TAOGRAPH: TASK-ORIENTED SCENE GRAPH LEARNING FOR EMBODIED PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied task planners are crucial for robotic navigation and manipulation. Among them, scene-graph-based approaches represent environments by semantically organizing objects into a spatial topology, supporting reasoning about object interactions and action prerequisites. However, constructing scene graphs that provide the spatial and procedural information needed for planning remains challenging. First, generic spatial representations lack task-aware adaptation, making it difficult to distinguish task-relevant spatial cues and construct relations needed for planning. Second, conventional graph supervision lacks sufficient procedural semantic guidance, making it difficult to infer action prerequisites and determine valid action orders. To address these challenges, we propose **TaoGraph**, a new task-oriented scene graph learning framework that transforms multi-view observations into task-oriented scene graphs through two key components. Semantics-Induced Spatial Reorganization (SISR) constructs task-conditioned spatial representations by modulating generic spatial features according to task instructions and scene semantics, preserving task-relevant information for graph generation. Change-Aware Procedure Distillation (CAPD) introduces procedure-guided graph supervision by distilling procedure-enriched graph predictions with guidance from discrepancies between two graph teachers, transferring procedural knowledge for action-order reasoning. Experiments demonstrate that SISR preserves task-relevant cues for graph reconstruction, while CAPD improves sequence-sensitive reasoning with comparable overall accuracy. Results on BLINK and MMSI-Bench further demonstrate TaoGraph's generalization to broader scene understanding and reasoning tasks.

1 INTRODUCTION

Embodied task planning determines the actions and execution order required to achieve task goals in robot navigation and object manipulation based on task instructions and environmental observations (Maggio et al., 2024). Existing task planning methods combine visual observations with task instructions for multimodal reasoning (Driess et al., 2023; Fu et al., 2024a), or use textual scene descriptions and execution feedback to guide language models in generating and refining plans (Huang et al., 2022). Scene graphs, as a promising alternative representation, organize environments into objects, attributes, and relations to connect perception with task reasoning and planning (Armeni et al., 2019; Wu et al., 2021; Hughes et al., 2022; Gu et al., 2024). For embodied tasks, they can encode object functions, affordances, and interactive parts that inform feasible actions and their prerequisites. Furthermore, the objects and relations required for planning vary across tasks. Task-oriented scene graphs therefore prioritize relevant entities, relations, and interaction cues according to the current goal, providing a compact, task-focused environmental representation for task planning (Agia et al., 2022; Rana et al., 2023; Gu et al., 2024; Ju et al., 2025).

However, constructing task-oriented scene graphs that effectively support embodied planning remains challenging. Existing frameworks do not sufficiently combine task goals, object functions, and spatial structure during graph generation. The resulting graphs identify main objects but miss task-relevant interactive parts and spatial relations, limiting their support for task reasoning about object interactions and action order. This problem arises from two key challenges. 1) **Insufficient Adaptability of Spatial Representations:** Task-oriented scene graphs require semantic evidence to identify objects and functions, together with spatial evidence to understand layout and structural

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.

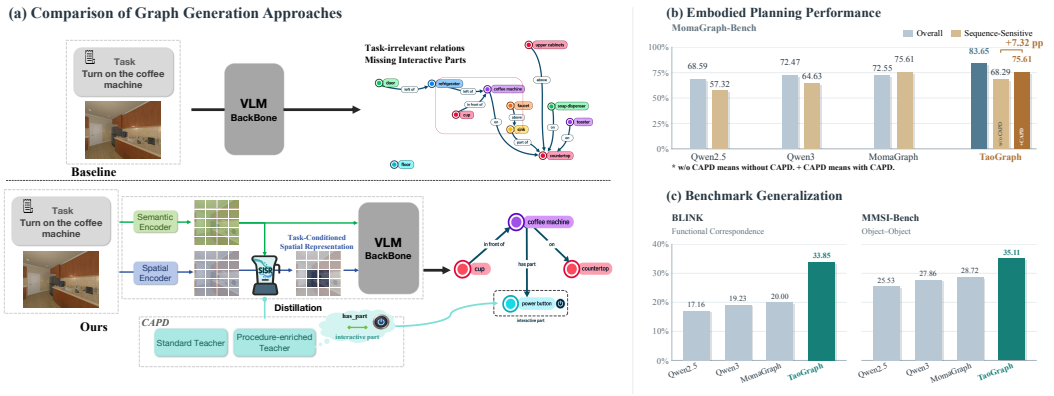

Figure 1: (a) SISR modulates generic spatial features using task instructions and scene semantics to construct task-conditioned spatial representations. CAPD combines procedure-enriched graph targets with change-aware training allocation. (b) On MomaGraph-Bench, TaoGraph achieves strong overall performance, and CAPD improves sequence-sensitive accuracy. (c) TaoGraph further generalizes to functional correspondence on BLINK and object–object spatial reasoning on MMSI-Bench.

relations. However, the relevance of spatial relations varies across tasks and scenes (Chen et al., 2022; Wang et al., 2025b; Li et al., 2025; Zhu et al., 2025). General geometry encoders capture scene geometry without explicitly distinguishing spatial evidence relevant to the current graph construction goal from irrelevant information. This lack of distinction limits the selective use of spatial evidence for constructing relations needed for task planning. 2) **Limited Support for Sequence-Sensitive Planning:** In sequence-sensitive planning, task success depends on performing actions in the correct order. Determining this order often requires identifying interactive object parts and understanding their functions (Agia et al., 2022; Rana et al., 2023; Honerkamp et al., 2024; Ju et al., 2025). For example, opening a cabinet secured by a latch requires releasing the latch before opening the door. The scene graph should capture the latch and its role in keeping the door closed to support this reasoning. However, conventional graph supervision lacks guidance on capturing the graph information needed to infer action prerequisites, limiting their support for action-order reasoning.

To address these challenges, we propose **TaoGraph**, a task-oriented scene graph learning framework for embodied task planning. TaoGraph transforms multi-view observations and task instructions into more task-relevant scene graphs by explicitly introducing semantic and spatial information. As shown in Figure 1, the framework extracts complementary evidence through semantic and spatial encoders and combines task-conditioned representation construction with graph supervision that supports action-order reasoning, helping the generated graphs preserve entities, relations, and interaction cues needed for embodied planning. Within this framework, to address insufficient adaptability of spatial representations, we propose **Semantics-Induced Spatial Reorganization (SISR)**, a task-oriented spatial representation construction mechanism. SISR incorporates task instructions and scene semantics into the construction of spatial representations, allowing generic spatial representations to be reorganized around the current task. It thereby produces spatial representations aligned with the task, providing task-relevant spatial information for scene graph generation. To address limited supervision for sequence-sensitive planning, we further propose **Change-Aware Procedure Distillation (CAPD)**, a procedure-guided supervision paradigm. CAPD incorporates procedural knowledge into graph supervision and uses changes in graph predictions under procedure-enriched supervision to guide learning toward information needed to infer action prerequisites, strengthening the generated graphs' support for action-order reasoning.

In summary, we propose TaoGraph, a new task-oriented scene graph learning framework for embodied task planning that combines task-conditioned semantic-spatial representation construction with graph supervision supporting action-order reasoning. We make the following contributions:

- We introduce Semantics-Induced Spatial Reorganization (SISR), which uses task instructions and scene semantics to reorganize generic spatial representations into task-conditioned spatial representations that preserve task-relevant information for scene graph generation.

- We introduce Change-Aware Procedure Distillation (CAPD), a procedure-guided graph supervision paradigm that incorporates procedural knowledge and uses prediction changes to guide graph learning toward information needed for action-order reasoning.
- Experiments on MomaGraph-Bench show that SISR helps preserve task-relevant information for graph reconstruction, while CAPD enhances sequence-sensitive reasoning with comparable overall accuracy. Evaluations on BLINK and MMSI-Bench further assess the framework's generalization in functional and spatial reasoning.

