arXiv:2602.02456v1 [cs.RO] 2 Feb 2026

面向任务推理的关系感知分层三维场景图

Albert Gassol Puigjaner、Angelos Zacharia、Kostas Alexis

摘要—以结构化方式表示和理解三维环境，对于自主智能体在周围环境中导航并进行推理至关重要。尽管传统的同时定位与建图（SLAM）方法能够生成度量重建，并可扩展为度量-语义建图，但它们缺乏更高层次的抽象和关系推理。为弥补这一不足，三维场景图应运而生，成为捕捉层次结构和对象关系的强大表示形式。在这项工作中，我们提出了一种增强型分层三维场景图，在多个抽象层级上融合开放词汇特征，并支持对象关系推理。我们的方法利用视觉语言模型（VLM）推断语义关系。值得注意的是，我们引入了一个任务推理模块，将大语言模型（LLM）与视觉语言模型（VLM）相结合，以解释场景图的语义和关系信息，使智能体能够更智能地对任务进行推理并与其环境交互。我们在多个环境和任务中将该方法部署于一台四足机器人上进行验证，展现了其对相关任务进行推理的能力。

I. 引言

机器人空间感知面临的一项核心挑战，是构建兼具结构性与语义意义的三维表示。人类能够自然地通过识别物体、其属性及相互关系来感知和操纵场景，其中包括房间位于楼层或建筑物内等层次结构。对于自主智能体而言，这需要可扩展的在线表示，以支持多个层次的抽象，并对对象关系进行推理。

传统的同时定位与建图（SLAM）方法利用相机 [1], 激光雷达 [2], 雷达 [3], 或惯性测量单元（IMU） [4], 等传感器重建度量地图 [5] ，并可借助视觉基础模型 [7], [8]扩展为封闭词汇表 [6] 或开放词汇表的度量语义地图。然而，这些方法缺乏更高层次的抽象和对象推理。

三维场景图 [5], [9], [10], [11], [12], [13] 通过捕捉层次信息和关系信息来弥补这一不足。层次模型 [9], [10], [11]表示室内多个抽象层级的场景（例如，物体、房间、建筑物），同时还编码层间和层内的几何关系（例如，物体位于房间内部）。其他研究 [12], [13]从 RGB点云生成对象级场景图，预测几何、比较和语义关系；近期方法 [14] 则利用视觉语言模型（VLM）引入开放词汇的对象关系。更近期的增量式——

Autonomous RobotsLab，挪威科技大学（NTNU），挪威特隆赫姆，albert.g.puigjaner@ntnu.no本研究得到了欧盟委员会地平线欧洲计划 SYNERGISE 项目（EC 101121321）的资助。

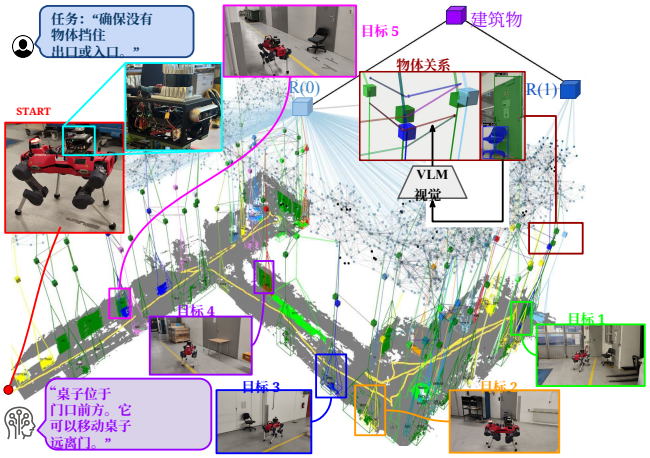


图 1：任务推理示例。我们在四足机器人上部署 REASONINGGRAPH，该机器人在自主探索过程中逐步构建环境的开放词汇、关系感知分层场景图。借助开放词汇嵌入和物体关系嵌入，REASONINGGRAPH 能够识别与任务相关的物体，并推理它们之间的交互。在此示例中，它识别出所有阻挡出口的物体（椅子、一张桌子和一个垃圾桶）。

方法 [15], [16] 构建开放词汇场景图，但不包含显式的物体级关系。

在这项工作中，我们提出 REASONINGGRAPH，这是一种用于增量式构建推理增强型分层三维场景图的框架，可整合多个抽象层级的开放词汇特征，并支持对象关系推理。此外，我们还引入了一个任务推理模块。对于可能需要物体交互推理的任务（例如，“为会议准备房间”、“确保出口未被堵塞”），该模块会利用图中的语义和关系信息，将任务分解为子任务、识别相关物体，并评估哪些子任务需要执行。我们的贡献如下：

- 我们为分层三维场景图扩展了开放词汇特征，使其覆盖多个抽象层级。
- 我们利用视觉语言模型推断物体关系特征，从而获得更丰富的上下文感知表征。
- 我们提出了一种推理模块，结合大语言模型（LLM）和视觉语言模型（VLM）来处理自然语言任务、预测相关物体，并评估物体交互的可行性。
- 我们定量评估了 REASONINGGRAPH 对开放词汇物体进行编码的能力，结果表明其性能可与强基线方法相媲美。
- 我们展示了将物体关系与我们的推理模块相结合，对于推理复杂任务的益处。此外，我们在一台四足机器人上部署了我们的方法，展示了其在线构建场景图并对任务进行推理的能力。

本文其余部分的安排如下：首先回顾相关文献（第二节），然后定义所要解决的问题（第三节）。接着详细介绍所提出的方法（第四节），并对其性能进行评估（第五节）。最后在第六节给出结论。

II. 相关工作

度量语义表示。封闭词汇的语义同步定位与建图方法旨在构建带有语义标注的环境三维地图。这些方法通常依赖语义与全景分割网络 [17],[18] 来增强三维表示 [5], [19]。随着CLIP [7]和SAM [8], 等视觉基础模型的引入，近期工作聚焦于构建富含开放词汇特征的三维表示，这些特征易于查询和/或聚类 [6]。这些方法使用视觉基础模型从二维图像中提取开放词汇嵌入，并将其投影到三维空间。然而，由于它们在点级分配特征，因此需要大量内存，且无法高效扩展。此外，这些方法缺乏抽象、层次结构和对象级推理，限制了其支持更高层次场景理解的能力。

三维场景图。早期工作 [5], [9]引入三维场景图来建模室内多层级抽象，从而能够跨越智能体姿态、物体、房间和建筑物进行空间推理。这些方法还建立了层间与层内关系，例如对象包含关系和空间邻近性，以提供更丰富的场景结构理解。后续工作 [10], [11] 采用了这一层次化框架，并将其扩展为实时增量构建三维场景图。与此同时，已有研究提出对象级场景图 [12], [13] ，用于从RGB点云推断几何关系、比较关系和语义关系，从而进一步提升上下文推理。最近，视觉语言模型被用于引入开放词汇关系，使场景图中的语义更加灵活、可适应。Koch等人 [14] 提出将InstructBLIP [20] 视觉编码器的知识蒸馏到图卷积神经网络中，而Chen 等人 [21] 则采用类似方法并结合CLIP [7]。然而，这些方法通常离线构建对象级场景图，需要场景的完整点云。尽管取得了这些进展，现有方法仍存在若干局限。具体而言，它们要么（a）缺乏开放词汇与关系推理，（b）未能纳入层次化表示，要么（c）不适用于在线场景图构建。

