GaLa: Hypergraph-Guided Visual Language Models for Procedural Planning

Kun Wang[1,3†], **Yiming Li**[1,3†], **Mingcheng Qu**[1,3†], **Aqiang Zhang**[1,3], **Guang Yang**[1], **Tonghua Su**[1,2,3*]
[1] Harbin Institute of Technology, Harbin, China
[2] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China
[3] Chongqing Research Institute of HIT, Chongqing, China

Abstract

Implicit spatial relations and deep semantic structures encoded in object attributes are crucial for procedural planning in embodied AI systems. However, existing approaches often over-rely on the reasoning capabilities of vision language models (VLMs) themselves, while overlooking the rich structured semantic information that can be mined from multimodal inputs. As a result, models struggle to effectively understand functional spatial relationships in complex scenes. To fully exploit implicit spatial relations and deep semantic structures in multimodal data, we propose GaLa, a vision–language framework for multimodal procedural planning. GaLa introduces a hypergraph-based representation, where object instances in the image are modeled as nodes, and region-level hyperedges are constructed by aggregating objects according to their attributes and functional semantics. This design explicitly captures implicit semantic relations among objects as well as the hierarchical organization of functional regions. Furthermore, we design a Tri-View HyperGraph Encoder that enforces semantic consistency across the node view, area view, and node–area association view via contrastive learning, enabling hypergraph semantics to be more effectively injected into downstream VLM reasoning. Extensive experiments on the ActPlan-1K and ALFRED benchmarks demonstrate that GaLa significantly outperforms existing methods in terms of execution success rate, LCS, and planning correctness.

1 Introduction

In recent years, research in procedural planning has significantly advanced the capability of embodied agents to execute complex instructions by understanding multimodal information (Wan et al., 2025; Li et al., 2023a; Sun et al., 2025). Procedural planning refers to the process of progressively decomposing high-level linguistic commands into a sequence of precise actions required to accomplish a task (Wang et al., 2025a), ultimately enabling embodied agents to perform task planning and execution in real-world environments (Huang et al., 2025). During this process, textual instructions enable the agent to effectively understand the required task (Zhao et al., 2024), while visual inputs allow the agent to perceive its surroundings with precision (Lin et al., 2025). Through multimodal fusion (Yang et al., 2024), the embodied agent is ultimately able to execute the specified instructions.

Recent research in procedural planning can be categorized into two main streams based on the modality they primarily rely on: Large Language Models (LLMs) (Guan et al., 2023; Silver et al., 2024; Li et al., 2024b) and Vision-Language Models (VLMs) (Mu et al., 2023; Zhai et al., 2024; Yang et al., 2025b). VLM-based methods, by contrast, integrate visual observations with language input, enabling improved spatial perception and grounding. This capability allows VLMs (Yang et al., 2026) to generate action plans that better respect physical constraints in the environment. Despite this advantage, existing VLM-based approaches predominantly operate on explicit spatial representations, such as bounding boxes or object coordinates, and focus primarily on image-level semantic cues. As a result, they remain limited in capturing deeper semantic structures that are implicitly embedded in visual scenes. In particular, two critical limitations emerge. First, implicit spatial relations conveyed through object attributes—such as "can-be-placed-on", "is-supported-by" are often overlooked, as they are not explicitly encoded in geometric representations. Second, hierarchical semantics arising from groups of objects forming functional regions (e.g., "dining area", "kitchen area") are rarely modeled in a structured manner. Consequently, the agent's understanding of the scene is

[†]Equal contribution.
[*]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 19590–19603

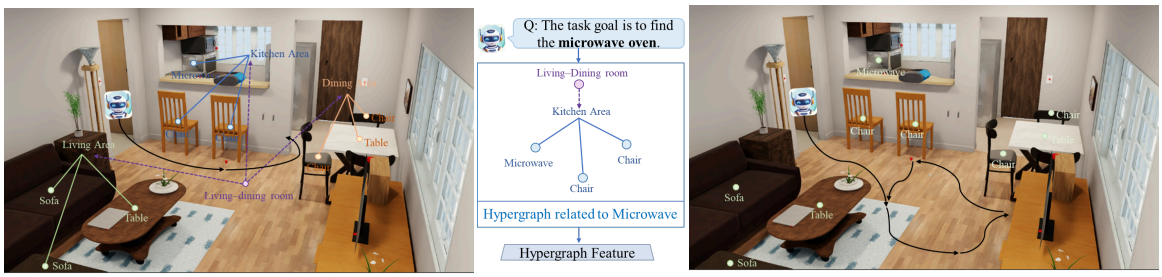


Figure 1: On the left, when the hypergraph is introduced, the deep semantic information contained in the visual data of the room is extracted, enabling the execution of the procedural planning process. In the middle, the abstraction process of building the hypergraph is shown. On the right, without the introduction of the hypergraph, the procedural planning results in a logical deadlock.

reduced to a collection of isolated objects, which can lead to suboptimal or even degenerate planning behaviors, such as repetitive or oscillatory actions in ambiguous contexts (as illustrated in Figure 1).

This phenomenon raises a critical question: Are current approaches overly reliant on the explicit spatial reasoning capabilities of VLMs, while neglecting the opportunity to mine deeper semantic structures from images?

To address this challenge, we propose GaLa, a framework that introduces a hypergraph-based representation as an intermediate semantic abstraction between vision and language. Rather than treating spatial reasoning as purely geometric, GaLa models the scene as a hypergraph that explicitly captures both object-level semantic attributes and area-level functional organization. Specifically, GaLa constructs a semantic hypergraph in which nodes correspond to individual object instances, while hyperedges represent functional or spatial regions inferred from object attributes (e.g., "dining area", "kitchen area"). By aggregating object-level semantics into structured area-level representations, the hypergraph elevates implicit spatial relations—originally latent in textual and semantic cues—into an explicit, reasoning-friendly form. To effectively integrate this structured knowledge into downstream reasoning, we further introduce a "Tri-View HyperGraph Encoder", which employs contrastive learning to enforce semantic consistency across three complementary views: object-level (node-view), region-level (area-view), and object–region association (all-view). This tri-view objective encourages the model to learn robust representations that preserve both fine-grained object semantics and higher-order relational structure, enabling more coherent and spatially grounded procedural planning. Through this design, GaLa provides VLMs with richer semantic context and mitigates common planning failures arising from ambiguous or under-structured scene representations.

Experimental validation on the ActPlan-1k benchmark dataset shows that the GaLa model achieves significant performance improvements. The results indicate that, thanks to the explicit modeling of implicit spatial relations and object cluster semantics via the hypergraph, our framework can guide the VLM to generate action sequences that are more spatially accurate and semantically consistent, demonstrating superior performance in complex procedural planning tasks.

