C.

Baselines. Task experiments compare 3DSSG (Wald et al., 2020), MonoSSG (Wu et al., 2023), 3DSU (Fei et al., 2026), BBQ (Linok et al., 2025), VLSAT (Wang et al., 2023), and TG-OSG. Each predicted graph is connected to the same task interpreter and symbolic planner. Open-setting point-cloud generation is compared among 3DSU, BBQ, and TG-OSG. Closed-setting experiments directly provide point clouds under three point-cloud and node-prior conditions; their results are reported in Appendix F.

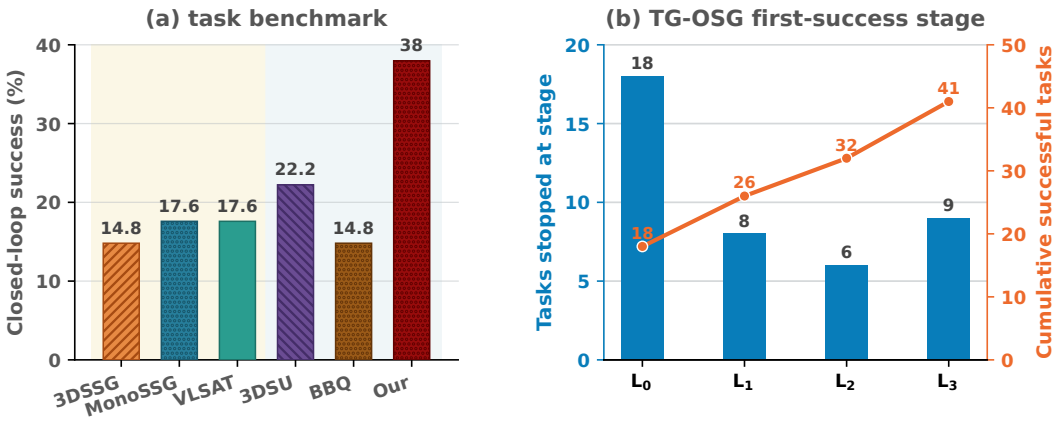

Figure 3: Main task-level results. (a) Closed-loop success rate across observation levels L_t on the cross-dataset task-planning benchmark. (b) Number of TG-OSG tasks that first succeed at each observation level L_t and the corresponding cumulative count.

Metrics. The primary task metric is closed-loop task success rate. Task success follows the solving criterion in Section 3.5. Let $s_i(t)$ indicate whether task i succeeds at L_t (where L_t denotes the cumulative observation level or stage length at step t). We define

$$\tau_i = \min\{t \mid s_i(t) = 1\}, \qquad \mathrm{SR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[\tau_i < \infty], \tag{6}$$

where τ_i represents the earliest task-sufficient stage for task i. Scene-graph quality is measured by Object R@1/5/10, Predicate R@3/5, and Relationship R@50; input conditions and scoring rules are specified in Appendix D. Let $\mathcal{Q}_x$ denote the corresponding ground-truth query set. The three recall families share the form

$$\mathrm{R}_x@K = \frac{1}{|\mathcal{Q}_x|}\sum_{q\in\mathcal{Q}_x} \mathbb{I}[r_x(q) \le K], \qquad x \in \{O, P, R\}, \tag{7}$$

where $r_x(q)$ represents the rank of the ground-truth query q in prediction results.

4.2 Task Success and Observation Efficiency

Figure 3 (a) shows that TG-OSG achieves a closed-loop success rate of 37.96% on our cross-dataset task-planning benchmark, compared with 22.22% for the strongest baseline, 3DSU. This corresponds to an absolute gain of 15.74 percentage points and a relative improvement of 70.83%. Because all methods use the same planning interface, the comparison measures the combined utility of object discovery, relational constraints, and symbolic planning.

Figure 3 (b) further shows that, among successful tasks, 43.90% are solved directly on the initial graph L_0, while 19.51% and 14.63% first succeed at L_1 and L_2, respectively. Only 21.95% require the full budget. Thus, 78.05% of successful tasks terminate before L_3, with a mean budget of 40.73% of the maximum. Even when unresolved tasks are conservatively charged the full budget, the mean over the entire benchmark is 77.50%. Moreover, static evaluation at fixed L_3 yields a 33.33% success rate, whereas the dynamic loop reaches 37.96%, a relative gain of 13.89%. Task-solvability-based termination therefore preserves valid solutions obtained at earlier stages and reduces exposure to accumulated back-projection, registration, and instance-association errors.

4.3 Open-Setting Scene Graph Comparison

Table 1 reports open-setting point-cloud generation results at the full L_3 budget. Every method generates its own point cloud and object graph from observations and is evaluated using the same visible ground truth and joint scoring protocol.

Table 1: Open-setting scene graph results at the full L_3 budget (%). Predicate and Relationship use joint scoring. The best result within each dataset is shown in bold.

Dataset	Method	Object R@1	Object R@5	Object R@10	Predicate R@3	Predicate R@5	Relationship R@50
Replica	3DSU	19.82	35.67	40.55	21.60	24.83	10.84
	BBQ	15.55	32.32	35.98	18.89	22.36	8.60
	TG-OSG	**36.28**	**57.93**	**60.67**	**52.55**	**61.26**	**34.37**
ScanNet	3DSU	19.57	39.67	45.38	40.32	44.67	16.62
	BBQ	15.76	39.67	48.37	45.97	53.34	15.04
	TG-OSG	**22.83**	**51.36**	**62.50**	**63.86**	**72.42**	**26.29**
3RScan	3DSU	13.45	32.16	37.72	40.25	46.15	9.86
	BBQ	13.74	29.82	34.21	**41.04**	**46.37**	9.07
	TG-OSG	**19.88**	**37.72**	**43.27**	36.51	40.59	**14.06**
Macro avg.	3DSU	17.61	35.83	41.22	34.06	38.55	12.44
	BBQ	15.02	33.94	39.52	35.30	40.69	10.90
	TG-OSG	**26.33**	**49.00**	**55.48**	**50.97**	**58.09**	**24.91**

TG-OSG ranks first on all six macro-average metrics. It reaches 26.33%, 49.00%, and 55.48% on Object R@1/5/10, improving over the strongest baseline by 49.5%, 36.8%, and 34.6% relatively. Predicate R@3/5 reach 50.97% and 58.09%, relative improvements of 44.4% and 42.8%. Relationship R@50 rises from 12.44% for the strongest baseline to 24.91%, an absolute gain of 12.47 points and a $2.00\times$ improvement. Because this metric requires the relation endpoints and predicate to be jointly correct, the gain supports our central claim: progressive object discovery and cross-stage identity persistence improve not only node recall but also the completeness of relational structures used for planning. Overall, TG-OSG ranks first in 16 of the 18 dataset–metric combinations and on all six metrics after macro-averaging across datasets.

Stage-wise curves, low-budget comparisons, and closed-setting point-cloud results are provided in Appendices E–F.

4.4 Task-Graph Evolution and Evaluation Scope

Figure 4 illustrates how the persistent graph is progressively completed for representative R1–R3 tasks. Gray nodes denote environmental context, blue and orange nodes with thick edges form the task subgraph, and green outlines identify nodes newly introduced at the current stage. The complete constraint chain first emerges at a different stage for each task. This behavior shows that TG-OSG maintains a reusable open graph and performs task grounding and planning over that graph, rather than constructing a task-specific fragment alone.