开放词汇三维场景图。近期研究 [15],[16]探索了开放词汇场景图，以实现语言引导导航。ConceptGraphs [15] 通过聚类三维投影的 CLIP嵌入，逐步构建物体级开放词汇三维场景图。它还通过向 GPT-4 提供物体的几何信息和汇总的图像描述来预测物体关系。此外，它还通过向 GPT-4 查询物体的几何数据，实现语言引导的物体搜索和描述。不过，这种方法仅适用于小规模场景，且其表示缺少层次结构。

在类似思想的基础上，HOV-SG [16] 引入了一种分层开放词汇三维场景图，其组织结构包括建筑物、楼层、房间、物体和导航图。除了将 CLIP 嵌入附加到检测到的物体上（其中，物体嵌入通过深度图像投影到三维空间，并在多个视角间取平均），HOV-SG 还将开放词汇特征扩展到了其层级结构中的楼层和房间。在物体搜索过程中，GPT-4 将自然语言查询解析为结构化属性：查询中提及的房间、楼层和物体。随后，通过计算这些属性与开放词汇嵌入之间的余弦相似度，在层级结构中对其进行匹配。尽管该方法整合了层级表示和开放词汇表示，但它缺乏关系推理；通过捕捉物体之间的相关交互，这种推理本可以进一步提升语言落地的任务推理能力。

III. 问题陈述

分层三维场景图 $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$由 N 层级组成，其定义包括一组节点$\mathcal{N} = \{\mathcal{N}_{L_n}\}_{n=1}^{N}$和一组边 $\mathcal{E}$ 。节点集合包含各层中的所有节点，边集合表示节点之间的几何或语义关系。层级按照从底到顶的顺序组织，每一层依次表示抽象程度更高的实体。每个节点都具有几何属性，例如位置或质心，并且可能包含语义属性和/或开放词汇属性。每一层 L_n都包含层内边 $\mathcal{E}_{L_n}^{L_n}$，用于编码关系信息。此外，层间边 $\mathcal{E}_{L_n}^{L_{n+1}}$ 连接不同层之间的节点。形式化地，图的节点集合和边集合定义如下：

$$\mathcal{N} = \bigcup_{n=1}^{N} \mathcal{N}_{L_n}, \tag{1}$$

$$\mathcal{E} = \left(\bigcup_{n=1}^{N-1} \left(\mathcal{E}_{L_n}^{L_n} \cup \mathcal{E}_{L_n}^{L_{n+1}} \right) \right) \cup \mathcal{E}_{L_N}^{L_N}. \tag{2}$$

目标*1*：给定一系列室内场景的 $M \in \mathbb{N}$ RGB-D帧 $\mathcal{I} = \{I_i\}_{i=1}^{M}$，其中每帧 $I_i = \{I_i^{\text{RGB}}, I_i^{\text{Depth}}\}$，以及对应的里程计估计 $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{M}$ ，目标是增量式地构建 $\mathcal{G}$。此外，我们旨在逐步为 $\mathcal{G}$ 补充开放词汇语义和关系信息。

目标*2*：给定场景图 $\mathcal{G}$，目标是利用其开放词汇和关系表示，对涉及物体交互或搜索的任务进行推理：识别相关物体，并确定是否需要它们之间进行交互才能完成该任务。

IV. 方法

本节简要介绍我们在 IV-A 节中对场景图的定义及其构建，随后在 IV-B 节详细讨论开放词汇特征和关系。最后，我们介绍 IV-C 节中的推理模块，该模块利用场景图信息

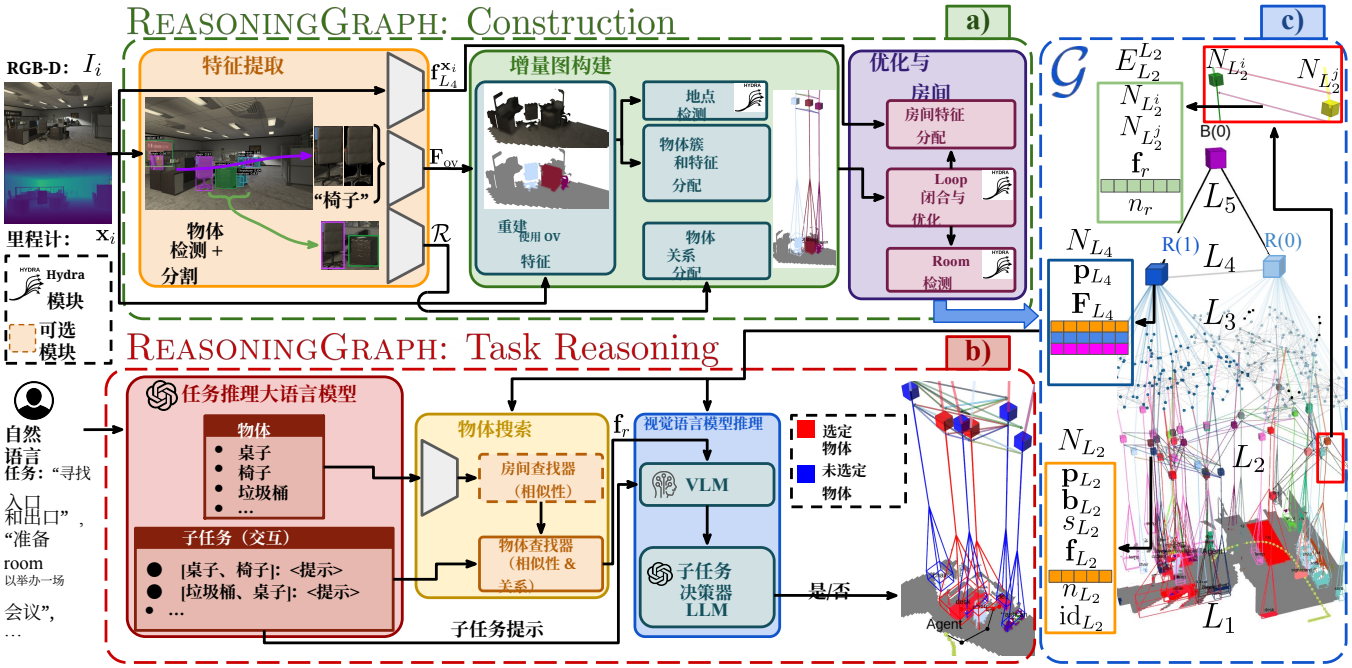


图 2: 推理图概览。 **a)** 推理图逐步构建分层三维场景图 $\mathcal{G}$ **(c))** 基于 RGB-D 帧和位姿，利用开放词汇检测器 [22] 和 CLIP [7] 嵌入来表示物体。物体关系由视觉语言模型 [23] 视觉编码器推导，而 Hydra [10] 重建语义网格（L_1），对物体进行聚类（L_2），并检测地点和房间（L_3，L_4）。随后将开放词汇特征和关系赋予 $\mathcal{G}$。 **b)** 任务推理模块利用两个大语言模型和一个视觉语言模型。给定一项任务，大语言模型会识别相关物体，并制定需要评估的子任务。视觉语言模型会评估这些子任务的可行性，并使用 CLIP 相似度进行物体检索。

