# ReSem3D: 通过精细语义基础进行可改进的三维空间约束以实现可推广的机器人操作

Chenyu Su, Weiwei Shang, Chen Qian, Fei Zhang, and Shuang Cong

***Abstract*—语义驱动的三维空间约束将高层语义表示与低层动作空间对齐，促进了任务理解与执行在机器人操作中的统一。多模态大型语言模型（MLLMs）和视觉基础模型（VFMs）的协同推理使跨模态的三维空间约束构建成为可能。然而，现有方法存在三个主要局限性：（1）约束建模中语义粒度较粗，（2）缺乏实时闭环规划，（3）在语义多样环境中稳健性下降。为了解决这些问题，我们提出了 ReSem3D，这是一种用于语义多样环境的统一操作框架，利用 VFMs 和 MLLMs 之间的协同作用，实现细粒度的视觉定位，并动态构建层次化三维空间约束以进行实时操作。具体而言，该框架通过 MLLMs 中的层次递归推理驱动，与 VFMs 交互，自动从自然语言指令和 RGB-D 观测中分两个阶段构建三维空间约束：部件级提取和区域级细化。随后，这些约束在关节空间中被编码为实时优化目标，使其能对动态干扰作出反应。我们在语义丰富的家庭和稀疏的化学实验室环境中进行了广泛的仿真和真实世界实验。结果表明，ReSem3D 能够在零样本条件下执行多样的操作任务，表现出强大的适应性和泛化能力。代码和视频见 ReSem3D.github.io 。对实践者的注意事项——本研究旨在解决机器人在家庭和化学实验室等语义多样的环境中实现稳健和可泛化操控的实际挑战。现有系统在空间约束的细粒度语义建模和实时适应性上往往不足，限制了其在实际应用中的有效性。我们提出了一个统一的框架，该框架利用多模态语言和视觉模型的最新进展，从自然语言指令和视觉观测中自动构建 3D 空间约束。这些约束在层次上得到更细化，并整合到支持反应和上下文感知行为的实时控制系统中。所提出的方法在模拟和物理环境中进行了验证，展示了在对象类别和任务场景上的强泛化能力，使机器人系统能够在动态和语义多样的环境中执行复杂的零样本任务，克服了之前在粗略约束建模中的局限性。该框架为实现适应性操控系统提供了实用解决方案，而无需繁琐的任务特定手动编程。**

***Index Terms*—Multimodal Large Language Models, Vision Foundation Models, Task and Motion Planning, Manipulation.**

随着机器人自主能力在非结构化环境中的进步，语义描述的多样性增加以及视觉建模的复杂性已成为限制系统普遍化和鲁棒性的关键挑战。空间约束作为高层语义表示与低层动作空间之间的关键接口，使得能够将抽象语义映射为可执行的动作。由多模态大型语言模型（MLLMs）和视觉基础模型（VFMs）驱动的跨模态推理的最新进展，显著提高了视觉对准能力，这对于构建准确的三维空间约束至关重要。例如，使用镊子抓取需要定位粗略的语义抓取区域并推理镊子尖端和抓取点之间的三维几何关系。基于这些推断关系，空间约束模型可以实时由约束优化器构建和解析，从而在动态环境中实现闭环控制和细粒度操作。

然而，当前的方法缺乏一种可优化的中间表示，这种表示能够有效地将语言意图与动作空间连接起来，从而阻碍了语义目标与物理执行之间的一致对齐。作为核心的中间表示，三维空间约束通过一组任务基础的姿态参数化了从语义意图到可执行动作的映射。然而，MLLM 的几何推理和空间定位能力仍然不足以直接从语言中推导出细粒度的约束。先前的努力已经利用 MLLM 和 VFM 来增强语言与视觉的语义对齐 [1], [2] ，促进了用于运动规划的三维空间约束的构建 [3], [4] 。这些约束往往仅限于粗略定位，缺乏语义引导的细化。此外，下游控制器通常依赖于任务空间的规划，其受限于逆运动学求解器的稳定性和效率。这些限制提出了两个关键挑战：1）如何设计一个灵活且可优化的三维空间约束模型，以自动将语义意图映射到机器人动作；2）如何确保在语义多样化环境中在低级动作空间内实现稳定的闭环和实时执行。

为了解决第一个挑战，我们提出了一种由 VFMs 和 MLLMs 协同推理驱动的两阶段层次化 3D 空间约束建模方法：部件级约束提取和区域级约束细化。部件级提取利用 VFMs 从 RGB 观测中识别语义相关的部件级区域，同时生成明确的数字标记作为视觉提示，以便 MLLMs 进行粗粒度约束提取。区域级细化则在遮罩区域内建立密集的语义网格，使 MLLMs 能够对几何特征进行细粒度的语义定位。随后，3D 空间约束被编码为实时解析的代价函数，并提供给下游优化器。为了解决第二个挑战，受基于通用物理模拟器 [5], [6] 的有限视界最优控制方法的启发，我们在 Isaac Gym 平台 [7] 中实现了一种实时模型预测路径积分（MPPI）控制算法，该算法提供 15 Hz 的关节空间速度命令而不是任务空间命令，以实现稳定的闭环执行。

此外，我们开发了一种任务和运动规划（TAMP）框架，该框架扩展了 Liang 等人的分层递归提示结构 [8] ，基于 MLLM 的推理驱动，实现了从感知到行动的自主和闭环操作。该框架通过自然语言指令自主确定 3D 空间约束所需的粒度，展示了语义驱动的适应性。它分解多阶段任务，并生成每个子任务操作所需的前置条件检查、后置条件评估和可优化成本函数。在此基础上，制定了子任务级的回溯策略，利用前置条件的可行性来启动跨阶段的重新计划，从而在动态干扰下实现行为恢复和反应控制。

本文的贡献总结如下：

1) 我们提出了一种两阶段层次约束建模框架，该框架整合了部分体积模型（VFMs）和多层次学习模型（MLLMs），以构建语义自适应的多粒度 3D 空间约束，有效弥合高级语义和低级动作之间的差距。
2) 首次通过新开发的由 MLLM 驱动的 TAMP 框架实现了对复杂长时间任务的闭环控制，该框架执行带有条件推理和成本优化的自主多阶段任务分解，并具有细粒度的语义基础。
3) 通过在语义丰富的家庭环境和稀疏的化学实验室环境中进行广泛的模拟和真实世界实验，我们成功验证了 ReSem3D 的强大零样本能力，展示了其良好的泛化性和反应特性。

本论文的其余部分组织如下：在第 I 节，我们回顾相关

The authors are with the Department of Automation, University of Science and Technology of China, Hefei 230027, China (e-mail: suchenyu@mail.ustc.edu.cn; wwshang@ustc.edu.cn; qian_chen@ustc.edu.cn; zfei@ustc.edu.cn; scong@ustc.edu.cn). Corresponding author: Weiwei Shang.

工作，包括基础模型和机器人操作。在第 ?? 节，我们详细介绍所提出的系统架构。在第 II 节，我们通过模拟和现实实验展示我们的方法在语义多样化环境中的表现，包括化学环境和家庭环境。在第 ?? 节，我们讨论优势、局限性和未来工作。最后，我们在第 III 节总结这项工作。

## I. 相关工作

### A. 使用基础模型进行视觉定位

视觉定位指的是将自然语言中的语义特征映射到图像区域和空间实体，随着基础模型的进步，从低级感知发展到语义空间推理。先前的视觉基础模型专注于稳健的图像感知，利用自监督学习 [9], [10] 和包括 CNNs [11], [12] 和 Vision Transformers [13] 的架构，生成用于下游任务的视觉编码，如物体检测 [14], [15] 、分割 [16], [17] 和特征提取 [18] 。此外，通过潜在特征编码和嵌入视觉提示 [19], [20] 提升了像素级解析。随后，视觉-语言模型（VLMs）通过视觉-语言表示执行多模态语义定位，在图文检索 [21] 和开放词汇物体检测 [22] 方面取得显著进展，但在细粒度视觉定位方面仍面临挑战。在此基础上，研究人员将大语言模型（LLMs）[23] 引入视觉-语言系统，形成了多模态大语言模型（MLLMs）[24] ，将视觉特征作为多模态推理的上下文引入。尽管如此，MLLMs 仍依赖于前端视觉编码器，限制了空间精度和语义理解。最近的研究通过嵌入视觉提示进行语义对齐 [1] ，探索了 VFMs 和 MLLMs 之间的协同作用，但细粒度视觉定位仍然是一个关键限制。为解决这一挑战，我们提出了一个多粒度语义理解框架，具有层次多模态推理，以实现部件级提取和区域级细化，从而实现细粒度视觉定位。