Functional-task success means that the symbolic action sequence passes state-transition and goal checks; physical navigation and manipulation execution are outside the scope of this metric. Appendix G provides simulations for all three task levels, qualitatively illustrating progressive observation, instance grounding, and task-planning paths. “‘latex

5 Conclusion

We study incremental 3D scene graph generation for robot task planning, where exhaustive scene reconstruction can introduce unnecessary observations. We show that task solvability can serve as a stopping criterion for scene graph construction, and introduce TG-OSG to progressively acquire observations and update a persistent open-vocabulary scene graph until the current representation supports the task. Experiments on Replica, ScanNet, and 3RScan show that TG-OSG achieves a 37.96% closed-loop task success rate, improving over the strongest baseline by 15.74 percentage points, while 78.05% of successful tasks terminate before the full observation budget. TG-OSG also achieves 24.91% Relationship R@50 in open-vocabulary scene graph evaluation.

While these results demonstrate the effectiveness of task-grounded incremental construction, the current observation schedule does not explicitly model task-conditioned information gain. Future work will explore task-aware observation policies and validate the framework in closed-loop navigation and manipulation on physical robots.

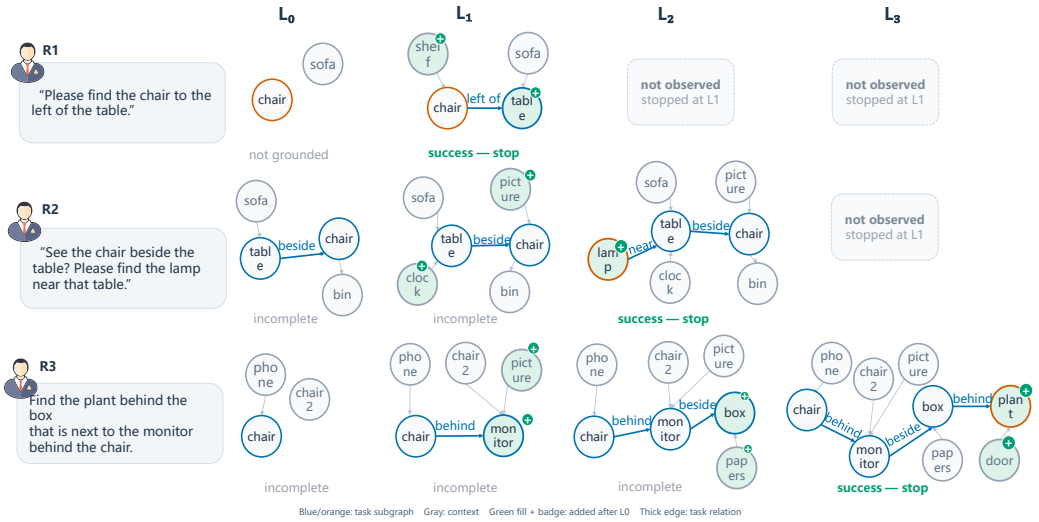

Figure 4: Persistent graphs and task-planning paths for representative R1–R3 tasks. New observations introduce missing entities and relations; the system stops once the constraint chain is complete and instance bindings are unique.

6 AI USE STATEMENT

In this work, we used generative AI tools to assist with editing the manuscript for clarity and readability and with software development, including code completion, debugging, and refactoring. The research questions, core conceptual contributions, methodology, experimental design, and scientific conclusions were developed by the authors. Generative AI tools were not used to generate synthetic datasets, formulate mathematical claims or proofs, propose or refine research hypotheses, design the research methodology or experiments, perform qualitative data analysis, or interpret experimental results. All AI-assisted text and code were reviewed by the authors, and AI-assisted code was manually inspected and tested for correctness. We take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

REFERENCES

Christopher Agia, Krishna Murthy Jatavallabhula, Mohamed Khodeir, Ondrej Miksik, Vibhav Vineet, Mustafa Mukadam, Liam Paull, and Florian Shkurti. Taskography: Evaluating robot task planning over large 3D scene graphs. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 46–58. PMLR, 2022.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Cristian Cortes, Byron David, Chelsea Finn, Keerthana Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Ruano, Daniel Xu, Andy Zeng, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the Conference on Robot Learning*, pp. 878–897, 2022.

Saeid Amiri, Kishan Chandan, and Shiqi Zhang. Reasoning with scene graphs for robot planning under partial observability. *IEEE Robotics and Automation Letters*, 7(2):5560–5567, 2022. doi: 10.1109/LRA.2022.3157567.

Haonan Chang, Kowndinya Boyalakuntla, Shiyang Lu, Siwei Cai, Eric Pu Jing, Shreesh Keskar, Shijie Geng, Adeeb Abbas, Lifeng Zhou, Kostas Bekris, and Abdeslam Boularias. Context-aware entity grounding with open-vocabulary 3D scene graphs. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1950–1974. PMLR, 2023.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017. doi: 10.1109/CVPR.2017.261.

Yu Fei, Quan Deng, Shengeng Tang, Yuehua Li, and Lechao Cheng. Open-world 3D scene graph generation for retrieval-augmented reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3894–3902, 2026. doi: 10.1609/aaai.v40i5.37391.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation*, pp. 5021–5028, 2024. doi: 10.1109/ICRA57147.2024.10610243.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Proceedings of the 6th Conference on Robot Learning*, volume 205, pp. 1769–1782, 2023.

Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. ConceptFusion: Open-set multimodal 3D mapping. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.066.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14183–14193, 2024. doi: 10.1109/CVPR52733.2024.01345.

Sergey Linok, Tatiana Zemskova, Svetlana Ladanova, Roman Titkov, Dmitry Yudin, Maxim Monastyrny, and Aleksei Valenkov. Beyond bare queries: Open-vocabulary object grounding with 3D scene graph. In *2025 IEEE International Conference on Robotics and Automation*, pp. 13582–13589, 2025. doi: 10.1109/ICRA55743.2025.11128059.

Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 815–824, 2023. doi: 10.1109/CVPR52729.2023.00085.

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. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 23–72. PMLR, 2023.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10740–10749, 2020.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530, 2023.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaoqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales,

Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Ayca Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-vocabulary 3D instance segmentation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68367–68390, 2023. doi: 10.52202/075280-2989.

Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Niessner. RIO: 3D object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7658–7667, 2019. doi: 10.1109/ICCV.2019.00775.

Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D semantic scene graphs from 3D indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3961–3970, 2020. doi: 10.1109/CVPR42600.2020.00402.

Ziqin Wang, Bowen Cheng, Lichen Zhao, Dong Xu, Yang Tang, and Lu Sheng. VL-SAT: Visual-linguistic semantics assisted training for 3D semantic scene graph prediction in point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21560–21569, 2023. doi: 10.1109/CVPR52729.2023.02065.

Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. SceneGraphFusion: Incremental 3D scene graph prediction from RGB-D sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7515–7525, 2021. doi: 10.1109/CVPR46437.2021.00743.

Shun-Cheng Wu, Keisuke Tateno, Nassir Navab, and Federico Tombari. Incremental 3D semantic scene graph prediction from RGB sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5064–5074, 2023. doi: 10.1109/CVPR52729.2023.00490.

