GSR：面向具身操作的结构化推理学习

Kewei Hu#，Michael Zhang#，Wei Ying、Tianhao Liu、Guoqiang Hao、
Zimeng Li、Wanchan Yu、Jiajian Jing、Fangwen Chen、Hanwen Kang*
https://klmmotion.github.io/gsr.github.io/

arXiv:2602.01693v1 [cs.RO] 2 Feb 2026



图 1：本文介绍了 GSR 和 Manip-Cognition 数据集。GSR 通过理性推理能力推动机器人操作的发展，而 Manip-Cognition 提供了 160 万个样本，涵盖多种原子操作技能中的场景落地、目标解释和动作规划。

摘要——**尽管取得了快速进展，具身智能体在长时程操作任务中仍面临挑战，因为这类任务要求保持空间一致性、因果依赖和目标约束。现有方法的一个关键局限在于，任务推理被隐式地嵌入高维潜在表示中，因此难以将任务结构与感知变化区分开来。我们提出基于场景图的推理（GSR），这是一种结构化推理范式，它将世界状态演化显式建模为语义落地场景图上的状态转移。GSR 逐步推理物体状态和空间关系，而非直接将感知映射为动作，从而能够在物理落地空间中显式推理动作前提条件、后果及目标满足情况。为支持此类推理的学习，我们构建了 Manip-Cognition-1.6M，这是一个大规模数据集，可联合监督世界理解、动作规划和目标解释。在 RLBench、LIBERO、GSR-benchmark 以及真实世界机器人任务上的大量评估表明，与基于提示的基线方法相比，GSR 显著提升了零样本泛化能力和长时程任务完成能力。这些结果凸显了显式世界状态表示作为可扩展具身推理关键归纳偏置的重要性。**

I. 引言

EMBODIED 学习已成为构建整合感知、推理、规划与执行的机器人系统的重要方法。尽管取得了显著进展，现有方法在适应现实世界变化以及在多样化语义下完成长时程任务方面仍面临困难 [1]。这些挑战与模型如何解读观测、理解语言指令，以及将高层意图与低层动作联系起来密切相关。许多现有方法并未明确区分高层任务推理。因此，任务逻辑往往与视觉变化纠缠在一起，使模型对环境变化过于敏感。当智能体需要超越训练数据的范围运行，以解决需要对长序列进行规划的任务时，这一局限便显而易见 [2]。

具身智能体依赖两项核心能力：具身推理和动作控制。具身推理根据感知输入 [3] 推断意图导向的序列化动作（例如“倾斜冰箱”或“把杯子放在盘子上”）。这要求智能体感知环境、理解人类指令所表达的意图，并生成顺序正确的动作以完成任务。世界表征方式在这一过程中发挥着关键作用。世界表征受多种因素影响，包括物体外观、空间关系和任务语义。即使在同一场景中，不同目标也可能产生完全不同的动作序列；反过来，在目标相同的情况下，场景配置的变化也可能要求采取不同的行为。然而，当前的隐式神经网络模型，例如卷积神经网络

（CNN）和视觉 Transformer（ViT），难以提取有效推理所需的此类高层语义和关系信息。因此，它们跨场景和任务泛化的能力仍然有限 [4]。因此，迁移到新任务或适应哪怕细微的环境变化，通常都需要大量额外的数据收集和重新训练。经典的任务与运动规划（TAMP）框架为此类规划能力提供了结构化解决方案，但通常依赖预定义的符号表征（例如 PDDL 和 BDDL）。对于变化多端、开放且复杂的环境，这些方法往往难以扩展 [5]。受这些局限的启发，本研究探讨：**具身智能体能否借助结构化中间表征，提升其在一般操作任务中的推理能力?**

为此，我们提出**接地场景图推理（GSR）**，这是一种具身推理框架，其原则是智能体应基于抽象的世界表征进行规划，而非依赖原始视觉信息。GSR 利用语义落地场景图，从观测中提取稳定的因果结构，并明确区分高层概念推理与低层动作执行。这种设计赋予系统持久且可迁移至不同任务的能力，使其能够灵活组合一系列动作，并在不同任务间实现稳健适应。为训练 GSR，我们构建了 **Manip-Cognition-1.6M**，该数据集针对多种操作任务，对世界理解、意图解读和动作规划进行联合监督。

本文其余部分的结构如下：第III 节介绍研究方法；第IV 节介绍实验设置与结果；第V节总结全文。

II. 相关工作

A. 面向机器人学习的世界表征

当前的具身学习模型通常采用端到端或分层模块化设计。端到端方法，包括 RT-2 [6]，OpenVLA [7]，和 π_0 [1]，利用大规模数据将原始视觉观测直接映射为动作，展现出强大的规模扩展能力。然而，处理高维感官输入会引入表征瓶颈 [8]，，使与任务无关的视觉变化与潜在的物理动力学纠缠在一起。缺乏显式的状态抽象，会使学习偏向由外观驱动的相关性，而非因果结构，导致模型在域外场景和长时程操作任务中的泛化能力下降 [9]。为缓解这些问题，RT-H 等分层视觉语言动作模型 [10] 将决策分解为高层规划和低层控制。尽管这提升了模块化程度，但许多此类方法仍依赖潜在嵌入，所提供的空间和语义监督有限，也缺乏针对物理结构的显式归纳偏置，因此其表征仍容易受到感知变化的影响[11]。这些局限表明，仅靠架构分解并不足够；稳健的泛化从根本上取决于用于衔接感知、推理与动作的信息选择。

有效的表征会引入归纳偏置，以将任务结构与感知变化分离开来 [8]。像素级表征和潜在表征虽然表达能力强，且广泛用于端到端视觉语言动作模型 [6], ，却缺乏关于物理交互和组合推理的显式结构先验。近年来，场景图已成为机器人应用中的一种结构化分层表征，能够显式抽象物体、属性及其关系 [12]。除了紧凑的场景编码外，场景图还提供了与语言对齐的因果抽象，将空间理解与动作推理和规划联系起来 [13]。通过以符号形式呈现物体状态和关系谓词，场景图使模型，尤其是基于语言的推理模型，能够推断动作后果、前置条件以及跨时间的依赖关系。这种因果基础有助于长时程规划，使智能体能够对多步动作进行推理，同时衔接低层感知与高层语言概念，从而实现可泛化的推理（例如，推断杯子位于微波炉内部）。

B. 与近期工作的关系

我们的工作与基于提示的空间推理方法有关，例如 VoxPoser [14],ReKep [15],ConceptGraphs[16], 和 ENACT [17]。VoxPoser 和 ReKep 将语言指令转换为运动级约束或关系关键点，但任务语义和世界状态仍是隐式编码的。ConceptGraphs 和 ENACT 利用场景图进行感知、映射或提供辅助规划支持；然而，决策仍由基于提示或隐式推理的方式驱动，并未显式建模世界状态转移。相比之下，GSR 将接地场景图视为**决策的主要状态空间**，并将具身操作表述为对以对象为中心的状态转移进行显式推理。GSR 在统一的具身框架中联合学习前向状态演化、目标状态推断和下一步动作推理，从而能够逐步建模动作前提条件、因果效应和目标满足情况。这使 GSR 有别于以往将场景图用作辅助表示、而非具身推理核心基础的系统。

III. 方法

A. GSR框架概述

1）世界表示：给定一个RGB-D观测 $\mathcal{I}$，我们构建一个三维场景图 $M_{sg} = (O_t, E_t)$ 来编码工作空间和物体状态，其中$O_t = \{o_j\}_{j=1,\dots,J}$表示物体集合，而 $E_t = \{e_k\}_{k=1,\dots,K}$ 表示关系边集合。图2 展示了从原始视觉输入到结构化表示的转换过程，以及由此得到的场景图。每个物体 o_j 都表示为一个结构化实体，由功能关键点以及（如适用）关节式子组件组成。例如，**马克杯**包含一个对应于其功能关键点的**把手**，而**橱柜**则被建模为一个关节式物体，包含多个子元素，例如**抽屉**。边e_k 编码物体对之间的空间关系，表示