2 Related Work

Vision-Language Models for Scene Understanding. Vision-language models (VLMs) align visual content with natural language and have improved open-vocabulary recognition, visual reasoning, and embodied perception. Recent studies extend these capabilities to spatial and embodied reasoning by incorporating geometric information and 3D scene representations (Chen et al., 2024b; Hong et al., 2023; Chen et al., 2024a; Cheng et al., 2024; Song et al., 2025). These advances provide useful foundations for task-oriented graph construction, which requires both semantic and spatial evidence. However, incorporating geometric information alone does not ensure that spatial representations are organized around the requirements of the task. Task-oriented graph construction further requires the selective use of spatial evidence according to task goals and scene semantics. TaoGraph addresses this requirement through Semantics-Induced Spatial Reorganization (SISR), which uses task instructions and scene semantics to reorganize generic spatial representations into task-conditioned spatial representations, helping preserve task-relevant information for scene graph generation.

Scene Graphs for Embodied Planning. Scene graphs organize objects, attributes, and relations to support scene understanding and embodied reasoning (Armeni et al., 2019; Wu et al., 2021; Hughes et al., 2022). Open-vocabulary mapping and scene graph methods further connect scene representations with language, supporting more flexible queries and reasoning (Jatavallabhula et al., 2023; Koch et al., 2024; Gu et al., 2024). For embodied planning, task-oriented scene graphs provide planners with relevant objects, relations, and interaction information (Agia et al., 2022; Rana et al., 2023; Ju et al., 2025). However, sequence-sensitive planning requires more than identifying relevant objects and spatial relations: it requires information about interactive parts and their functions to infer action prerequisites and determine action order. Conventional graph supervision does not explicitly target these reasoning requirements. TaoGraph addresses this gap through CAPD, a procedure-guided graph supervision paradigm that incorporates procedural knowledge into scene graph learning to support action-order reasoning.

3 Method

TaoGraph is a task-oriented scene graph learning framework for embodied task planning. As shown in Figure 2, the framework uses semantic and spatial encoders to extract complementary evidence from multi-view observations. Semantics-Induced Spatial Reorganization (SISR) reorganizes spatial representations using the task instruction and scene semantics for subsequent graph generation alongside the original semantic stream (Section 3.2). During training, Change-Aware Procedure Distillation (CAPD) uses differences between teacher graph predictions to guide graph learning, improving the generated graphs' support for action-order reasoning (Section 3.3). At inference time, following the Graph-then-Plan approach (Appendix A), TaoGraph generates a spatially grounded, task-oriented scene graph and then uses it for embodied task planning.

3.1 Problem Formulation

Given multi-view visual observations $\mathcal{I} = \{I_v\}_{v=1}^{N}$ and a task instruction T, TaoGraph generates a task-oriented scene graph $G = (V, E, A)$, where V contains object and interactive-part nodes, E encodes semantic and spatial relations, including contact information, and A contains graph-level action and function attributes. Rather than exhaustively describing the scene, G preserves planning-relevant entities and relations and serves as the intermediate representation for embodied task planning.

3.2 Scene Graph Generation with Semantics-Induced Spatial Representations

To generate scene graphs that support embodied task planning, we extract scene semantic representations and generic spatial representations. Since the latter are not organized around a specific task,

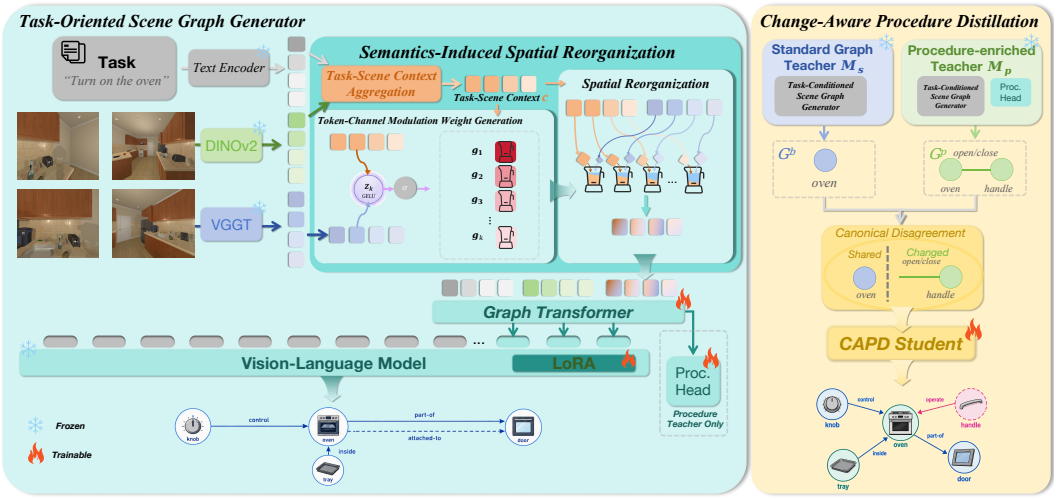

Figure 2: Overview of TaoGraph. Semantics-Induced Spatial Reorganization (SISR) uses task instructions and scene semantics to reorganize generic spatial representations into task-conditioned representations. Change-Aware Procedure Distillation (CAPD) turns canonical changes between standard and procedure-enriched teacher predictions into procedure-sensitive supervision, while preserving the graph schema and inference architecture.

SISR uses the task instruction and scene semantics to reorganize them into task-conditioned spatial representations. The resulting spatial stream is jointly encoded with the original semantic stream to construct graph tokens for task-oriented scene graph generation.

3.2.1 Semantic and Spatial Encoding

Given multi-view observations $\mathcal{I} = \{I_v\}_{v=1}^{N}$, we employ DINOv2 (Oquab et al., 2023) and VGGT (Wang et al., 2025a) to extract semantic features and generic geometry-aware spatial features. For each view I_v, the semantic feature is $x_{\text{sem},v} = \text{DINOv2}(I_v)$. For the spatial branch, VGGT jointly processes all views in $\mathcal{I}$. Let $f_{\text{VGGT},v}$ denote VGGT feature associated with the v-th view after multi-view aggregation. We augment it with the normalized layout descriptor of the corresponding view: $x_{\text{spa},v} = f_{\text{VGGT},v} + P_{\text{layout}}(\ell_v)$, where ℓ_v denotes the normalized layout descriptor and P_{layout} is a learnable projector.

Let $X_{\text{sem}} = \{x_{\text{sem},v}\}_{v=1}^{N}$ and $X_{\text{spa}} = \{x_{\text{spa},v}\}_{v=1}^{N}$. The two streams are projected into a shared D-dimensional space as $H_{\text{sem}} = P_{\text{sem}}(X_{\text{sem}}) + e_{\text{sem}}$ and $H_{\text{spa}} = P_{\text{spa}}(X_{\text{spa}}) + e_{\text{spa}}$, where P_{sem} and P_{spa} are learnable projectors for the semantic and spatial streams, with e_{sem} and e_{spa} denoting the corresponding modality-type embeddings. Here, H_{spa} denotes the generic spatial representation that is subsequently reorganized by SISR.

3.2.2 Semantics-Induced Spatial Reorganization

To address insufficient task awareness in generic spatial representations, SISR consists of two stages: Task-Scene Context Aggregation combines the task condition and scene semantic summary into a task-scene context. Task-Conditioned Spatial Reorganization uses this context to reorganize generic spatial tokens into task-conditioned spatial representations.

Task-Scene Context Aggregation. For a given multi-view sample and its task instruction, we first obtain a task condition $t \in \mathbb{R}^D$ by mean-pooling the valid prompt embeddings, and summarize the semantic stream as a scene semantic summary $s = \text{MaskedMean}(H_{\text{sem}})$. The two representations are combined to form the task-scene context:

$$c = \text{LN}_c\left(\text{LN}_{\text{sem}}(s) + \text{LN}_{\text{text}}(t)\right), \qquad c \in \mathbb{R}^D \tag{1}$$

where LN denotes layer normalization. The task-scene context c conditions the subsequent spatial representation construction.

Task-Conditioned Spatial Modulation. For each generic spatial token $H_{\text{spa},k} \in \mathbb{R}^D$, we first apply layer normalization: $\bar{H}_{\text{spa},k} = \text{LN}_{\text{spa}}\left(H_{\text{spa},k}\right)$. We then compute token-specific, channel-

wise modulation coefficients from the task-scene context and the spatial token:

$$z_k = \mathrm{GELU}\left(W_c c + W_s \bar{H}_{\mathrm{spa},k}\right) \tag{2}$$

$$g_k = \sigma\left(W_o z_k\right), \qquad g_k \in (0,1)^D \tag{3}$$

Here, W_c and W_s are learnable projections into a shared hidden space, z_k is the resulting hidden representation, and W_o maps it to D channel-wise coefficients. The functions GELU and σ denote the GELU activation and sigmoid function, respectively. Each coefficient $g_{k,d}$ controls the relative contributions of the task–scene context and the normalized spatial token at channel d.