### B. 机器人操作中的空间约束

空间约束对机器人操作至关重要，它在高层语义和低层动作之间架起桥梁，以确保任务的可行性。传统方法通过对象形状、接触动力学和环境结构的显式几何建模 [25], [26] 来构建约束，以实现运动规划。虽然在结构化场景中物理上可解释且有效，但它们对准确建模的依赖限制了其在复杂环境中的灵活性。最近，依赖感知的学习方法被用来从视觉观测中推断潜在的操作区域，包括可操作性识别 [27] 、抓取姿势检测 [28], [29] 、目标姿态估计 [30] 和语义关键点生成 [31] 。这些方法进一步扩展到包含基础模型推理的多模态操作框架 [32], [33] 。虽然能够适应对象变形和语义变化，但它们仍需要人工数据收集，并在细粒度语义解析和复杂几何关系的精确建模方面存在困难。当前方法整合语言和视觉模型以实现零样本语义区域感知 [34], [3] 。然而，上述方法通常缺乏细粒度语义定位和可执行的几何规范。为克服这些限制，我们提出了一个两阶段的 3D 空间约束建模框架，该框架分层地将自然语言指令映射到语义目标，从而实现细粒度和可执行的空间约束。

### C. 任务和运动规划

任务和运动规划（TAMP）提供了一个关键的框架，将高层次任务推理与低层次运动执行连接起来。传统的方法通常利用诸如 PDDL 和 HTN 等形式化语言来建模符号任务，通过逻辑-几何和混合整数编程 [35], [36] 解决任务序列和运动轨迹。尽管这些方法具有较强的可解释性，但它们依赖于手动设计的任务模型和动作原语，限制了在开放环境中的适应性。随着 LLMs 的进步，最近的研究探索了基于预定义运动原语的零次任务规划 [37] 。然而，这些方法缺乏对几何约束和环境动态的有效建模。在这些进展的基础上，最近的研究探索了将 LLMs 与运动规划相结合，以实现高层次任务分解和连续动作生成，而无需预定义原语 [4] 。尽管此类方法增强了 TAMP 的灵活性，它们通常依赖于任务空间中的在线优化，这对实时执行提出了挑战。为了应对这些挑战，我们采用了以 Isaac Gym 为基础的实时 MPPI 优化器，通过关节速度控制在执行过程中增强反应行为。同时，子任务级别的约束和优化目标由 MLLM 自动生成，使得在动态环境中实现有效的闭环控制和零次自适应。

在本节中，我们首先将 ReSem3D 表述为一个实时约束优化问题，以标准化任务目标和约束。随后，我们提出了一种具有语义适应性的两阶段分层空间约束模型，用于多粒度 3D 推理。之后，我们推导出一种实时闭环控制策略来解决该优化问题。最后，我们整合了 MLLM 驱动的推理来自动化 TAMP，实现端到端的闭环执行。

### D. *ReSem3D* 问题表述

如图 1 所示，给定来自用户的自由形式自然语言指令 $\mathcal{L} = \{L_1, L_2, \ldots, L_n\}$ ，$\mathcal{L}$ 通常表现出显著的长期依赖关系和复杂的语义推理。直接生成完整的三维空间约束并对其进行优化，会损害推理精确性并使约束建模复杂化。因此，ReSem3D 通过将任务分解为顺序子任务并逐步对每个 $L_i$ ，$i \in \{1, 2, \ldots, n\}$ 进行约束推理和优化，采用分层建模策略。

具体来说，对于第 $i$ 个子任务指令 $L_i$ ，3D 空间约束模型的实例化定义为

$$f : (L_i, O_i) \to \mathcal{C}_i^{\text{init}} \tag{1}$$

其中 $O_i$ 表示 RGB-D 传感器观测，$\mathcal{C}_i^{\text{init}} \in SE(3)$ 表示由 $f$ 映射的初始空间约束集。为了捕捉动态环境中的不确定性，我们引入了外部动态演变和干扰 $\mathcal{E}_i$ 。另外，我们将 $\boldsymbol{T}_e(t) \in SE(3)$ 定义为时间 $t$ 时的末端执行器姿态，$S$ 则为已知的运动和物理模型。在每个阶段 $i$ ，ReSem3D 通过实时解决受限优化问题来处理任务执行：

$$\begin{aligned} \underset{\boldsymbol{v}(t)}{\text{argmin}} \quad & \mathcal{J}\left(\boldsymbol{T}_e(t), \mathcal{C}_i\right) \\ \text{subject to} \quad & \mathcal{C}_i^{\text{init}} = f(L_i, O_i), \\ & \mathcal{C}_i = g(\mathcal{C}_i^{\text{init}}, \mathcal{E}_i), \\ & \epsilon_{\text{pre}}\left(\boldsymbol{T}_e(t),\ \mathcal{C}_i\right) \le \varepsilon_{\text{pre}}, \\ & \epsilon_{\text{post}}\left(\boldsymbol{T}_e(t), \mathcal{C}_i\right) \le \varepsilon_{\text{post}}, \\ & \boldsymbol{v}(t),\ \boldsymbol{T}_e(t) \in \text{Feasible}\left(S\right) \end{aligned} \tag{2}$$

，其中 $\mathcal{J}\left(\boldsymbol{T}_e(t), \mathcal{C}_i\right)$ 是实时优化的代价函数，$g(\cdot)$ 表示实例映射，它基于外部扰动更新空间约束，$\epsilon_{\text{pre}}\left(\boldsymbol{T}_e(t), \mathcal{C}_i\right)$ 和 $\epsilon_{\text{post}}\left(\boldsymbol{T}_e(t), \mathcal{C}_i\right)$ 表示子任务执行的前置条件和后置条件约束，$\varepsilon_{\text{pre}}$ 和 $\varepsilon_{\text{post}}$ 指定其容忍阈值。集合 $\text{Feasible}\left(S\right)$ 强制与机器人运动学和物理模型保持一致，$\boldsymbol{v}(t) \in \mathbb{R}^n$ 是关节空间速度，作为决策变量，$n$ 表示自由度。

基于前面描述的配置，ReSem3D 将长时间视野的操作任务分解成多阶段实时约束优化问题，从而使每个子任务实现闭环动态控制。随后，处理后的决策变量 $\boldsymbol{v}(t)$ 直接作为关节速度命令，使机器人能够在动态环境中有效执行任务。

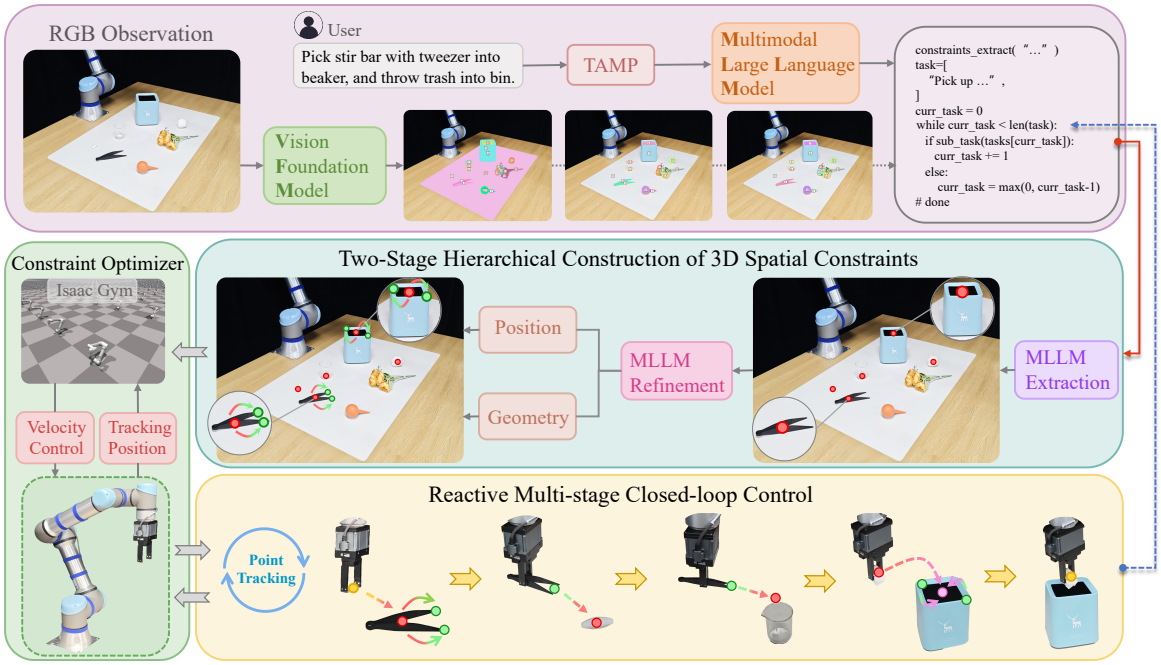


Fig. 1. 总体框架。给定自然语言指令和 RGB-D 观测，VFM 分割语义相关的部件级区域，并叠加视觉提示以促进初始约束生成。在 MLLM 驱动的 TAMP 框架下，约束建模分两个阶段分层进行：部件级提取和区域级细化。结果生成的 3D 空间约束被编码为代价函数，进行实时解析，并在 Isaac Gym 中通过基于 MPPI 的优化器闭环求解，从而实现带点跟踪的关节空间速度控制。