APPENDIX CONTENTS

1. Appendix A: Supplementary Details for Method ..... p. 11;
2. Appendix B: Dynamic Observation and Persistent Graph Updates ..... p. 12;
3. Appendix C: Task-Planning Benchmark and Task Hierarchy ..... p. 13;
4. Appendix D: Scene Graph Generation Evaluation Protocol ..... p. 14;
5. Appendix E: Open-Vocabulary Scene Graph Generation Results ..... p. 16;
6. Appendix F: Point-Cloud Input Diagnostics for Closed-Vocabulary Models ..... p. 17;
7. Appendix G: Simulation Platform and R1–R3 Case Studies ..... p. 19.

This appendix supplements the dynamic observation mechanism of TG-OSG, the task and evaluation protocols, the open-vocabulary comparison, the closed-vocabulary diagnostics, and the simulation results. The system starts from a coarse open scene graph, expands its observations only when the current evidence is insufficient for task grounding and planning, and terminates upon obtaining the first valid solution. Thus, L_0–L_3 denote progressively richer evidence states visited on demand by the closed loop, rather than a fixed pipeline that every task must exhaust.

A SUPPLEMENTARY DETAILS FOR METHOD

A.1 PROGRESSIVE OBSERVATION DETAILS

To obtain a spatially distributed observation sequence, each camera pose $T_i = \begin{bmatrix} R_i & t_i \\ \mathbf{0}^\top & 1 \end{bmatrix} \in SE(3)$ is represented by:

$$x_i = \left[\widetilde{t}_i; \lambda \widehat{d}_i\right], \tag{8}$$

where $\widetilde{t}_i$ is the normalized camera position, $\widehat{d}_i$ is the normalized viewing direction from orientation R_i, and λ controls the trade-off. We iteratively select frames maximizing the minimum distance:

$$i^* = \arg\max_{i \notin \mathcal{S}} \min_{j \in \mathcal{S}} \|x_i - x_j\|_2, \tag{9}$$

where $\mathcal{S}$ is the selected frame set.

A.2 3D OBJECT EVIDENCE AND ASSOCIATION DETAILS

Depth pixels $u = [u_x, u_y, 1]^\top$ are back-projected into the global coordinate system via camera intrinsics K and pose T_i:

$$p_i(u) = T_i \begin{bmatrix} D_i(u) K^{-1} u \\ 1 \end{bmatrix}. \tag{10}$$

The resulting object observation is $o = (g, f, p)$, where g is geometry, f is visual-semantic feature, and p is semantic prediction.

A.3 PERSISTENT CROSS-STAGE GRAPH UPDATE DETAILS

Identity consistency between a newly detected object v_j^t and an existing node v_i^{t-1} is evaluated by:

$$s_{\mathrm{id}}(i,j) = w_1 s_{\mathrm{hist}} + w_2 s_{\mathrm{geo}} + w_3 s_{\mathrm{sem}} + w_4 s_{\mathrm{dist}}, \tag{11}$$

where s_{hist}, s_{geo}, s_{sem}, and s_{dist} capture historical observation consistency, geometric overlap, semantic similarity, and spatial proximity, respectively, weighted by $w_1, \ldots, w_4$.

Table 2: Task-triggered progressive observation protocol. The cumulative budget is measured relative to the maximum observation budget of a scene.

Stage	Cumulative budget	Activation condition
L_0	10%	Construct the initial coarse scene graph
L_1	32.5%	The L_0 graph does not satisfy task success
L_2	55%	The L_1 graph does not satisfy task success
L_3	100%	The L_2 graph does not satisfy task success

B Dynamic Observation and Persistent Graph Updates

B.1 Open-Vocabulary Perception and Incremental Object Fusion

New RGB-D observations are converted into object-level 2D evidence by open-vocabulary detection and instance segmentation, and then back-projected into 3D instances. The system associates new instances with the existing object map using geometric and visual evidence: consistent observations are fused into existing nodes, whereas unmatched observations create new nodes. Free-form object semantics and multi-view visual features are accumulated over time. Canonical dataset categories are used only by the unified task interface and quantitative evaluation; they neither constrain object proposals nor serve as hard conditions for instance association. Additional observations therefore extend a single object map instead of constructing independent graphs at different evidence states. All perception backbones remain frozen throughout the experiments.

B.2 Deterministic Pose-FPS Ordering

For camera pose $T_i = [R_i, t_i] \in \mathrm{SE}(3)$, we form the pose descriptor $x_i = [\widetilde{t}_i; \lambda \widehat{d}_i]$ from the normalized camera position $\widetilde{t}_i$ and viewing direction $\widehat{d}_i$. Given the set $\mathcal{S}_k$ of viewpoints selected after k iterations, the next viewpoint is chosen by

$$\pi_{k+1} = \arg\max_{i \notin \mathcal{S}_k} \min_{j \in \mathcal{S}_k} \|x_i - x_j\|_2. \tag{12}$$

Pose-FPS measures viewpoint complementarity from camera position and viewing direction and uses Eq. equation 12 to produce a deterministic, coverage-prioritized ordering. The first viewpoint is the candidate farthest from the mean pose; each subsequent viewpoint maximizes its minimum distance to the selected set, with frame order used to break distance ties. The ordering depends only on camera poses and is independent of task text, graph predictions, and ground-truth locations. It is fixed across all methods and tasks within the same scene. Its nested prefixes therefore provide comparable, progressively increasing evidence budgets while avoiding the local concentration induced by chronological video prefixes.

B.3 Observation Budgets and Online Termination

Let $n_0 = \lceil 0.1N \rceil$ and $N_H = N - n_0$. The remaining cumulative observation counts are defined as

$$n(\rho) = n_0 + \lceil \rho N_H \rceil, \qquad \rho \in \{0.25, 0.50, 1.00\}. \tag{13}$$

This rule produces four strictly nested Pose-FPS prefixes, L_0–L_3. Each level corresponds to an increasing evidence budget and a potential task-checking point. A task is first solved on the L_0 graph. The controller enters the next evidence state only when the current graph cannot yield a unique grounding or valid plan, and terminates at the first state satisfying the success criterion. These prefixes thus support evidence acquisition on demand rather than defining a fixed four-stage procedure that every task must traverse.

B.4 Persistent Identity and Relation Updates

The stage graph is associated one-to-one with the historical graph according to the consistency of observation provenance and 3D geometry. Successfully associated objects inherit persistent identifiers and accumulate new geometric and semantic evidence. Unmatched objects are inserted as new nodes, while historical nodes not observed again at the current stage are retained. This mechanism

Table 3: Coverage of the cross-dataset task-planning benchmark.

Dimension	Coverage
Datasets	Replica, ScanNet, and 3RScan
Scenes	Multiple indoor scenes from the three datasets
Task types	locate, open, close, place-in, and place-on
Relation depth	R1, R2, and R3

allows missing entities and relations to enter the same graph progressively while avoiding the direct use of category predictions as identity-merging criteria.

After object fusion, directed edges are recomputed from the updated bounding boxes, surface distances, directions, and support or containment configurations. The relation update covers the spatial predicates required for task planning, including proximity, relative direction, support, and containment. Re-estimating relations instead of simply accumulating historical edges enables new observations both to add missing relations and to correct relation estimates induced by incomplete early geometry. Persistent nodes and dynamically updated relations jointly define the scene state used by every subsequent task-solving attempt.