Using these coefficients, we construct the task-conditioned spatial token:

$$\widetilde{H}_{\mathrm{spa},k} = \mathrm{LN}_{\mathrm{out}}\left(g_k \odot c + (1 - g_k) \odot \bar{H}_{\mathrm{spa},k}\right) + e_{\mathrm{tc}}, \tag{4}$$

where $\widetilde{H}_{\mathrm{spa},k} \in \mathbb{R}^D$, $\odot$ denotes element-wise multiplication, and $e_{\mathrm{tc}} \in \mathbb{R}^D$ is a learnable type embedding. Larger values of $g_{k,d}$ give greater weight to the task–scene context, whereas smaller values give greater weight to the corresponding spatial feature. This token-specific, channel-wise modulation reorganizes the spatial representations under task and scene conditioning. The resulting tokens form the task-conditioned spatial stream $\widetilde{H}_{\mathrm{spa}}$, which is processed together with the original semantic stream H_{sem} for graph-token aggregation.

3.2.3 Graph Token Construction and Decoding

We aggregate the original semantic stream H_{sem} and the task-conditioned spatial stream $\widetilde{H}_{\mathrm{spa}}$ produced by SISR into a compact set of graph tokens. We achieve this using the *Graph Transformer* [1], a two-layer Transformer encoder (Vaswani et al., 2017). Specifically, we introduce m learnable graph queries $Q_G \in \mathbb{R}^{m \times D}$ and compute $H_G = \Phi_{\mathrm{G}}([Q_G; H_{\mathrm{sem}}; \widetilde{H}_{\mathrm{spa}}])_{1:m} \in \mathbb{R}^{m \times D}$, where $[\cdot;\cdot]$ denotes token concatenation and Φ_{G} denotes the Graph Transformer. The outputs at the m graph-query positions form H_G.

The graph tokens H_G are injected into the vision-language model by replacing the embeddings at dedicated graph-token positions. Conditioned on H_G and the task instruction T, the model generates a task-oriented scene graph following the schema defined in Section 3.1.

3.2.4 Graph Generation Loss

The graph generator is trained with an autoregressive graph-generation loss, following standard maximum-likelihood training for conditional sequence generation (Vinyals et al., 2015), together with auxiliary supervision on graph-level attributes and graph structure. For a ground-truth serialized graph sequence $(y_{i,1}, \ldots, y_{i,L_i})$, the graph language-modeling loss is defined as $\mathcal{L}_{\mathrm{LM}}^{\mathrm{graph}} = -\sum_i \sum_{\ell=1}^{L_i} \log p_\theta(y_{i,\ell} \mid y_{i,<\ell}, H_{G,i}, T_i)$.

We further apply an action-function loss $\mathcal{L}_{\mathrm{af}} = \alpha_{\mathrm{act}}\mathcal{L}_{\mathrm{act}} + \alpha_{\mathrm{func}}\mathcal{L}_{\mathrm{func}}$ to supervise graph-level action and function types, together with a structural loss $\mathcal{L}_{\mathrm{str}} = \alpha_{\mathrm{edge}}\mathcal{L}_{\mathrm{edge}} + \alpha_{\mathrm{count}}\mathcal{L}_{\mathrm{count}} + \alpha_{\mathrm{spatial}}\mathcal{L}_{\mathrm{spatial}} + \alpha_{\mathrm{touch}}\mathcal{L}_{\mathrm{touch}}$ for edge presence, edge count, spatial relations, and touching states.

The base graph-generation loss is $\mathcal{L}_{\mathrm{base}} = \mathcal{L}_{\mathrm{LM}}^{\mathrm{graph}} + \lambda_{\mathrm{af}}\mathcal{L}_{\mathrm{af}} + \lambda_{\mathrm{str}}\mathcal{L}_{\mathrm{str}}$, where λ_{af} and λ_{str} control the contributions of the two auxiliary losses relative to the graph language-modeling loss, while the α coefficients weight their individual supervision terms. The specific loss weights used in our experiments are provided in Appendix B.3.

3.3 Change-Aware Procedure Distillation

To provide scene graph learning with supervision for action-order reasoning, we propose CAPD, a procedure-guided graph supervision paradigm. CAPD incorporates procedural knowledge into student learning through procedure-enriched graph targets and uses prediction differences between a Standard Graph Teacher and a Procedure-Enriched Graph Teacher to guide training emphasis. This process guides the student toward graph information needed to infer action prerequisites.

[1] In this paper, *Graph Transformer* refers to a standard Transformer encoder used to construct graph tokens, rather than a Transformer designed to operate directly on graph-structured data.

3.3.1 PROCEDURE-ENRICHED GRAPH TEACHER SUPERVISION

To incorporate procedural information into graph supervision, we train a Standard Graph Teacher M_s with the base graph-generation loss $\mathcal{L}_{\text{base}}$ and a Procedure-Enriched Graph Teacher M_p with additional procedure-category supervision. This additional supervision encourages the graph representations of M_p to encode procedural interaction patterns relevant to task execution.

For each training sample, we derive a procedure label y_i^{proc} from the graph annotation and task instruction using action type, function type, and functional relations. We use seven coarse categories, detailed in Appendix C, to describe interaction patterns such as operating a part before opening an object. These labels provide procedural guidance without specifying complete action sequences.

The Procedure-Enriched Graph Teacher predicts the procedure category from its graph tokens using an auxiliary classification head: $z_i^{\text{proc}} = h_{\text{proc}}\left(\frac{1}{m}\sum_{j=1}^{m} H_{G,i,j}^{p}\right)$, where $H_{G,i,j}^{p}$ is the j-th graph token produced by M_p for sample i, m is the number of graph tokens, h_{proc} is the classification head, and z_i^{proc} denotes the procedure-category logits. The auxiliary loss is$\mathcal{L}_{\text{proc}} = \sum_i \text{CE}\left(z_i^{\text{proc}}, y_i^{\text{proc}}\right)$, where CE denotes cross-entropy. The training loss for M_p is

$$\mathcal{L}_{\text{teacher}}^{p} = \mathcal{L}_{\text{base}} + \lambda_{\text{proc}}\mathcal{L}_{\text{proc}}, \tag{5}$$

where λ_{proc} controls the contribution of procedure-category supervision. By applying this supervision to the graph tokens, we couple graph-generation training with the recognition of procedural interaction patterns. After training, the teachers generate graph predictions for each training sample $(\mathcal{I}_i, T_i)$: $G_i^s = M_s(\mathcal{I}_i, T_i)$ and $G_i^p = M_p(\mathcal{I}_i, T_i)$. The procedure-enriched predictions provide graph targets for student learning, while the standard predictions serve as a reference for detecting graph changes. These predictions are canonicalized and compared to determine the training allocation described next.

3.3.2 PROCEDURE-ENRICHED SUPERVISION AND CHANGE-AWARE ALLOCATION

CAPD first extracts graph prediction changes between standard and procedure-enriched supervision, then uses the resulting change signal to guide the allocation of procedure-enriched graph supervision during student training.

Graph Change Extraction. For each training sample $(\mathcal{I}_i, T_i)$, we compare the graph predictions from two teachers. We first canonicalize both predictions:$\widetilde{G}_i^q = \mathcal{C}\left(M_q(\mathcal{I}_i, T_i)\right), q \in \{s, p\}$ where $\mathcal{C}$ imposes a deterministic ordering on graph elements, preventing ordering differences from being treated as changes in graph content. We then extract a sample-level graph change signal: $\Delta_i = \mathbb{I}\left[\widetilde{G}_i^p \neq \widetilde{G}_i^s\right]$, where $\mathbb{I}[\cdot]$ denotes the indicator function. The signal Δ_i identifies whether the canonical graph predictions differ between standard and procedure-enriched supervision.