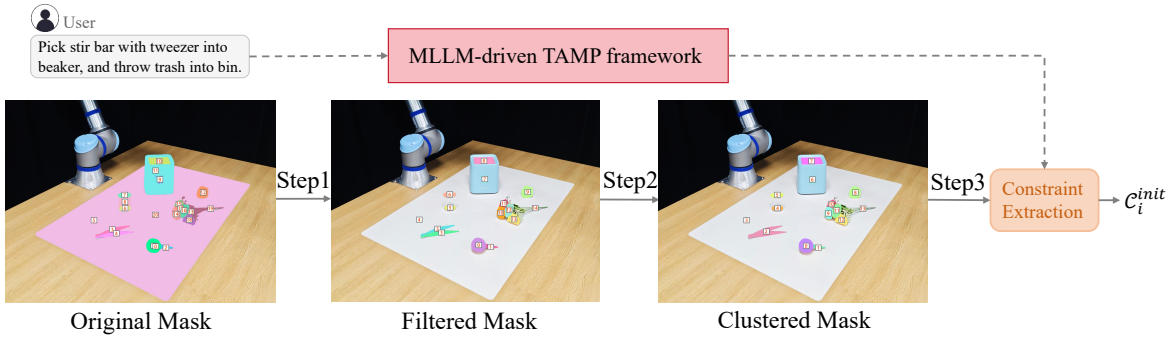


Fig. 2. 部分层级约束提取：原始掩码通过快速分割任何物体（FastSAM）生成，随后进行掩码过滤、部分层级语义一致性聚类和中心点注释，以在 MLLM 驱动的 TAMP 框架内提取部分层级空间约束。

### *E. 两阶段分层 3D 空间约束模型*

为实例化映射 $f:(L_i,O_i)\to\mathcal{C}_i^{\text{init}}$，我们利用 VFM 提示 MLLM 进行细粒度的视觉定位，从而自动构建由部件级约束提取和区域级约束细化组成的两阶段分层 3D 空间约束模型。

*1) 部件级约束提取：* 如图 2 所示，给定子任务指令 $L_i$ 和 RGB-D 观测 $O_i\in\mathbb{R}^{H\times W\times 4}$，我们应用预训练的图像分割模型 [38] 生成初始的掩码集 $M=\{m_1,m_2,\ldots,m_g\}$，其中 $m_j\in\{0,\ 1\}^{H\times W}$ 和 $\forall j\in\{1,\ldots,g\}$。这里，$H$ 和 $W$ 分别表示图像的高度和宽度。为了提取高质量和有区别性的部件级约束，我们设计了以下三步后处理策略。

步骤 1：掩码过滤。应用一个双重过滤流程，以促进有效且可区分的掩码，从而得到精炼的掩码集 $M_{\text{filtered}}$。

区域过滤：我们保留在有效区域范围内的掩码，并将掩码集定义为

$$M_{\text{area}}=\{m_j\in M\mid \alpha\cdot A_{\text{img}}\le \text{Area}(m_j)\le \beta\cdot A_{\text{img}}\} \quad (3)$$

，其中 $A_{\text{img}}$ 表示图像区域，$\alpha$ 和 $\beta$ 是阈值。

b) 结构独立性过滤：我们省略包含多于指定数量其他有效掩码的任何掩码，并将子集定义为 $M_{\text{contain}}$

$$M_{\text{contain}}=\{m_j\in M|N_{\text{sub}}\,(m_j)<3\} \quad (4)$$

，其中 $N_{\text{sub}}\,(m_j)$ 表示包含在遮罩 $m_j$ 内的遮罩数量，定义为

$$N_{\text{sub}}\,(m_j)=|\{m_k\in M|m_k\subset m_j\}| \quad (5)$$

最后，在经过双重过滤步骤后的有效掩码集被定义为

$$M_{\text{filtered}}=M_{\text{area}}\cap M_{\text{contain}} \quad (6)$$

步骤 2：部件级语义一致性聚类。对过滤后的掩码集合进行基于密度的聚类（DBSCAN）[39]，以进一步减少视觉歧义和冗余，在每个聚类内合并具有相同语义标签的掩码。聚类后的掩码集合表示为

$$M_{\text{cluster}}=\text{DBSCAN}(M_{\text{filtered}}) \quad (7)$$

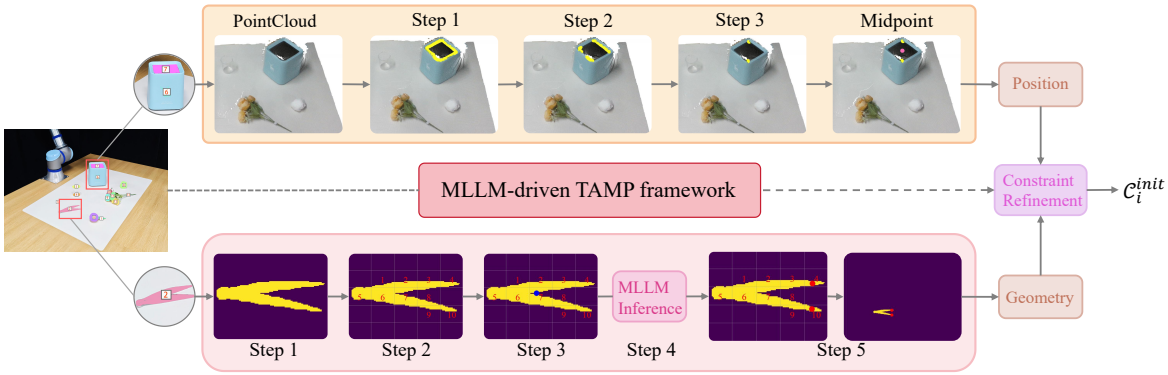


Fig. 3. 区域级约束优化。在基于 MLLM 的 TAMP 框架中，该模块包含两个策略：几何和位置约束优化。几何优化将镊子的抓取点从中心调整到两个端点，从而引入详细的几何先验信息。位置优化将垃圾桶的位置定位于对称中心的中点，从而提高空间精度。

其中 $M_{\text{cluster}} = \{\widetilde{m}_1, \widetilde{m}_2, \ldots, \widetilde{m}_e\}$ ，而 $e$ 是聚类掩码的数量。为了进一步提取有效的部分级约束，我们为每个聚类掩码 $\widetilde{m}_j \in M_{\text{cluster}}$ 定义质心坐标 $\boldsymbol{c}_j$ 如下

$$\boldsymbol{c}_j = \frac{1}{|\widetilde{m}_j|} \sum_{(x,y)\in\widetilde{m}_j} \begin{bmatrix} x \\ y \end{bmatrix} \tag{8}$$

，其中 $(x, y)$ 表示遮罩 $\widetilde{m}_j$ 中像素的二维坐标，$|\widetilde{m}_j|$ 是 $\widetilde{m}_j$ 中像素的数量。

步骤三：部件级约束提取。坐标集 $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_q\}$ 中的每个质心被顺序分配一个来自标签集 $\mathcal{Z} = \{0, 1, \ldots, q-1\}$ 的数字标签，以增强视觉语义模型的视觉基础。这些标签被叠加到原始 RGB 图像 $I_{\text{rgb}} \in \mathbb{R}^{H\times W\times 3}$ 上，形成一个显性视觉提示。与子任务指令 $L_i$ 一起，生成的多模态输入被送入 MLLM，MLLM 利用语义基础提取与指令相关的标签。最后，提取的标签通过深度信息 $I_{\text{depth}} \in \mathbb{R}^{H\times W}$ 投射到三维空间中，完成初始约束 $\mathcal{C}_i^{\text{init}}$ 的位置确定。

*2) 区域级约束精炼：* 如图 3 所示，为了增强空间约束的细粒度和适应性，我们提出了一种语义自适应区域级别的优化方法，该方法自动提取语义信息，并在两种优化策略之间进行选择：几何约束优化和位置约束优化。前者侧重于遮罩区域内的细粒度结构先验，而后者强调语义映射区域的空间布局。

几何约束细化：给定与 MLLM 提取的中心坐标 $\boldsymbol{c}_j \in C$ 相关联的遮罩 $\widetilde{m}_j$ ，通过几何约束细化来增强细粒度的视觉定位，并通过以下五个步骤进行。

步骤 1：掩膜归一化。通过裁剪每个掩膜的有效区域并对其进行统一缩放，来缓解不同掩膜 $\widetilde{m}_j$ 之间的比例差异，从而将所有掩膜一致地嵌入到固定分辨率的标准画布上。具体来说，我们首先提取出 $\widetilde{m}_j$ 中前景像素的坐标，其表示为

$$\mathcal{S}_j = \{(x, y) \mid \widetilde{m}_j(x, y) = 1\} \tag{9}$$

，其中 $\mathcal{S}_j$ 表示前景像素坐标的集合。随后，我们通过计算 $\mathcal{S}_j$ 的边界参数来确定 $\widetilde{m}_j$ 的有效区域，其表示为

$$\begin{aligned} t_j &= \min_{(x,y)\in\mathcal{S}_j} y, \quad b_j = \max_{(x,y)\in\mathcal{S}_j} y \\ l_j &= \min_{(x,y)\in\mathcal{S}_j} x, \quad r_j = \max_{(x,y)\in\mathcal{S}_j} x \end{aligned} \tag{10}$$