Our key contributions are summarized as follows:

• We propose GaLa, the first framework that introduces graph-theoretic structural semantic information into a VLM-based procedural planning architecture.

• We design a HyperGraph Semantic Encoder that models visual information to construct a hypergraph enriched with semantic information, thereby enhancing the structural semantics inherent in the image.

• We introduce a Tri-View HyperGraph Encoder that employs a contrastive learning approach to ensure that the hypergraph information is fully preserved and effectively transmitted into the VLM.

2 Related Work

2.1 Vision-Language Models (VLMs)

VLMs jointly model visual and linguistic information and are widely used in multimodal tasks such as image captioning (Cheng et al., 2025), visual question answering (Bhat et al., 2025; Lim et al., 2025), and image–text retrieval (Liu et al., 2024; Chen et al., 2015; Plummer et al., 2015). Early con-

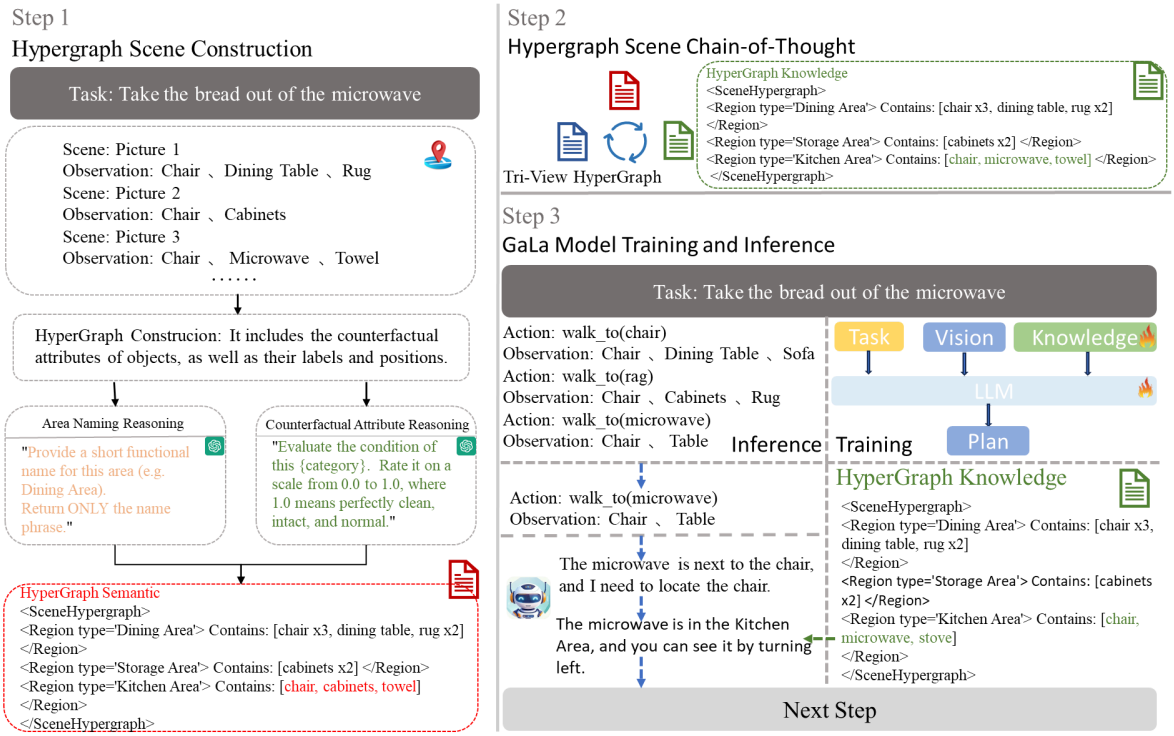


Figure 2: We present the model pipeline for GaLa. In *Step 1*, we initially build the hypergraph semantic information. In *Step 2*, we optimize the hypergraph information for CoT using the semantic information from *Step 1* and the Tri-View HyperGraph. In *Step 3*, we train the constructed knowledge, enabling the model to predict the next action more accurately.

trastive methods, including CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), established large-scale image–text alignment.

Modern VLMs integrate visual features into LLM pipelines with powerful backbones (GPT-style, LLaMA-family and Qwen-series (Yang et al., 2025a)) and visual instruction tuning. Representative methods include Flamingo (Alayrac et al., 2022), BLIP-2 (Li et al., 2023b), and LLaVA (Liu et al., 2023), using different mechanisms to inject visual information. Despite strong performance, most VLMs treat images as unstructured tokens, limiting explicit modeling of object relations and higher-order structure, motivating structured visual representations.

2.2 Procedural Planning.

Procedural planning decomposes high-level goals into executable action sequences. Early symbolic planners with manually defined state and action spaces often struggle to generalize to perceptually rich, open-world environments (Lu et al., 2022; Srivastava et al., 2014).

Recent large language models support learning-based planning from natural language, using chain-of-thought reasoning, tool invocation, and iterative refinement (Wei et al., 2022; Schick et al., 2023; Shinn et al., 2023). VLM-based planners further ground language reasoning in visual observations. Benchmarks like ActPlan-1K (Su et al., 2024) show that integrating vision improves plan feasibility and coherence.

Most methods encode visual inputs implicitly, lacking explicit modeling of object-level structure and spatial relations, motivating structured semantic abstractions for more interpretable and reliable VLM-based planning.

2.3 Hypergraph-based Contrastive Learning

Hypergraph-based contrastive learning captures higher-order relationships beyond pairwise connections. Recent work introduces a critical node-aware hypergraph contrastive learning method (Li et al., 2025) and applies similar ideas to multi-interest fairness in recommender systems (Zheng et al., 2025). These studies demonstrate that explicitly modeling high-order structure can enrich representations, which inspires our HyperGraph Semantic