Change-Aware Supervision Transfer. We use the procedure-enriched graph prediction as the student target and the extracted change signal to determine its training exposure $G_i^\star = \widetilde{G}_i^p, w_i = 1+\Delta_i$. The target $G_i^\star$ transfers graph supervision from the procedure-enriched teacher, while w_i assigns additional exposure to samples with changed predictions. Specifically, each changed sample appears twice during training, and each unchanged sample appears once. When the predictions agree, $G_i^\star$ is also their shared graph target.

3.3.3 STUDENT DISTILLATION

The student is initialized from the graph generator trained with SISR and optimized using the graph targets $G_i^\star$ with the change-aware sample weights w_i defined above. The teachers are used offline to construct graph targets and determine training exposure, and the auxiliary classification head of M_p is not included in the student.

Let $\text{seq}(G_i^\star) = (y_{i,1}^\star, \ldots, y_{i,L_i}^\star)$ denote the serialized target with L_i tokens. The weighted graph language-modeling loss is $\mathcal{L}_{\text{LM}}^{\text{CAPD}} = -\sum_i w_i \sum_{\ell=1}^{L_i} \log p_\theta\left(y_{i,\ell}^\star \mid y_{i,<\ell}^\star, H_{G,i}, T_i\right)$, where $H_{G,i}$ is the student's graph-token representation. We retain the graph-level attribute and structural losses, using supervision derived from $G_i^\star$ and the same sample weights: $\mathcal{L}_{\text{af}}^{\text{CAPD}} = \sum_i w_i \ell_{\text{af},i}^\star$, $\mathcal{L}_{\text{str}}^{\text{CAPD}} = \sum_i w_i \ell_{\text{str},i}^\star$, where $\ell_{\text{af},i}^\star$ and $\ell_{\text{str},i}^\star$ are the corresponding per-sample losses. The overall distillation loss is $\mathcal{L}_{\text{CAPD}} = \mathcal{L}_{\text{LM}}^{\text{CAPD}} + \lambda_{\text{af}}\mathcal{L}_{\text{af}}^{\text{CAPD}} + \lambda_{\text{str}}\mathcal{L}_{\text{str}}^{\text{CAPD}}$.

At inference time, TaoGraph follows the Graph-then-Plan approach used in MomaGraph (Ju et al., 2025), first generating a scene graph and then using it for embodied task planning. Details of the inference process and the prompt templates used in both stages are provided in Appendices A and E.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We conduct our experiments on MomaGraph-Bench (Ju et al., 2025), covering spatial, object, action, affordance, effect, and sequence reasoning. Graph generation is evaluated on the held-out graph validation split, while task reasoning is evaluated on the benchmark evaluation split. To evaluate generalization beyond MomaGraph-Bench, we further conduct external evaluation on BLINK (Fu et al., 2024b) and MMSI-Bench (Yang et al., 2026). We evaluate Functional Correspondence, Object Localization, and Spatial Relation on BLINK, and object–object positional relation reasoning on MMSI-Bench. More details about Datasets are provided in Appendix D.1.

Evaluation Metrics. We report multiple-choice accuracy for task planning reasoning and additionally evaluate the sequence-sensitive subset (Appendix D.2), non-sequential subset, and Sequence Planning category. Graph generation is evaluated using action and function accuracy, node F1, edge F1, and spatial relation F1. External benchmarks are evaluated by multiple-choice accuracy. Detailed metric definitions are provided in Appendix D.3.

Baselines. We compare TaoGraph with closed-source VLMs like Gemini 3 (Google DeepMind, 2026), GPT-5.5 (OpenAI, 2026) and some open-source models like Qwen2.5-VL (Yang et al., 2024), Qwen3-VL (Yang et al., 2025) and MomaGraph-R1 (Ju et al., 2025) under the graph-plan setting.

4.2 MAIN RESULTS

4.2.1 COMPARISON ON MOMAGRAPH-BENCH

We compare TaoGraph with general-purpose open-source VLMs, MomaGraph-R1, and closed-source VLMs on MomaGraph-Bench. As shown in Table 1, our method achieves 83.65% overall accuracy, outperforming all listed methods. It exceeds the strongest baseline, GPT-5.5, by 1.52 percentage points and MomaGraph-R1 by 11.10 percentage points. These results demonstrate TaoGraph's effectiveness in embodied task planning reasoning on this benchmark.

Table 1: Performance comparison on MomaGraph-Bench (%). **Bold** and underlined values denote the best and second-best results within each comparison group, respectively. This convention is used throughout the paper unless otherwise specified.

Type	Models	Params	Overall
Closed Source	Gemini 3	–	75.89
	GPT-5.5	–	<u>82.13</u>
Open Source	Qwen2.5-VL-7B-Instruct	7B	68.59
	Qwen3-VL-8B-Instruct	8B	72.47
	MomaGraph-R1	7B	72.55
	Ours	9B	**83.65**

4.2.2 GRAPH GENERATION QUALITY

We evaluate scene graph generation on the held-out validation split containing 54 samples. Action and Function denote graph-level action and function classification accuracy, respectively. Node F1 and Edge F1 measure node and complete-edge recovery, while Spatial F1 measures the recovery of spatial relation types at the graph level. As shown in Table 2, our method achieves the best performance across all reported metrics. These results show that its advantages extend beyond graph-level attribute prediction to recovering objects and their relations.

Several baselines obtain zero Edge F1 under the strict edge-matching criterion: a predicted edge is counted as correct only when its object pair, functional relation, complete spatial relation set, and touching state all match the reference edge. A zero score therefore indicates that no predicted edge satisfies all matching conditions, rather than an absence of predicted relations.

Table 2: Graph generation quality on the held-out validation split (%).

Type	Models	Action	Function	Node F1	Edge F1	Spatial F1
Closed Source	Gemini 3	50.00	72.22	41.41	0.00	54.61
	GPT-5.5	62.96	94.44	48.57	0.00	70.03
Open Source	Qwen2.5-VL-7B-Instruct	53.70	74.07	32.34	0.00	31.80
	Qwen3-VL-8B-Instruct	48.15	87.04	37.92	0.00	46.51
	MomaGraph-R1	77.78	88.89	56.37	2.80	62.95
	Ours	**90.74**	**98.15**	**81.59**	**42.58**	**77.13**

4.2.3 Sequence-Sensitive Evaluation

We evaluate action-order reasoning on the sequence-sensitive subset and the benchmark-defined Sequence Planning category. Their construction is detailed in Appendix D.2. As shown in Table 3, adding CAPD improves sequence-sensitive subset accuracy from 68.29% to 75.61% and Sequence Planning accuracy from 65.33% to 73.33%. Non-sequential accuracy decreases from 85.00% to 84.18%. These results show that CAPD improves sequence-sensitive reasoning while retaining strong performance on non-sequential questions. Compared with other methods, TaoGraph outperforms all evaluated closed-source models on both sequence-focused metrics and matches MomaGraph-R1, while exceeding its non-sequential accuracy by 11.84 percentage points. Figure 3 further illustrates changes in the generated graph: compared with the model without CAPD, the full model includes the interactive part *handle* and its functional relation to the trashcan, and selects the correct action sequence during subsequent planning. This example illustrates the interaction information relevant to action-order reasoning that CAPD aims to capture.

Table 3: Sequence-sensitive evaluation on MomaGraph-Bench (%).

Models	Seq. Sub.	Seq. Plan.	Non-Seq.
Gemini 3	50.00	46.67	77.62
GPT-5.5	70.73	68.00	82.89
Qwen2.5-VL-7B-Instruct	57.32	53.33	69.34
Qwen3-VL-8B-Instruct	64.63	61.33	72.99
MomaGraph-R1	75.61	73.33	72.34
Ours w/o CAPD	68.29	65.33	**85.00**
Ours	**75.61**	**73.33**	84.18

4.2.4 External Evaluation

To examine whether the learned graph representation transfers beyond MomaGraph-Bench, we further evaluate TaoGraph on BLINK and MMSI-Bench. As shown in Table 4, TaoGraph achieves the best performance on BLINK Functional Correspondence and MMSI-Bench object–object positional reasoning, while remaining competitive on BLINK Object Localization and Spatial Relation. These results indicate that the learned graph representation transfers beyond MomaGraph-Bench, particularly in functional correspondence and object-centric spatial reasoning.