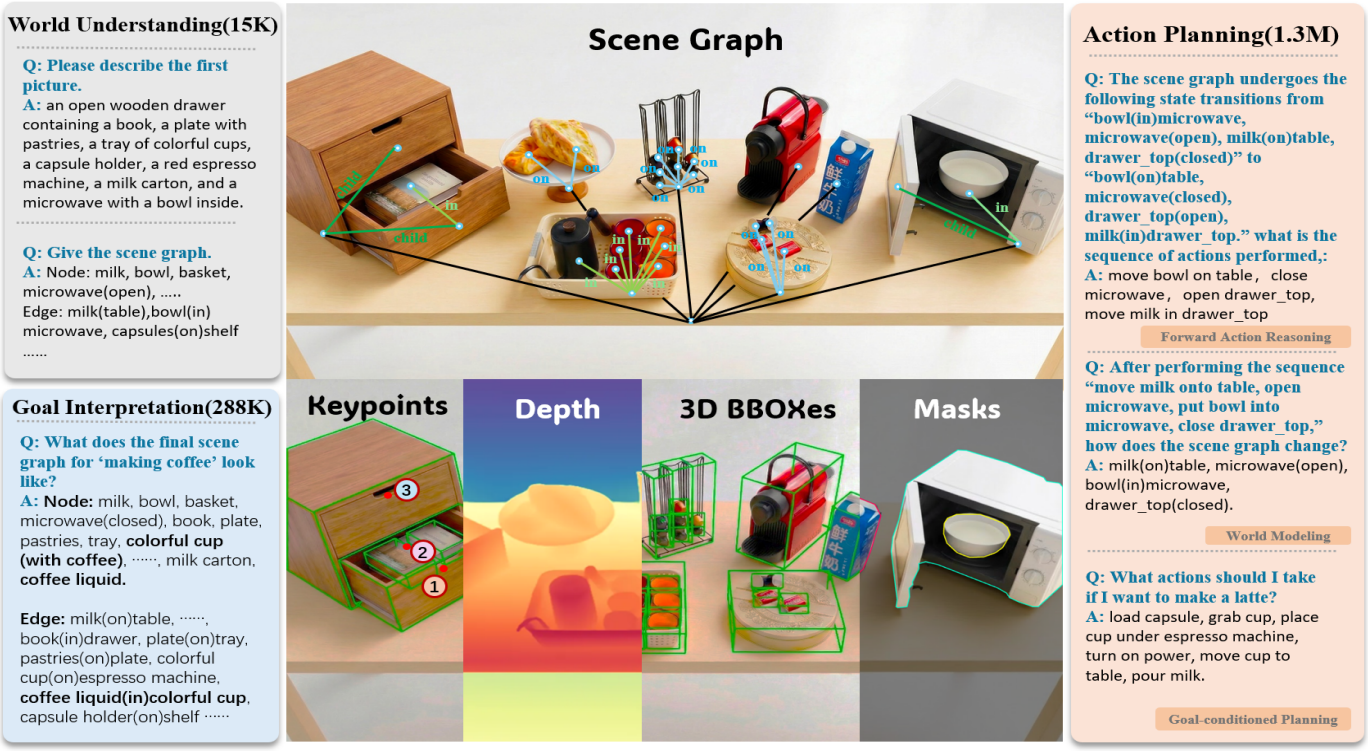

图 2：Manip-Cognition 数据示例，包括场景定位数据、规划数据和目标解释数据

诸如位于.....上、内部或相邻于之类的谓词（例如，一个**马克杯**位于一张**桌子**上）。附录A中提供了不同任务场景下的其他场景图示例。

2) 模型结构： GSR 是经过微调的大语言模型（LLM），旨在对场景图表示进行常识推理。为将 GSR 应用于物理实体化场景，我们将其与感知前端和动作专家后端集成。该物理系统由两个组件构成：用于决策的感知-推理模块，以及负责执行低层控制的动作专家。感知-推理模块根据原始观测构建场景图，使 GSR 能够基于这些信息进行推理并生成动作序列。为构建场景图，我们采用**视觉基础模型（VFM）**（见附录 A）。动作专家利用元技能库，详见第III-D节。

B. Manip-Cognition 数据引擎

本节介绍**Manip-Cognition-1.6M**，该数据集围绕三项核心能力设计：场景到图的理解、动作规划和目标解释，如图2所示。数据引擎将原始视觉演示处理为结构化任务轨迹，使模型能够生成合理、循序渐进的动作，以完成长时程的指令驱动操作任务。**此外，我们对原始轨迹进行系统性数据增强，最终构建出包含 1.6M 个样本的训练语料库**。详细的数据处理流程见附录B。

1) 世界理解数据：该数据组件训练模型将视觉观测抽象为语义落地的场景图（SG）。数据来源于 Behavior-1K 等高质量交互基准测试**Behavior-1K** [18] 和**Enact**[17]，使模型学会解读结构化场景图，并从自然语言描述中提取空间关系。最终表示形式为标准化的文本 $\rightarrow$ —场景图配对。经过语言与结构增强后，**该子集包含 15k 个接地配对**，为模型提供必要的监督，使其能够弥合感知与符号推理之间的差距。

2) 动作规划数据： 该数据集旨在训练模型掌握物体与动作之间的因果关系，这对于在各种空间约束下进行动作排序和推理至关重要。我们将这一能力分解为三项互补的推理任务：

(a)**前向动作推理**：此任务侧重于目标条件规划和因果推断。不同于简单的单步模仿，模型经过训练，需要推断导致观测到的或期望的状态转移的底层动作序列。给定初始状态 SG_t 和目标状态 SG_{t+n}（其中 $n \geq 1$），模型必须预测弥合这一差距所需的多步动作$\mathcal{A} = \{A_t, A_{t+1}, \ldots, A_{t+n-1}\}$。

$$(SG_t, SG_{t+n}) \rightarrow \{A_i\}_{i=t}^{t+n-1} \tag{1}$$

(b) **世界建模**：此任务充当内部物理引擎，训练智能体预测其动作所产生的环境后果。通过分析任务轨迹中的连续场景图对，我们提取其中的结构

变化，即动作前状态（SG_t）与动作后状态（SG_{t+1}）之间边的变化（ΔSG_{edge}）。模型经过训练，能够根据当前的 SG_t 和特定的候选动作 Action A_t，预测这一 ΔSG_{edge} （由此产生的状态变化）。

$$(SG_t, A_t) \rightarrow \Delta SG_{\text{edge}} \tag{2}$$

(c) 目标条件规划：该数据集将高层级的语言指令与结构化世界状态对齐。为确保智能体能够根据用户意图执行任务，模型经过训练，可根据当前环境配置 SG_t预测为实现文本任务目标 G而需要执行的下一个即时符号动作 A_t 。

$$(SG_t, G) \rightarrow A_t \tag{3}$$

增强后，规划数据集总计包含 130 万个样本，覆盖 6,000 条轨迹。

3）目标解释数据：该数据集旨在让模型深入理解任务要求和终态条件，使模型能够在结构化空间中内化“已完成的任务是什么样的”。不同于逐步进行的世界建模，该能力侧重于长时程目标推演：根据当前状态（SG_t）和自然语言指令，预测最终环境配置（SG_{goal}）。

$$(SG_t, \text{Goal Instruction}) \rightarrow SG_{goal} \tag{4}$$

这为智能体提供了稳定的语义目标，以指导多步规划。数据主要来源于**Epic-Kitchens-100 [19]**，使用来自**EgoPlan [20]**的第一人称视角动作标注。最终生成的数据集将（SG_t、指令）与目标 SG_{goal}配对，训练模型推断满足用户目标所需的动作。**增强后，该数据集包含 28.8 万个样本**。

*C. GSR*模型训练

GSR基于Qwen3-8B架构 [21]。我们采用分组查询注意力，其中32个查询头共享8个键值头。旋转位置嵌入用于改进长上下文建模，这对于处理冗长的场景图表示是必要的。在推理时，GSR接收场景图和用户指令，并输出下一个动作命令。训练分两个独立阶段进行：（i）监督微调（SFT）以激活结构化推理，随后（ii）强化微调（RFT）以使GSR与执行约束对齐。