C Task-Planning Benchmark and Task Hierarchy

C.1 Task Composition and Success Criteria

The cross-dataset task-planning benchmark is constructed from multiple indoor scenes in Replica, ScanNet, and 3RScan. It contains five task types: *locate*, *open*, *close*, *place-in*, and *place-on*. A natural-language instruction is issued and parsed once at L_0; the resulting roles, semantic constraints, relational constraints, and goal state remain fixed in subsequent evidence states. Every method is connected to the same task interpreter and symbolic planner, thereby isolating the effect of the scene-graph representation on downstream task solving.

Localization and functional tasks use the success criteria defined in Section 3.5. A task terminates immediately after succeeding at the current evidence state; a task that remains unsuccessful after L_3 is marked unresolved.

C.2 Task Representation and Dynamic Solving

Each natural-language task is converted into a symbolic representation consisting of object roles, directed relational constraints, initial facts, and goal facts. This representation is fixed when the task is issued and remains unchanged as the graph is updated. Re-solving therefore reflects newly acquired scene evidence rather than a reinterpretation of task semantics. Parsing uses only the task text and the unified object and relation vocabularies, without access to scene ground truth or target instance identifiers. The relational constraints belong to {*near*, *left-of*, *right-of*, *in-front-of*, *behind*, *above*, *under*, *on*, *supports*, *inside*, *contains*}. Instructions that cannot be represented completely by this task domain are excluded from the planning evaluation.

At every activated evidence state, the system retrieves candidate nodes satisfying the semantic constraints from the persistent graph and jointly disambiguates them using role distinctness and directed relation chains. A localization task succeeds only when every role admits a unique and mutually consistent instance binding; a functional task enters symbolic planning only after this condition is met. If the current graph lacks sufficient object or relation evidence, the failed solve triggers another graph expansion. Tasks for which a valid solution has already been found consume no subsequent observations.

C.3 Symbolic Planning and Success Evaluation

The planning state comprises the robot's abstract location, the currently held object, a set of symbolic facts, and a termination flag. The planner searches for a valid plan satisfying the goal facts under the action preconditions and state effects. Table 4 summarizes the semantics of the action domain.

Table 4: Symbolic action domain used in the task-planning evaluation.

Action	Main preconditions	State effects
`MoveTo(x)`	x is grounded	Set abstract location to x
`PickUp(x)`	At x; x is movable; hand is empty	Hold x; remove its prior placement fact
`PlaceOn(x,y)`	Hold x; at support surface y	Add $on(x, y)$
`PlaceIn(x,y)`	Hold x; at container y; open if openable	Add $inside(x, y)$
`Open/Close(x)`	At openable x; opposite state holds	Toggle $open/closed(x)$
`Finish`	All goal facts hold	Terminate the plan

Table 5: Task relation depth and representative examples.

Level	Relational constraints	Primary capability	Representative task
R1	1	Single-hop localization	Locate the chair to the left of the table
R2	2	Relation-chain and functional planning	Locate and open the specified door
R3	3	Multi-hop joint disambiguation	Locate the plant behind the box

In addition to unique grounding, a functional task requires all action preconditions, state transitions, and goal facts to be valid. When the current evidence is insufficient to verify these conditions, the system continues expanding the scene graph. A task that still lacks a unique binding or valid plan at the maximum observation budget is marked unresolved. This criterion separates insufficient evidence from physical infeasibility and avoids interpreting epistemic uncertainty as evidence that the task itself is infeasible.

C.4 R1–R3 Relation Depth

Task levels are determined by the number of directed spatial constraints required for grounding. R1 evaluates target grounding under a single-hop relation. R2 introduces a two-hop relation chain or a functional goal conditioned on such a chain. R3 requires joint disambiguation under three relational constraints. The hierarchy changes the evidence structure required by a task without changing the object vocabulary, relation set, or symbolic action domain.

D Scene Graph Generation Evaluation Protocol

D.1 Open-Vocabulary and Closed-Vocabulary Methods

We distinguish open-vocabulary and closed-vocabulary methods by their semantic prediction spaces, not by whether point clouds are generated by the method or supplied by the evaluation protocol. TG-OSG, 3DSU, and BBQ are open-vocabulary methods: they construct object representations from RGB-D observations, allow node semantics to be expressed as open-ended text, and project these semantics into a unified category vocabulary only for quantitative evaluation. 3DSSG, MonoSSG, and VLSAT are closed-vocabulary methods: their object and predicate output spaces are determined by fixed training categories, and they predict fixed categories and relations from object-level 3D inputs. Mapping all outputs to the unified evaluation vocabulary enables common scoring but does not alter the original vocabulary regime of a method.

D.2 Three Point-Cloud Conditions for Closed-Vocabulary Models

To analyze the dependence of closed-vocabulary back ends on node coverage and 3D geometry, we define three controlled input conditions at the full L_3 budget. The first condition selects predicted nodes from the persistent TG-OSG graph that admit frozen geometric correspondences to visible ground-truth instances and retains their reconstructed point clouds. The second condition keeps exactly the same node set, ordering, and instance correspondences, while replacing each reconstructed point cloud with the corresponding ground-truth instance point cloud supplied by the dataset. The difference between the first two conditions therefore reflects only input geometry quality, without changing node cardinality or instance correspondences. The third condition further expands the input to all ground-truth nodes within the evaluation scope and characterizes closed-vocabulary back-end performance under complete node coverage. Frozen correspondences are used only for input

Table 6: Vocabulary regimes and input forms of the scene graph generation methods.

Vocabulary regime	Methods	Semantic space	Experimental input
Open vocabulary	TG-OSG, 3DSU, BBQ	Open-ended text semantics	RGB-D and poses
Closed vocabulary	3DSSG, MonoSSG, VLSAT	Fixed object/predicate classes	Object-level point clouds

Table 7: Three controlled node point-cloud conditions for closed-vocabulary models.

Condition	Node set	Node geometry	Diagnostic factor
TG-OSG reconstruction	Frozen matched predictions	TG-OSG reconstruction	Back-end performance on system geometry
Matched-node GT geometry	Identical to preceding condition	GT instance point clouds	Reconstruction error
Complete GT nodes	All visible GT nodes	GT instance point clouds	Node coverage and back-end upper bound

alignment and scoring, not as network features; none of the three conditions provides ground-truth object categories or relations to the model.

D.3 Unified Metrics

All SGG experiments report Object R@1/5/10, Predicate R@3/5, and Relationship R@50. Predicate and Relationship use joint end-to-end scoring: if an object or relation endpoint is unmatched, the rank of the corresponding query is set to infinity. Object evaluates object discovery and semantic ranking, Predicate evaluates predicate ranking, and Relationship requires a joint match of subject, predicate, and object. These definitions instantiate Eq. equation 7.

Each dataset is scored against its visible ground-truth objects and relations. Hits and denominators are first accumulated across scenes within each dataset to compute dataset-level recall. The reported cross-dataset aggregate is then the equally weighted macro average of the three dataset-level recalls. This order prevents differences in scene or dataset size from dominating the aggregate result.