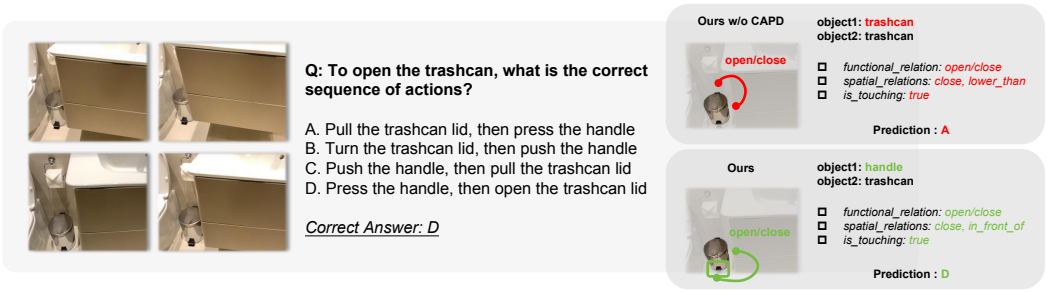

Figure 3: Qualitative illustration of sequence-sensitive graph reasoning. Compared with the TaoGraph w/o CAPD, TaoGraph includes the interactive part *handle* and its *open/close* relation to the trashcan, enabling the model to select the correct action sequence.

Table 4: External generalization on BLINK and MMSI-Bench (%).

Models	BLINK			MMSI-Bench
	Func. Corr.	Obj. Loc.	Spatial Rel.	Obj.–Obj.
Qwen2.5-VL-7B-Instruct	17.16	47.54	75.42	25.53
Qwen3-VL-8B-Instruct	19.23	**58.20**	**83.92**	27.86
MomaGraph-R1	20.00	40.16	79.02	28.72
Ours *	**33.85**	**58.20**	79.72	**35.11**

* *Ours uses Qwen2.5-VL-7B-Instruct as vision-language model backbone.*

4.3 Ablation Studies

4.3.1 Ablation of Semantics-Induced Spatial Reorganization

We compare alternative spatial representation construction schemes, including single-branch representations, direct spatial feature injection, cross-modal interaction, task-conditioned spatial selection, scalar modulation, and SISR. Single-branch and direct-injection variants perform worse than SISR, indicating that generic spatial features alone do not provide sufficient task-relevant evidence for scene graph generation. SISR achieves the best results on all reported metrics. These results show that modulating generic spatial features with task instructions and scene semantics helps construct spatial representations that preserve structural information needed for graph generation.

Table 5: Ablation study of SISR (%).

Construction	Node	Edge	Spatial	Bench.
Semantic	73.80	17.11	70.27	81.22
Spatial	75.29	18.79	66.30	78.71
Direct	68.68	10.60	68.27	76.50
Cross-Attn.	59.46	9.79	63.13	76.50
Task-Select.	75.68	19.05	72.11	82.05
Scalar-Mod.	78.10	27.10	74.79	82.43
SISR (Ours)	**81.59**	**42.58**	**77.13**	**83.95**

4.3.2 Effect of Change-Aware Procedure Distillation

We disentangle the effects of procedure-enriched graph supervision and change-aware training allocation. The procedure-teacher variant uses procedure-enriched graph targets for all samples but removes CAPD's change-aware allocation. The alternative target-selection variant evaluates another way of using the two teacher predictions. As shown in Table 6, both variants improve sequence-sensitive performance over the model without CAPD, confirming the value of procedure-enriched supervision. The full CAPD configuration achieves the best results on both sequence metrics. This result shows that change-aware allocation provides an additional benefit beyond procedure-enriched graph targets while retaining strong non-sequential performance.

Table 6: Ablation study of CAPD (%).

Method	Seq. Sub.	Seq. Plan.	Non-Seq.
Ours w/o CAPD	68.29	65.33	**85.00**
Alternative Target Selection	70.73	68.00	84.18
Procedure-Teacher Supervision	71.95	69.33	83.37
Ours	**75.61**	**73.33**	84.18

5 Conclusion

We present TaoGraph, a new task-oriented scene graph learning framework for embodied task planning. Semantics-Induced Spatial Reorganization (SISR) modulates generic spatial features using task instructions and scene semantics to construct task-conditioned spatial representations that preserve task-relevant information. Change-Aware Procedure Distillation (CAPD) combines procedure-enriched graph targets with change-aware training allocation to improve graphs' support for action-order reasoning. On MomaGraph-Bench, SISR improves graph quality, while CAPD improves sequence-sensitive planning with comparable overall accuracy. Evaluations on BLINK and MMSI-Bench show effectiveness in functional correspondence and object-centric spatial reasoning. These results can inspire future work on extending TaoGraph to more complex interactions and a broader range of embodied tasks.

6 AI USE STATEMENT

AI tools were used to assist with drafting sections of this paper and polishing the writing for grammar, clarity, and readability. They were also used to support the retrieval and discovery of related work and to generate boilerplate code for experimental and testing scripts. The authors reviewed and revised all AI-assisted text, independently checked relevant references, and verified AI-assisted code against the implemented methods and reported experiments. The authors take full responsibility for the final paper and all associated code and artifacts.

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 *Conference on Robot Learning*, pp. 46–58. PMLR, 2022.

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.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14455–14465, 2024a. doi: 10.1109/CVPR52733.2024.01370.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Language conditioned spatial relation reasoning for 3d object grounding. *Advances in neural information processing systems*, 35:20522–20535, 2022.

Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26428–26438, 2024b.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024a.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pp. 148–166. Springer, 2024b.

Google DeepMind. Gemini 3 Pro Model Card, 5 2026. URL `https://deepmind.google/models/model-cards/gemini-3-pro/`.

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.

Daniel Honerkamp, Martin Büchner, Fabien Despinoy, Tim Welschehold, and Abhinav Valada. Language-grounded dynamic scene graphs for interactive object search with mobile manipulation. *IEEE Robotics and Automation Letters*, 9(10):8298–8305, 2024.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

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.

Yuanchen Ju, Yongyuan Liang, Yen-Jen Wang, Nandiraju Gireesh, Yuanliang Ju, Seungjae Lee, Qiao Gu, Elvis Hsieh, Furong Huang, and Koushil Sreenath. Momagraph: State-aware unified scene graphs with vision-language model for embodied task planning. *arXiv preprint arXiv:2512.16909*, 2025.

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.

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3707–3717. IEEE, 2025.

Dominic Maggio, Yun Chang, Nathan Hughes, Matthew Trang, Dan Griffith, Carlyn Dougherty, Eric Cristofalo, Lukas Schmid, and Luca Carlone. Clio: Real-time task-driven open-set 3d scene graphs. *IEEE Robotics and Automation Letters*, 9(10):8921–8928, 2024.

OpenAI. GPT-5.5 System Card, 4 2026. URL `https://openai.com/index/gpt-5-5-system-card/`.

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.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.

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.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15768–15780, 2025.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *Robotics: Science and Systems*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025a.

Xinyi Wang, Na Zhao, Zhiyuan Han, Dan Guo, and Xun Yang. Augrefer: Advancing 3d visual grounding via cross-modal augmentation and spatial relation-based referring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8006–8014, 2025b.

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.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, et al. Mmsi-bench: A benchmark for multi-image spatial intelligence. In *International Conference on Learning Representations*, volume 2026, pp. 157051–157088, 2026.

Yihang Zhu, Jinhao Zhang, Yuxuan Wang, Aming Wu, and Cheng Deng. Vgmamba: Attribute-to-location clue reasoning for quantity-agnostic 3d visual grounding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5295–5304. IEEE, 2025.

A GRAPH-THEN-PLAN INFERENCE

At inference time, TaoGraph follows the two-stage Graph-then-Plan process illustrated in Figure 4. Given multi-view observations $\mathcal{I}$ and a task instruction T, the graph-generation stage applies Semantics-Induced Spatial Reorganization (SISR) to construct task-conditioned spatial representations, which are combined with the original semantic representations to generate a spatially grounded, task-oriented scene graph:

$$\widehat{G} = \arg\max_{G} p_\theta(G \mid \mathcal{I}, T). \tag{6}$$

The generated graph follows the schema defined in Section 3.1 and provides a compact representation of the observed scene.