1）第一阶段训练（监督微调）：第一阶段训练侧重于学习明确的场景理解、目标解释和动作推理。我们使用大量转换为指令遵循格式的轨迹混合数据进行训练。我们采用参数高效微调（PEFT），使用应用于所有线性层的低秩适配（LoRA）。由此得到的监督策略 π_{SFT} 经过训练，可以先生成结构化的中间推理，再生成动作命令。然而，我们观察到，经过监督微调的模型往往会出现推理伪影，例如幻觉生成的对象标识符，因此有必要进行第二阶段训练。

表I：GSR模型的训练超参数

超参数	**Qwen3-8B**
模型规模	8.2 B
非嵌入参数	6.95 B
最大上下文长度	131,072 (YaRN)
训练序列长度	2,048
精度	bfloat16
训练方法	低秩适配（所有线性层）
LoRA 秩（r）	8
LoRA Alpha（α）	32
优化器	AdamW
学习率	1×10^{-4}
学习率调度器	余弦预热
预热比例	0.05
训练轮数	20
每设备批量大小	4
梯度累积步数	16
全局批量大小	128

2）第二阶段训练（强化微调）：为了解决监督微调的局限性，并使模型更好地适应物理执行约束，我们引入了第二阶段的强化微调，采用组相对策略优化（GRPO）。优化目标是最大化任务特定奖励函数 R，以针对逐步具身推理中的三种常见失败模式：（i）多步幻觉；（ii）物体定位错误；（iii）错误的终止决策。总奖励 R_{total} 定义为这三个组成部分的加权和。

$$R_{total} = \lambda_1 R_S + \lambda_2 R_G + \lambda_3 R_T \tag{5}$$

a) 逐步约束奖励：此项旨在解决监督微调模型倾向于通过在单次输出中生成多个动作（例如“拿起 A，然后放置 B”）来臆造未来计划的问题，而执行引擎要求严格逐步生成动作。我们施加严格的格式惩罚：给定预测的动作词元序列 A_{pred}，奖励 R_S 会鼓励模型在推理过程之后恰好输出一个原子动作元组。

$$R_S = \begin{cases} 1, & \text{if } a_{\text{pred}} = \texttt{END} \wedge \text{Satisfied}(S, G) \\ -\alpha, & \text{if } N(A_{\text{pred}}) > 1 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

其中，$N(\cdot)$表示响应中可执行动作原语的数量。这会促使模型关注紧接着要执行的步骤，而非开环规划。

b) 场景图接地奖励：该项针对物体幻觉，这是一种常见的失败模式，即模型会生成并不存在的物体，例如排版变体（如用“$apple$”代替“$apple\ 01$”）。该奖励强制执行严格的物体接地：给定预测的目标物体标识符 obj_{pred} 以及当前场景图中的有效节点标签集合 V_{SG}，仅当 $obj_{\text{pred}} \in V_{SG}$时模型才会获得奖励。

$$R_G = \mathbb{I}(obj_{pred} \in V_{SG}) \tag{7}$$



图3：我们的评估包含三个部分：(1) 来自 RLBench 和 LIBERO、包含230个任务的开源基准；(2) GSR任务套件，包含180个难度各异的长时程任务，目标涉及语义物体消歧、空间感知序列规划和目标条件泛化；以及(3) 涵盖多种任务的真实世界评估。

其中 $\mathbb{I}(\cdot)$为指示函数。这一硬约束消除了由无效物体引用引起的执行错误。

c) 终止奖励： 该项用于处理终止错误，包括过早终止（在目标实现前输出“任务结束”）或无限循环（任务完成后仍继续生成动作）。

$$R_T = \begin{cases} 1, & \text{if } a_{\text{pred}} = \texttt{END} \text{ and } \text{Satisfied}(S_{\text{env}}, G) \\ -\beta, & \text{if } a_{\text{pred}} = \texttt{END} \text{ and } \neg\,\text{Satisfied}(S_{\text{env}}, G) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

其中， S_{env} 表示当前环境状态， G 表示目标条件。该奖励用于区分正确终止行为与错误的停止行为。

D. 真实世界机器人部署

为了在实体机器人上部署GSR框架，我们实现了一个连接高层推理与低层执行的元技能库。在前端，给定一个动作命令，VFM会计算一组定向三维关键点，用于指定起始状态、合理的交互点和目标。在后端，我们采用基于几何织物的行为生成模块。每个由几何织物编码的行为都遵循以下形式：

$$M_f(q_f, \dot{q}_f)\ddot{q}_f + f_f(q_f, \dot{q}_f) + f_\pi(a) = 0 \quad (9)$$

其中， q_f 和 $\dot{q}_f$ 表示织物空间中的广义坐标和速度（例如关节空间或末端执行器空间）。状态相关度量 $M_f(q_f, \dot{q}_f)$ 通过编码任务优先级来塑造运动几何，而 $f_f(q_f, \dot{q}_f)$ 则捕捉内在织物动力学，例如吸引力、阻尼和障碍物规避。由所选动作原语 a 指定的动作条件项 $f_\pi(a)$ 注入任务意图（例如目标、交互力约束和可操作性项）。我们的物理系统采用 Tianji Marvin-6 七自由度双臂机器人，并在头部配备一台 Intel RealSense D435 RGB-D相机。

IV. 实验

我们对 GSR 进行了全面评估，包括在**RLBench** 和 **LIBERO**中针对不同任务配置的单次推理、在 **GSR-Bench**中的空间感知、目标条件和长时程推理，以及通过真实世界的机器人实现来评估性能，如图3所示。

A. 在 *RLBench* 和 *LIBERO* 上的评估

表 II：RLBench LIBERO

平台	Category	数量
RLBench	厨房操作 **(KO)**	25
	抓取与放置 **(PP)**	27
	容器开关操作 **(SC)**	31
	控制器操作 **(CO)**	7
	堆叠与组装 **(SA)**	10
LIBERO	LIBERO_物体	10
	LIBERO_空间	10
	LIBERO_目标	10
	LIBERO_长程	10
	LIBERO Short	90

本实验评估该模型的**一般推理能力**在**未经特定任务训练的操作任务中**。该模型必须生成**指令**，并以给定的**初始状态**为条件，推断其潜在的**目标**，并预测正确的**动作**。我们选取五个大语言模型作为基线，包括`GPT-5` [22],`Gemini-2.5-Pro` [23], `DeepSeek-V3`[24],`Qwen3-8B` [21], 以及Claude-Sonnet-4。`5` [25]。为在操作任务中部署大语言模型，我们将基于场景图的感知模块与大语言模型控制接口相结合（见附录C），该接口将语言指令映射为相应的低层动作。我们在零样本设置下，于 RLBench 和 LIBERO（表II）上评估 GSR，且不进行任何微调。对于其他模型，我们为每个大语言模型设计特定任务提示，以确保评估过程中其性能得到充分优化。在**RLBench**上，我们评估了 100

表 III：RLBench 和 LIBERO基准上的性能比较。

方法	RLBench					LIBERO				
	KO	PP	SC	CO	SA	物体	空间	Goal	短时	Long
GPT-5	80.00	58.20	45.00	33.30	15.00	81.00	79.00	69.00	80.00	57.00
Gemini-2.5-Pro	**95.00**	**85.60**	51.50	46.70	30.00	**87.00**	86.00	**74.00**	86.00	**68.00**
DeepSeek-V3	90.00	70.00	64.30	42.00	32.50	82.00	77.00	73.00	80.00	61.00
Claude-sonnet-4-5	30.00	51.00	63.20	36.70	20.00	82.00	**88.00**	74.00	83.00	62.00
Qwen3-8B	25.00	51.00	56.40	33.30	26.70	84.00	81.00	72.00	82.00	54.00
GSR-8B（我们的方法）	92.50	82.00	**68.50**	**52.50**	**41.20**	86.50	**89.20**	81.00	85.00	**82.40**