来推理给定的任务。我们的方法的整体设计如图 2 所示。

A. 分层场景图的定义与构建

遵循 Hydra [10], [11], ，我们定义了一个五层层次结构（$N = 5$），如图 2c 所示：

- **度量-语义网格层** L_1: 每个节点为 $N_{L_1} = \{\mathbf{v},\ \mathbf{c}\ s\} \in \mathcal{N}_{L_1}$，其中 $\mathbf{v} \in \mathbb{R}^3$ 是一个网格顶点，$\mathbf{c} \in [0, 255]^3$ 是其颜色，$s \in \mathbb{N}$ 是其语义标签。如果一个顶点属于某个物体，我们就添加一条图边 $\{N_{L_1}, N_{L_2}\} \in \mathcal{E}_{L_1}^{L_2}$ 。

- 物体层 L_2: 如图 2c 所示，每个对象节点为 $N_{L_2} = \{\mathbf{p}_{L_2},\ \mathbf{b}_{L_2},\ s_{L_2},\ \mathbf{f}_{L_2},\ n_{L_2},\ \mathrm{id}_{L_2}\} \in \mathcal{N}_{L_2}$，其中 $\mathbf{p}_{L_2} \in \mathbb{R}^3$ 是物体质心，$\mathbf{b}_{L_2} \in \mathbb{R}^6$ 是其边界框，$s_{L_2} \in \mathbb{N}$ 是其语义标签，$\mathbf{f}_{L_2} \in \mathbb{R}^{|\mathbf{f}_{L_2}|}$ 是开放词汇特征，$n_{L_2} \in \mathbb{N}$ 是特征更新次数，$\mathrm{id}_{L_2} \in \mathbb{N}$ 是用于分配物体级关系的变量标识符。物体通过 $E_{L_2}^{L_3} = \{N_{L_2}, N_{L_3}\} \in \mathcal{E}_{L_2}^{L_3}$ 与最近地点相连。两个物体（$N_{L_2^i}$ 和 $N_{L_2^j}$）之间的物体级语义关系（例如"一个玻璃杯在桌子上"）通过 $E_{L_2^i}^{L_2^j} = \{N_{L_2^i}, N_{L_2^j}, \mathbf{f}_r,\ n_r\} \in \mathcal{E}_{L_2}^{L_2}$ 表示，其中 $\mathbf{f}_r \in \mathbb{R}^{|\mathbf{f}_r|}$ 编码该关系，$n_r \in \mathbb{N}$ 记录其更新次数。这些关系可在图 2c 中可视化。

- **地点层** L_3: 每个节点均为 $N_{L_3} = \{\mathbf{p}_{L_3}\} \in \mathcal{N}_{L_3}$，以 $\mathbf{p}_{L_3} \in \mathbb{R}^3$ 作为质心。地点通过图边 $E_{L_3}^{L_4} = \{N_{L_3}, N_{L_4}\} \in \mathcal{E}_{L_3}^{L_4}$ 与其所属房间相连。

- **房间层** L_4 $N_{L_4} = \{\ _{L_4}\ \ _{L_4}\} \in$: 每个节点为 $\mathcal{N}_{L_4}$ $_{L_4} \in \mathbb{R}^3$ $\mathbf{p}$，$\mathbf{F}$，其中 $\mathbf{p}$ 是房间质心，$\mathbf{F}$ $_{L_4} =$ $[\ ^1_{L_4} \ \cdots\ \ ^K_{L_4}]^T \in \mathbb{R}^{K\times|\ _{L_4}|}$ K $\mathbf{f}$，，$\mathbf{f}$ 是一组开放词汇特征簇，其中 $\mathbf{f}_{L_4}^i \in \mathbb{R}^{|\mathbf{f}_{L_4}|}$ 是该集合中的第 i 个开放词汇特征簇，（房间节点的可视化表示见图 2c）。房间通过边 $E_{L_4}^{L_5} = \{N_{L_4}, N_{L_5}\} \in \mathcal{E}_{L_4}^{L_5}$ 与建筑物相连。

- **建筑层** L_5 $N_{L_5} = \{\ _{L_5}\} \in$: 每个节点均为 $\mathbf{p}$，$\mathcal{N}_{L_5}$ $_{L_5} \in \mathbb{R}^3$ 以 $\mathbf{p}$ 作为质心。

定义分层场景图后，我们现在介绍如何根据传感器数据构建该图（图2a）。我们使用 Hydra [10] 重建场景并提取分层图层（L_1 - L_5）。Hydra 在线增量构建网格（L_1）、物体（L_2）和地点（L_3）层，而房间层（L_4）、网格细化和位姿优化则以较低频率更新。由此生成的层次结构构成开放词汇特征和关系推理增强的基础。

语义网格（$\mathcal{N}_{L_1}$）使用 Kimera [5]进行语义分割，并通过窗口化的 Voxblox [24]提取截断符号距离场（TSDF）、欧几里得符号距离场（ESDF），以及通过移动立方体算法生成的网格。

物体（$\mathcal{N}_{L_2}$）通过对具有相同语义标签的顶点进行欧几里得聚类来提取，并逐步融合相互重叠且属于同一类别的物体。地点（$\mathcal{N}_{L_3}$）通过稀疏化基于 ESDF 得到的广义Voronoi图（GVD）获得，并将所得体素连接起来形成图。房间（$\mathcal{N}_{L_4}$）通过膨胀体素地图并修剪相应的地点子图来检测，使相连的地点节点对应于房间。

B. 开放词汇与推理增强 *ent*

我们通过为物体（$\mathcal{N}_{L_2}$）和房间（$\mathcal{N}_{L_4}$）附加开放词汇特征，增强图 $\mathcal{G}$。此外，我们计算物体之间的关系特征，并将其纳入

图边（$\mathcal{E}_{L_2}^{L_2}$）。这一增强过程可在图 2c 中直观呈现。在我们的框架中，引入此类特征需要若干步骤，具体步骤见算法 1 至 4。接下来，我们将详细介绍如何计算这些特征并将其纳入 $\mathcal{G}$。

物体特征。在算法1（第2行）中，我们首先使用开放集检测与分割方法（如 YOLOe）从输入的 RGB-D 帧 $I = \{I^{\text{RGB}}, I^{\text{Depth}}\}$（为简洁起见，省略帧索引）中检测物体边界框、分割掩码和语义标签（$\mathcal{B}$，I^{seg}，$\mathcal{S}$）。对于每个检测到的物体$i \in \{1, \ldots, |\mathcal{B}|\}$，我们生成两种图像裁剪：（a）掩码图像 $g_{\text{mask}}(I^{\text{RGB}}, I_i^{\text{seg}})$，其中物体被隔离在黑色背景上；（b）边界框裁剪图$g_{\mathcal{B}}(I^{\text{RGB}}, \mathcal{B}_i)$。然后，我们计算 CLIP [7] 嵌入的平均值：

$$\mathbf{f}_{\text{ov}}^i = \alpha_{\text{mask}} \mathbf{f}_{\text{CLIP}}(g_{\text{mask}}(I^{\text{RGB}}, I_i^{\text{seg}})) + \alpha_{\mathcal{B}} \mathbf{f}_{\text{CLIP}}(g_{\mathcal{B}}(I^{\text{RGB}}, \mathcal{B}_i)) + \alpha_s \mathbf{f}_{\text{CLIP}}(\mathcal{S}_i), \tag{3}$$