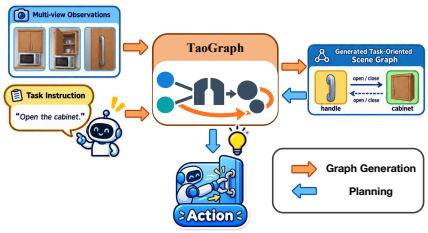

Figure 4: Graph-then-plan inference with TaoGraph.

In the planning stage, the serialized graph $\widehat{G}$ is provided together with the task instruction T, planning question Q, and candidate set $\mathcal{C}$:

$$\widehat{c} = \arg\max_{c_k \in \mathcal{C}} p_\theta(c_k \mid \widehat{G}, T, Q, \mathcal{C}). \tag{7}$$

After $\widehat{G}$ is generated, its serialized form serves as the planner's only visual evidence, and the original images are no longer used. The final prediction therefore depends on the scene information preserved in the generated graph. CAPD affects only the training supervision and introduces no teacher model or additional module during inference.

B IMPLEMENTATION DETAILS

B.1 MODEL ARCHITECTURE

TaoGraph uses Qwen2.5-VL-7B-Instruct as the autoregressive graph decoder, with LoRA adapters applied to the language model. Visual evidence is extracted by two frozen visual encoders. The semantic branch uses DINOv2 to obtain semantic representations, while the spatial branch uses VGGT to obtain geometry-aware spatial representations from the same multi-view observations. The language model does not directly receive raw image tokens during graph generation.

For each view, we use the DINOv2 CLS representation as the semantic feature. For the spatial branch, the views of each sample are jointly processed by the VGGT aggregator, after which the final-stage patch features are mean-pooled for each view. We further augment each pooled spatial feature with a normalized view-layout descriptor,

$$\ell_v = \left[\frac{x_1}{W}, \frac{y_1}{H}, \frac{x_2}{W}, \frac{y_2}{H}, \frac{w}{W}, \frac{h}{H}, \frac{\text{order}}{N-1}\right],$$

where (x_1, y_1, x_2, y_2) denotes the view bounding box, W and H are the dimensions of the composed observation, and N is the number of views. The layout descriptor is mapped through a learnable projection and added to the pooled VGGT feature. The semantic and spatial streams are then projected into the language-model hidden dimension and assigned modality-type embeddings.

Semantics-Induced Spatial Reorganization (SISR) modulates each generic spatial token using the task instruction and the scene semantic context described in Section 3.2.2. The modulation network uses a hidden dimension of 128 and produces channel-wise coefficients in the language-model hidden space. The resulting task-conditioned spatial representations are retained together with the original semantic representations for graph-token construction.

We use 16 learnable graph queries and a two-layer Transformer encoder to aggregate the semantic and task-conditioned spatial representations. The Graph Transformer has a hidden dimension of 3584, 8 attention heads, and a feed-forward dimension of 6144. Its outputs at the 16 graph-query

positions form the graph-token representation H_G. These graph tokens replace the input embeddings at the dedicated <graph_0>, . . ., <graph_15> positions in the Qwen prompt. The decoder subsequently generates the serialized scene graph autoregressively. Thus, the graph tokens serve as compact visual memory rather than as cross-attention memory.

B.2 Training Configuration

The DINOv2 semantic encoder, VGGT spatial encoder, and original Qwen language-model parameters are frozen during graph training. Trainable parameters include the semantic and spatial projectors, the view-layout projector, the SISR module, modality-type embeddings, graph queries, the Graph Transformer, auxiliary graph-prediction heads, and LoRA adapters applied to the language model. LoRA is applied to the q_proj, k_proj, v_proj, and o_proj modules, with rank 4, scaling factor 8, and dropout 0.05.

Training is performed with bfloat16 precision, gradient checkpointing, and ZeRO-3 optimization (Rajbhandari et al., 2020). Table 7 summarizes the main optimization hyperparameters. The CAPD student is initialized from this trained generator and uses a smaller learning rate during continued training.

Table 7: Training configuration used for TaoGraph and the CAPD student.

Configuration	Base Generator	CAPD Student
Main learning rate	1×10^{-4}	5×10^{-5}
Projector learning rate	1×10^{-4}	5×10^{-5}
LoRA learning rate	5×10^{-5}	2×10^{-5}
Weight decay	0.01	0.01
Maximum gradient norm	1.0	1.0
LoRA rank	4	4
LoRA alpha	8	8
LoRA dropout	0.05	0.05
Precision	bfloat16	bfloat16
Optimization	ZeRO-3	ZeRO-3

During training, we additionally apply spatial-stream corruption as data augmentation. Each training sample is selected for corruption with probability 0.2. For a selected sample, a severity value $\rho \sim \mathcal{U}[0.25, 1.0]$ is sampled, and either noise interpolation or feature attenuation is applied to the projected spatial representations. For noise interpolation,

$$H'_{\text{spa}} = (1 - \rho)H_{\text{spa}} + \rho\epsilon,$$

where the scale of ϵ is determined by the root-mean-square magnitude of the valid spatial features. For feature attenuation,

$$H'_{\text{spa}} = (1 - \rho)H_{\text{spa}}.$$

The same augmentation configuration is used during CAPD student training.

B.3 Loss Configuration

We use the loss formulation defined in the main paper and determine the relative weights of its auxiliary objectives based on preliminary experiments. We found that equal weighting within each auxiliary objective provides the best overall balance among the corresponding graph attributes. Accordingly, we set $\alpha_{\text{act}} = \alpha_{\text{func}} = 1$ for the action-function objective. This avoids favoring either graph-level attribute and allows action and function supervision to contribute equally to the learned graph representation.

We follow the same principle for the structural objective and set $\alpha_{\text{edge}} = \alpha_{\text{count}} = \alpha_{\text{spatial}} = \alpha_{\text{touch}} = 1$. Among the tested relative-weighting configurations, equal weighting provided the most effective overall supervision across edge presence, edge count, spatial relations, and touching states. This configuration allows the structural objective to capture complementary aspects of the graph without overemphasizing a particular structural property.

For the outer auxiliary weights, we set $\lambda_{\mathrm{af}} = 0.1$ and $\lambda_{\mathrm{str}} = 0.05$. We found that relatively small auxiliary coefficients provide a better balance with the autoregressive graph-generation objective. Since the action and function types are explicit graph-level attributes and provide direct global supervision, we assign the action-function objective a weight of 0.1. The structural objective combines four auxiliary prediction terms and is therefore assigned a more conservative weight of 0.05, which provides useful structural guidance without overwhelming the graph-generation objective.

For the edge-count prediction, the target count is clipped at four. Since TaoGraph generates compact, task-oriented graphs rather than exhaustive scene descriptions, this setting keeps the auxiliary classification focused on the structural complexity most relevant to the generated graphs while limiting the influence of unusually dense targets.

The standard graph teacher uses the same loss configuration as the base graph generator. For the procedure-enriched graph teacher, we set $\lambda_{\mathrm{proc}} = 0.05$. The procedure-category objective is intended to provide lightweight additional supervision rather than replace the graph-generation objective. In our experiments, this weight provides useful procedure-sensitive guidance while maintaining the teacher's primary graph-generation behavior. The CAPD student does not use the procedure-classification loss and retains the same loss coefficients as the base graph generator. Table 8 summarizes the loss hyperparameters used in the main experiments.

Table 8: Loss hyperparameters used in the main experiments.

Hyperparameter	Value
λ_{af}	0.1
λ_{str}	0.05
α_{act}	1.0
α_{func}	1.0
α_{edge}	1.0
α_{count}	1.0
$\alpha_{\mathrm{spatial}}$	1.0
α_{touch}	1.0
Maximum edge-count target	4
λ_{proc} (procedure teacher only)	0.05

C CAPD Training Details

This section provides additional implementation details for Change-Aware Procedure Distillation (CAPD), including procedure-label construction, teacher-graph processing, and changed-sample oversampling.

Procedure Category Construction. The Procedure-Enriched Graph Teacher uses an auxiliary classification head to predict one of seven coarse procedure categories: `none`, `power_then_activate`, `operate_part_then_open`, `enable_water_flow`, `adjust_control`, `pair_or_insert`, and `device_start`. These labels are derived automatically from the graph-training annotations and task instructions using deterministic rules over the graph-level action type, function type, functional relations, and task words. They represent coarse interaction patterns rather than complete action sequences and require no additional manually annotated procedure labels.