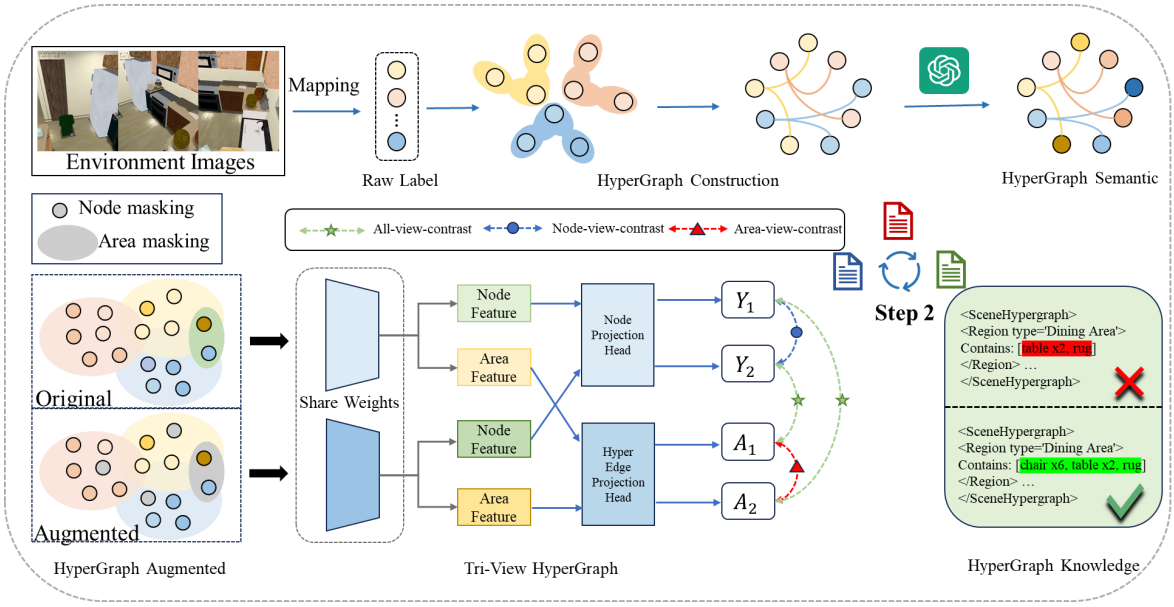


Figure 3: The detailed architecture of Step 2 is illustrated in Figure 2. The upper part depicts the process of Hypergraph Semantic Construction, while the lower-left part shows the Tri-View HyperGraph module, and the lower-right part presents the resulting HyperGraph Knowledges. The entire pipeline is organized as a chain-of-thought (CoT) process, which progressively refines and optimizes the HyperGraph Knowledges.

Encoder and Tri-View HyperGraph Encoder to capture semantic structures in images and preserve them within VLM-based procedural planning.

3 Method

3.1 Overview

Problem Definition. The workflow for performing procedural planning using a multimodal model is described as follows. We define the visual features as $I = \{i_1, i_2, ..., i_n\}$, which represent the environment perceived during the exploration process. The textual features are defined as $L = \{l_{goal}, l_1, ..., l_m\}$, where l_{goal} denotes the execution goal of the task, and the remaining elements represent semantic guidance instructions required for task execution. The generated execution objective is defined as $A = \{a_1, a_2, ..., a_t\}$, which represents the sequence of actions required to accomplish the task.

GaLa Pipeline. The overall pipeline of GaLa can be summarized into three main steps, as illustrated in Figure 2. **Step.1 : Hypergraph Semantic Construction.** Object nodes $x \in \mathbb{R}$ are first generated from visual images $v \in \mathbb{R}^{W \times H \times 3}$ obtained in a simulated environment. Each object node x contains its position $p \in \mathbb{R}^{W \times H}$, semantic label $c \in \mathbb{R}$, and a textual description of its attributes $s \in \mathbb{R}$. The object labels c are treated as graph nodes, while corresponding Area attributes $z \in \mathbb{R}$ generated by GPT from these labels are used as hyperedges C to construct the hypergraph $\mathcal{G}$. In addition, a large language model is employed to score object attributes s to determine whether they correspond to normal or counterfactual attributes $r \in [0, 1]$. This process ultimately produces the hypergraph semantic information $\mathcal{G}$. **Step.2 : Hypergraph Scene Chain-of Thought.** The overall architecture of this step is depicted in Figure 3. The hypergraph semantic information $\mathcal{G}$ generated in Step 1, together with the Tri-View Hypergraph Encoder $\mathcal{G}_t$ and Hypergraph Knowledge $\mathcal{G}_k$, forms a chain-of-thought reasoning process to progressively refine the hypergraph semantics. This step results in a comprehensive hypergraph knowledge $\mathcal{G}_k$ representation of the visual scene. **Step.3 : GaLa Training and Inference with Hypergraph Knowledge.** The hypergraph knowledge $\mathcal{G}_k$ produced in Step 2, along with visual inputs v and task information l, is fed into the VLM for training. During inference, the next action a is predicted based on the visual input, task instruction, and the hypergraph knowledge distilled from the scene. This structured knowledge enhances the textual representation by incorporating explicit spatial and semantic relationships, enabling the model to better capture the underlying scene structure. As a result, the model can generate actions with improved spatial awareness and avoid

logical inconsistencies, leading to more coherent and reasonable action sequences.

3.2 HyperGraph Semantic Encoder

The HyperGraph Semantic Encoder module is illustrated in the upper part of Figure 3. We first obtain a set of images from the simulated environment $V = \{v_1, v_2, ..., v_{|M|}\}$, where each image $v_i \in \mathbb{R}^{W \times H \times 3}$. We apply the YOLO-World model to infer a set of objects, denoted as: $\mathcal{X} = \{x_1, x_2, ..., x_{|N|}\}$. Each object instance x_i is represented by a tuple $x_i = (p_i, c_i, s_i)$, where $p_i \in \mathbb{R}^{W \times H}$ denotes the spatial position of the object, c_i it its category label and s_i is a textual description of its attributes. To capture area-level spatial structure, we perform density-based spatial clustering over object positions. Specifically, we apply a DBSCAN-based (Ester et al., 1996) clustering algorithm to the set of object coordinates $\{p_i\}_{i=1}^{|N|}$, yielding a collection of spatial clusters:

$$\mathcal{C} = D(\{p_i\}_{i=1}^{|N|}, \epsilon, MinPts), \tag{1}$$

where $\mathcal{C} = \{C_1, C_2, ..., C_K\}$, each cluster C_k corresponds to a group of spatially coherent objects and is defined as:

$$C_k = \{x_i \in \mathcal{X} | f(x_i) = k\}, \tag{2}$$

where $f : \mathcal{X} \to \{1, \ldots, K\}$ denotes the cluster assignment function. Finally, we construct a hypergraph by treating each object instance x_i as a node and each spatial cluster C_k as a hyperedge, resulting in the hypergraph $\mathcal{G}_F = (\mathcal{X}, \mathcal{C})$. To enhance the extraction of semantic and spatial structures from visual scenes, we perform area-level semantic reasoning over the constructed hypergraph. For each hyperedge C_i, we collect the category labels of its constituent object nodes:

$$L_i = \{c_j | x_j = (p_j, c_j, s_j) \in C_i\}, \tag{3}$$

where c_j denotes the catagory label of object x_j. We feed the aggregated category labels $\mathcal{L}_i$ into the large language model $\mathcal{M}_{LLM}$ (InternVL-3) with a task-specific prompt to infer an area-level semantic label: $z_i = \mathcal{M}_{LLM}(\mathcal{L}_i; \mathcal{P})$, where $\mathcal{P}$ denotes the prompt used to instruct the LLM for area semantic reasoning, and z_i represents the area-level semantic label associated with hyperedge C_i (e.g., Kitchen Area, Dining Area). **Counterfactual Attribute Scoring with LLM.** For each detected object instance x_j, we further assess whether its attribute description corresponds to a normal or counterfactual state. We leverage the large language model $\mathcal{M}_{LLM}$ (InternVL-3) with a task-specific prompt $\mathcal{P}_{cf}$ to assign a counterfactual score to each attribute: $r_j = \mathcal{M}_{LLM}(s_j; \mathcal{P}_{cf}), r_j \in [0, 1]$.