其中，$\mathbf{f}_{\text{CLIP}}(g_{\text{mask}}(I^{\text{RGB}}, I_i^{\text{seg}}))$ 是掩码物体的嵌入，$\mathbf{f}_{\text{CLIP}}(g_{\mathcal{B}}(I^{\text{RGB}}, \mathcal{B}_i))$ 是由其边界框定义的裁剪物体的嵌入，而$\mathbf{f}_{\text{CLIP}}(\mathcal{S}_i)$ 是物体语义标签的嵌入。按照 [16]，我们结合裁剪嵌入和掩码嵌入，因为这能增强 CLIP 表征的鲁棒性；语义标签嵌入则提供了互补的文本线索。权重满足 $\alpha_{\text{mask}} + \alpha_{\mathcal{B}} + \alpha_s = 1$。最后，将所有物体的开放词汇特征汇集为一个由特征向量构成的向量，记为$\mathbf{F}_{\text{ov}}$（第6行）。这一过程见图2a中的特征提取模块。

因此，在算法 2 中，使用 Voxblox [24] 和移动立方体进行三维重建时，会将物体特征暂时附加到网格（$\mathcal{N}_{L_1}$）上（第 2 行）。对于聚类网格中的每个物体 C^i（第 3 行），设 $i \in \{1, \ldots, N_C\}$和 $N_C \in \mathbb{N}$分别为簇的数量，我们判断其顶点是否包含开放词汇特征。如果包含，则对这些特征取平均，得到 $\mathbf{f}_{\text{mesh}}^i$（第 7 行）。如果 C^i 对应图（$N_{L_2^i}$）中的一个节点，则对其开放词汇特征取平均（第 10、11 行）：

$$\mathbf{f}_{L_2^i} = \frac{n_{L_2^i} \mathbf{f}_{L_2^i} + \mathbf{f}_{\text{mesh}}^i}{n_{L_2^i} + 1}, \quad \mathbf{f}_{L_2^i}, n_{L_2^i} \in N_{L_2^i}. \tag{4}$$

否则，我们根据该簇创建一个新节点 L_2^i 来表示 C^i，并将其添加到图中（第 13 行）。随后，从网格（$\mathcal{N}_{L_1}$）中移除检测到的开放词汇特征，以减少内存使用量（第 16 行）。

房间特征。与物体类似，我们为房间提取开放词汇特征。在特征提取步骤（算法1）中，系统持续计算完整 RGB 帧（I^{RGB}）的 CLIP嵌入（$\mathbf{f}_{L_4}^{\mathbf{x}}$），并将每个嵌入与智能体对应的位姿（$\mathbf{x}$）相关联（第7行）。在房间特征分配模块（算法 3；另请参见图2a中的“优化与房间”）中，我们根据空间包含关系，将时间步 n时当前计算得到的所有完整 RGB 嵌入$\{\mathbf{f}_{L_4}^{\mathbf{x}_0}$、……、$\mathbf{f}_{L_4}^{\mathbf{x}_n}\}$与房间相关联。对于每个检测到的房间（第3行），我们收集与位姿相关联且位于该房间边界内的所有 CLIP嵌入（第5行和第6行）。由于 CLIP嵌入

算法1 特征提取

1: **输入：** $\mathbf{x}$，I
2: $\mathcal{B}$，I^{seg}，$\mathcal{S}$ = 检测(I)　　▷物体检测
3: **对于** $i = 1, \ldots, |\mathcal{B}|$ **执行**　▷ 每个物体的平均CLIP
4: $\mathbf{f}_{\text{ov}}^i$ = 平均CLIP(I^{RGB}，I_i^{seg}，$\mathcal{B}_i$，$\mathcal{S}_i$)　　▷公式 (3)
5: **结束 for**
6: $\mathbf{F}_{\text{ov}} = \{\mathbf{f}_{\text{ov}}^0, \ldots, \mathbf{f}_{\text{ov}}^{|\mathcal{B}|}\}$
7: $\mathbf{f}_{L_4}^{\mathbf{x}} = \mathbf{f}_{\text{CLIP}}(I^{\text{RGB}})$　▷ 房间完整输入的CLIP
8: $\mathcal{R} = \{(i, j) : \mathbf{f}_{\text{VLM}}(I^{\text{RGB}}[\mathcal{B}_i \cup \mathcal{B}_j])\}$ $\forall i, j \in \{1, \ldots, |\mathcal{B}|\}$，$i \neq j$　▷ 视觉语言模型视觉编码器以获取物体关系特征。存储在物体字典中对
9: **返回** I^{seg}，$\mathcal{S}$，$\mathbf{F}_{\text{ov}}$，$\mathbf{f}_{L_4}^{\mathbf{x}}$，$\mathcal{R}$

算法 2 物体特征分配

1: **Input:** $\mathbf{x}$, I, I^{seg}, $\mathcal{S}$, $\mathbf{F}_{\text{ov}}$, $\mathcal{R}$
2: $\mathcal{N}_{L_1} = \texttt{reconstruct}_{\text{ov}}(\mathcal{N}_{L_1}, \mathbf{x}, I, I^{\text{seg}}, \mathbf{F}_{\text{ov}}, \mathcal{S}, \mathcal{R})$
3: $C = \{C^1, \ldots, C^{N_C}\} = \texttt{cluster}(\mathcal{N}_{L_1})$
4: **for** $i = 1, \ldots, N_C$ **do**　▷ Iterate over object clusters
5: $\mathbf{f}_{\text{mesh}}^i = \emptyset$, $n = 0$　▷ Empty feature
6: **if** $\mathcal{N}_{L_1}$.has_features(C^i) **then**　▷ Fill feature
7: $\mathbf{f}_{\text{mesh}}^i = \texttt{average}(\mathcal{N}_{L_1}.\texttt{features}(C^i))$, $n = 1$
8: **end if**
9: **if** $N_{L_2^i} \in \mathcal{N}_{L_2}$ **then**　▷ Running average of features
10: $\mathbf{f}_{L_2^i} = \texttt{average}_{L_2}(N_{L_2^i}, \mathbf{f}_{\text{mesh}}^i)$　▷ Eq. (4)
11: $n_{L_2^i}$ += 1, $\text{id}_{L_2^i} = L_2^i$.id
12: **else**　▷ Add new object node to the graph
13: $\mathcal{N}_{L_2} \leftarrow \{C^i.\mathbf{p}_{L_2}, C^i.\mathbf{b}_{L_2}, C^i.s_{L_2}, \mathbf{f}_{\text{mesh}}^i, n, C^i.\text{id}\}$
14: **end if**
15: **end for**
16: $\mathcal{N}_{L_1}$.remove($\mathbf{F}_{\text{ov}}$)　▷ Delete mesh features

房间的 CLIP嵌入在不同视角下可能存在显著差异，因此我们使用 K-均值将收集到的嵌入聚类为 K 组（第9行）。这将生成一组与第 i个房间$N_{L_4^i}$相关联的开放词汇特征簇$\mathbf{F}^{L_4^i}$。通过对嵌入进行聚类，我们假设具有相似视锥体的图像会产生相似的 CLIP嵌入。

物体级关系。我们利用视觉语言模型的强大表达能力，显式建模物体之间的关系，以增强图的表示。在特征提取模块中检测输入 RGB-D 帧（I）中的物体后（参见图2a和算法1），我们使用视觉语言模型的视觉编码器（$\mathbf{f}_{\text{VLM}}$）提取每对检测物体的视觉特征，并将其存储在字典（$\mathcal{R}$）中，字典的键为检测到的物体的索引（第8行）。我们的框架旨在兼容任何视觉语言模型，因为所有架构都包含视觉编码器、视觉语言适配器和大语言模型。在我们的图 $\mathcal{G}$中，我们附加视觉编码器的输出，而视觉语言适配器和大语言模型则用于推理模块（IV-C节）。