The Procedure-Enriched Graph Teacher predicts these categories from its mean-pooled graph tokens using an auxiliary classification head. The head consists of a LayerNorm layer followed by a two-layer MLP with a GELU activation between the linear layers. Its cross-entropy loss is weighted by $\lambda_{\mathrm{proc}} = 0.05$ and jointly optimized with the graph-generation loss, providing procedure supervision to the graph-token representations. The classification head is used only during procedure-enriched teacher training and is not included in the CAPD student.

Teacher Graph Processing and Change Detection. After training, the Standard Graph Teacher and Procedure-Enriched Graph Teacher are frozen and used to generate offline graph predictions for all samples in the MomaGraph-Scene training split. The parsed predictions are normalized to

a common graph schema through label normalization and repair, including normalization of action and function labels, removal of invalid spatial relations, replacement of invalid functional relations, and completion of node entries referenced by graph edges.

The processed graphs are then canonicalized by imposing a deterministic ordering on graph elements, preventing serialization-order differences from being treated as graph-content changes. Comparing the two canonicalized graphs yields a sample-level change indicator Δ_i, which equals one when the predictions differ and zero otherwise. The complete canonicalized prediction of the Procedure-Enriched Graph Teacher serves as the student target $G_i^\star$ for every sample. When the teachers agree, this target is also their shared prediction. The change indicator determines training exposure rather than teacher confidence or prediction quality.

Change-Aware Training Allocation. Teacher comparison partitions the 951 MomaGraph-Scene training samples into 540 *changed* samples and 411 *unchanged* samples. During CAPD student training, each changed sample appears twice per epoch, while each unchanged sample appears once, yielding $540 \times 2 + 411 = 1{,}491$ sample presentations per epoch. This implements the allocation weight $w_i = 1 + \Delta_i$ defined in the main paper, increasing training exposure for samples on which the two teachers produce different graph predictions while retaining all training samples.

Student Training. The CAPD student is initialized from the base graph generator trained with SISR and retains its architecture and graph schema. It is trained using the procedure-enriched graph targets and change-aware training allocation described above. The training loss includes the graph language-modeling loss and the action-function and structural auxiliary losses, with coefficients specified in Appendix B.3. The student does not include the procedure-classification head or use its auxiliary loss. Optimization settings are provided in Appendix B.2. CAPD therefore transfers procedure-enriched graph supervision through change-aware training allocation without introducing additional inference-time components.

D Experimental Setup and Evaluation Protocol

D.1 Datasets

We evaluate TaoGraph on four datasets with distinct supervision and evaluation roles: MomaGraph-Scene, MomaGraph-Bench, BLINK, and MMSI-Bench.

MomaGraph-Scene. MomaGraph-Scene is a graph-supervised dataset for task-oriented scene graph learning. Each sample contains multi-view observations, a task instruction, and a corresponding scene graph annotation. We use its training portion to train the base graph generator, the Standard Graph Teacher, the Procedure-Enriched Graph Teacher, and the CAPD student. A fixed held-out portion is used to evaluate graph reconstruction. The validation samples are excluded from teacher target construction and CAPD student training.

MomaGraph-Bench. MomaGraph-Bench is an embodied task-planning benchmark. Each sample provides multi-view observations, a task instruction, a natural-language planning question, multiple answer candidates, and a ground-truth answer. It does not provide scene graph supervision. We therefore use it only for downstream Graph-then-Plan evaluation: the trained graph generator first produces a task-oriented scene graph, which is then serialized and provided to the planner as the scene evidence for answering the planning question. No answer supervision from this benchmark is used during graph-model training.

BLINK. BLINK is used to evaluate the transfer of the learned visual representations to broader visual reasoning tasks. We report its Functional Correspondence, Object Localization, and Spatial Relation settings, which assess functional matching, object localization, and spatial-relation understanding, respectively. The benchmark is used only for external evaluation and provides no supervision for training TaoGraph.

MMSI-Bench. MMSI-Bench is used to evaluate object-centric spatial reasoning beyond the MomaGraph benchmarks. We report its object–object positional reasoning task, which measures whether a model can determine the relative spatial relationship between objects. MMSI-Bench is used only for external evaluation and is not involved in graph-model training.

D.2 SEQUENCE-SENSITIVE SUBSET

The benchmark-defined *Sequence Planning* category consists of questions assigned this reasoning type by MomaGraph-Bench. Its membership is determined solely by the benchmark metadata.

To additionally capture questions that explicitly involve action ordering, we construct a broader sequence-sensitive subset using deterministic metadata- and question-based rules. A question is included if it belongs to the Sequence Planning category or contains an explicit ordering cue, such as "correct sequence," "sequence of actions," "correct order," or a "first ... then ..." pattern.

We report accuracy on the broader subset as **Seq. Sub.** and on the benchmark-defined category as **Seq. Plan.**. The Sequence Planning category is contained within the sequence-sensitive subset, so the two metrics evaluate overlapping sets of questions. **Non-Seq.** denotes accuracy on the complement of the sequence-sensitive subset within MomaGraph-Bench. The same subset definitions are used for all evaluated methods.

D.3 EVALUATION METRICS

We evaluate TaoGraph at two levels: the quality of the generated scene graph and its downstream utility for task planning.

Graph Generation Metrics. Let $\hat{G}_i$ and G_i denote the predicted and reference graphs for sample i, respectively. We report graph-level attribute accuracy together with structural reconstruction metrics.

Action Accuracy and **Function Accuracy** measure the fraction of samples for which the predicted graph-level action type and function type match the corresponding reference labels:

$$\mathrm{Acc}_{\mathrm{act}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\hat{a}_i^{\mathrm{act}} = a_i^{\mathrm{act}}], \qquad \mathrm{Acc}_{\mathrm{func}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\hat{a}_i^{\mathrm{func}} = a_i^{\mathrm{func}}].$$

An unparsable graph prediction is counted as incorrect for these metrics.

Node F1 measures the recovery of task-relevant graph nodes. For each sample, the predicted and reference node sets are accumulated over the evaluation set, and precision, recall, and F1 are computed using micro-averaged true-positive, false-positive, and false-negative counts.

Edge F1 evaluates complete graph edges using the edge signature

$$(\mathrm{object1}, \mathrm{object2}, \mathrm{functional\ relationship}, \mathrm{spatial\ relation\ set}, \mathrm{touching}).$$

A predicted edge is therefore considered correct only when the object pair, functional relation, complete set of spatial relations, and touching state all match the reference edge. Edge F1 is computed as micro-F1 over the complete evaluation set.

Spatial Relation F1 evaluates the recovery of spatial-relation types present in the graph. For each sample, we form the set of spatial relation labels appearing across its edges and additionally include a `touching` label when at least one edge has a positive touching state. Precision, recall, and F1 are then computed from the aggregated relation-set matches over the evaluation set. This metric evaluates graph-level relation-type recovery and does not require a spatial relation to be associated with the correct object pair.

All graph-generation methods are evaluated with the same graph parser, label space, and metric implementation.

Planning Metrics. Downstream task planning is evaluated as multiple-choice prediction. Let $\hat{a}_i$ and a_i denote the parsed prediction and ground-truth option for sample i. Overall accuracy is computed as

$$\mathrm{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\hat{a}_i = a_i].$$

Outputs that cannot be parsed into a valid option are counted as incorrect.

To evaluate action-order reasoning separately, we additionally report accuracy on the **Sequence-Sensitive Subset** (Seq. Sub.) and the benchmark-defined **Sequence Planning** category (Seq. Plan.). The former is a broader subset identified using sequence-related metadata and question cues, whereas the latter contains only samples assigned to the Sequence Planning reasoning category by the benchmark. We also report **Non-Sequential Accuracy** (Non-Seq.) on the complement of the sequence-sensitive subset. The construction of the sequence-sensitive subset is described in Appendix D.2.

For answer parsing, we first search for an explicit `Final Choice: (X)`-style prediction and otherwise apply the same fallback parsing rules to all evaluated methods. The same multiple-choice scoring protocol is used for TaoGraph and all comparison methods.