Here, r_j approaching 0 indicates that the attribute is more likely to be normal, while values closer to 1 indicate a higher degree of abnormality. Through the above process, we obtain the hypergraph semantics of the visual information.

3.3 Tri-View HyperGraph

Through the above procedure, we obtain a hypergraph $\mathcal{G}_F = (\mathcal{X}, \mathcal{C})$. The corresponding incidence matrix is defined as $\mathbf{H} \in \{0, 1\}^{|\mathcal{X}| \times |\mathcal{C}|}$. We then apply random masking to both hypergraph nodes and hyperedges. Specifically, the masked node- and hyperedge-level textual representations are given by:

$$c_i^{(k)} = Mask^{(k)}(c_i), z_j^{(k)} = Mask^{(k)}(z_i). \tag{4}$$

Based on this formulation, both the constructed hypergraph and its augmented variants can be derived as follows:

$$\begin{aligned} \mathcal{G}_F^{(1)} &= (\mathcal{X}, \mathcal{C}, \mathbf{H}^{(1)}, c_i^{(1)}, z_j^{(1)}), \\ \mathcal{G}_F^{(2)} &= (\mathcal{X}, \mathcal{C}, \mathbf{H}^{(2)}, c_i^{(2)}, z_j^{(2)}), \end{aligned} \tag{5}$$

where the augmented incidence matrices are defined as:

$$\mathbf{H}^{(k)} = \mathbf{M}^{(k)} \odot \mathbf{H}, k \in \{1, 2\}, \tag{6}$$

$\mathbf{M}^{(k)}$ denotes the all-view masking matrix. After applying data augmentation, we employ the text encoder of the InternVL model to separately encode node-level texts and hyperedge-level texts, obtaining the corresponding node and hyperedge representations.

$$\begin{aligned} P^{(k)} &= \mathcal{E}_\theta(\{c_i^{(k)}\}_{i=1}^{N}) \in \mathbb{R}^{N \times d}, \\ Q^{(k)} &= \mathcal{E}_\theta(\{z_j^{(k)}\}_{j=1}^{K}) \in \mathbb{R}^{K \times d}. \end{aligned} \tag{7}$$

To decouple semantic encoding from contrastive optimization, we employ projection heads on top of the text-encoded node and hyperedge representations. The projection heads map semantic embeddings into a contrastive space, preventing the contrastive loss from directly constraining the semantic representation space.

$$W^{(k)} = g_\phi(P^{(k)}), D^{(k)} = g_\psi(Q^{(k)}), \tag{8}$$

both g_ϕ and g_ψ are implemented as two-layer multilayer perceptrons (MLPs) with ELU activation to introduce nonlinearity.

We then adopt three contrastive learning objectives: node-level contrast, which aims to distinguish the representation of the same node across two augmented views from those of other nodes; group-level contrast, which differentiates the representations of the same hyperedge (area-level semantics) across two views from those of other hyperedges; and all-level contrast, which seeks to discriminate "real" node–hyperedge memberships from "fake" ones across different views. In this work, we employ the InfoNCE loss (Oord et al., 2018) as the contrastive learning objective.

For node-level contrast, given any node w_i, its representation in the first view, $w_i^{(1)}$, is treated as the anchor, while the corresponding representation in the second view, $w_i^{(2)}$, is regarded as the postive sample. The representations of all other nodes are considered negative samples. In this work, we adopt cosine similarity as the scoring function, defined as $s(u,v) = \frac{u^T v}{||u||||v||}$. Then the loss function for each positive node pair is defined as:

$$\mathbf{l}_n(w_i^{(1)}, w_i^{(2)}) = -\log \frac{\exp(s(w_i^{(1)}, w_i^{(2)})/\tau_n)}{\sum_{k=1}^{|N|} \exp(s(w_i^{(1)}, w_k^{(2)})/\tau_n)}, \tag{9}$$

where $w_i^{(2)}$ denotes the representations of other nodes in the second view with $k \neq i$, and τ_n is a temperature parameter. We then symmetrize this loss function and compute the average over all positive sample pairs, as shown below:

$$\mathcal{L}_n = \frac{1}{2|N|} \sum_{i=1}^{|N|} (\mathbf{l}_n(w_i^{(1)}, w_i^{(2)}) + \mathbf{l}_n(w_i^{(2)}, w_i^{(1)})). \tag{10}$$

For area-level contrast, given any hyperedge z_j, the representation $d_j^{(1)}$ from the first view is treated as the anchor, while the corresponding representation $d_j^{(2)}$ from the other view is regarded as the positive sample. The representations of all remaining hyperedges are considered negative samples. The loss for each positive hyperedge pair is defined as follows:

$$\mathbf{l}_g(d_j^{(1)}, d_j^{(2)}) = -\log \frac{\exp(s(d_j^{(1)}, d_j^{(2)})/\tau_g)}{\sum_{k=1}^{|K|} \exp(s(d_j^{(1)}, d_j^{(2)})/\tau_g)}, \tag{11}$$

where $d_j^{(2)}$ denotes the representations of other hyperedges in the second view with $k \neq j$, and τ_g is a temperature parameter. We then symmetrize this loss function and compute the average over all positive sample pairs, as shown below:

$$\mathcal{L}_g = \frac{1}{2|K|} \sum_{j=1}^{|K|} (\mathbf{l}_g(d_j^{(1)}, d_j^{(2)}) + \mathbf{l}_g(d_j^{(2)}, d_j^{(1)})). \tag{12}$$