。

这里，$t_j$ ，$b_j$ ，$l_j$ 和 $r_j$ 分别表示上、下、左和右边界。然后，根据这些边界裁剪遮罩 $\widetilde{m}_j$ ，并调整到固定的分辨率 $H_r \times W_r$ 。裁剪后的遮罩 $\widetilde{m}_j^c = \widetilde{m}_j(t_j : b_j, l_j : r_j)$ 的宽度为 $W_c = r_j - l_j + 1$ ，高度为 $H_c = b_j - t_j + 1$ ，其中缩放因子定义为

$$\alpha_x = \frac{W_r}{W_c}, \quad \alpha_y = \frac{H_r}{H_c} \tag{11}$$

。

其中 $\alpha_x$ 和 $\alpha_y$ 分别表示水平方向和垂直方向的缩放因子。随后，我们应用最近邻插值将掩码调整到目标尺寸，如下所示

$$\widetilde{m}_j^r = \mathcal{R}(\widetilde{m}_j^c, \alpha_x, \alpha_y) \tag{12}$$

，其中 $\mathcal{R}$ 表示最近邻插值，$\widetilde{m}_j^r$ 是调整大小后的掩码。为了保持清晰的掩码边界并防止过度拟合图像边缘，调整的分辨率通常小于目标画布的分辨率。因此，调整大小后的掩码会进一步对齐到标准化画布。给定目标分辨率为 $H_t \times W_t$ ，目标掩码 $\widetilde{m}_j^t$ 定义为

$$\widetilde{m}_j^t = \widetilde{m}_j^r(y - \Delta_y, x - \Delta_x) \tag{13}$$

。

其中，$\Delta_x = W_t - W_r/2$ 和 $\Delta_y = H_t - H_r/2$ 分别表示水平和垂直偏移，确保掩码在图像中保持居中。

步骤 2：网格构建和标签嵌入。为了提高几何结构的视觉定位，在目标掩码图像上执行网格构建和标签嵌入。具体而言，第 $j$ 个掩码 $\widetilde{m}_j^t$ 被分成 $G_h \times G_w$ 个网格，其中每个网格的宽度 $\omega_c$ 和高度 $h_c$ 由目标分辨率 $H_t \times W_t$ 确定，公式如下：

$$\omega_c = \frac{W_t}{G_w}, \quad h_c = \frac{H_t}{G_h} \tag{14}$$

对于任意网格位于第 $k$ 行和第 $s$ 列，其中 $k \in \{0, \ldots, G_h - 1\}$ ，$s \in \{0, \ldots, G_w - 1\}$ ，该网格所覆盖的坐标区域表示为

$$\mathcal{G}_{k,s}^j = \left\{ (x, y) \;\middle|\; \begin{matrix} k \cdot h_c \le y < (k+1) \cdot h_c, \\ s \cdot w_c \le x < (s+1) \cdot w_c \end{matrix} \right\} \tag{15}$$

对于每个 $\mathcal{G}_{k,s}^j$ ，前景密度 $\rho_{k,s}^j$ 由

$$\rho_{k,s}^j = \frac{\mathcal{P}_{k,s}^j}{w_c \cdot h_c} \tag{16}$$

确定

其中 $\mathcal{P}_{k,s}^j = \sum_{(x,y)\in\mathcal{G}_{k,s}} [\widetilde{m}_j^t(x, y) = 1]$ 表示 $\mathcal{G}_{k,s}^j$ 中的前景像素集合，$|\mathcal{P}_{k,s}^j|$ 是像素的数量。当前景密度 $\rho_{k,s}^j$ 超过

分配阈值 $\tau$ 时，语义标签 $\ell_{k,s}^{j} \in \{0,\ 1, \ldots, N_{\text{valid}}-1\}$ 被分配给网格。我们定义映射 $\mathcal{U}:\ \mathcal{G}_{k,s}^{j} \rightarrow \ell_{k,s}^{j}$ 以将每个网格与语义标签关联，其中 $N_{valid}$ 表示满足阈值条件的网格总数。然后计算网格内的中心点为

$$\boldsymbol{c}_{j}^{k,s}=\frac{1}{|\mathcal{P}_{k,s}^{j}|}\sum_{(x,y)\in\mathcal{P}_{k,s}^{j}}\begin{bmatrix}x\\y\end{bmatrix} \tag{17}$$

步骤三：质心映射和标签关联。从部件级约束中提取的质心坐标 $\boldsymbol{c}_j \in C$ 被映射到目标掩码并与网格语义标签 $\ell_{k,s}^{j}$ 相关联，这些标签在目标掩码上作为文本提示叠加，以增强 MLLM 的细粒度几何理解。随后，我们将质心 $\boldsymbol{c}_j=\begin{bmatrix}x_j\\y_j\end{bmatrix}$ 映射到目标掩码，如下所示：

$$\bar{x}_j=\frac{x_j-l_j}{\alpha_x}+\Delta_x,\ \bar{y}_j=\frac{y_j-t_j}{\alpha_y}+\Delta_y \tag{18}$$

其中 $\bar{\boldsymbol{c}}_j=\begin{bmatrix}\bar{x}_j\\\bar{y}_j\end{bmatrix}$ 表示目标掩码中的坐标。为了建立与网格标签的对应关系，$\bar{\boldsymbol{c}}_j$ 通过方程 (15) 被分配到其对应的网格 $\mathcal{G}_{k,s}^{j}$，然后通过方程 (17) 投影到网格质心 $\boldsymbol{c}_j^{k,s}$，从而将其关联到语义标签 $\ell_{k,s}^{j}$。

步骤 4：通过 MLLM 进行细粒度语义推理。实现区域级别的约束优化，并基于目标几何推导新的标签 $\hat{\ell}_{k,s}^{j}$，涉及将原始 RGB 图像 $I_{\text{rgb}}$、原始掩码 $\widetilde{m}_j$ 和目标掩码 $\widetilde{m}_j$ 作为视觉提示输入到 MLLM 中。随后，数值标签 $z_j$ 和语义标签 $\ell_{k,s}^{j}$ 以文本提示的形式提供给 MLLM 进行推理。

步骤 5：区域级别约束提取。通过实例化映射 $\mathcal{U}:\mathcal{G}_{k,s}^{j}\rightarrow\ell_{k,s}^{j}$ 将标签 $\hat{\ell}_{k,s}^{j}$ 逆向映射到其对应的网格 $\mathcal{G}_{k,s}^{j}$ 来构建精炼的约束。然后，相关网格中心点 $\boldsymbol{c}_j^{k,s}$ 使用方程 (17) 和 (18) 投影到原始蒙版坐标作为 $\boldsymbol{c}_j$。最后，在深度信息 $I_{\text{depth}}\in\mathbb{R}^{H\times W}$ 下，2D 坐标被转换到 3D 空间，从而形成初始约束 $\mathcal{C}_i^{\text{init}}$。

位置约束细化：为解决具有明显边界特征的对象在语义定位中出现的位置差异问题，包括开顶结构，我们提出了一种利用边缘点和空间几何的三步细化方法。通过结合密度峰值估计和对称分析，该方法有效减轻了 MLLM 在位置建模中的局限性，提高了语义定位的实际可行性。

步骤 1: 沿着蒙版边缘提取三维坐标。关联蒙版 $\widetilde{m}_j^t$ 的边缘点集合 $\mathcal{M}_j$ 定义如下，以提取物体的边缘点：

$$\mathcal{M}_j=\{(x,y)\ \mid \widetilde{m}_j^t(x,y)=1\} \tag{19}$$

在以下条件下：

$$\exists(x\prime,y\prime)\in\mathcal{N}_4(x,y)\quad \text{s.t. } \widetilde{m}_j^t(x\prime,y\prime)=0$$

，其中 $\mathcal{N}_4(x,y)$ 表示像素坐标 $(x,y)$ 的 4 邻域，定义为

$$\mathcal{N}_4(x,y)=\{(x\pm1,y),(x,y\pm1)\} \tag{20}$$

随后，利用深度信息 $I_{\text{depth}}\in\mathbb{R}^{H\times W}$，映射得到 3D 边缘点集 $\mathcal{P}_j$，具体如下：

$$\mathcal{P}_j=\{\mathcal{F}_{3\text{D}}(x,y,I_{\text{depth}})\,|\,\forall(x,y)\in\mathcal{M}_j\} \tag{21}$$