输入$\mathbf{f}_{\text{VLM}}$ 是由两个物体边界框的并集所定义的裁剪区域。在该区域内，使用每个标签对应的独特颜色将边界框图像修复到图像中（见图4），以明确告知视觉语言模型

算法 3 房间特征分配

输入： $\{\mathbf{f}_{L_4}^{\mathbf{x}_0}, \ldots, \mathbf{f}_{L_4}^{\mathbf{x}_n}\}$, $\{\mathbf{x}_0 \ldots, \mathbf{x}_n\}$
房间特征 $= \{\}$
$\mathcal{N}_{L_4}$ =检测房间($\mathcal{G}$)
对于 $i = 0, \ldots, n$ **do**
  N_{L_4} = 查找房间($\mathcal{N}_{L_4}$,$\mathbf{x}_i$) ▷ 包含$\mathbf{x}_i$的房间
  room_特征$[N_{L_4}]$.append($\mathbf{f}_{\mathbf{x}_{L_4}^i}$)
结束循环
对于 $i = 0, \ldots, |\mathcal{N}_{L_4}|$ **do** ▷ 房间特征簇
  $\mathbf{F}_{L_4^i}$ =K 均值（房间特征$[N_{L_4^i}]$）
结束循环

算法 4 物体关系分配

输入: $\mathcal{R}$
对于 $i = 1, \ldots, |\mathcal{N}_{L_2}|$; $j = 1, \ldots, |\mathcal{N}_{L_2}|$; $j \neq i$ **执行**
  $\mathbf{f}_{\text{VLM}}^{i,j} = \mathcal{R}[$(编号$_{L^i}$, 编号$_{L_2^j}$)$]$
  如果 $\mathbf{f}_{\text{VLM}}^{i,j} == \emptyset$ **：继续** ▷ 添加/更新关系
  如果 $E_{L^i}^{L_2^j} \in \mathcal{E}_{L_2}^{L_2}$ **则** ▷更新现有关系
    $\mathbf{f}_r^{i,j} =$ 平均$_r(E_{L_2^i}^{L_2^j}, \mathbf{f}^{i,j}$ 视觉语言模型) ▷ Eq. (5)
    $n_r^{i,j} += 1$
  else ▷ 向图中添加新关系
    $\mathcal{E}_{L_2}^{L_2} \leftarrow \{N_{L_2^i}, N_{L_2^j}, \mathbf{f}^{i,j}$ 视觉语言模型 $1\}$
  结束 if
结束 for
$\mathcal{N}_{L_1}$.remove($\mathcal{R}$.keys()) ▷ 删除关系 ID

需要进行推理的重要物体。随后，这些特征可以与文本提示结合，以自然语言描述图中物体之间的关系。

在重建网格时（第 2 行，算法 2），我们将检测到的对象索引（称为 ID）附加到网格上，随后用于为每个对象节点分配一个 ID（第 11 行和第 13 行）。在算法 4 中，我们使用这些 ID 来分配或更新物体级关系。如果物体之间的边已存在于图中，我们使用以下运行平均来更新其关系特征（第 5 至 7 行）：

$$\mathbf{f}_r^{i,j} = \frac{n_r^{i,j}\mathbf{f}_r^{i,j} + \mathbf{f}_{\text{VLM}}^{i,j}}{n_r^{i,j} + 1}, \quad \mathbf{f}_r^{i,j}, n_r^{i,j} \in E_{L_2^i}^{L_2^j}. \tag{5}$$

否则，我们创建一条包含视觉关系特征的新边（第 9 行）。最后，我们从网格中移除 ID（$\mathcal{N}_{L_1}$），以免它们在下一次迭代分配关系特征时产生干扰（第 12 行）。

C. 任务推理

在本节中，我们提出一种方法，利用编码在 $\mathcal{G}$ 中的开放词汇语义和物体关系信息来对任务进行推理。目标是评估任务可行性，并生成能够指导下游规划和场景交互方法的高层计划。给定一个自然语言任务，我们的方法由一个三阶段推理流程组成，如图 2b 所示。

任务推理。我们使用大语言模型解析输入任务。大语言模型会识别出一组 $N_{\text{LLM}} \in \mathbb{N}$ 任务相关物体；如果任务要求这些物体之间进行交互，还会将这些交互列为子任务。对于每个

任务推理大语言模型系统提示词

你会收到一段关于某项任务的自然语言描述 · 你的目标是：
1. **识别与**该任务相关的物体。
2. 判断该任务是否需要对**物体交互**进行推理（例如，移动、放置、使用等）。
你的输出必须采用**JSON**格式，包含：
物体：相关物体列表。
交互：用于描述物体交互子任务的提示（如果任务仅涉及定位、检测或分类物体，则留空）。
示例 1。 提示：“找个地方坐下。” 输出：{"objects": ["沙发", "椅子", "床"], "interactions": {}}
示例 2。 提示："把食物保存起来以备后用。" 输出：{"objects": ["食物", "冰箱", "冷冻柜"], "interactions": {"1": {"objects": ["食物", "冰箱"], "prompt": "检查食物是否已妥善存放；如果没有，能否将其放入冰箱？"}, "2": {"objects": ["食物", "冷冻柜"], "prompt": "检查食物是否已妥善存放；如果没有，能否将其放入冷冻柜？"}}}
...

子任务决策器大语言模型系统提示

你是一名决策助手。给定：
- **视觉语言模型的提示：**关于某项任务的问题或指令。
- **视觉语言模型的回答：**来自视觉语言模型的答案。
判断该**任务是否应当执行**，评估其**必要性、合理性**以及**可行性**，依据**视觉语言模型**。
若应当执行该任务，请回答**"Yes"**；若不应执行，请回答**"No"**。

图 3：用于任务推理和子任务决策大语言模型的系统提示。

子任务，大语言模型会生成一条自然语言提示，用于评估其可行性。为确保输出一致且结构化，我们设计了一个系统提示，如图 3 所示，引导大语言模型遵循这一推理流程，并返回一个包含相关物体、其子任务（如有需要）及对应提示的 JSON 文件。我们还通过提供任务示例（图 3 下半部分）来进一步引导大语言模型。

对象与房间搜索。根据任务推理大语言模型提供的 N_{LLM} 与任务相关的物体列表，下一步是在 $\mathcal{G}$中定位这些物体。为此，我们首先计算所提供物体名称的 N_{LLM} CLIP [7]嵌入。然后，通过计算这些嵌入与场景图中已有物体的开放词汇特征$\mathbf{f}_{L_2}$之间的余弦相似度，将两者进行比较。如果某个物体的相似性超过预设阈值，则认为已找到该物体。如果任务推理大语言模型指定了物体之间的子任务，我们会在匹配物体后验证图中对应节点之间是否存在关系边。视觉语言模型执行子任务推理步骤时需要这些关系信息。虽然 $\mathcal{G}$ 中的物体带有语义标签，但为了减轻对象检测器造成的标签分类错误，我们依靠其 CLIP 嵌入来搜索物体。