For all-level contrast, the node representation $w_i^{(1)}$ from the first view is treated as the anchor, while the hyperedge representation $d_j^{(2)}$ from the other view that has a membership relation with the node is regarded as the positive sample. Negative samples are drawn from the representations of other hyperedges that are not associated with the node. To distinguish "real" node–hyperedge memberships from "fake" ones, we introduce a discriminator $\mathcal{D} : \mathbb{R}^{F'} \times \mathbb{R}^{F''} \to \mathbb{R}$, which outputs a probability score for a given node–hyperedge representation pair. We optimize the following objective function:

$$\mathbf{l}_m = -\log \frac{\exp(D(w_i, d_j/\tau_m)}{\exp(D(w_i, d_j/\tau_m) + \sum_{j' \in \mathcal{N}_i^-} \exp(D(w_i^{(1)}, d_{j'})/\tau_m)}, \tag{13}$$

where the negative sample set $\mathcal{N}_i^-$ is defined as: $\mathcal{N}_i^- = \{j' | h_{ij'=0}\}$, where τ_m is a temperature parameter. For each node–hyperedge membership, two node–hyperedge representation pairs can be obtained from the two symmetric views. The final objective function for all-level contrast is defined as follows:

$$\mathcal{L}_m = \frac{1}{2|K|} \sum_{i=1}^{|N|} \sum_{j=1}^{|K|} \mathbb{I}[h_{ij} = 1](\mathbf{l}_m(w_i^{(1)}, d_j^{(2)}) + \mathbf{l}_m(w_i^{(2)}, d_j^{(1)})). \tag{14}$$

Finally, contrastive loss is formulated as:

$$\mathcal{L} = \mathcal{L}_n + \alpha_g \mathcal{L}_g + \alpha_m \mathcal{L}_m, \tag{15}$$

where α_g and α_m are the weights of $\mathcal{L}_g$ and $\mathcal{L}_m$, respectively. In all experiments, we simply set $\alpha_g = \alpha_m = 1$.

Based on this strategy, we further refine the hypergraph construction process, resulting in semantic hypergraph knowledge that serves as structured input for the VLM.

4 Experiments

4.1 Experimental Setup

Dataset. To evaluate the model performance, we conducted experiments on two benchmarks: ActPlan-1K (Su et al., 2024) and ALFRED (Shridhar et al., 2020). ActPlan-1K is currently the only

Method	Type	Size	ActPlan-1K(ctrf.)			ActPlan-1K(norm.)			ALFRED		
			Exec.↑	LCS↑	Corr.↑	Exec.↑	LCS↑	Corr.↑	Exec.↑	LCS↑	Corr.↑
GPT-4o	Close-set	-	49.2	0.48	21.4	56.5	0.59	39.8	**90.4**	0.60	47.4
Gemini-Pro-1.5		-	46.0	0.51	25.8	51.0	0.55	38.6	84.0	0.60	46.6
LLaVa-OV	Open-set	7B	37.3	0.49	24.6	50.6	0.55	35.5	78.2	0.42	29.8
VideoLLaMA2		7B	30.2	0.43	20.3	40.6	0.48	31.9	73.1	0.57	43.9
DeepSeek-VL2		4.5B	34.9	0.44	23.8	38.2	0.50	24.4	72.1	0.54	32.1
Qwen2-VL		7B	43.7	0.53	27.9	57.8	0.59	37.9	81.6	0.56	43.8
InternVL2		8B	44.9	0.52	26.1	52.8	0.57	38.2	81.9	0.58	45.3
InternVL2 (Pla)		8B	48.2	0.53	30.2	53.9	0.58	39.0	81.8	0.58	45.7
InternVL3		8B	45.2	0.53	28.2	52.9	0.57	38.4	81.9	0.58	45.5
InternVL3.5		8B	45.1	0.52	27.8	53.1	0.58	38.5	82.0	0.59	45.6
Embodied-GPT	Specialized	7B	39.8	0.48	21.5	54.3	0.57	36.6	70.7	0.56	41.6
LLaPa		8B	53.2	0.57	36.1	62.9	0.62	45.2	85.3	0.62	48.6
GaLa(**ours**)		8B	**55.1**	**0.59**	**37.2**	**65.8**	**0.66**	**47.5**	89.9	**0.67**	**50.1**

Table 1: The table reports a quantitative comparison of different models on the ActPlan-1K dataset (including both counterfactual and normal activities) and the ALFRED dataset.

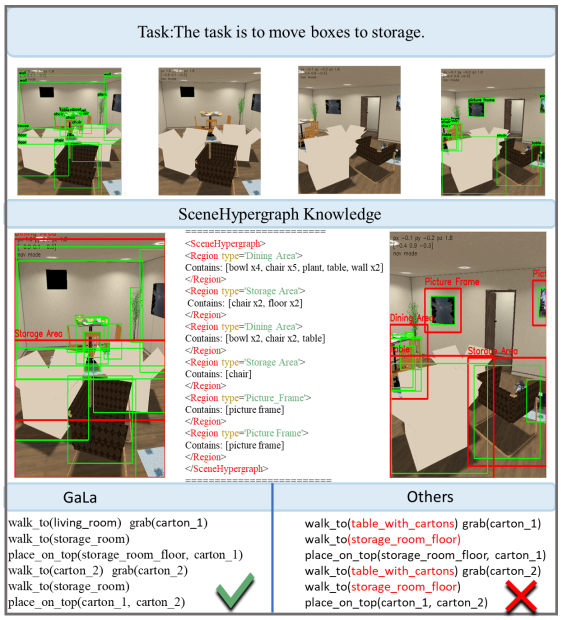


Figure 4: This figure illustrates the instruction decomposition process of GaLa.

multimodal planning dataset that supports counterfactual reasoning, used to examine the model's adaptive planning capability under unconventional conditions. ALFRED is employed to test the model's zero-shot generalization performance in embodied tasks. More detailed information about the dataset can be found in the Appendix B.1.

Baselines. We compare the GaLa model with a wide range of state-of-the-art baseline methods, which are divided into three categories. The first category consists of close-set VLM-based model methods.: GPT-4o (Hurst et al., 2024); Gemini-Pro-1.5 (Team et al., 2023). The second category consists of open-set VLM-based model methods: LLaVa-OV (Li et al., 2024a); VideoLLaMA2 (Cheng et al., 2024b); DeepSeek-VL2 (Wu et al., 2024); Qwen2-VL (Wang et al., 2024); InternVL2 (Chen et al., 2024); InternVLwithPlasma's (Brahman et al., 2024); InternVL3 (Zhu et al., 2025); InternVL3.5 (Wang et al., 2025b). The third category consists of specialized procedural planning model methods: Embodied-GPT (Mu et al., 2023); LLaPa (Sun et al., 2025). We uniformly set the maximum generation length to 512 tokens.

Implementation Details. Our model GaLa uses InternVL3-8B as its backbone, where the visual encoder adopted the InternViT architecture, and the language model was based on InternVL3. Our model is trained for 30 epochs on benchmark hardware (Intel Xeon 8558@4.00GHz, NVIDIA HGX H200 141G, Ubuntu22.04). Training is conducted using the Adam optimizer with an initial learning rate of 2e-5, no weight decay, and a batch size of 128. More detailed information about the implementation details can be found in Appendix B.2.