D.4 Visible Ground Truth, Node Matching, and Ranking Spaces

At each evidence state, metric denominators include only ground-truth instances supported by the cumulative source frames available at that state, and visibility filtering is independent of the predicted graph. This design aligns graph quality with the evidence actually available to the closed loop: objects outside the current observation budget are not counted as missed, whereas objects that are sufficiently observed but absent from the graph remain errors. Stage-wise recall therefore measures representation quality under the current evidence rather than conflating it with differences in observation coverage. For Replica, the evaluation reads native per-frame instance boxes and camera poses. An instance box must occupy at least 0.5% of the image, have a short side of at least 5 pixels, and be visible in at least three frames from at least two camera view groups. View groups are constructed in frame order; a new group begins when the camera moves more than 0.20 m from the current group anchor or its forward direction changes by more than 12°. A relation fact enters the denominator only when both endpoints pass the visibility filter at the current evidence state. The entire filtering procedure is independent of the predicted graph.

Predicted and ground-truth nodes are matched through a greedy one-to-one geometric procedure that does not use semantic labels. A candidate pair is retained if its 3D bounding-box IoU is at least 0.10, or if its center distance is no greater than $\mathrm{clip}(0.5l_{\mathrm{GT}}, 0.25, 0.75)$ m, where l_{GT} is the diagonal length of the ground-truth bounding box. Candidate pairs are sorted in descending order by $\max(\mathrm{IoU}_{3D}, 0.5s_{\mathrm{ctr}})$ and greedily assigned to unmatched predicted and ground-truth nodes, where s_{ctr} is the linear center-proximity score within the adaptive threshold. Predicted categories are not used during matching, preventing semantic correctness from influencing its own evaluation.

Object ranks the ground-truth class within a fixed dataset evaluation vocabulary. Open-ended text labels are projected into this vocabulary in a frozen OpenCLIP text space, without manual task-specific synonym correction. Predicate ranks 11 directed predicate classes on geometrically matched ordered node pairs. Relationship ranks triplets in the joint $|\mathcal{C}| \times 11 \times |\mathcal{C}|$ subject–predicate–object space. If a ground-truth object is not matched to a predicted node, or if either endpoint of a relation is absent, the corresponding query rank is set to infinity. Predicate and Relationship therefore include errors from node discovery, instance matching, and relation prediction rather than measuring predicate classification only under known ground-truth endpoints.

Table 8: Open-vocabulary TG-OSG results under progressively richer evidence states (%). Predicate and Relationship use joint end-to-end scoring.

Dataset	Stage	Object R@1	Object R@5	Object R@10	Predicate R@3	Predicate R@5	Relationship R@50
Replica	L_0	33.54	49.22	53.29	38.15	44.34	28.08
	L_1	32.62	53.05	57.01	44.31	51.95	30.75
	L_2	34.76	55.18	59.15	48.81	57.09	32.05
	L_3	36.28	57.93	60.67	52.55	61.26	34.37
ScanNet	L_0	22.87	42.15	50.96	32.76	37.64	17.34
	L_1	23.91	51.63	59.51	53.19	60.78	24.76
	L_2	23.91	53.26	61.68	57.15	65.22	26.44
	L_3	22.83	51.36	62.50	63.86	72.42	26.29
3RScan	L_0	5.28	8.68	8.68	1.23	1.54	0.00
	L_1	12.95	23.80	25.30	18.63	20.52	6.72
	L_2	17.06	31.18	33.82	26.31	29.16	8.43
	L_3	19.88	37.72	43.27	36.51	40.59	14.06

D.5 Unified Comparison and Diagnostic Boundaries

The main open-vocabulary comparison requires every method to construct its object graph independently from the same type of RGB-D and pose input, and therefore jointly evaluates object discovery, 3D representation, semantic prediction, and relation generation. Closed-vocabulary experiments instead use the three object-level point-cloud conditions in Table 7. The TG-OSG reconstruction condition evaluates closed-vocabulary back ends on system-reconstructed nodes. The matched-node GT condition changes only node geometry, so its difference from the first condition measures the effect of reconstruction quality. The complete-GT-node condition additionally removes incomplete node coverage and exposes the remaining semantic and relation-prediction errors. The latter two are diagnostic conditions rather than same-input leaderboards against the end-to-end TG-OSG system. Each baseline retains its original network and output space, with only the input adaptation and category mapping required for unified scoring. In downstream task comparisons, all scene graphs are connected to the same task representation, grounding rules, symbolic action domain, and termination criteria.

E Open-Vocabulary Scene Graph Generation Results

Table 1 reports full-budget L_3 results for the open-vocabulary methods. This section further analyzes the evidence states that may be visited by the closed loop. The L_0–L_3 results characterize how the representation changes when an unresolved task acquires additional observations. Online execution visits only the prefix up to the first successful state and does not continue graph construction for an already solvable task.

E.1 Graph Quality under Progressively Richer Evidence

As a diagnostic of closed-loop graph updates, Table 8 compares progressively richer evidence states along the same nested observation sequence. Complementary observations generally improve object and relation recall. Across the three datasets, the macro averages of Object R@5, Predicate R@3, and Relationship R@50 increase from 33.35%, 24.05%, and 15.14% at L_0 to 49.00%, 50.97%, and 24.91% at L_3, corresponding to absolute gains of 15.65, 26.92, and 9.77 percentage points. Several ScanNet object metrics fluctuate slightly at later evidence states, indicating that additional observations do not guarantee monotonic gains. By retaining the current solution at the first solvable state, the closed loop avoids exposing a representation that already satisfies the task constraints to unnecessary subsequent fusion errors.

E.2 Low-Budget Quality Comparison

As shown in Figure 6, at L_1 TG-OSG uses 32.5% of the observation budget and reaches 42.83% Object R@5, 38.71% Predicate R@3, and 20.74% Relationship R@50. Each value exceeds the corresponding full-budget L_3 result of both 3DSU and BBQ. Relative to the stronger of the two baselines for each metric, the absolute gains are 7.00, 3.41, and 8.30 percentage points. Together

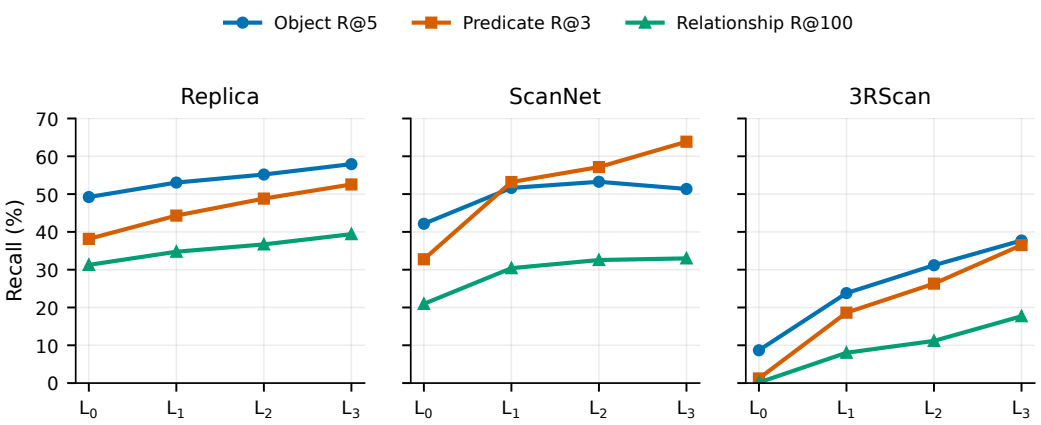

Figure 5: Scene-graph quality of open-vocabulary TG-OSG as evidence expands. The curves report Object R@5, Predicate R@3, and Relationship R@50. Online execution terminates at the first evidence state in which the task succeeds.