个任务，涵盖五个代表性类别：**厨房操作（KO）**、**抓取与放置（PP）**、**容器开关操作（SC）**、**控制器操作（CO）**以及**堆叠与组装（SA）（见附录C）**。在**LIBERO**上，我们评估了 100 个任务，涵盖四个子集：**LIBERO-Spatial**、**LIBERO-Object**、**LIBERO-Goal**以及**LIBERO-Long**。我们对每个任务运行十次试验，并报告**成功率**。详细结果见表III和图4。

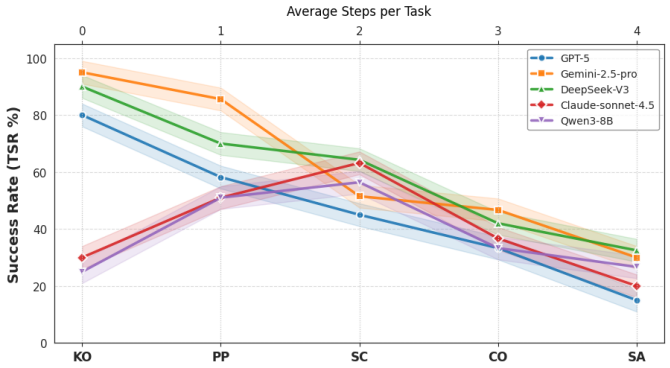

图 4：RLBench 上的模型性能比较，展示了成功率与任务复杂度之间的关系。

从结果中可以看出几种一致的趋势。对于所有模型，随着任务涉及更复杂的空间推理和更长的动作序列，性能都会下降。简单的抓取与放置任务所施加的空间约束有限，而堆叠与装配则需要对物体关系和中间状态进行推理，导致性能大幅下降（例如，Gemini-2.5-Pro 在 PP 任务上的成功率从 95% 降至 SA 任务上的 30%）。随着任务时域增加，这种性能下降会愈发明显，因为更长的序列会放大推理错误并妨碍恢复；例如，随着平均步数从 1.85 增至 3.9，DeepSeek-V3 的成功率从 90% 降至 32.5%。在 LIBERO 上也观察到了类似趋势：对于所有模型，时域更长的任务以及目标条件任务（LIBERO-Long 和 LIBERO-Goal）的成功率始终较低。

B. 在 *GSR-bench* 上的评估

GSR-Bench评估在空间和语义约束下进行的长时域任务推理。它包含 180 项任务，平均规划时域超过 10 步。我们的评估聚焦于三个方面：**(1) 语义物体消歧（SOD）**，用于评估在物体语义多变情况下的推理能力；**(2) 空间感知序列规划（SAS）**，用于评估基于物理因果关系和空间约束的推理能力；以及**(3) 目标条件泛化（GCG）**，用于衡量针对多种抽象目标进行推理的能力。对于每个方面，我们定义了三个难度等级：简单、中等和困难（见附录D）。我们使用**任务进度（TP）**作为评估指标，实验结果见图5。

表 IV：GSR 对场景图噪声的分析。

任务类型	噪声	Easy	Med.	Hard
任务-SOD	0%	1.000	0.980	0.969
	5%	0.960	0.940	0.910
	10%	0.900	0.880	0.850
任务-SAS	0%	0.975	0.980	0.940
	5%	0.920	0.910	0.860
	10%	0.850	0.830	0.750
Task-GCG	0%	0.917	0.367	0.250
	5%	0.800	0.300	0.180
	10%	0.680	0.220	0.120

在所有评测任务中，随着任务难度从**简单**提升到困难，*TP*均会持续下降。例如，GPT-5 在空间感知序列规划上的 TP 从 0.11（简单）大幅降至 0.02（困难）。这一趋势在不同模型和任务类别中均一致：随着物体密度、空间遮挡和指令歧义增加，具身推理性能会下降。在 RLBench 和 LIBERO 上也观察到了类似的性能下滑，因为这些任务需要更复杂的空间推理和更长的动作序列。这种影响在长时程任务中最为明显：基于提示的大语言模型无法在多个步骤中始终维持空间约束和目标约束。

在所有任务中，目标条件任务最具挑战性。这主要有两个原因。首先，指令通常更加抽象且模糊（例如“将不同颜色的物体分类放入不同的抽屉”），这要求更强的语言理解和目标解释能力。其次，目标条件任务通常涉及更复杂的空间约束和更长的动作序列，需要进行多步规划，并对中间状态进行推理。定量结果清晰地反映了这一趋势。例如，Gemini-2.5-Pro 在物体、空间和目标任务上的 TP 分别为 0.259、0.163 和 0.101。

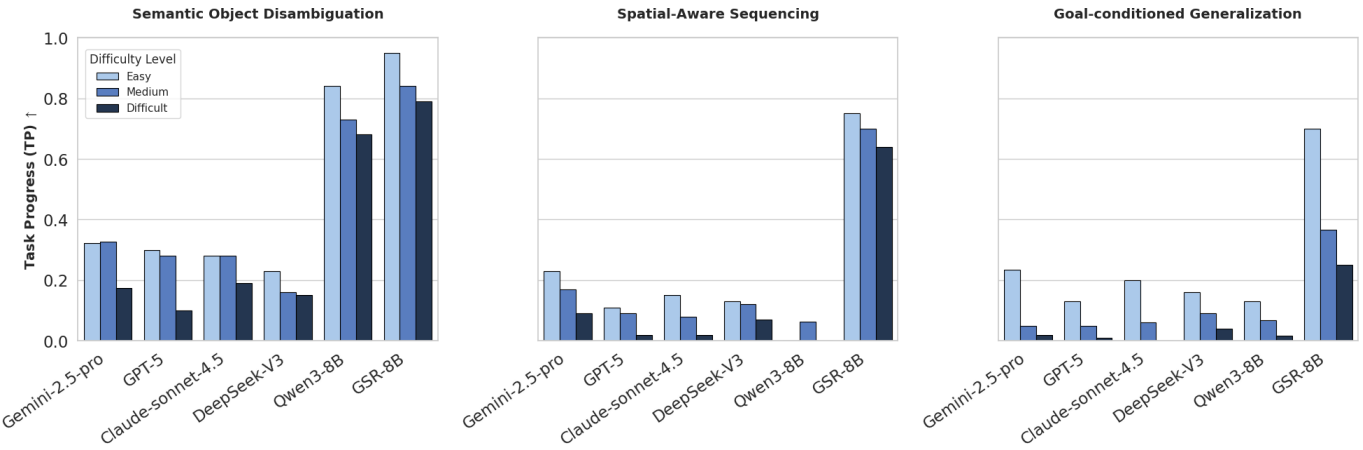

图 5：通过任务进度（TP %）评估 GSR-Bench，结果表明，GSR 在全部三个评估方面均达到最先进的性能，并且在涉及复杂空间约束和目标推理的长时程任务上优势显著。

GPT-5 展现出类似的模式，其 TP 值分别为 0.227、0.073 和 0.063。有趣的是，一些模型在目标-简单任务上的表现略优于空间-简单任务。例如，Claude-Sonnet 4.5 在空间-简单任务上的 TP 为 15%，而在目标-简单任务上提升至 20%。这是因为目标-简单任务的空间复杂度较低：有些案例不涉及对堆叠容器或关节式物体进行操作。总体而言，目标条件任务最具挑战性，因为它们要求在进行空间推理的同时理解抽象指令，而 GSR 始终优于基于提示的基线方法。

C. 对感知噪声的鲁棒性分析

为评估对感知错误的鲁棒性，我们在推理时对 GSR-8B 的输入场景图进行扰动，无需重新训练，如表 IV 所示。在 GSR-Bench 中，我们以 5% 和 10% 的噪声比例随机删除或翻转场景图关系，并使用任务进度（TP）评估性能。GSR 在中等噪声下仍具鲁棒性：在 5% 扰动下，各项任务的 TP 仅略有下降（例如，语义物体消歧在简单/中等/困难任务上的得分分别从 1.00 / 0.98 / 0.97 降至 0.96 / 0.94 / 0.91）。噪声达到 10% 时，性能进一步下降，但总体仍保持稳定。对噪声的敏感程度因任务类型而异：目标条件泛化受影响最大（困难任务的 TP 从 0.25 降至 0.12），而语义物体消歧和空间感知序列规划在相同噪声水平下仍保持较高性能。