Method	ActPlan-1K(ctrf.)			ActPlan-1K(norm.)		
	Exec.↑	LCS↑	Corr.↑	Exec.↑	LCS↑	Corr.↑
GaLa	**55.1**	**0.59**	**37.2**	**65.8**	**0.66**	**47.5**
w/o Hyper	51.7	0.56	32.2	61.5	0.62	44.3
w/o $H_{\mathcal{L}_n}$	52.1	0.56	33.6	62.3	0.63	45.1
w/o $H_{\mathcal{L}_g}$	52.7	0.57	34.0	63.2	0.64	45.6
w/o $H_{\mathcal{L}_m}$	52.4	0.57	33.9	63.5	0.64	46.1

Table 2: Ablation Results of GaLa on ActPlan-1K.

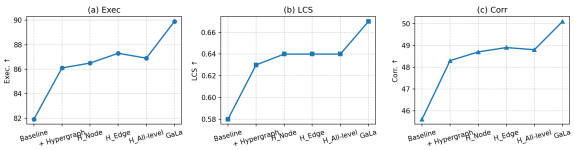


Figure 5: Ablation studies of GaLa on the ALFRED dataset in terms of Exec., LCS, and Corr. metrics.

Evaluation Metrics. To comprehensively evaluate the quality of the generated action sequences, this study adopts a multidimensional evaluation framework that integrates three key aspects: the executability of the plan, its sequential similarity to reference plans, and the task correctness. This aligns with common practices in the field (Puig et al., 2018; Song et al., 2023; Brahman et al., 2024). Specifically: (i) Executability quantifies the proportion of generated actions that can be successfully executed in the simulated environment; (ii) Sequential similarity is measured by calculating the Longest Common Subsequence (LCS) between the generated plan and human-annotated references; (iii) Task correctness assesses whether the action sequence ultimately accomplishes the predefined task objective.

4.2 Overall Performance

Quantitative Results. As shown in Table 1, GaLa is trained on the ActPlan-1K and ALFRED datasets and evaluated using three metrics. On the ActPlan-1K dataset, for counterfactual activities, GaLa achieves 55.1% Executability, 0.59 LCS, and 37.2% Correctness. For normal activities, it attains 65.8% Executability, 0.66 LCS, and 47.5% Correctness. We observe that the performance gains on normal activities are larger than those on counterfactual activities, indicating that counterfactual tasks are inherently more challenging. Compared with its baselines, GaLa achieves substantial improvements across all evaluation metrics. On the ALFRED dataset, GaLa reaches 89.9% Executability, 0.67 LCS, and 50.1% Correctness.

These results validate that explicitly modeling implicit spatial relations and object-cluster semantics through hypergraphs can effectively guide VLMs to generate more spatially aware and accurate action sequence

Qualitative Results. Figure 4 and 7 present a case study of the GaLa model, illustrating how it constructs hypergraph-based knowledge from images and ultimately generates a correct action sequence. The figures also compares GaLa with other models that do not incorporate hypergraph construction, highlighting the differences in the generated action sequences.

4.3 Ablation Study.

To systematically validate the design choices of GaLa, we conduct a series of ablation studies to evaluate the effectiveness of the hypergraph information and the Tri-View Hypergraph mechanism.

As shown in Table 2, we analyze the impact of incorporating hypergraph construction information (w/o Hyper) into this domain on procedural planning performance. We also conduct ablation studies to examine the effects of introducing hypergraph-based contrastive learning on nodes ($H_{\mathcal{L}_n}$), edges ($H_{\mathcal{L}_g}$), and both nodes and edges ($H_{\mathcal{L}_m}$), respectively.

Through ablation studies on the ActPlan-1K dataset, we find that introducing the hyperedge-based graph construction strategy (w/o Hyper) into the procedural planning framework effectively models implicit relations among objects and enables the extraction of object-cluster attributes, thereby substantially improving the performance of VLM-based models. Under counterfactual activity settings, the Exec. score increases from 45.2% to 51.7%, the LCS score improves from 0.53 to 0.56, and the Corr. score rises from 28.2% to 34.2%. For normal activities, similar gains are observed, with the Exec. score improving from 52.9% to 61.5%, the LCS score increasing from 0.57 to 0.62, and the Corr. score rising from 38.4% to 44.3%.

Moreover, building upon the constructed hypergraph, the incorporation of the Tri-view strategy consistently yields further improvements across all three evaluation metrics, demonstrating its effectiveness in enhancing structured semantic representations.

As shown in Figure 5, we conduct corresponding ablation experiments on the ALFRED dataset.

The results demonstrate that incorporating the hypergraph-based strategy effectively improves the model's performance across all evaluation metrics. Furthermore, introducing the three contrastive learning objectives leads to additional and consistent performance gains.

5 Conclusion

We presented GaLa, a Graph-augmented vision–language framework for multimodal procedural planning in embodied AI. By introducing a hypergraph-based intermediate representation, GaLa explicitly models implicit spatial relations and region-level semantic structures that are often overlooked by existing VLM-based approaches. Furthermore, the proposed Tri-View HyperGraph Encoder enforces semantic consistency across node-level, area-level, and all-level views via contrastive learning, enabling effective injection of structured scene knowledge into downstream reasoning. Experimental results on ActPlan-1K and ALFRED demonstrate that GaLa significantly improves execution success, plan consistency, and correctness, validating the effectiveness of explicitly modeling implicit semantic structures for robust procedural planning.

6 Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62277011), Project of Chongqing MEITC(Grant No. YJX-2025001001009) , Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) (Grant No.GML-KF-24-18) and CAAI-CANN Open Fund, developed on OpenI Community.

7 Limitations

Although GaLa achieves substantial performance improvements over InternVL3 on both the ActPlan-1K and ALFRED benchmarks, it still has several limitations. First, GaLa improves model performance by constructing and refining hypergraphs from visual inputs; however, since the ActPlan-1K dataset provides only a limited set of images associated with each action, the constructed hypergraphs may be incomplete, potentially leading to the loss of important semantic information. Second, while GaLa represents an initial attempt to introduce hypergraph-based modeling into the procedural planning domain and demonstrates consistent gains over baseline methods, it does not fully explore the potential of more advanced graph-theoretic techniques that could further enhance model performance.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, and Malcolm Reynolds. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Nagaraj N Bhat, Joydeb Mondal, and Srijon Sarkar. 2025. Expertneurons at scivqa-2025: Retrieval augmented vqa with vision language model (ravqa-vlm). In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 221–229.

Faeze Brahman, Chandra Bhagavatula, Valentina Pyatkin, Jena D Hwang, Xiang Lorraine Li, Hirona Jacqueline Arai, Soumya Sanyal, Keisuke Sakaguchi, Xiang Ren, and Yejin Choi. 2024. Plasma: Procedural knowledge models for language-based planning and re-planning. In *The Twelfth International Conference on Learning Representations*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, and Zheng Ma. 2024. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101.