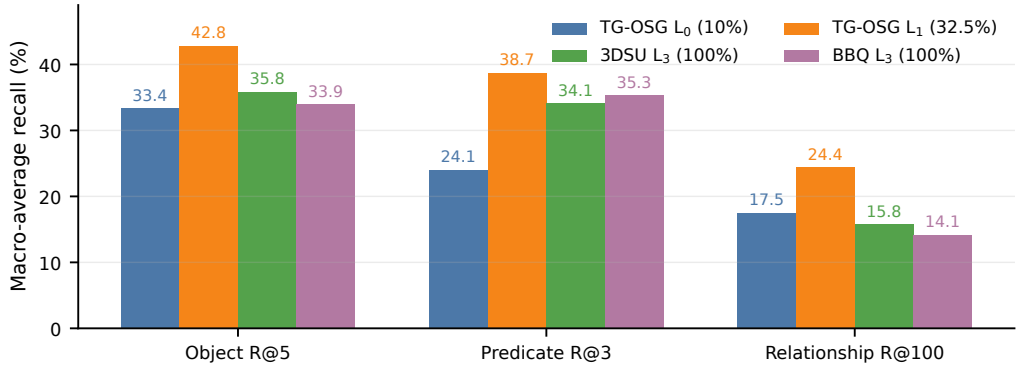

Figure 6: Macro-average recall of partial-budget TG-OSG and full-budget open-vocabulary baselines. All values are equally weighted averages over Replica, ScanNet, and 3RScan.

with the early-termination distribution in the task evaluation, these results show that coverage-complementary early observations can already form an effective scene representation for task solving.

F Point-Cloud Input Diagnostics for Closed-Vocabulary Models

This section evaluates closed-vocabulary models under three controlled node point-cloud conditions to separate the effects of reconstruction geometry, node coverage, and back-end prediction. Every table uses the six metrics defined in Section D.

F.1 TG-OSG Reconstructed Node Point Clouds

Under TG-OSG reconstructed node geometry, the closed-vocabulary models predict objects and relations for the frozen matched predicted nodes; the corresponding TG-OSG system result is also reported as a reference. TG-OSG ranks first in 15 of the 18 dataset–metric combinations. It leads all six metrics on Replica and ScanNet. On 3RScan, MonoSSG is stronger on the three Object metrics, whereas TG-OSG remains best on Predicate R@3/5 and Relationship R@50. The advantage is particularly consistent for relational structure: TG-OSG ranks first in all nine Predicate/Relationship comparisons across the three datasets. Relative to the strongest conventional method in each dataset, Relationship R@50 increases from 21.46% to 34.37% on Replica, from 3.63% to 37.10% on ScanNet, and from 9.52% to 14.06% on 3RScan. These results indicate that the point clouds and persis-

Table 9: Results of closed-vocabulary models using TG-OSG reconstructed node point clouds (%). The TG-OSG row provides a system-level reference under the same metrics. Bold denotes the best result within each dataset; only reported method–dataset combinations are shown.

Dataset	Method	Object R@1	Object R@5	Object R@10	Predicate R@3	Predicate R@5	Relationship R@50
Replica	3DSSG	17.38	29.88	40.85	27.77	44.15	8.38
	MonoSSG	32.01	48.48	53.35	27.36	43.12	21.46
	VLSAT	14.94	30.79	39.02	25.10	37.66	4.54
	TG-OSG	**36.28**	**57.93**	**60.67**	**52.55**	**61.26**	**34.37**
ScanNet	3DSSG	11.41	21.20	28.26	32.89	51.98	3.63
	MonoSSG	11.96	22.28	30.16	32.81	50.34	3.41
	VLSAT	3.53	9.24	17.39	24.53	39.94	0.38
	TG-OSG	**32.88**	**64.13**	**70.38**	**63.86**	**72.42**	**37.10**
3RScan	3DSSG	5.85	17.54	26.61	26.87	34.92	1.81
	MonoSSG	**20.47**	**39.77**	**45.91**	23.58	31.86	9.52
	TG-OSG	19.88	37.72	43.27	**36.51**	**40.59**	**14.06**

Table 10: Results of closed-vocabulary models using ground-truth point clouds for matched nodes (%). The node set, ordering, and instance correspondences are identical to Table 9; only node geometry is replaced.

Dataset	Method	Object R@1	Object R@5	Object R@10	Predicate R@3	Predicate R@5	Relationship R@50
Replica	3DSSG	30.79	42.38	48.17	28.34	43.82	18.18
	MonoSSG	**39.94**	**51.83**	**55.18**	27.66	43.43	**28.59**
	VLSAT	23.48	39.94	47.87	24.78	36.31	10.69
ScanNet	3DSSG	**24.73**	**40.49**	**49.18**	**34.48**	**53.16**	**13.05**
	MonoSSG	21.47	34.51	42.66	33.90	50.62	11.28
	VLSAT	9.78	25.27	35.05	26.55	39.37	2.98
3RScan	3DSSG	16.96	40.06	50.58	48.07	57.03	17.57
	MonoSSG	**29.24**	**52.92**	**59.65**	41.95	52.61	**28.46**
	VLSAT	11.40	33.63	43.57	**55.33**	**58.62**	11.90

tent object identities produced by the complete TG-OSG pipeline retain the information needed to recover relation endpoints and predicates jointly, rather than improving object classification alone. Because the closed-vocabulary models receive only the frozen matched node subset, this table is also affected by node coverage; the next two conditions further separate geometry quality from node coverage.

F.2 Ground-Truth Point Clouds for Matched Nodes

This condition neither adds nodes nor changes the frozen instance correspondences. Instead, it replaces the TG-OSG reconstruction of each node with the ground-truth instance surface for the same node set. Its difference from Table 9 therefore measures the sensitivity of closed-vocabulary back ends to point-cloud completeness and geometric noise; it cannot be interpreted as a change in object discovery or instance-correspondence ability. MonoSSG achieves the highest Relationship R@50 on Replica and 3RScan, whereas 3DSSG performs best on all six ScanNet metrics. The changes vary across models and datasets, indicating that more complete instance geometry can alleviate some relation-prediction errors but does not remove the limitations of fixed-vocabulary semantic and relation classification.

F.3 Complete Ground-Truth Node Input

The complete-GT-node condition expands the input from the frozen matched subset to all visible ground-truth instances and uses their ground-truth instance point clouds, while object categories and relations remain predictions of the closed-vocabulary models. This condition is an idealized diagnostic under complete node coverage rather than an input available to the end-to-end TG-OSG system. For example, when the input changes from TG-OSG reconstructed point clouds over the frozen matched nodes to complete ground-truth nodes, MonoSSG's Relationship R@50 rises from 21.46% to 41.50% on Replica, from 3.41% to 33.87% on ScanNet, and from 9.52% to 43.20% on 3RScan. Tables 9–11 therefore constitute a controlled error decomposition rather than three directly comparable leaderboards. Their central implication is that a substantial fraction of full-system errors

Table 11: Results of closed-vocabulary models using complete ground-truth nodes and their instance point clouds (%). Ground-truth object categories and relations are not provided as model inputs.