D. 真实世界任务演示

本实验在三种真实场景中评估 GSR，每种场景分别针对其推理和泛化能力的一个不同方面。首先，我们评估 GSR 在涉及常见物体的一般抓取与放置任务中的表现，考察其根据自然语言指令进行推理并生成恰当行为的能力。其次，我们在 GSR-Bench 中的长时序排序任务上评估 GSR，考察其在多种空间配置和目标条件下的表现，这些条件需要长程规划。第三，我们展示 GSR 在四项日常生活任务中的应用，详细结果见补充视频。

V. 结论

本研究提出了 GSR，这是一种基于显式结构化世界表征进行推理的具身推理框架。GSR 通过对编码物体状态和关系的场景图进行推理，而非直接采用从感知到动作的映射，从而实现了可迁移的物理常识推理，并提升了操作中的泛化能力。在仿真环境和实体机器人上的大量评估表明，GSR 具有出色的零样本性能、长时程推理能力，以及应对真实世界扰动的鲁棒性。

局限性。我们的工作存在若干局限。首先，当前框架依赖预定义的动作原语，这限制了长时程任务中灵活的动作生成，以及连续步骤之间的自适应能力。其次，用于构建场景图的视觉表征在捕捉精细几何结构、细微物体状态和复杂物理属性方面仍有不足，这可能会影响视觉环境复杂或杂乱时的推理准确性。**影响**。本研究结果表明，以显式世界表征为基础进行决策，有助于在开放世界具身场景中实现更系统的泛化、持续推理和自适应行为。未来工作将探索表达能力更强、适应性更高的动作策略，以及能够生成更精确世界表征、以支持更细粒度推理与控制的改进型感知模型。

参考文献

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter等，“Pi0：一种用于通用机器人控制的视觉-语言-动作流模型”，*arXiv*预印本 *arXiv:2410.24164*，2024。

[2] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn 等，“Cot-vla：面向视觉-语言-动作模型的视觉思维链推理”，载于计算机视觉与模式识别会议论文集，2025，第1702–1713页。[3] J. Chen, Y. Mu, Q. Yu, T. Wei, S. Wu, Z. Yuan, Z. Liang, C. Yang, K. Zhang, W. Shao 等，“Roboscript：面向真实环境与仿真环境中自由形式操作任务的代码生成”，*arXiv*预印本 *arXiv:2402.14623*，2024。[4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark等，“通过自然语言监督学习可迁移视觉模型”，载于国际机器学习会议。PmLR，2021，第8748–8763页。[5]Y. Ji, H. Tan, J. Shi, X. Hao, Y. Zhang, H. Zhang, P. Wang, M. Zhao, Y. Mu, P. An 等，“Robobrain：一个将抽象概念转化为具体操作、用于机器人操作的统一大脑模型”，载于计算机视觉与模式识别会议论文集，2025，第1724–1734页。[6]B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid 等，“Rt-2：视觉-语言-动作模型将网络知识迁移至机器人控制”，载于机器人学习会议。PMLR，2023，第2165–2183页。[7] M. J. Kim, K. Pertsch, S.Karamcheti, T. Xiao, A. Balakr- ishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi等，“Openvla：一种开源视觉-语言-动作模型”，*arXiv*预印本 *arXiv:2406.09246*，2024。[8] B. Ai, S. Tian, H. Shi, Y. Wang, T. Pfaff, C. Tan, H. I. Christe nsen, H. Su, J. Wu, and Y. Li，“机器人操作中基于学习的动力学模型综述”，《科学·机器人》，第10卷，第106期，文章编号 eadt1497，2025。[9] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu 等，“Rt-1：用于大规模真实世界控制的机器人变换器”，*arXiv*预印本 *arXiv:2212.06817*，2022。[10] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh，“Rt-h：基于语言的动作层级”，*arXiv*预印本 *arXiv:2403.01823*，2024。[11] S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans，“用于决策的基础模型：问题、方法与机遇”，*arXiv*预印本 *arXiv:2303.04129*，2023。[12] H. Li, G. Zhu, L. Zhang, Y. Jiang, Y. Dang, H. Hou, P. Shen, X. Zhao, S. A. A. Shah, and M. Bennamoun，“场景图生成：全面综述”，《神经计算》，第566卷，文章编号127052，2024。[13] Y. Zhu, A. Lim, P. Stone, and Y. Zhu，“借助开放世界对象图，从单个人类视频实现基于视觉的操作”，*arXiv*预印本 *arXiv:2405.20321*，2024。[14] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei- Fei，“Voxposer：借助语言模型实现机器人操作的可组合三维价值图”，*arXiv*预印本 arXiv:2307.05973, 2023.[15] W. Huang、C. Wang、Y. Li、R. Zhang 和 L. Fei-Fei，“Rekep：用于机器人操作的关系关键点约束时空推理”，arXiv 预印本 arXiv:2409.01652，2024.[16] Q. Gu、A. Kuwajerwala、S. Morin、K. M. Jatavallabhula、B. Sen、A. Agarwal、C. Rivera、W. Paul、K. Ellis、R. Chellappa 等，“ConceptGraphs：用于感知与规划的开放词汇三维场景图”，载于 2024年IEEE机器人与自动化国际会议（ICRA）。IEEE，2024年，第 5021–5028 页。[17] Q. Wang、W. Huang、Y. Zhou、H. Yin、T. Bao、J. Lyu、W. Liu、R. Zhang、J. Wu、L. Fei-Fei 等，“Enact：通过自我中心交互的世界建模评估具身认知”，arXiv 预印本 arXiv:2511.20937，2025年。[18] C. Li、R. Zhang、J. Wong、C. Gokmen、S. Srivastava、R. Mart´ín-Mart´ín、C. Wang、G. Levine、W. Ai、B. Martinez 等，“Behavior-1K：以人为中心的具身 AI 基准测试，涵盖 1,000 项日常活动和逼真模拟”，arXiv 预印本 arXiv:2403.09227，2024。[19]D. Damen、H. Doughty、G. M. Farinella、S. Fidler、A. Furnari、E. Kazakos、D. Moltisanti、J. Munro、T. Perrett、W. Price 等，“EPIC-KITCHENS 数据集：收集、挑战与基线”，IEEE模式分析与机器智能汇刊，第 43 卷，第 11 期，第 4125–4141 页，2020年。[20] Y. Chen、Y. Ge、Y. Ge、M. Ding、B. Li、R. Wang、R. Xu、Y. Shan 和 X. Liu，“EgoPlan-Bench：多模态大语言模型类人规划能力基准测试”，arXiv 预印本 arXiv:2312.06722，2023年。[21] Qwen团队，“Qwen3：深思熟虑，快速行动”，2025年4月，访问日期：2025-04-29。 [在线]。网址：https: // openai.com/zh-Hans-CN/index/gpt-5-system-card/[22] OpenAI，“GPT-5系统卡”，2025年8月，访问日期：2025-09-16。 [在线]。网址：https://openai.com/zh-Hans-CN/index/gpt-5-system-card/[23] G. Comanici、E. Bieber、M. Schaekermann、I. Pasupat、N. Sachdeva、I. Dhillon、M. Blistein、O. Ram、D. Zhang、E. Rosen 等，“Gemini 2.5：以高级推理、多模态能力、长上下文和新一代智能体能力推动前沿发展”，arXiv 预印本 arXiv:2507.06261，2025年。[24]DeepSeek-AI，“DeepSeek-V3技术报告”，2024。[在线]。网址：https://arxiv.org/abs/2412.19437[25] Anthropic，“Claude Sonnet 4.5 正式发布”，2025年9月，发布日期：2025-09-29。 [在线]。网址：https://www.anthropic.com/news/claude-sonnet-4-5