步骤 2: 密度峰估计和最大高度滤波。采用核密度估计 (KDE) 对边缘点的 $z$ 轴分布进行建模，从而有助于消除离群值并提取显著的边缘特征。给定集合 $\{z_i\}_{i=1}^{N}$，其表示从由 $\widetilde{m}_j^t$ 表示的掩码中获得的 $N$ 边缘点的高度，概率密度函数近似为

$$\hat{f}(z)=\frac{1}{Nh}\sum_{i=1}^{N}K(\frac{z-z_i}{h}) \tag{22}$$

。

其中 $K(\cdot)$ 表示高斯核函数，$h$ 是带宽。密度峰值被定义为 $z^*=\arg\max_z\hat{f}(z)$。在高度范围 $|z_i-z^*|\le\delta$ 内的所有点被保留，结果得到过滤后的 3D 和 2D 点集 $\mathcal{P}_j^{\text{KDE}}\subset\mathcal{P}_j$ 和 $\mathcal{M}_j^{\text{KDE}}\subset\mathcal{M}_j$。随后，识别出 $\mathcal{P}_j^{\text{KDE}}$ 中沿 $z$ 轴的最大高度为 $z_{max}$。最后，进一步选择高度满足 $z_i\ge z_{max}-\eta$ 的点，从而得到最终的精炼点集 $\mathcal{P}_j^{\text{peak}}\subset\mathcal{P}_j^{\text{KDE}}$ 和 $\mathcal{M}_j^{\text{peak}}\subset\mathcal{M}_j^{\text{KDE}}$。

步骤 3：提取中心对称点对。通过识别二维空间中的中心对称点对，进一步集成对象位置和结构信息，这些点对被用来推导细化后的三维空间约束，记为 $\mathcal{C}_i^{\text{init}}$。给定中心点 $\boldsymbol{c}_j$，我们列举所有点对 $\boldsymbol{u}_a,\boldsymbol{u}_b\in\mathcal{M}_j^{\text{peak}}$，并识别中点最接近 $\boldsymbol{c}_j$ 的点对，该点对定义为

$$(\boldsymbol{u}_a^*,\boldsymbol{u}_b^*)=\underset{\boldsymbol{u}_a,\boldsymbol{u}_b\in\mathcal{M}_j^{\text{peak}}}{\arg\min}\left\|\frac{\boldsymbol{u}_a+\boldsymbol{u}_b}{2}-\boldsymbol{c}_j\right\|_2 \tag{23}$$

，其中 $(\boldsymbol{u}_a^*,\boldsymbol{u}_b^*)$ 表示过滤后的点对，具有 3D 坐标 $\boldsymbol{p}_a,\boldsymbol{p}_b\in\mathcal{P}_j^{\text{peak}}$。最后，精细的 3D 空间约束 $\mathcal{C}_i^{\text{init}}$ 表示为

$$\mathcal{C}_i^{\text{init}}=\frac{1}{2}(\boldsymbol{p}_a+\boldsymbol{p}_b) \tag{24}$$

### F. 实时闭环控制策略

为了以实时性能解决方程 (2) 中的约束优化问题，使用了 Isaac Gym 仿真平台上的模型预测路径积分（MPPI）控制算法。MPPI 是一种为动态和不确定环境中的机器人控制设计的随机优化方法。基本上，该算法从控制输入空间中采样多个候选控制序列，基于模拟状态轨迹评估成本函数，并通过重要加权平均更新控制策略。Isaac Gym 启用的 GPU 加速并行仿真显著增强了采样效率，从而支持在动态环境中的实时执行。有关于 MPPI 的详细解释，请参考 [40], [41]。在这项工作中，我们推导了解决方程 (2) 中呈现的约束优化问题的具体公式。

首先，机器人的当前关节状态记为 $\boldsymbol{x}(t)=\begin{bmatrix}\boldsymbol{\theta}\\\dot{\boldsymbol{\theta}}\end{bmatrix}\in\mathbb{R}^{2n}$，包括关节位置和速度。使用速度控制输入 $\boldsymbol{v}(t)\in\mathbb{R}^n$，其中 $n$ 表示自由度，非线性状态转移函数可以表示为

$$\boldsymbol{x}(t+1)=S(\boldsymbol{x}(t),\boldsymbol{v}(t)),\quad \boldsymbol{v}(t)\sim\mathcal{N}(\mu(t),\Sigma) \tag{25}$$

中，$\mathcal{N}(\cdot)$ 表示高斯采样，$\mu(t)$ 是采样均值，$\Sigma$ 是协方差矩阵，$S$ 代表前向运动学和物理模型。在每个时间步，MPPI 采样 $K$ 个不同的控制序列 $\{\boldsymbol{V}_k\}_{k=1}^{K}$，其中每个序列由时间范围内的速度输入组成，表示为 $\boldsymbol{V}_k=[\boldsymbol{v}_k(0),\boldsymbol{v}_k(1),\ldots,\boldsymbol{v}_k(T-1)]^\top$。随后，每个状态轨迹通过将初始状态 $\boldsymbol{x}(0)$ 与前向模型结合推导为

$$\mathcal{X}_k=[\boldsymbol{x}_k(0),\boldsymbol{x}_k(1),\ldots,\boldsymbol{x}_k(T)]^\top \tag{26}$$

。

其中 $\boldsymbol{x}_k(t+1)=S(\boldsymbol{x}_k(t),\boldsymbol{v}_k(t))$。为了进一步解决这个问题，我们引入了前向运动学映射 $\varphi:\ \mathbb{R}^{2n}\rightarrow SE(3)$，它将关节空间状态映射到末端执行器姿态，表示为 $\boldsymbol{T}_e^k(t)=\varphi(\boldsymbol{x}_k(t))$。此时，$\boldsymbol{T}_e^k(t)\in SE(3)$ 表示在时间 $t$ 的 $k$ -th 轨

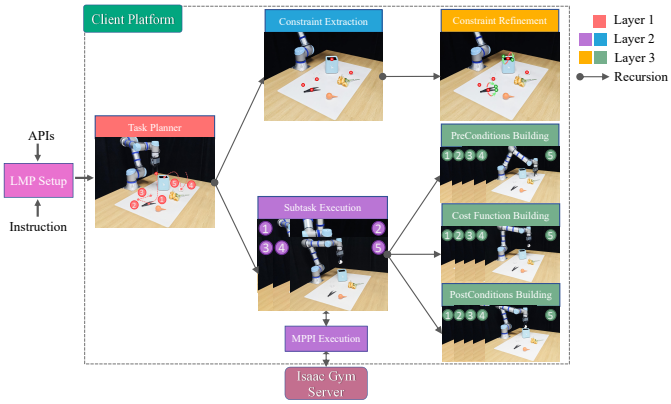


Fig. 4. 基于 MLLM 的用于 TAMP 的自动化建模和通信框架。

迹的末端执行器姿态。对于每一个轨迹 $\mathcal{X}_k$ ，总的代价是目标函数 $\mathcal{J}(\cdot)$ 在每个时间步的累加和，定义为

$$C_k = \sum_{t=0}^{T-1} \mathcal{J}\left(\boldsymbol{T}_e^k(t), \mathcal{C}_i\right) \tag{27}$$

。

为了便于分析，我们定义位置和方向误差如下：

$$\mathcal{D}_p = d_p\left(P(\boldsymbol{T}_e^k(t)), P(\mathcal{C}_i)\right) \tag{28}$$

$$\mathcal{D}_r = d_r\left(R(\boldsymbol{T}_e^k(t)), R(\mathcal{C}_i)\right) \tag{29}$$

，其中 $P(\cdot)$ 和 $R(\cdot)$ 表示从空间姿态中提取的位置和四元数分量。误差计算为 $d_p(\boldsymbol{p}_1, \boldsymbol{p}_2) = \|\boldsymbol{p}_1 - \boldsymbol{p}_2\|_2$ 和 $d_r(\boldsymbol{q}_1, \boldsymbol{q}_2) = 2\arccos(\max(-1, \min(1, \langle \boldsymbol{q}_1, \boldsymbol{q}_2 \rangle)))$ 。随后，目标函数定义为

$$\mathcal{J}\left(\boldsymbol{T}_e^k(t), \mathcal{C}_i\right) = \lambda_p \cdot \mathcal{D}_p + \lambda_r \cdot \mathcal{D}_r + \lambda_c \cdot \mathcal{D}_c \tag{30}$$

。

其中 $\lambda_p$ 和 $\lambda_r$ 分别为位置和方向代价的权重，$\lambda_c$ 表示碰撞代价 $\mathcal{D}_c$ 的权重，该代价是基于 ESDF 表示形式制定的，如 [3] 所述。在这项工作中，目标函数受到如方程 (2) 中所指定的前提条件和后条件的限制。前提条件通常确保末端执行器在执行阶段前位于指定区域内，而后条件则评估末端执行器是否已成功达到所需姿态，定义为

$$\epsilon_{\text{pre}}\left(\boldsymbol{T}_e^k(t),\ \mathcal{C}_i\right) = \mathcal{D}_p(k, t, i) \tag{31}$$

$$\epsilon_{\text{post}}\left(\boldsymbol{T}_e^k(t), \mathcal{C}_i\right) = \mathcal{D}_p(k, t, i) + \mathcal{D}_r(k, t, i) \tag{32}$$