Dataset	Method	Object R@1	Object R@5	Object R@10	Predicate R@3	Predicate R@5	Relationship R@50
Replica	3DSSG	35.06	52.74	61.59	40.46	64.69	21.02
	MonoSSG	**50.30**	**69.21**	**75.30**	**40.54**	64.59	**41.50**
	VLSAT	28.66	48.78	59.15	36.78	55.26	13.15
ScanNet	3DSSG	27.45	43.75	53.26	**43.24**	**67.32**	15.40
	MonoSSG	**44.57**	**63.86**	**71.47**	42.53	64.79	**33.87**
	VLSAT	16.30	36.41	44.84	38.03	55.86	6.32
3RScan	3DSSG	22.81	52.05	68.71	74.26	92.86	24.04
	MonoSSG	**37.72**	**69.01**	**78.95**	63.04	83.33	**43.20**
	VLSAT	14.62	48.83	60.82	**89.80**	**97.62**	17.46

occurs before relation classification, at object discovery and cross-view instance association—the two components targeted by progressive observation and persistent graph updates in TG-OSG.

G Simulation Platform and R1–R3 Case Studies

G.1 Platform and Visualization Protocol

The simulation platform uses Habitat-Sim with Replica indoor scenes and the Bullet physics back end at a $1/60$-second time step. The interaction configuration uses 960×540 RGB observations with a 90° horizontal field of view. All three cases use the same mobile-manipulator appearance, task interface, and stage controller, and replay the evidence states actually reached during task solving from L_0 to L_3. Each case begins with the initial graph, introduces the next observation batch only when the current graph cannot complete grounding or planning, and ends at the first evidence state yielding a task solution.

The keyframes show how new observations complete the objects and relations required by a task, while the planning path on the right reports the grounding state under the corresponding evidence. Arm pointing denotes the final localization result, and the door-opening sequence visualizes the symbolic state transition of a functional task. The three cases exhibit different relation depths and termination states, illustrating how the same dynamic closed loop allocates different observation budgets according to task evidence requirements. They are not included in the quantitative task-success statistics.

G.2 Evaluation Scope and Limitations

When the target location is not yet known, room-level coverage provides a conservative observation scope for potential task evidence, and the task-solving outcome determines how far the coverage sequence is consumed. The system can still fail because of missed small or occluded objects, incorrect cross-view instance association, non-unique grounding among nearby objects of the same category, or sensitivity to geometric relation thresholds. Additional observations may also introduce semantic-fusion noise, consistent with the non-monotonic changes in several object metrics in Table 8.

Quantitative task success is bounded by instance grounding and symbolic state-transition verification; it does not include continuous navigation, contact dynamics, or low-level control errors. Visual content in the simulation cases corresponds to entities that have entered the persistent graph at each evidence state and illustrates the system's knowledge state and its effect on the planning path, rather than serving as an additional quantitative result. Our conclusions are therefore limited to task-driven scene graph construction, instance grounding, and symbolic planning; real-robot navigation and manipulation success remain for future work.

G.3 Qualitative R1–R3 Case Studies

R1: Single-Relation Target Localization. Figure 7 shows a localization task whose initial graph contains the candidate chair but lacks the table required to verify the *left-of* relation. The comple-

mentary L_1 evidence grounds the table, completes the relational constraint, and terminates observation after the chair is uniquely identified.

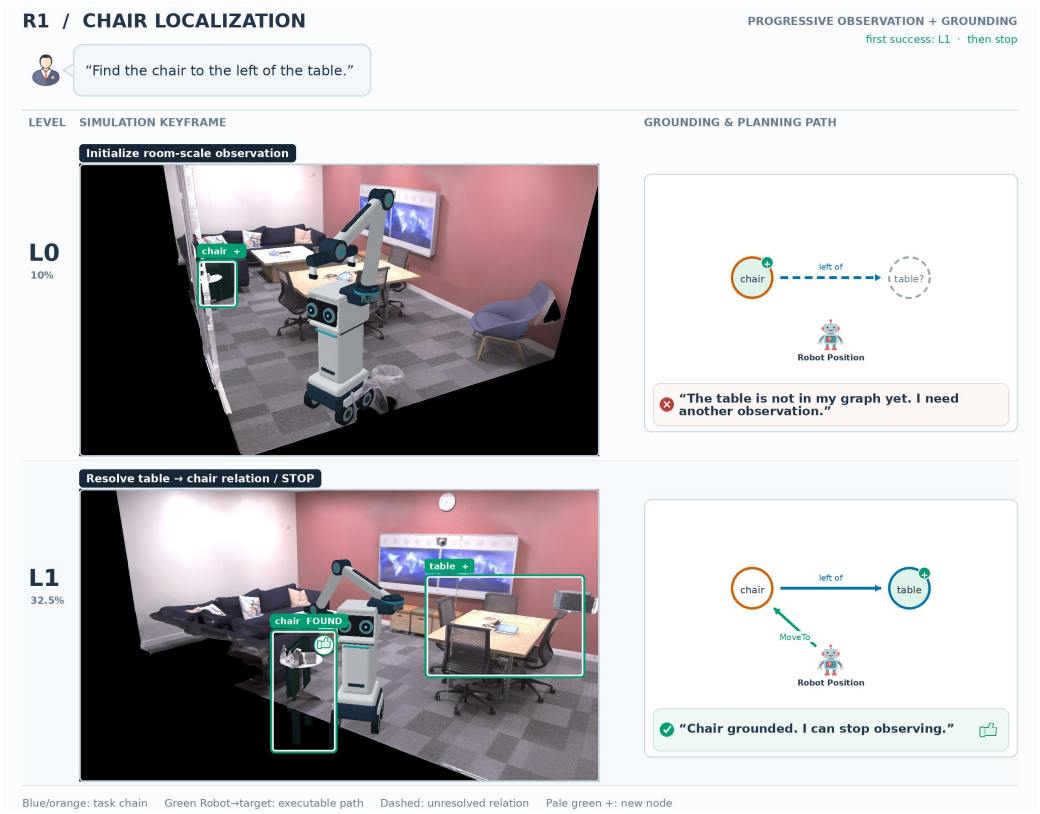

Figure 7: R1 chair-localization task. The system uses the *left-of* relation to locate the chair to the left of the table. Once the new L_1 observation yields a unique instance binding, the arm points to the target and subsequent observation terminates.

R2: Relation-Chain-Guided Door Opening. Figure 8 illustrates a functional task in which the door is grounded through the two-relation chain connecting the door, candle, and vase. The missing candle keeps the task unresolved at L_0 and L_1; once it enters the persistent graph at L_2, the planner verifies the relation chain and produces the door-opening action sequence.