附录

A. 视觉基础模型的详细信息

本节详细介绍我们使用的开放世界视觉基础模型。为了准确表征物体的全面信息，我们设计了一个可解释的视觉信息提取框架。通过分析任务目标，我们为模型检索初始化参数，并建立相应的信息提取工作流。完成初始化后，我们使用 YOLO-E 高效地进行开放世界目标检测和分割。为了提取物体关键点信息，我们对检测头进行了微调，使其能够提取特定物体的关键点数据。为了持续跟踪点，我们简化了 Spatial-TrackerV2 并优化了特征提取模块。这确保点跟踪器在机器人操作或移动过程中保持稳定跟踪，同时与整个模型保持固定频率。

我们还考虑到，实时更新场景图需要快速获取物体的三维空间坐标，并进行高频更新。我们基于 Gen-pose++ 开发了一种轻量级六维物体姿态估计模型。

1）不同物体的检测细节：我们的检测模型设计基于实验部分所述的三种物体类型。

刚体：对于刚体，例如基本几何形状（立方体），无需进行特殊的功能设计。我们仅采用基本的六维姿态检测，为 GSR 提供稳定的空间检测结果，从而支持加速场景图构建。对于彩色立方体排序任务，我们只需准确检测不同颜色立方体的位置及其目标放置区域。这样便能获取所有物体之间的空间关系并计算场景图，最终将其交由 GSR 进行操作规划。为有效表示动态场景关系，检测模型必须保持固定的检测帧率（15 Hz）。因此，我们开发了一种更高频的检测方法，以确保 GSR 基于最新的空间关系信息处理决策。

可动关节物体：对于涉及关节式物体的精细操作任务，我们引入关键点检测来引导 GSR 实现精确规划。具体而言，当机器人需要操作咖啡机时，仅依靠物体级六维姿态估计或最先进的视觉语言模型（VLMs），往往不足以准确定位机器上的关键可操作部件（例如按钮和开关）。为解决这一问题，我们提出一种融合表示方法，将检测到的部件关键点信息以显式编码的形式与物体的六维姿态估计结果相结合，并将这一集成方法统一纳入场景图计算框架。实验表明，这种融合信息表示为后续操作规划提供了更稳定、准确的几何与语义约束，显著提升了系统在复杂交互任务中的表现。

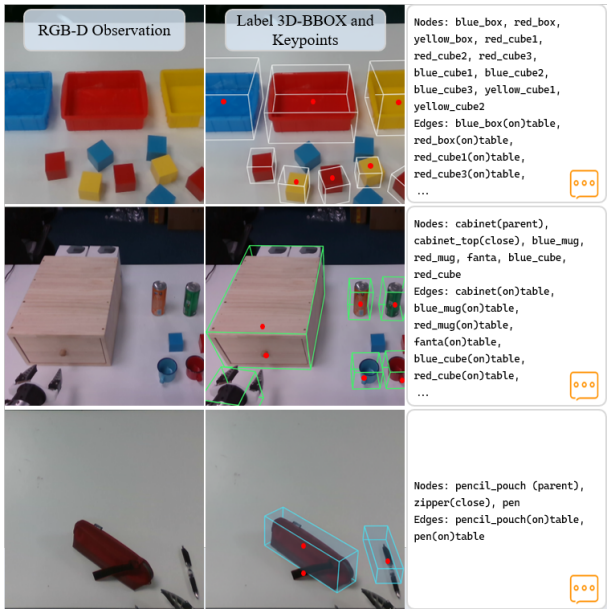

图 6：视觉模型检测与真实场景图生成结果

软体物体：基于刚体假设的传统六维姿态估计方法在处理软体物体时往往效果不佳。为解决这一问题，我们采用关键点跟踪策略，实时检测并跟踪物体表面的语义关键点，逐帧提供用于操作的几何状态更新。这种方法能够适应物体变形和姿态变化，同时持续输出稳定的关键点信息，并支持动态场景图的实时维护与更新。在笔袋开合任务中，该方法展现出很强的鲁棒性。具体而言，系统能够可靠地检测拉链位置及其开合状态，实时估计笔袋的开口程度，引导机械臂放入书写工具，并持续跟踪拉链运动直至闭合。实验表明，这一关键点跟踪框架有效克服了软体物体几何不确定性带来的操作挑战，在保持任务准确性的同时，维持了场景理解的一致性。

2) 真实场景图的实现：我们在三个真实场景中测试了视觉融合模型的动态场景图生成能力。图6中的结果表明，该模型能够持续提供场景图生成所需的信息。通过显式纳入物体关键点状态信息，我们有效增强了大语言模型对图像中细粒度组件的几何和功能状态的理解。关于从真实场景中的视觉信息生成场景图的具体内容，由于本文的核心工作并未涉及场景图生成，我们将在未来工作中进行详细的实验数据比较。

B. Manip-Cognition 数据引擎详情

Manip-Cognition数据集是基于场景图的推理（GSR）框架的主要训练来源。它弥合了原始视觉演示与结构化符号推理之间的鸿沟。在以下章节中，我们将详细介绍数据选择与来源（第B1节）、数据清理与结构规范化（第B2节）、标注流程（第B3节）以及我们的数据增强策略（第B4节）。

1) 数据选择与来源：为涵盖多样的任务结构和物理交互类型，我们从 4 个开源具身数据集和基准测试中收集了数据。

- **Behavior-1K [18] 和 Enact [17]**：我们采用了 Behavior-1K 中的 29 个任务场景，这些场景提供完整的演示视频和场景图真实标注。通过利用 Enact 基准测试中的关键帧标注，我们提取出**856 个独特的场景图（SG）**，用于表示不同的动作边界。经过清理和增强后，这些数据构成了基础性的**场景定位数据**（12,000 对）。

- **EPIC-KITCHENS-100 [19] 和 Ego-Plan [20]**：为扩展长时程推理，我们处理了 42 个厨房场景中的 614 个长视频。根据 Ego-Plan 标注的 9,138 个独特任务和 69,253 个动作关键帧，我们进行了严格筛选，保留了**6,000 条完整的情节轨迹**，每个任务平均包含 6 个子步骤。这些轨迹被分解为特定的推理模态：

  – **世界建模**：使用连续场景图对（SG_t，SG_{t+1}）及其间的动作 A_t 来预测边变化：（SG_t，A_t）$\rightarrow$ ΔSG_{edge}。– **前向动作推理**：利用跨越多个步骤的序列来弥合状态之间的差距：（SG_t，SG_{t+n}）$\rightarrow \{A_i\}_{i=t}^{t+n-1}$。– **目标条件规划**：将当前场景图与高层任务目标 G 配对，以预测紧接着的下一步动作：（SG_t，G）$\rightarrow A_t$。– **目标解释**：将初始状态 SG_{start} 与最终任务完成状态 SG_{goal} 和全局指令配对，以训练“目标构想”：（SG_t，目标指令）$\rightarrow SG_{\text{goal}}$。

2) 数据清理与结构规范化：为确保结构推理可靠并尽量减少模型幻觉，我们根据每种数据源的具体特征实施了有针对性的清理流程。

- **Behavior-1K 场景图**：从 Behavior-1K 中提取的原始场景图存在两个主要的结构问题：(1) 关系语法不一致：空间关系的表示方式不一致，有时会使用冗余的双向标签（例如同时使用 '$A(ontop)B$' 和 '$B(under)A$'），有时则只使用单个单向谓词。这种表达缺乏一致性，会产生相互冲突的监督信号。(2) 节点-边不一致：经常出现

Data filtering Prompt

Instructions: You are a task verification assistant. Check if the given next_actions are appropriate and logically correspond to achieving the task_goal.