Kanzhi Cheng, Wenpo Song, Jiaxin Fan, Zheng Ma, Qiushi Sun, Fangzhi Xu, Chenyang Yan, Nuo Chen, Jianbing Zhang, and Jiajun Chen. 2025. CapArena: Benchmarking and analyzing detailed image captioning in the LLM era. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14077–14094, Vienna, Austria. Association for Computational Linguistics.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. 2024a. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16901–16911.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, and Deli Zhao. 2024b. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. Density-based spatial clustering of applications with noise. In *Int. Conf. knowledge discovery and data mining*, volume 240.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Leveraging pretrained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36:79081–79094.

Kung-Hsiang Huang, Akshara Prabhakar, Sidharth Dhawan, Yixin Mao, Huan Wang, Silvio Savarese, Caiming Xiong, Philippe Laban, and Chien-Sheng Wu. 2025. Crmarena: Understanding the capacity of llm agents to perform professional crm tasks in realistic environments. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3830–3850.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, and Alec Radford. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, and Ziwei Liu. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, and Jiankai Sun. 2023a. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, and Ruohan Zhang. 2024b. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534.

Zhuo Li, Yuena Lin, Yipeng Wang, Wenmao Liu, Mingliang Yu, Zhen Yang, and Gengyu Lyu. 2025. Critical node-aware augmentation for hypergraph contrastive learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 5671–5679.

Hyeonseok Lim, Dongjae Shin, Seohyun Song, Inho Won, Minjun Kim, Junghun Yuk, Haneol Jang, and KyungTae Lim. 2025. Vlr-bench: Multilingual benchmark dataset for vision-language retrieval augmented generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6150–6168.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. 2025. Showui: One vision-language-action model for gui visual agent. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19498–19508.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, and Ziwei Liu. 2024. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Yujie Lu, Weixi Feng, Wanrong Zhu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. 2022. Neuro-symbolic procedural planning with commonsense prompting. *arXiv preprint arXiv:2206.02928*.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. 2023. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36:25081–25094.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark.

2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

Tom Silver, Soham Dan, Kavitha Srinivas, Joshua B Tenenbaum, Leslie Kaelbling, and Michael Katz. 2024. Generalized planning in pddl domains with pretrained large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 20256–20264.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2998–3009.

Siddharth Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stuart Russell, and Pieter Abbeel. 2014. Combined task and motion planning through an extensible planner-independent interface layer. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 639–646. IEEE.

Ying Su, Zhan Ling, Haochen Shi, Cheng Jiayang, Yauwai Yim, and Yangqiu Song. 2024. Actplan-1k: Benchmarking the procedural planning ability of visual language models in household activities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14953–14965.

Shibo Sun, Xue Li, Donglin Di, Mingjie Wei, Lanshun Nie, Wei-Nan Zhang, Dechen Zhan, Yang Song, and Lei Fan. 2025. Llapa: A vision-language model framework for counterfactual-aware procedural planning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 5020–5029.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and Katie Millican. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Zishen Wan, Yuhang Du, Mohamed Ibrahim, Jiayi Qian, Jason Jabbour, Yang Zhao, Tushar Krishna, Arijit Raychowdhury, and Vijay Janapa Reddi. 2025. Reca: Integrated acceleration for real-time and efficient cooperative embodied autonomous agents. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 982–997.

Dong Wang, Xinghang Li, Zhengshen Zhang, Jirong Liu, Xiao Ma, Hanbo Zhang, Tao Kong, and Huaping Liu. 2025a. Procworld: Benchmarking large model planning in reachability-constrained environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12575–12605.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, and Wenbin Ge. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, and Jie Shao. 2025b. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, and Bingxuan Wang. 2024. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*.

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

Fuxiang Yang, Donglin Di, Lulu Tang, Xuancheng Zhang, Lei Fan, Hao Li, Chen Wei, Tonghua Su, and Baorui Ma. 2026. Chain of world: World model thinking in latent motion. *arXiv preprint arXiv:2603.03195*.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, and Manling Li. 2025b. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*.

Yijun Yang, Tianyi Zhou, Kanxue Li, Dapeng Tao, Lusong Li, Li Shen, Xiaodong He, Jing Jiang, and Yuhui Shi. 2024. Embodied multi-modal agent trained by an llm from a parallel textworld. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 26275–26285.

Simon Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Peter Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, and Yi Ma. 2024. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37:110935–110971.

Zhonghan Zhao, Wenhao Chai, Xuan Wang, Boyi Li, Shengyu Hao, Shidong Cao, Tian Ye, and Gaoang Wang. 2024. See and think: Embodied agent in virtual environment. In *European Conference on Computer Vision*, pages 187–204. Springer.

Yongsen Zheng, Zongxuan Xie, Guohua Wang, Ziyao Liu, Liang Lin, and Kwok-Yan Lam. 2025. Why multi-interest fairness matters: Hypergraph contrastive multi-interest learning for fair conversational recommender system. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25673–25684, Vienna, Austria. Association for Computational Linguistics.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, and Jie Shao. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Appendix: Method

A.1 Prompt

As shown in Figure 6, we first apply YOLO-World (Cheng et al., 2024a) to detect objects in the image and extract their attributes, based on which a hypergraph is constructed. We then feed each object cluster together with a predefined prompt into a large language model to generate semantically meaningful area-level attributes, enabling the model to further uncover hierarchical region-level relationships within the image.

For the ActPlan-1K dataset, we additionally design a specific prompt to send individual object attributes to the language model, which outputs a scalar value in the range [0,1] to indicate the degree of normality of each attribute.

B Appendix: Experiments

B.1 Dataset

The main components of the ActPlan-1K dataset are located in the annotation directory. This directory contains multiple scenes, and each scene includes several tasks. For each task, there are both counterfactual activities and normal activities. Each activity consists of multiple samples that share the same task description. Each sample contains several scene-frame images and two gold files with different styles. The ActPlan-1K dataset includes 703 training samples, 243 test samples, and 243 validation samples.

In our experiments, we use the Lite version of the ALFRED dataset, which includes task descriptions and scene information. The dataset also provides high-level actions in PDDL format. By running the provided scripts, image frames corresponding to the task execution process can be generated. To align with our model, we manually select a small number of key scene images based on the actions in the dataset, allowing most viewpoints of the scene to be covered with fewer images. The ALFRED dataset contains 2,435 training samples, 483 test samples, and 242 validation samples.

B.2 Implementation Details.