在评估总成本 $C_k$ 之后，MPPI 为每个采样的轨迹分配权重如下：

$$\omega_k = \frac{1}{\eta} \exp\left(-\frac{1}{\beta}\left(C_k - \rho\right)\right), \quad \sum_{k=1}^{K} \omega_k = 1 \tag{33}$$

，其中 $\beta$ 是控制样本权重方差的温度参数，$\rho = \min_k C_k$ 是稳定性偏移。近似最优控制序列 $\boldsymbol{U}^* = \left[\boldsymbol{u}_0^*, \boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_{T-1}^*\right]$ 表示为

$$\boldsymbol{U}^* = \sum_{k=1}^{K} \omega_k \cdot \boldsymbol{V}_k \tag{34}$$

最后，序列 $\boldsymbol{U}^*$ 的第一个控制输入 $\boldsymbol{u}_0^*$ 被用作系统的速度命令，优化过程是迭代进行的。

### G. MLLM 驱动的 TAMP 自动建模

如图 4 所示，语言模型程序（LMP）框架基于分层递归架构 [8] 构建，支持结构化调用和多层提示嵌套。基于该框架，我们在 ReSem3D 中设计并封装了七个 LMP 模块，分别对应任务规划器、约束提取、约束细化、子任务执行、前提条件构建、成本函数构建和后置条件构建。每个模块实现了独立的提示结构，并通过标准化 API 与系统组件接口，实现了任务和运动规划的自动化建模。

在 LMP 设置阶段，LMP 层之间的依赖关系建立起来，同时根据自然语言指令配置对 API 接口的映射。随后，任务规划器执行全局调度和意图解析，协同约束提取和子任务执行来建模空间约束并执行任务。最后，考虑到在空间约束建模中对语义理解的依赖，引入了约束细化，通过递归参数更新反馈给任务规划器来实现两阶段建模。

在子任务执行阶段，每个任务被分解为一系列受约束的动作执行单元。具体来说，动作执行的启动条件、优化目标和终止标准分别通过前置条件建立、成本函数建立和后置条件建立来制定。随后，每个动作执行单元激活 MPPI 执行，建立客户端与 Isaac Gym 服务器之间的通信循环。客户端传输实时关节状态和感知融合成本函数，而服务器则执行 GPU 加速的 MPPI 采样和优化，以合成连续的关节空间速度指令，这些指令被传回客户端，从而实现逐步的闭环控制。

在任务执行过程中，前提条件和后置条件会被监控。如果前提条件被违反，系统会回溯并重新启动先前的子任务，而后置条件的满足将触发向下一个子任务的进展。最终，ReSem3D 建立了一个以语义理解为中心的闭环 TAMP 框架，该框架由层次递归引导，并基于约束反馈。

## II. 实验

在本节中，我们首先详细介绍实验设置。随后，我们评估 ReSem3D 在模拟环境中的跨平台泛化能力。然后，我们展示了其在语义多样化的真实环境中的操作泛化能力。此外，我们分析了 ReSem3D 的核心机制，展示了其相对于基线方法的优势。最后，我们识别出导致执行失败的关键因素。

### A. 实验设置

硬件配置。如图 6 所示，我们的真实世界实验平台是围绕 6 自由度的 Universal Robots UR5e 协作机器人手臂构建的，配备了 PGI-140-80 两指平行抓手和 Intel RealSense D435i RGB-D 摄像头。对于仿真，我们采用斯坦福大学开发的 Omnigibson [42] 平台，并利用 UR5e 和 7 自由度的 Franka Emika Panda 协作手臂。

任务和指标。如图 5 所示，我们设计了 12 个实际任务（6 个化学实验室场景和 6 个家庭场景）来评估在静态条件和动态干扰下的泛化性和稳定性。在仿真中，我们执行 10 个任务（6 个化学实验室场景和 4 个家庭场景），以评估在不同机器人平台上的静态条件下的泛化性。此外，我们设计了三组模拟实验和四组真实环境实验来评估在视觉定位中的有效性。所有任务都在 10 次试验中进行评估，并报告了成功率。

基线方法。我们比较了三个具有代表性的框架：VoxPoser [4] 、ReKep [3] 和 CoPa [34] 。VoxPoser 集成了大型语言和视觉模型，以生成机器人可解释的三维价值地图，实现无预定义运动原语的闭环轨迹合成。它利用 GPT-4 和视觉前端来实现任务对象的语义定位，支持在不同任务中进

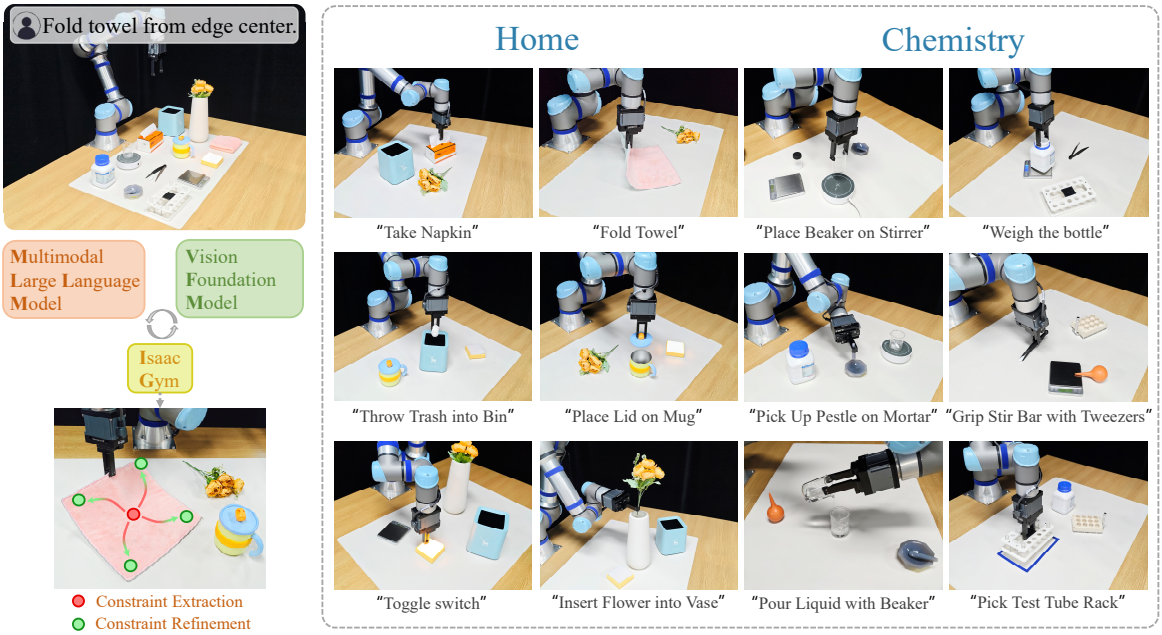


Fig. 5. ReSem3D 是一个面向语义多样化环境的统一机器人操作框架。它利用 MLLMs 和 VFMs 之间的协同作用构建语义驱动的两阶段层次化三维空间约束，并将其映射到关节空间中的实时优化目标，以实现闭环感知-动作控制。

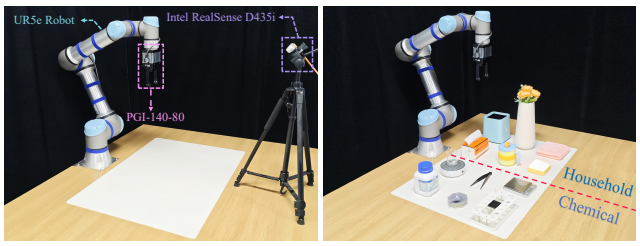


Fig. 6. 硬件平台配置。


Fig. 7. Rekep 和 ReSem3D（我们的方法）在约束建模中的比较。每种方法有两列，分别展示 RGB 和点云的可视化效果，两行分别对应 150 % 和 40 % 对象尺度。紫色、绿色和红色的数字标签分别表示 MLLM 在烤面包机、花瓶和杵上提取的初始 2D 约束，而彩色箭头表示几何和位置 3D 约束的细化。

行灵活操作。ReKep 融合了 VLM 和 MLLM 以推断结构化任务的语义关键点约束，并通过分层优化实时解决末端执行器轨迹。CoPa 利用基础视觉语言模型提出了一个两阶段的任务执行过程，在第一阶段通过粗到细的视觉定位进行任务导向的抓取，在第二阶段识别与任务相关的空间和几何约束以供机器人规划。所有三个框架都深度整合了大型语言和视觉模型，利用语义驱动的视觉定位和控制优化来实现高效且稳健的机器人操作。