此外，我们的方法支持聚焦于特定房间中的物体。为此，我们计算任务推理大语言模型提供的物体名称的 CLIP 嵌入与各个检测到的房间内物体的嵌入之间的余弦相似度。对于每个房间，我们对其$K \times N_{\text{LLM}}$ 余弦相似度取平均值，并选择平均值超过给定阈值的房间。这样，我们便能将物体搜索和推理范围缩小至相关区域。

视觉语言模型推理。如果任务需要推理物体之间的交互，且相关物体已在 $\mathcal{G}$ 中定位，我们会使用视觉语言模型评估每个子任务是否必要且可行。我们将任务推理大语言模型生成的子任务与存储在 $\mathcal{G}$ 中的物体级关系特征（$\mathbf{f}_r$）结合起来。除了任务推理大语言模型生成的子任务外，我们

	HM3DSem [25]								Replica [26]							
	Acc_5	Acc_{10}	Acc_{25}	Acc_{100}	Acc_{250}	Acc_{500}	AUC_k^{Acc}	#物体	Acc_5	Acc_{10}	Acc_{25}	Acc_{100}	Acc_{250}	Acc_{500}	AUC_k^{Acc}	#物体
ConceptGraphs [15]	**18.5**	**24.70**	**38.05**	**60.29**	**76.58**	**85.98**	**87.27**	**237.78**	**20.70**	**30.02**	**41.72**	**57.47**	**72.85**	**79.94**	**80.50**	**40.5**
HOV-SG [16]	16.09	20.98	32.62	56.78	71.79	81.98	84.46	**405.33**	3.66	6.56	11.26	24.00	43.65	63.27	66.82	**72.63**
REASONINGGRAPH (我们的方法)	**37.72**	**45.18**	**56.52**	**74.47**	**85.51**	**92.95**	**92.49**	88.33	**53.99**	**58.23**	**63.16**	**73.74**	**87.29**	**88.80**	**87.23**	13.25

表 I：物体检索定量结果。 **最佳结果**，**次佳结果**。REASONINGGRAPH 在各项指标和数据集上均优于基线方法。请注意，ConceptGraphs [15] 和 HOV-SG [16] 基于完全开放词汇的分割运行，这些分割通过对其点云的嵌入进行聚类计算得出。

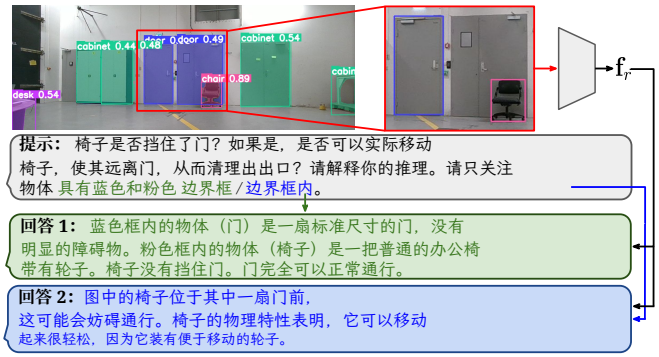


图4：通过边界框颜色引导视觉语言模型。视觉语言模型会收到一张包含一对物体及其图像修复边界框的图像，以及任务推理大语言模型的子任务提示。在此示例中，任务是推理一扇门和一把椅子。若提示中不包含边界框颜色（蓝色），视觉语言模型会关注错误的物体，转而推理椅子后面的门。加入颜色信息后（绿色），视觉语言模型便会受到引导，关注相关物体。

还会附加一条提示，引导视觉语言模型关注图像中的正确物体。具体而言，这条人工编写的提示为：仅关注具有 X 和 Y 边界框的物体，其中 X 和 Y 是根据物体标签 s_{L_2} 推断出的边界框颜色。为使其有效，集合中的每个标签都关联一种独特的颜色。如图4所示，提供这项额外信息有助于视觉语言模型关注感兴趣的物体。相比之下，如果提示仅指定带有通用边界框的物体，视觉语言模型可能难以聚焦于预期物体。

最后，第二个大语言模型解读视觉语言模型的输出，以决定是否应执行这些子任务。该大语言模型由专用的系统提示引导（见图 3）。

V. 实验

在本节中，我们首先介绍我们的方法的实现细节（第 V-A 节）。随后，我们评估 REASONINGGRAPH 构建富含物体级开放词汇特征的场景图的能力（第 V-B 节）。接下来，我们利用四足机器人以及由人工操作员采集的数据，在多种真实场景中评估任务推理模块（第 V-C 节）。最后，我们报告我们的方法的运行时性能（第 V-D 节）。

A. 实现细节

我们采用 YOLOe [22] 作为对象检测器，同时输出边界框和分割掩码。对于开放词汇特征提取，我们使用 CLIP [7], ，具体选用 OpenAI 的 ViT-L/14 模型，该模型在图像分类任务中具有很强的泛化能力。我们选择 Deepseek VL2 [23] 作为视觉语言模型，因为它达到了最先进水平（SOTA）的性能。此外，我们使用 OpenAI 的 o3 大语言模型进行任务推理和解析，并使用 GPT-4o 作为子任务决策大语言模型。

我们的传感器和计算配置包括一台 Realsense D455 RGB-D相机、一台 Ouster OS0激光雷达，以及一台用于里程计的 VectorNav VN100 IMU，并集成 NVIDIA Jetson Orin AGX 进行机载处理（见图 1）。在 Orin AGX 上，我们以 1 Hz 在线运行特征提取模块（YOLOe、CLIP 和 VL2视觉编码器）以及场景图构建；房间检测、网格细化和位姿优化则以 0.5 Hz 运行。大语言模型和 VL2 的语言组件托管在云端，仅在接收到任务时才会调用。

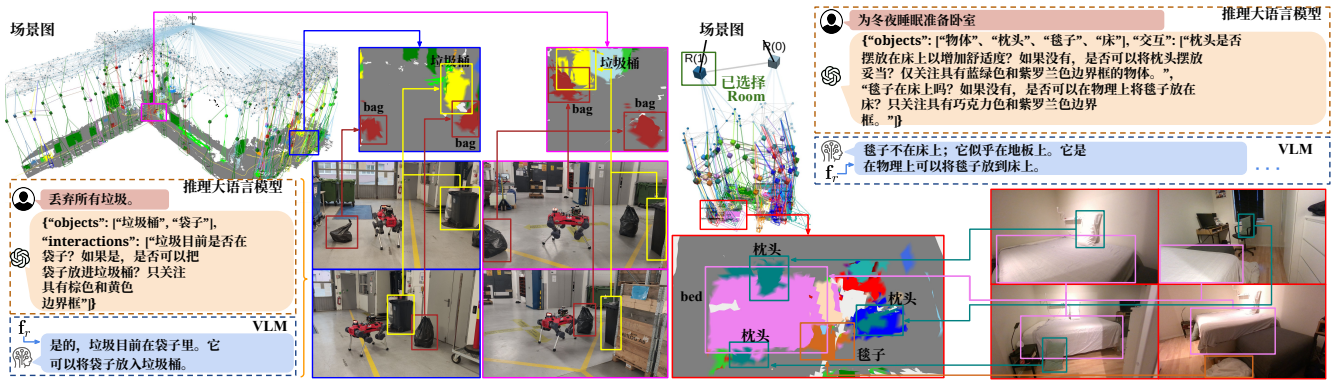
B. 开放词汇目标检索

为评估开放词汇目标特征，我们计算每个物体特征与数据集标签之间的余弦相似度。随后，我们按相似性对标签进行排序，并衡量排名前 k 的预测的准确率（Acc_k）。遵循 HOV-SG [16]，我们还计算前 k 准确率曲线下的面积（AUC_k^{Acc}），该指标刻画了在不同 k 取值下预测类别与真实目标类别之间的对齐程度。我们将我们的方法与两种强有力的开放词汇场景图基线方法 HOV-SG [16] 和 ConceptGraphs [15]，在 Replica [26] 与 Habitat Semantics（HM3DSem） [25] 数据集上进行比较。