The GaLa model is trained for 30 epochs on 8 NVIDIA HGX H200 (141 GB) GPUs. Training is conducted using the Adam optimizer with an initial learning rate of 2e-5, no weight decay, and a batch size of 128. The original input images have

Area Attribute Construction for Hypergraph Object Clusters:

```
question = (
        f"{task_ctx} Carefully analyze the image.
        List the visible objects, furniture, and tools. "
        "Be precise. Output ONLY a comma-separated
        list of names." )

question = (
        f"{task_ctx} This specific image crop contains:
        {objs_str}. "
        "Provide a short, functional name for this
        specific area/zone. "
        "Return ONLY the name phrase, nothing else."
    )
```

Scoring counterfactual and normal attributes of objects:

```
question = (
        f"Evaluate the condition of this {category}. "
        "Rate it on a scale from 0.0 to 1.0, where 1.0
         means perfectly clean, intact, and normal, "
        "and 0.0 means very dirty, broken, or abnormal. "
        "Return ONLY the numeric value."
    )
```

Figure 6: This figure illustrates the instruction decomposition process of GaLa.

a resolution of 600 × 600, which is forcibly resized to 448 × 448 before being fed into the model.

The TriCL module consists of 2 layers with a projection dimension of 512. The InfoNCE temperature is set to 0.07, and the TriCL loss weight is 0.5. The visual encoder follows the InternViT architecture, with a ViT patch size of 14, while the text encoder is Qwen3-8B.

B.3 Case Study

As shown in Figure 7, this figure illustrates the overall hypergraph construction process of the GaLa model as well as typical errors that may occur in other models. The first column presents different types of errors commonly made by existing approaches, including mistakes of object location, mistakes of event cause, mistakes of object number, and mistakes of relational blindness. The second column contains the corresponding instructional task descriptions. The third column shows the input images, while the fourth column visualizes the constructed hypergraphs. The fifth column provides the XML-formatted representation of the constructed hypergraph. The last two columns compare the action predictions generated by GaLa and by other baseline models.

From these examples, we observe that constructing a hypergraph not only improves the overall performance of the model, but also enables more effective mining of implicit spatial relations and deep semantic information from multimodal inputs. As a result, GaLa can alleviate logical deadlocks in action sequence prediction—particularly those caused by relational blindness—to a considerable extent.

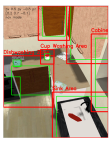
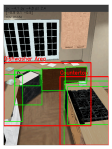
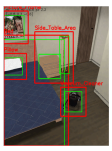
Error Type	Task	Image Input	Hypergragh Image	Hypergragh XML	Gala	Others
mistake of object's location	The task is to move boxes to storage.			========== <SceneHypergraph> <Region type="Dining Area"> Contains: [bowl x4, chair x5, plant, table, wall x2] </Region> <Region type="Storage Area"> Contains: [chair x2, floor x2] </Region> <Region type="Dining Area"> Contains: [bowl x2, chair x2, table] </Region> <Region type="Picture_Frame"> Contains: [picture frame] </Region> </SceneHypergraph> ==========	walk_to(carton_1) grab(living_room) walk_to(storage_room) place_on_top(storage_room_floor, carton_1) walk_to(carton_2) grab(carton_2) walk_to(storage_room) place_on_top(carton_1, carton_2)	walk_to(table_with_cartons) grab(carton_1) walk_to(storage_room_floor) place_on_top(storage_room_floor, carton_1) walk_to(table_with_cartons) grab(carton_2) walk_to(storage_room_floor) place_on_top(carton_1, carton_2)
mistake of event cause/result	The task is to wash dishes in the kitchen.			========== <SceneHypergraph> <Region type="Sink Area"> Contains: [knife, cutting board, dishcloth x3, cup, faucet, countertop, sink, cabinets x2] </Region> <Region type="Dishwashing Zone"> Contains: [cabinet, bowl, oven] </Region> <Region type="Cabinet"> Contains: [cabinet] </Region> <Region type="Cup Washing Area"> Contains: [cup] </Region> <Region type="Cabinet"> Contains: [cabinet x2] </Region> </SceneHypergraph> ==========	... walk_to(countertop) grab(bowl) walk_to(kitchen_sink) open(kitchen_sink) switch-on(faucet) place_inside(bowl, kitchen_sink) grab(scrub_brush) place_inside(bowl, scrub_brush) switch-off(faucet) walk_to(countertop) place_on_top(countertop, bowl) ...	... walk_to(bowl) place_on_top(bowl, scrub_brush) grab(bowl) switch_on(faucet) place_inside(sink, bowl) grab(bowl) walk_to(cup) place_on_top(cup, scrub_brush) grab(cup) switch_on(faucet) place_inside(sink, cup) grab(cup) ...
mistake of object's number	The task is to wash dishes in the kitchen.			========== <SceneHypergraph> <Region type="Countertop"> Contains: [countertop, stove, plates x4] </Region> <Region type="Dishwasher Area"> Contains: [sink x2, cabinets x3] </Region> <Region type="Cabinet"> Contains: [cabinets] </Region> </SceneHypergraph> ==========	... place_inside(cabinet, plate_1_from_countertop_1) grab(plate_2_from_countertop_1) place_inside(cabinet, plate_2_from_countertop_1) grab(plate_3_from_countertop_1) place_inside(cabinet, plate_3_from_countertop_1) grab(plate_4_from_countertop_1) place_inside(cabinet, plate_4_from_countertop_1) close(cabinet) ...	... grab(plate1) walk_to(cabinet) open(cabinet) place_inside(cabinet, plate1) walk_to(countertop2) grab(plate2) walk_to(cabinet) place_inside(cabinet, plate2) walk_to(countertop3) ...
mistake of relational blindness	The task is to vacuum the floors in the bedroom.			========== <SceneHypergraph> <Region type="Hallway Storage Area"> Contains: [trash can, box x2, dresser] </Region> <Region type="Lamp"> Contains: [lamp] </Region> <Region type="Bed"> Contains: [bed] </Region> <Region type="Pillow"> Contains: [pillow] </Region> <Region type="Side_Table_Area"> Contains: [nightstand, stool x2] </Region> <Region type="Picture_Frame"> Contains: [picture frame] </Region> <Region type="Vacuum_Cleaner"> Contains: [vacuum] </Region> </SceneHypergraph> ==========	... walk_to(vacuum) grab(vacuum) switch_on(vacuum) walk_to(bedroom_floor_start) walk_to(bedroom_floor_end) switch_off(vacuum) ...	... walk_to(vacuum) grab(vacuum) walk_to(bedroom_floor) switch_on(vacuum) walk_to(ashcan) grab(ashcan) place_on_top(floor, ashcan) walk_to(vacuum) switch_off(vacuum) grab(ashcan) place_on_top(floor, ashcan) ...

Figure 7: A comparison of action instruction predictions between the GaLa model and other models.