VFM 和 MLLM。我们使用来自 OpenAI 的可调用 MLLM——GPT-4o [4]，并将其与视觉基础模型 FastSAM [38] 结合，以实现高效率的多模态推理（提示设计可在 ReSem3D.github.io 时获得）。在动态干扰阶段，我们结合 Google DeepMind 的 TAPNext 模型 [43] 以及实时 RGB 帧输入，以大约 20 Hz 的速度实现在线点追踪。这有助于实时推断动态环境中不断变化的空间约束，从而增强任务执行的稳定性和鲁棒性。

### *B. ReSem3D 在模拟环境中的评估*

如表 I 所总结，我们系统地评估了 ReSem3D 在不同机器人平台和语义多样环境中的仿真泛化能力，包括静态条件下的 4 个家庭任务和 6 个化学实验室任务。在切换开关任务中，位置约束细化中的密度峰值被最大高度取代，以便更直接地提取对称点对。结果表明，ReSem3D 在零次实验条件下表现出较强的任务适应性和跨平台泛化能力，在家庭和化学实验室场景中分别实现了 70 % 和 65 % 的执行成功率，始终优于基线方法。这种性能可归因于提出的两阶段三维空间约束模型。具体而言，当物体遮罩的三维中心点在空间上与可行抓取点（例如，餐巾纸、试剂和搅拌器）对齐时，VoxPoser 实现了出色的性能。然而，当这种对齐不存在时，成功率显著下降。尽管 Rekep 始终在任务中生成语义上有意义的关键点，但由于预测关键点与任务相关关键点的不匹配其效果有限。值得注意的是，由于 Omnigibson 中可用透容器的深度信息，这些物体在仿真中被建模为不透明物体（例如，塑料瓶）。尽管与现实世界的视觉特性存在差异，这种方案保持了一致的约束生成和操作策略，确保了评估的严谨性。

对于进一步的分析，我们关注 ReKep 在开放结构容器和双端语义对象中采用的约束建模。如图 7 所示，这类对象的语义关键点通常由 MLLM 提取，要么计算多个开

TABLE I
模拟中静态任务的成功率

| Household Scenario | | Success Rate | | |
|---|---|---|---|---|
| Task | Robot Platform | Voxposer | Rekep | ReSem3D (Ours) |
| Toggle switch | UR5e | 0/10 | 3/10 | 9/10 |
| Fold Towel | Franka Panda | 0/10 | 3/10 | 5/10 |
| Take Napkin | Franka Panda | 8/10 | 6/10 | 7/10 |
| Cut Cake with Knife | Franka Panda | 0/10 | 5/10 | 7/10 |
| Total | | 20.0 % | 42.5 % | 70.0 % |
| Chemical Lab Scenario | | | | |
| Pour reagent into dish | UR5e | 7/10 | 0/10 | 8/10 |
| Weigh the Reagent Bottle | UR5e | 8/10 | 1/10 | 8/10 |
| Pick Up Pestle on Mortar | UR5e | 0/10 | 3/10 | 6/10 |
| Place Reagent Bottle on Stirrer | UR5e | 7/10 | 2/10 | 8/10 |
| Place Funnel on Iron Ring | Franka Panda | 0/10 | 0/10 | 4/10 |
| Grip Stir Bar with Tweezers | Franka Panda | 0/10 | 0/10 | 5/10 |
| Total | | 36.6 % | 10.0 % | 65.0 % |

TABLE II
仿真中的视觉定位成功率

| Task | Scale | ReKep | ReSem3D (Ours) |
|---|---|---|---|
| Insert Flower into Vase | 150 % | 3/10 | 7/10 |
| | 70 % | 1/10 | 6/10 |
| | 40 % | 0/10 | 4/10 |
| Insert Bread into Toaster | 150 % | 4/10 | 7/10 |
| | 70 % | 0/10 | 9/10 |
| | 40 % | 0/10 | 9/10 |
| Pick Up Pestle on Mortar | 150 % | 8/10 | 8/10 |
| | 70 % | 7/10 | 9/10 |
| | 40 % | 1/10 | 9/10 |

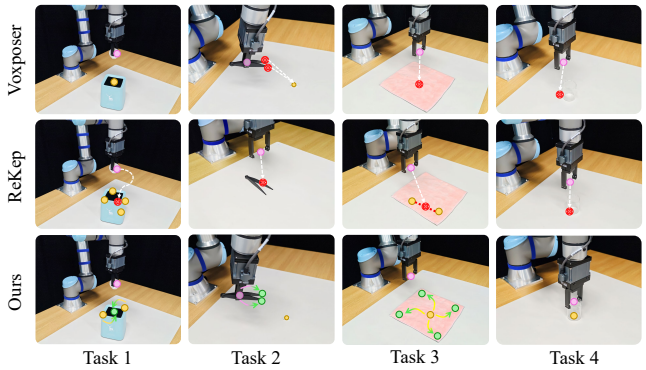


Fig. 8. 基线与 ReSem3D（我们的方法）在约束建模和闭环执行中的比较。任务 1 到 4 对应于对箱子、镊子、毛巾和烧杯的操作。粉红色标记表示末端执行器的位置或抓取的约束，黄色标记表示提取的约束，绿色标记表示精炼后的约束，红色标记突出显示不可执行的约束，五彩的箭头显示约束的精炼。

启边缘点的质心（例如花瓶和烤面包机），要么识别功能上不同的端点（例如杵的狭窄一端）。然而，ReKep 中的约束生成对关键点间距较为敏感，并且在很大程度上依赖于对象的尺度：尺度的增加往往会产生冗余约束，干扰准确识别，而尺度的减小通常会导致约束合并，从而导致语义关键点的丢失。相比之下，ReSem3D 在尺度变化下表现出显著更高的稳定性，对尺度扰动的鲁棒性增强。这一优势在表 II 中得到了证实，其中 ReSem3D 在按 150 %，70 %，和 40 % 缩放的花瓶、烤面包机和杵的可视化定位成功率上持续优于基线方法。这些结果表明，该方法有效缓解了开放结构容器和双端语义对象操作中尺度变化的影响，从而增强了零样本泛化。

### C. *ReSem3D 在真实世界环境中的评估*

在本节中，我们通过一组语义多样的操作任务系统地评估 ReSem3D 在现实环境中的泛化能力，这些任务包括 6 个家庭场景和 6 个化学实验室场景。如表 **??** 所示，M1、M2 和 M3 分别表示部件级约束提取、几何约束细化和位置约束细化。Static 表示没有外部干扰的环境，而 “Dist.” 则指在任务执行过程中通过手动移动物体产生的动态干扰。与基线相比，ReSem3D 在任务执行表现上取得了显著成绩，在家庭场景中实现 58.3 %（静态）和 43.3 %（动态），在化学实验室场景中实现 60 %（静态）和 45 %（动态），显示了卓越的零次泛化和对动态干扰的增强鲁棒性。下面提供了全面的分析。

具体来说，家庭场景的特点是丰富的语义结构和适度的约束精度，这使得基线能够在约束建模和物体定位中保留相当的泛化能力。Voxposer 在任务中表现出色，其中物体遮罩的 3D 中心点（例如餐巾、垃圾、垃圾桶）可以直接抓取，这与其模拟性能一致。对于其他任务，当定位结果不符合抓取要求时，就会发生失败。此外，Rekep 能够有效地提取丰富的语义关键点以供 MLLM 解释。然而，复杂的语义结构和细粒度的约束（例如定位毛巾的角或检测升起的开关边缘）超出了其解释能力。相反，化学实验室场景的特点是稀疏的语义结构和透明材料的普遍存在，这削弱了深度感知并需要更高的空间精度。这些条件进一步限制了基线泛化（例如，定位烧杯中心，识别杵的细尖），导致成功率持续下降。基于这些见解，ReSem3D 引入了一个两阶段 3D 空间约束模型，整合语义理解和空间推理，实现了任务成功率 1.7 %（静态）和 3.3 %（分布）的提高，并在语义多样化环境中保持强大的鲁棒性。

为了定性分析 ReSem3D 的优势，我们关注其在语义约束建模和闭环执行效率方面的表现，如图 8 所示。可以观察到，VoxPoser 的语义约束主要集中在物体的粗略中心，缺乏细粒度的几何意识和可操作性。此外，在化学场景中透明物体（例如，任务 4 中的烧杯）的点云缺失，对精确的 3D 中心定位提出了重大挑战。此外，5Hz 的 3D 值图更新

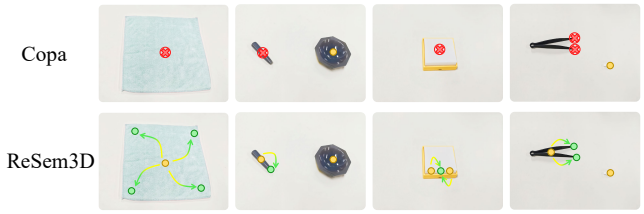


Fig. 9. 视觉定位中 Copa 与 ReSem3D（我们的）的比较。

率限制了闭环轨迹规划的响应性。类似地，虽然 ReKep 构建了语义导向的约束，但在约束生成的准确性（例如，烧杯、桶和镊子的中心）和可调性（例如，毛巾的角、镊子的尖端）方面有限。其性能依赖于 MLLM 生成约束和高层次推理的有效性。此外，10Hz 的闭环优化受逆运动学求解效率的限制，导致实时响应不稳定。