E Prompts and Output Formats

We provide the prompt templates and output formats used in the two stages of Graph-then-Plan inference. In the graph-generation stage, the model generates a task-oriented scene graph from the task instruction and structured visual memory. In the planning stage, the generated graph is serialized and used as scene evidence for answering the downstream multiple-choice question.

E.1 Graph Generation Prompt

During graph generation, the language model is conditioned on the task instruction and 16 graph memory tokens constructed from the multi-view observations. These tokens contain the fused semantic and spatial evidence produced by the graph-generation architecture. The model is instructed to generate a scene graph following the schema defined in Section 3.1, including task-relevant nodes, functional and spatial relations, touching states, and graph-level action and function attributes.

The prompt template used for graph generation is:

```
Structured Visual Memory: <graph_0> <graph_1> ... <graph_15>

You are given structured visual memory fused from all panels
from the same sample.

Task Instruction: {task_instruction}

Generate a valid scene graph JSON that follows the required schema.
Put the final scene graph on its own line after "Answer:".
```

The expected output follows the canonical graph schema:

```
Answer: {
  "task_instruction": "...",
  "nodes": [...],
  "edges": [
    {
      "object1": "...",
      "object2": "...",
      "functional_relationship": "...",
      "spatial_relations": [...],
      "is_touching": false
    }
  ],
  "action_type": "...",
  "function_type": "..."
}
```

The generated text is parsed into the structured graph representation used for graph evaluation and downstream planning. Restricting the output to this schema provides a consistent interface between graph generation and the subsequent planning stage.

E.2 Graph-Conditioned Planning Prompt

In the planning stage, the generated scene graph is serialized and provided to the language model together with the task instruction, question, and answer candidates. The original visual observations and graph memory tokens are not provided at this stage; the serialized graph serves as the scene evidence for downstream reasoning. The same language-model checkpoint used after graph-generation training is directly applied to the planning task without additional QA fine-tuning.

The prompt template used for graph-conditioned planning is:

```
Reference Pseudo Scene Graph: {pseudo_graph_text}

Task Instruction: {task_instruction}

Based on the scene graph analysis you performed above, answer the
following multiple-choice question.

Question: {question}
Options: {options}

After generating the scene graph as requested above, please provide
your final choice for the question.

Output your final choice in the format: "Final Choice: (X)"
where X is the option letter.
```

The planner generates its response autoregressively using greedy decoding. The final prediction is extracted from the option letter in `Final Choice: (X)` and evaluated using the multiple-choice scoring protocol described in Appendix D.3. No additional chain-of-thought supervision is used in the reported planning experiments.

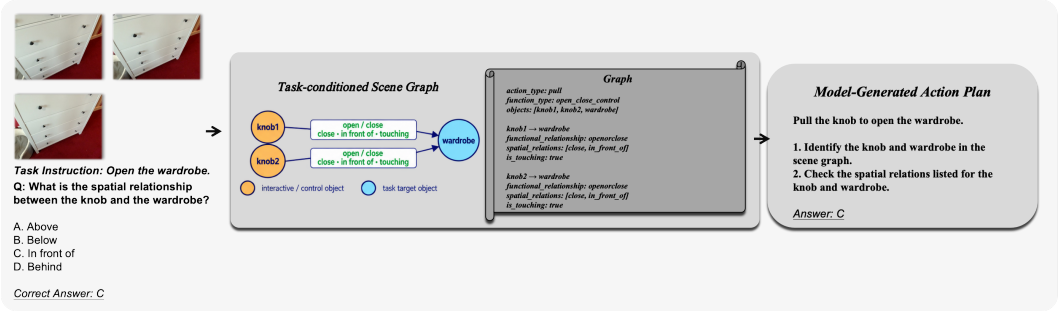

Figure 5: An example of MomaGraph-Bench performing the task "Open the wardrobe."

F More Visualization Results

F.1 Graph-then-Plan Examples on MomaGraph-Bench

Figure 5 and Figure 6 presents an example of our Graph-then-Plan pipeline. Given multi-view observations and a task instruction, TaoGraph generates a task-oriented scene graph containing task-relevant objects, functional relationships, and spatial relationships. The scene graph is then serialized and provided to the planner together with the task description and candidate answers. These examples illustrate how the generated scene graph supports task-oriented interaction and action-order reasoning.

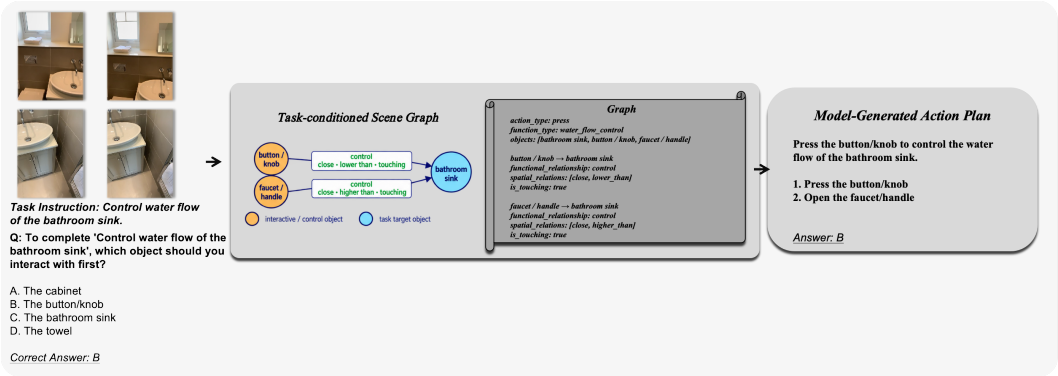

Figure 6: An example of MomaGraph-Bench performing the task "Control water flow of the bathroom sink."

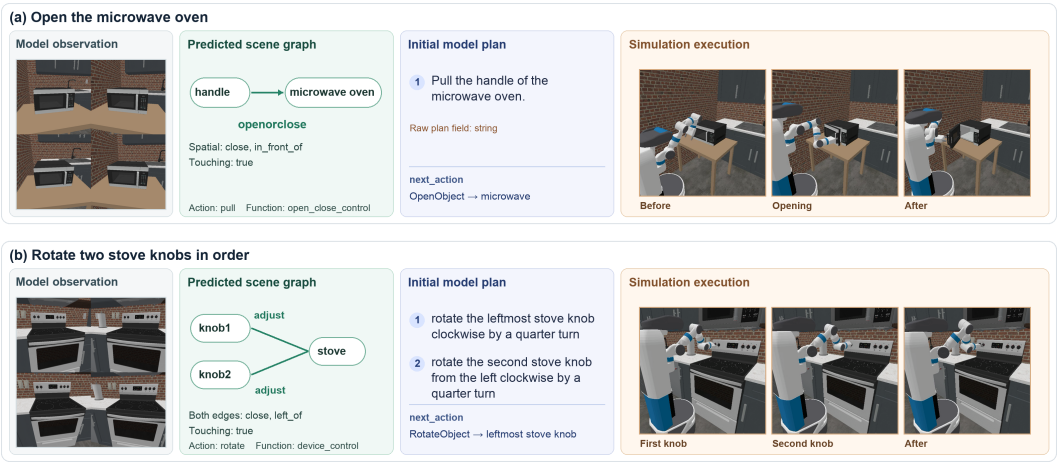

Figure 7: Qualitative examples of scene-graph-conditioned task planning in ManiSkill. (a) Opening a microwave oven. (b) Sequentially rotating two stove knobs.

F.2 Qualitative Task Planning Demonstrations

To examine the task-planning cues provided by generated scene graphs in embodied task planning, we conduct simulation tests in household scenes built in ManiSkill (Tao et al., 2025). TaoGraph generates a task-oriented scene graph from multi-view RGB images and a task instruction. During planning, The generated scene graph is then incorporated into the planning process to produce a corresponding task plan.

As shown in Figure 7, TaoGraph constructs an openorclose relation between the handle and the microwave oven in the door-opening task, and adjust relations between two knobs and the stove in the knob-operation task. These graphs preserve the task-relevant interactive parts and their functions. Using the generated graph as structured input, the planner combines it with the task instruction to determine the interaction targets and actions. For the two-knob task, it also produces an action order consistent with the instruction. These examples illustrate how task-oriented scene graphs provide structured scene information for task planning.