Task Goal: {task_goal}
Next Actions: {next_actions}
Requirements:
1. Analyze whether each action in next_actions is a reasonable step towards completing the task_goal
2. Check if the actions are in a logical order
3. Identify any actions that don't make sense for the task_goal
4. Determine if any important actions are missing

图7：**数据清洗提示示意图**。此提示旨在利用 GPT-5 的高级推理能力，自动进行数据集精炼并减少噪声。

场景图节点中列出的物体与边谓词中出现的物体不匹配的情况。若不加以纠正，这些问题可能导致训练不稳定性和严重的模型幻觉，例如生成逻辑上不可能存在的虚假节点或边。为解决这一问题，我们首先将所有关系统一规范为严格的单向 'ontop' 格式。随后，我们直接根据经过验证的边谓词，重建每个场景图的物体节点列表。

- **Ego-Plan任务完成**：原始 Ego-Plan 标签中常包含不完整的动作链，只展示较长任务的前几个步骤。我们使用 GPT-5 检查每条视频轨迹，并移除所有未能呈现明确、有效终点的片段。所用的具体提示见图 7。

3) 数据标注流水线：我们制定了分层标注策略，将原始视觉观察和场景图转化为 GSR框架所需的结构化推理对。

- **场景到图的定位标注**：为构建场景定位数据，我们使用了来自 Behavior-1K 和 Enact 的清理后场景图。由于这些来源缺乏丰富多样的自然语言描述，我们采用 GPT-5 为每个独特的场景图实例生成各具特色、语言多样的描述。这些描述严格保留对象身份和空间关系，同时提供视觉到符号定位所需的风格多样性。

- **情景与目标状态标注**：为将 EPIC-KITCHENS 的视觉关键帧转换为完整的推理轨迹，我们实施了一个三阶段流程。各阶段所用的提示见图 9：

  – **阶段 1：视觉语言模型描述生成**：本地部署的 Qwen3-VL-8B 会为每个动作边界关键帧的视觉状态生成详细的自然语言描述。

  – **阶段 2：目标状态合成**：由于 Ego-Plan 标注并未标明真实的任务完成状态，我们使用 GPT-5 根据任务指令和此前的动作描述进行推理，合成一份

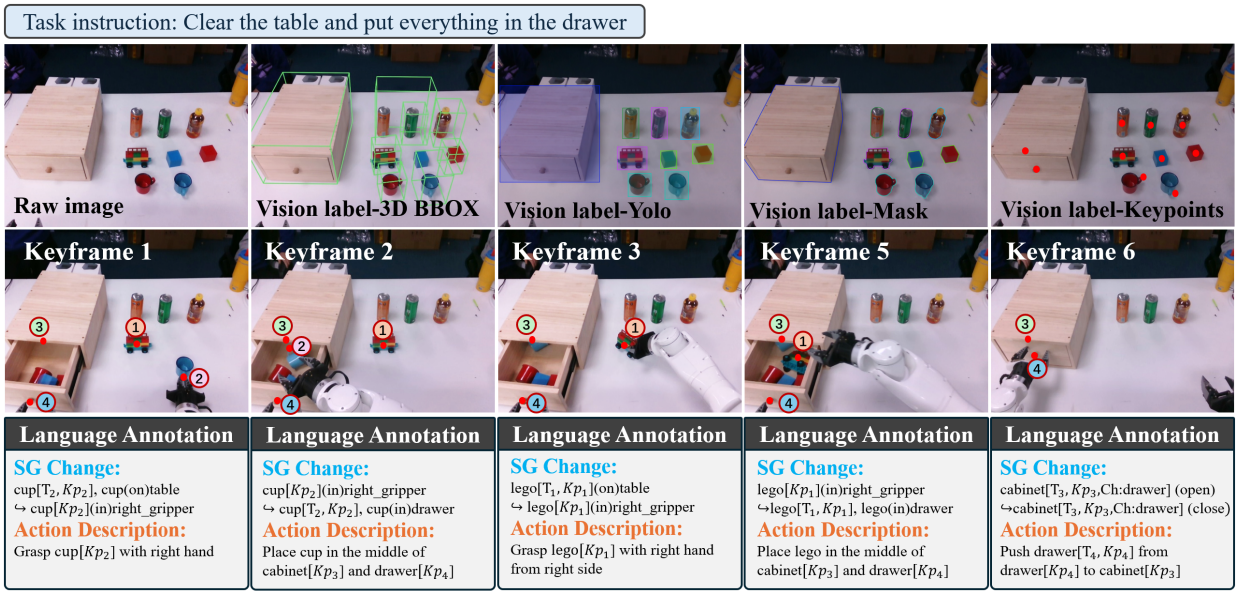

图 8：真实场景中采集的数据标注示例。 T_j 表示物体的三维位姿， Kp 表示物体上的语义关键点， Ch 表示子节点。

对预期最终状态的结构化描述（例如“折叠盒”的配置）。

– **阶段 3：场景到图转换**：所有生成的描述（来自关键帧和合成的目标状态）都会由专门的大语言模型（基于 Enact场景图描述对微调）处理，将文本转换为标准化的场景图（SG_t 和 SG_{goal}）。

– **真实场景评估数据标注**：对于第 IV-D 节所述的实体机器人实验，我们录制了同步的 RGB视频流和机器人内部状态。数据采集期间，我们主动触发并记录动作边界关键帧，以分离特定的子任务片段。该评估集的标注分为两个细粒度层次：

∗ **高保真视觉标注**：我们以 1/10 的抽取率对原始图像流进行采样。对每个采样帧进行人工标注，添加以物体为中心的视觉元数据，包括三维位姿（T_j）、YOLO 边界框、分割掩码和语义关键点（Kp）。

∗ **细粒度语言与符号标注**：对于每个关键帧，我们标注了自然语言动作描述以及相应的场景图变化。借助密集的视觉元数据，这些场景图采用了比我们大规模训练集更高的粒度。具体而言，每个物体节点表示为一个元组：

$o_j = [T_j, \mathcal{K}p_{1..n}, \mathcal{C}h_{1..m}]$。

其中， T_j 表示物体的三维位姿， $\mathcal{K}p$ 表示语义关键点， $\mathcal{C}h$ 表示层级子节点。这种详细的表征能够实现精确的动作落地，例如“将抽屉 $[T_4, Kp_4]$ 从抽屉$[Kp_4]$ 移至橱柜$[Kp_3]$”，从而能够更严格地评估智能体在真实场景中的空间推理能力（见图8）。

4) 数据增强与最终数据集构成：为增强GSR框架对语言变异和结构噪声的鲁棒性，我们在所有数据模态上应用了系统性数据增强。Manip-Cognition数据集的最终规模与构成亦详述如下。

• **场景落地数据增强**：从Behavior-1K和Enact的底座场景图描述对出发，我们应用了3类扩展：

– **描述改写**（×3）：每条描述使用GPT-5改写了两次，确保相同的语义和一致的对象身份，提示词如图10所示。
– **图顺序打乱**（×3）：为确保模型对输入顺序具有不变性，节点和场景图字符串中的边被随机重新排列。
– **替换对象名称**（×2）：我们使用了一个备选名称列表来替换常见的对象名称（例如，“杯子”替换为“马克杯”），并在整个场景图中保持一致及其描述。由此总共得到约 **15,000** 个接地配对。

• **规划与目标解释数据扩增**：基于来自Ego-Plan的6,000条清洗后情节轨迹（平均每个任务6步，共计36,000个状态-动作对），我们实施了多因素增强策略：

– **图顺序打乱**（×3）：我们随机重新

表V：数据增强因子和最终数据集规模汇总。

数据模态	基础规模	增强方法	乘数	最终数量
场景落地	856	释义 (3) × 打乱 (3) × 交换 (2)	18×	≈ 15,000
规划数据	36,000	打乱 (3) × 交换 (2) × 改写 (2) × 3 任务	36×	1,296,000
目标解释	6,000	打乱 (3) × 交换 (2) × 改写 (2) × 终态 (4)	48×	288,000

VLM Description Generation Prompt

Instructions:

Generate a detailed description of the current frame based on the provided image, task goal, and next action.

Requirements:

1. The description MUST include the position and state of objects mentioned in the next action, along with their spatial relationships to surrounding objects.
2. If an object in the next action is not clearly visible in the image, infer its location based on the task's progression.
3. Use specific spatial terminology to describe object locations.