ConceptGraphs [15]和HOV-SG [16] 均从完全开放词汇的目标分割片段构建场景图，这些片段通过对点云上的CLIP嵌入进行聚类获得，从而在其图中引入了背景目标。为确保公平比较，我们将这些背景目标过滤掉，因为我们的方法仅包含前景目标。背景目标通常会降低性能，因为其嵌入因缺乏纹理以及受到前景目标污染而更加模糊。具体而言，我们计算每个检测到的物体与一组背景类别（例如，“墙壁”、“地板”、“天花板”、“楼梯”）之间的余弦相似度，并丢弃具有高相似性的物体。表I中的结果报告了我们的方法与基线方法的top-k 准确率（Acc_k）和AUC_k^{Acc}分数。REASONINGGRAPH取得了显著更高的性能，表明其能够为物体分配正确的开放词汇特征。需要注意的是，基线方法检测到的物体数量更多，因为其完全开放词汇的设计通过在嵌入空间中聚类来检测物体，当同一物体在不同视角下的CLIP嵌入发生变化时，可能会将单个物体分割成多个片段。尽管如此，我们仍将这些方法作为基线方法纳入比较，因为它们代表了当前开放词汇场景图构建的最先进水平。

C. 真实世界任务推理评估

为评估 REASONING- GRAPH 处理复杂任务时的推理能力，我们设计了一系列评估——



a) 四足机器人正在执行**T1**。 b) 操作员正在执行**T3**。

图 5：REASONINGGRAPH 正在执行**T1** 和**T3**。探索过程中会构建场景图，之后大语言模型对任务进行推理，视觉语言模型则利用物体关系评估子任务（$\mathbf{f}_r$）。在两项单独实验中，成功率均达到 100%。在**T3**中，我们采用房间搜索方法，在全部 5 次评估中均达到 100% 的准确率。我们为每项任务展示一个视觉语言模型推理示例，但在实际应用中，视觉语言模型会对每个子任务进行推理。

任务，要求识别图中的物体，并在某些情况下推理它们之间的关系。由于没有专门的评估数据集，我们选择开展实际实验，这不仅提供了必要的评估数据，也使我们能够在实践中验证我们的方法的鲁棒性。我们使用两项指标衡量性能：成功率（SR%），定义为正确评估的子任务或成功识别的物体所占的百分比；以及假阳性（FP）的数量，定义为错误评估的子任务或误识别的物体的数量。对于每项任务，我们都会人工标注其子任务应被判定为正向的真实值物体或物体对。具体而言，我们考虑以下任务：

T1：处理所有垃圾：找出垃圾桶附近装满的垃圾袋，并判断是否可以丢弃。**T2：确保出口和入口没有被堵塞**：检测可能堵住门口的物体。**T3：为冬夜睡眠做好卧室准备**：检查枕头和毯子是否妥善放置在床上。**T4：为清洁地板做好会议室准备**：判断椅子和其他物体是否可以放到桌子上，以便于清洁。

T5：寻找背包、风扇、植物和垃圾桶。

对于 **T1**、**T2** 和 **T5**，REASONINGGRAPH 部署在一台四足机器人上，该机器人配备了 IV-A节中介绍的感知与计算模块。机器人使用基于图的路径规划方法 [27], 自主探索环境，同时逐步构建场景图。对于已检测到的物体，或评估为肯定的子任务所涉及的物体对，均使用同一规划器 [27]进行导航。相比之下，对于 **T3** 和**T4**，数据由人工操作员携带相同的感知模块采集。

场景图构建完成后，便开始进行任务推理。如 IV-C节所述，任务推理大语言模型会识别任务相关物体，然后使用基于 CLIP 的余弦相似度在场景图中定位这些物体。在**T5**如图6所示，在此检索步骤中已足够，因为无需对物体交互进行推理。对于其余任务，视觉语言模型会接收物体关系（$\mathbf{f}_r$）以及由任务推理大语言模型生成的子任务提示。

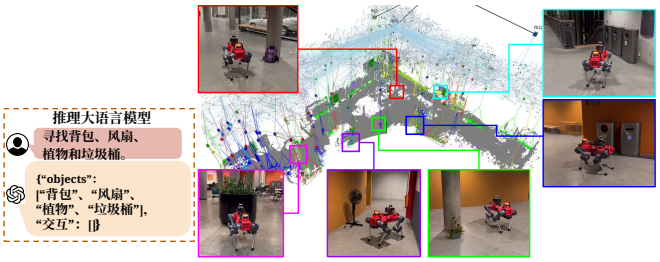


图6：物体搜索任务（**T5**）。在自主探索过程中构建场景图后，我们要求 REASONINGGRAPH 寻找背包、风扇、植物和垃圾桶。我们的方法能够找到所有物体。

表 II 展示了每项任务在五次评估中的成功率和假阳性数量及标准差，并比较了两个视觉语言模型：DeepSeek VL2 [23] 和 InstructBLIP [20]。REASONINGGRAPH（使用 VL2）在所有任务中均实现了稳定较高的成功率（84-100），且假阳性较少。DeepSeekVL2 始终优于 InstructBLIP，取得了更高的成功率和更少的假阳性，从而支持将其选为 REASONINGGRAPH 的主要视觉语言模型。#GT+表示每项任务中的正子任务数量。图5a展示了四足机器人上的**T1** ，其中系统对将四个垃圾袋放入附近垃圾桶的过程进行推理；图5b展示了**T3**，涉及在床上整理毯子和枕头。REASONING- GRAPH 在这两个示例中均取得了较高的成功率。

为了进一步评估物体级关系（$\mathbf{f}_r$）、视觉语言模型（VL2）和子任务决策大语言模型的性能，我们使用从任务 **T1** 到 **T4** 收集的数据，对所有子任务重新评估 100 次，以考虑这两个模型的随机性。如表III所示，组合系统在所有任务中均实现了较高的准确率，准确率定义为正确评估的子任务所占百分比。宏平均 F_1 分数兼顾精确率和召回率，表明系统在正向和负向子任务上均表现出色，即 应执行和不应执行的子任务。汇总结果凸显了我们的方法的鲁棒性，证实其能够在视觉语言模型和大语言模型存在随机性的情况下推理物体交互。*D.* 运行时间评估

为验证在线运行图构建的可行性，我们测量了其主要组件的运行时间，

	VLM	T1	T2	T3	T4	T5
SR%	VL2 [23]	90 ± 12.3	84 ± 15.0	90 ± 12.3	86 ± 7.1	100
	BLIP [20]	55 ± 24.5	48 ± 9.8	40 ± 12.2	53.3 ± 12.5	
FP	VL2 [23]	0.2 ± 0.4	0.8 ± 0.4	0.6 ± 0.5	0.8 ± 0.4	1.8 ± 0.4
	BLIP [20]	0.2 ± 0.4	1.8 ± 0.4	0.6 ± 0.49	1.0 ± 0.0	
	#GT+	4	5	4	6	7