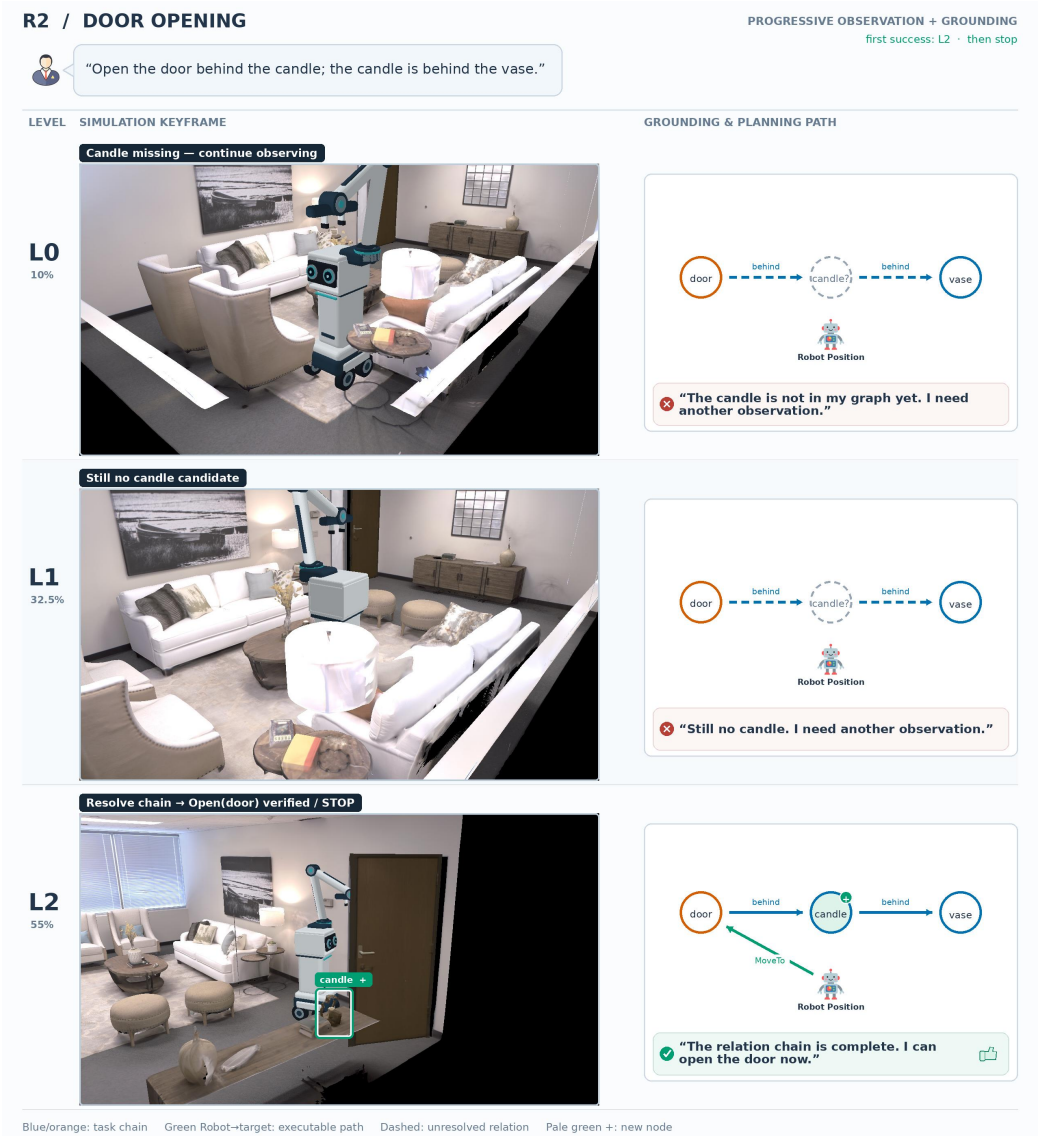

Figure 8: R2 door-opening task. The system identifies the target door from the relation chain among the door, candle, and vase, and at L_2 produces the plan `MoveTo(door)–Open(door)–Finish`.

R3: Multi-Hop Target Localization. Figure 9 presents the longest grounding chain. The chair is available in the initial graph, after which the monitor, box, and plant are incorporated by successive evidence updates. The target becomes uniquely identifiable at L_3, when the complete three-relation constraint chain is supported by the persistent graph.

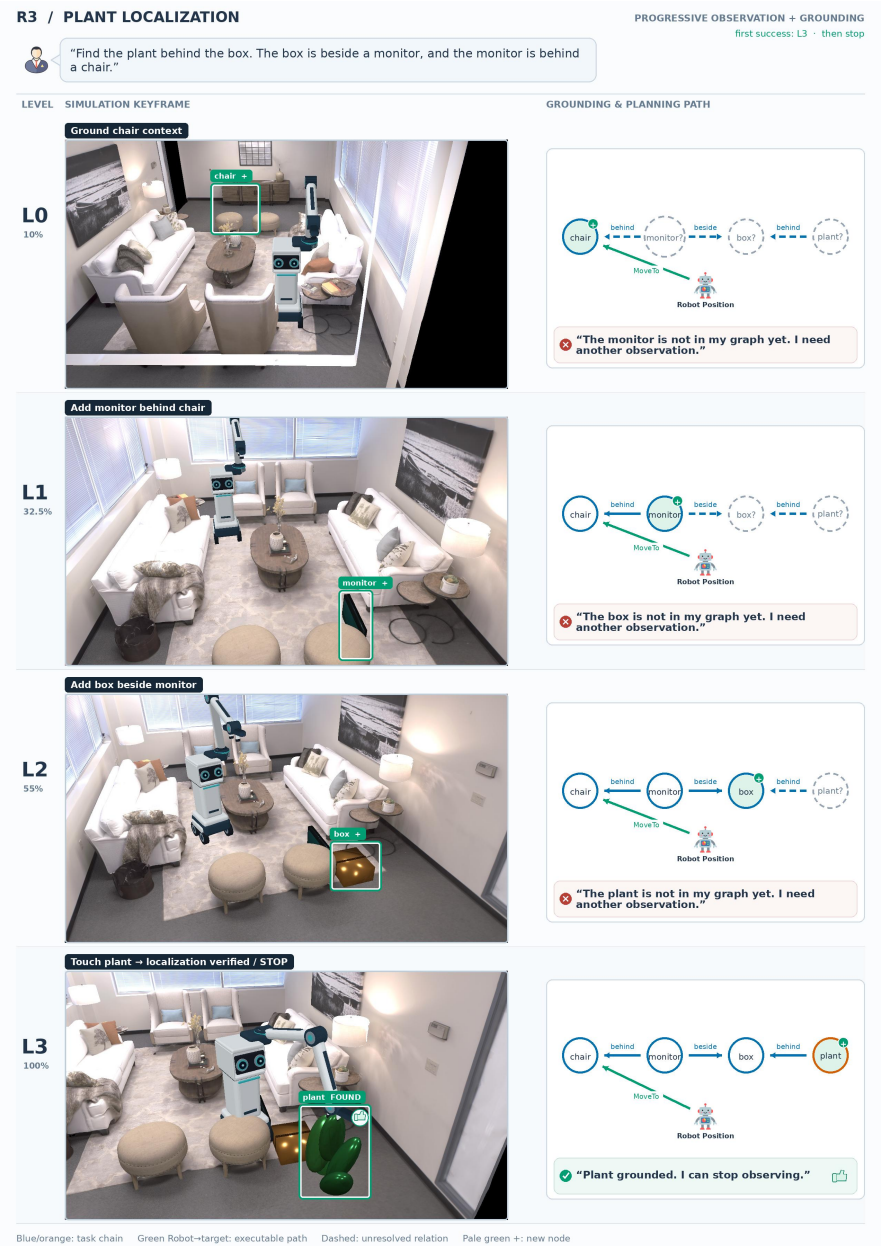

Figure 9: R3 plant-localization task. The system progressively grounds the chair, monitor, box, and plant and completes the multi-hop relation chain at L_3; the arm then points to the target plant.