此外，我们还与 CoPa 进行比较，它同样采用粗到细的视觉定位框架。如表格 **??** 所示，我们评估来自表格 **??** 中的四个代表性任务，以检查视觉定位是否足以成功执行任务。实验结果表明，ReSem3D 实现了 60% 的成功率，显著优于 CoPa。图 9 中的详细定性分析表明，尽管 CoPa 执行了分层定位，但其细粒度的定位精度仅限于部件级别，对于需要更细致空间推理的任务来说是不够的。

鉴于实现细粒度视觉定位和维持实时闭环控制的挑战，ReSem3D 通过三个策略实现更好的性能：（1）可精炼的两阶段 3D 空间约束模型。语义驱动的 3D 空间约束在两个阶段生成，能够在语义丰富和稀疏的环境中实现自动细化和适应。（2）关节空间中的实时闭环优化。语义驱动的 3D 空间约束被表述为关节空间的代价函数，并使用一个运行在 15 Hz 以上的实时约束求解器进行优化，从而在动态环境中实现反应性闭环控制。（3）MLLM 驱动的自动 TAMP 建模。ReSem3D 自动根据自然语言指令确定适合的 3D 空间约束建模粒度，并为实时约束优化器生成代价函数。此外，它能够分解多阶段任务并生成前置条件和后置条件，从而在执行过程中实现时间管理和动态回溯。

最终，ReSem3D 的稳定运行需要跨功能模块的有效协调。为了系统地识别故障原因，我们通过基于表格 **??** 计算故障率对每个子模块进行独立验证。具体而言，由于 VFM 模块缺乏语义指导，在 7 % 的情况下，它可能无法提供与任务相关的语义定位，从而生成不相关的候选项。约束提取和优化中的推理失败分别构成 16 % 和 28 % 的失败，源于跨模态语义定位不足和空间推理有限。由于多目标的复杂性，约束优化模块在 11 % 的情况下，由于显著的姿态变化而导致收敛缓慢。TAPNext 模块在 38 % 的情况下，由于遮挡或不准确的点云映射表现出较低的跟踪准确性，从而打破了感知-行动循环。

为了解决这些挑战，未来的工作将探讨将基于 MLLM 的自校正策略、全局优化算法和多视角融合方法进行整合，以进一步增强该框架的鲁棒性和适应性。

## III. 结论

在这项工作中，我们提出了 ReSem3D，一个用于语义多样环境的统一机器人操控框架。通过集成 MLLMs 和 VFMs，该框架自动构建语义基础的两阶段三维空间约束。这些约束嵌入到分层 TAMP 模型中，并被表述为联合空间成本函数，从而实现实时稳定的闭环控制。该方法促进了从语义丰富到稀疏环境的分层约束建模，支持多阶段任务执行和具备条件意识的动态回溯。大量的模拟和现实世界实验表明，ReSem3D 在零样本操作任务中，在语义多样的环境和不同的机器人平台上表现优于现有的基准。

## References

[1] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao, "Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v," *arXiv preprint arXiv:2310.11441*, 2023.

[2] J. Yang, R. Tan, Q. Wu, R. Zheng, B. Peng, Y. Liang, Y. Gu, M. Cai, S. Ye, J. Jang *et al.*, "Magma: A foundation model for multimodal ai agents," in *Proceedings of CVPR*, 2025, pp. 14 203–14 214.

[3] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.

[4] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," in *Proceedings of CoRL*. PMLR, 2023, pp. 540–562.

[5] Y. Zhang, C. Pezzato, E. Trevisan, C. Salmi, C. H. Corbato, and J. Alonso-Mora, "Multi-modal mppi and active inference for reactive task and motion planning," *IEEE Rob. Autom. Lett.*, 2024.

[6] C. Pezzato, C. Salmi, E. Trevisan, M. Spahn, J. Alonso-Mora, and C. H. Corbato, "Sampling-based model predictive control leveraging parallelizable physics simulations," *IEEE Rob. Autom. Lett.*, 2025.

[7] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[8] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *Proceedings of ICRA*. IEEE, 2023, pp. 9493–9500.

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of ICML*. PmLR, 2020, pp. 1597–1607.

[10] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of CVPR*, 2020, pp. 9729–9738.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 2002.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of ICLR*, 2020.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of CVPR*, 2014, pp. 580–587.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of CVPR*, 2016, pp. 779–788.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of CVPR*, 2015, pp. 3431–3440.

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[18] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of ICML*. PMLR, 2021, pp. 10 347–10 357.

[19] X. Chen, Z. Zhao, Y. Zhang, M. Duan, D. Qi, and H. Zhao, "Focalclick: Towards practical interactive image segmentation," in *Proceedings of CVPR*, 2022, pp. 1300–1309.

[20] X. Zou, J. Yang, H. Zhang, F. Li, L. Li, J. Wang, L. Wang, J. Gao, and Y. J. Lee, "Segment everything everywhere all at once," *Proceedings of NeurIPS*, vol. 36, pp. 19 769–19 782, 2023.

[21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of ICML*. PMLR, 2021, pp. 8748–8763.

[22] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *Proceedings of ECCV*. Springer, 2024, pp. 38–55.

[23] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[24] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Proceedings of NeurIPS*, vol. 36, pp. 34 892–34 916, 2023.

[25] L. P. Kaelbling and T. Lozano-Pérez, "Integrated task and motion planning in belief space," *Int. J. Rob. Res.*, vol. 32, no. 9-10, pp. 1194–1227, 2013.

[26] M. Kopicki, S. Zurek, R. Stolkin, T. Mörwald, and J. Wyatt, "Learning to predict how rigid objects behave under simple manipulation," in *Proceedings of ICRA*. IEEE, 2011, pp. 5722–5729.

[27] F.-J. Chu, R. Xu, and P. A. Vela, "Learning affordance segmentation for real-world robotic manipulation via synthetic images," *IEEE Rob. Autom. Lett.*, vol. 4, no. 2, pp. 1140–1147, 2019.

[28] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Trans. Rob.*, vol. 39, no. 5, pp. 3929–3945, 2023.

[29] D. Pavlichenko and S. Behnke, "Dexterous pre-grasp manipulation for human-like functional categorical grasping: Deep reinforcement learning and grasp representations," *IEEE Trans. Autom. Sci. Eng.*, 2025.

[30] L. Zheng, C. Wang, Y. Sun, E. Dasgupta, H. Chen, A. Leonardis, W. Zhang, and H. J. Chang, "Hs-pose: Hybrid scope feature extraction for category-level object pose estimation," in *Proceedings of CVPR*, 2023, pp. 17 163–17 173.

[31] T. Schmidt, R. Newcombe, and D. Fox, "Self-supervised visual descriptor learning for dense correspondence," *IEEE Rob. Autom. Lett.*, vol. 2, no. 2, pp. 420–427, 2016.

[32] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, "Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping," *IEEE Rob. Autom. Lett.*, vol. 8, no. 11, pp. 7551–7558, 2023.

[33] Z. Li, J. Liu, Z. Li, Z. Dong, T. Teng, Y. Ou, D. Caldwell, and F. Chen, "Language-guided dexterous functional grasping by llm generated grasp functionality and synergy for humanoid manipulation," *IEEE Trans. Autom. Sci. Eng.*, 2025.

[34] H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao, "Copa: General robotic manipulation through spatial constraints of parts with foundation models," in *Proceedings of IROS*. IEEE, 2024, pp. 9488–9495.

[35] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "Incremental task and motion planning: A constraint-based approach." in *Proceedings of RSS*, vol. 12. Ann Arbor, MI, USA, 2016, p. 00052.

[36] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Sampling-based methods for factored task and motion planning," *Int. J. Rob. Res.*, vol. 37, no. 13-14, pp. 1796–1825, 2018.

[37] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *Proceedings of ICRA*. IEEE, 2023, pp. 11 523–11 530.

[38] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[39] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of KDD*, vol. 96, no. 34, 1996, pp. 226–231.

[40] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *J. Guid. Control Dyn.*, vol. 40, no. 2, pp. 344–357, 2017.

[41] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Trans. Rob.*, vol. 34, no. 6, pp. 1603–1622.

[42] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun *et al.*, "Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation," in *Proceedings of CoRL*. PMLR, 2023, pp. 80–93.

[43] A. Zholus, C. Doersch, Y. Yang, S. Koppula, V. Patraucean, X. O. He, I. Rocco, M. S. Sajjadi, S. Chandar, and R. Goroshin, "Tapnext: Tracking any point (tap) as next token prediction," *arXiv preprint arXiv:2504.05579*, 2025.