表 II：使用两个视觉语言模型（DeepSeek VL2 [23] 和 Instruct BLIP [20]）进行任务推理性能比较。**T1 - T4** 需要关系推理，而**T5** 则不需要。每项任务执行 5 次，并对结果取平均值。我们的方法使用 VL2 时成功率更高、假阳性更少，因此选择它作为主要视觉语言模型。

任务	T1	T2	T3	T4	汇总
准确率（%）	88.81	84.43	83.11	74.91	79.30
F_1（%）	88.55	85.43	81.86	75.84	79.11
平均假阳性	0.27	1.04	0.69	1.12	0.78
# 正子任务	18	21	20	44	103
# 负向子任务	3	75	7	11	96

表 III：视觉语言模型和子任务决策大语言模型的性能。我们对 **T1 - T4** 中的每个子任务进行 100 次评估。这考虑了视觉语言模型和决策大语言模型带来的随机性。

结果如表 IV 所示。平均而言，YOLOe 和 CLIP 推理每帧耗时不到 160 ms，而 VL2视觉编码器是计算开销最大的组件，耗时为 214 - 278 ms。场景图构建仍然较为轻量，耗时在 145 - 276 ms 之间。总体而言，该方法每帧的运行时间不到一秒。这一运行速度支持该方法在安装于四足机器人上的 Orin AGX 上在线运行，使其能够以 1 - 2 Hz 的频率运行该方法，而不会遗漏重要信息。

VI. 结论

我们提出了 REASONINGGRAPH，这是一种用于构建分层三维场景图的框架，其中融合了开放词汇特征，并利用视觉语言模型编码物体级关系。我们的推理模块还结合大语言模型和视觉语言模型来理解复杂任务、识别与任务相关的物体，并推理它们可能发生的交互。大量实验表明，REASONINGGRAPH 在物体检索方面优于现有的开放词汇场景图基线方法，并且能够以较高的成功率和较低的假阳性率稳定地解决各种推理任务。在四足机器人上的部署展示了该框架在线增量式场景理解和任务推理的能力。这些结果表明，结合层次化表示与视觉语言推理，有望实现更丰富的上下文感知理解。

任务	YOLOe [22]	CLIP [7]	VL2 视觉 [23]	图
T1	44.9 ± 29.5	76.1 ± 56.6	214.8 ± 147	262.2 ± 105.5
T2	54.9 ± 43	97.4 ± 59	257.8 ± 132.4	231.1 ± 86.3
T3	70.7 ± 63.6	133.1 ± 95.5	277.8 ± 141.2	144.9 ± 91.8
T4	87.9 ± 33.8	157.9 ± 69.1	223.8 ± 145.3	213.5 ± 84.2
T5	54.8 ± 57.2	82.9 ± 66.9	247 ± 140.8	276 ± 92.2

表 IV：计时统计（均值 ± 标准差），单位为毫秒。

参考文献

[1] J. L. Sch¨onberger，E. Zheng 等，“非结构化多视图立体的逐像素视图选择”，欧洲计算机视觉会议，2016。[2] N. Khedekar，M. Kulkarni 等，“Mimosa：一种可抵御传感器退化、实现韧性自主能力的多模态SLAM框架”，*IEEE/RSJ*智能机器人与系统国际会议，2022。[3] M. Nissov，N. Khedekar 等，“具备抗退化能力的激光雷达-雷达-惯性里程计”，*IEEE*国际机器人与自动化会议，2024。[4] C. Forster，L. Carlone 等，“用于实时视觉惯性里程计的流形上预积分”，*IEEE*机器人学汇刊，2017。

[5] A. Rosinol，A. Violette 等，“Kimera：从SLAM到基于三维动态场景图的空间感知”，《国际机器人研究杂志》，2021。[6] S. Peng，K. Genova 等，“OpenScene：利用开放词汇进行三维场景理解”，IEEE/CVF计算机视觉与模式识别会议，2023。[7] A. Radford，J. Kim 等，“通过自然语言监督学习可迁移视觉模型”，国际机器学习会议，2021。[8] A. Kirillov，E. Mintun 等，“Segment Anything”，IEEE/CVF国际计算机视觉会议，2023。[9] I. Armeni，Z.-Y. He 等，“三维场景图：统一语义、三维空间与相机的结构”，IEEE/CVF国际计算机视觉会议，2019。[10] N. Hughes，Y. Chang 等，“Hydra：用于三维场景图构建与优化的实时空间感知系统”，机器人学：科学与系统，2022。[11] ——，“机器人空间感知基础：层次化表示与实时系统”，《国际机器人研究杂志》，2024。[12] J. Wald，H. Dhamo 等，“从三维室内重建中学习三维语义场景图”，IEEE/CVF计算机视觉与模式识别会议，2020。[13] S.-C. Wu，J. Wald 等，“SceneGraphFusion：从RGB-D序列进行增量式三维场景图预测”，IEEE/CVF计算机视觉与模式识别会议，2021。[14] S. Koch，N. Vaskevicius 等，“Open3dsg：从点云生成开放词汇三维场景图，支持可查询对象和开放集关系”，IEEE/CVF计算机视觉与模式识别会议，2024。[15] Q. Gu，A. Kuwajerwala 等，“Conceptgraphs：用于感知与规划的开放词汇三维场景图”，IEEE国际机器人与自动化会议，2024。[16] A. Werby，C. Huang 等，“用于语言引导机器人导航的层次化开放词汇三维场景图”，机器人学：科学与系统，2024。[17] Z. Zhang，H. Cai 等，“EfficientViT-SAM：在不损失性能的情况下加速Segment Anything模型”，IEEE/CVF计算机视觉与模式识别会议研讨会，2024。[18] F. Li，H. Zhang 等，“Mask DINO：迈向用于物体检测与分割的统一Transformer框架”，IEEE/CVF计算机视觉与模式识别会议，2023。[19] G. Narita，T. Seno 等，“PanopticFusion：在背景与物体层面进行在线体积语义映射”，IEEE/RSJ智能机器人与系统国际会议，2019。[20] W. Dai，J. Li 等，“InstructBLIP：迈向具备指令微调能力的通用视觉语言模型”，神经信息处理系统大会，2023。[21] L. Chen，X. Wang 等，“通过跨模态对比学习实现CLIP驱动的开放词汇三维场景图生成”，IEEE/CVF计算机视觉与模式识别会议，2024。[22] A. Wang，L. Liu 等，“YOLOe：实时看见万物”，IEEE/CVF国际计算机视觉会议，2025。[23] H. Lu，W. Liu 等，“迈向真实世界的视觉语言理解”，2024。[24] H. Oleynikova，Z. Taylor 等，“Voxblox：用于机载微型飞行器规划的增量式三维欧几里得有符号距离场”，IEEE/RSJ智能机器人与系统国际会议，2017。[25] K. Yadav，R. Ramrakhya 等，“Habitat-Matterport三维语义数据集”，IEEE/CVF计算机视觉与模式识别会议，2023。[26] J. Straub，T. Whelan 等，“Replica数据集：室内空间的数字副本”，2019。[27] T. Dang，M. Tranzatto 等，“使用空中机器人和足式机器人进行基于图的地下探索路径规划”，《野外机器人学杂志》，2020。