Goal State Synthesis Prompt

Instructions:

Based on the following information, generate a caption describing the final state of the scene after the last action is completed.
Task Goal: {task_goal}
Previous captions (scene descriptions): {captions}
Last action performed: {last_action}

Requirements:

1. Describe the scene state AFTER the last action has been completed
2. The description should clearly show that the task goal "{task_goal}" has been achieved
3. The description should be consistent with the previous captions style
4. Keep it concise but informative

Scene-to-Graph Translation Prompt

Instructions:

You are a scene graph generator. Based on the given scene description (caption) and next action, generate a scene graph.

Requirements:

1. Extract all OBJECTS mentioned in the caption as nodes.
2. The objects appearing in the next action MUST be included in the nodes list with exact name matching.
3. Generate edges representing spatial relationships between objects.
4. Edges format strictly: object1(preposition)object2, where preposition should be spatial relations like: on, in, under, near, beside, behind, in_front_of, next_to, above, below, holding, touching, inside, etc.
5. Both obj1 and obj2 in edges must exist in nodes.

图 9：**情景与目标状态标注的提示范式**。该框架有助于合成完整的推理路径，确保环境观测与最终任务目标之间的逻辑一致性。

Paraphrasing Description Prompt

Instructions:

Based on the following scene graph, generate {num_variations} NEW and DIFFERENT correct descriptions of the scene.
Scene Graph:
- Objects in the scene: {nodes}
- Relationships between objects: {edges}
Original description for reference (generate different ones): {original_description}

Requirements:

1. Generate {num_variations} CORRECT descriptions that accurately describe the scene
2. Include what objects are in the scene
3. Include spatial relationships between objects (e.g., bowl on table, book inside drawer)
4. IMPORTANT: The object names in the description must exactly match the names in the nodes list, including any suffixes or variations (e.g., if a node is "blue_container", use "blue_container" not "container")
5. Ensure that ALL relationships listed in the Relationships between objects are described in each description

Rephrasing Task Goals Prompt

Instructions:

Rephrase the following task goal into {num_variations} different way(s) of saying the same thing.
Original task goal: "{task_goal}"

Requirements:

1. Keep ALL object names EXACTLY the same
2. Only change the verb phrases and sentence structure
3. The meaning must remain identical
4. Be concise and natural

Diversifying End-State Descriptions Prompt

Instructions:

The task "{task_goal}" has been completed. Please write {num_variations} different brief sentences (one sentence each) that describe the task completion status.

Requirements:

1. Be concise and natural
2. Describe that the task "{task_goal}" has been done/finished/completed
3. Use different wording and phrasing for each sentence
4. Do NOT start with prefixes like "Task completed:", "Done:", "Finished:", etc.
5. Just write a natural sentence describing the completion

图 10：**数据增强提示分类**。该框架包含三种层级策略：改写描述以丰富场景落地、任务目标改写，以及终态多样化，从而促进规划和目标解释任务中的数据扩展。

排列场景图文本中的物体和关系顺序，但不改变实际的空间关系（例如，将“A 在 B 上，B 在 C 上”改为“B 在 C 上，A 在 B 上”）。这能教会模型关注底层结构，而非输入顺序。

– **替换对象名称**（×2）：我们使用备选名称列表来一致地替换 SG 及其描述中的常见对象名称（例如，将“cup”替换为“mug”）。这确保模型理解物体的功能类别，而不仅仅是记忆某个特定标签。

– **改写任务目标**（×2）：我们使用 GPT-5 以不同风格改写原始任务指令（例如，将“把书放进包里”改为“将教材放入背包内部”），提示词如图 10所示。这有助于模型

处理人类可能发出的各种指令方式。– **多样化终态描述**（×4）：对于目标解释任务，我们使用 GPT-5 为同一最终场景生成四种不同的文本描述，提示词如图10所示。这有助于模型更可靠地“想象”任务成功完成后的样子。

通过上述系统化的筛选、清理、标注和增强流程，Manip- Cognition 数据集得以扩充，为所有推理模态提供密集监督。最终的数据集构成以及各增强因子的具体影响汇总于表V。

C. 仿真评估详情

原版 RLBench 和 LIBERO 系统通过图像和关节状态信息执行任务，并通过低级控制命令直接与模拟器交互。为将大语言模型集成到 RLBench 和 LIBERO 中，并评估其在不同场景下的任务性能，如图11所示，我们通过加入 HCI （高级控制接口，参见 C3 ）和 SGG （场景图生成器，参见C2）来增强现有系统。这些增强使系统能够根据模型生成的动作命令有效控制机器人，同时利用场景图表示环境状态的变化。

表 VI：RLBench 和 LIBERO 状态

状态类型	RLBench 状态	LIBERO 状态
机器人状态	持有/空置	持有/空置
空间关系	下方/上方/旁边，内部/包含	上方/上方/内部/堆叠
物体状态	打开/关闭，开启/关闭，空/满，折叠/展开	打开/关闭，开启 开启/关闭

表 VII：RLBench 与 LIBERO 物体

类别	物体
物体操作与放置	长方体、平面、球体、圆柱体、立方体，三棱柱、书、瓶子、瓶盖、鸡，巧克力果冻、咖啡、电脑、饼干，杯子、盘子、一叠美元、罐子、罐盖、刀，笔记本电脑支架、盖子、月亮、芥末、手机，手机壳、植物、盘子、插头、垃圾，炖锅、炖锅盖、带锅铲的勺子，锅铲、鞋子、汤、午餐肉、星星、牛排、棍子，草莓果冻、糖、西红柿、金枪鱼、电视遥控器，雨伞 ，usb，酒瓶 ，桌子，底座、底部、抽屉支腿、物品、左侧板、左侧框架、左侧把手、支架
容器与开关操作	箱子、书架、箱底、箱盖、橱柜、橱柜底座、碗柜、碗碟架、门、门框 ，门把手 ，抽屉 ，抽屉框架、冰箱底座、烤架、刀座，微波炉、微波炉门、烤箱、烤箱门、盘架、置物架、保险箱、保险箱门，搁板、开关、水龙头、马桶、马桶座圈向上，tv，洗衣机，窗户
堆叠与组装任务	篮子、番茄酱、积木金字塔积木、积木金字塔干扰积木、叠叠乐长方体积木，形状分类玩具、堆叠积木干扰物、堆叠积木目标物
控制与操作任务	奶油奶酪、黄油、桌面收纳盒、酒架，灯泡、灯泡灯座、充电器、灯座、灯螺丝、灯罩、电灯泡、烤箱旋钮，浇水壶
烹饪与厨房操作	平板炉、Chefmate 煎锅、摩卡壶，木托盘、砧板、烤盘，放置物品、任务墙

1）模拟器信息：如表VII所示，我们在 RL-Bench 和 LIBERO 平台上系统评估了五类场景，涵盖物体操作与放置、容器与开关操作、堆叠以及

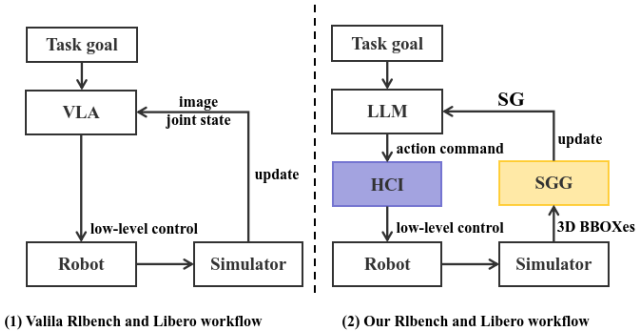

图 11：标准仿真环境与我们提出的框架的架构比较。

装配、控制与操作，以及典型的厨房烹饪场景。在 RLBench 中，这五类场景涵盖 70 项任务，涉及 107 种物体类别。在 LIBERO 中，这些场景涵盖 130 项任务，涉及 37 种物体类别。

算法 1： 场景图生成

输入： 关节值 J、夹爪值 G 和三维位姿 $P[i] = (x_{\min},\ x_{\max},\ y_{\min},\ y_{\max},\ z_{\min},\ z_{\max})$，其中 $i \in \{A, B, C, \dots\}$ 为物体编制索引。

输出： 场景图：

“节点”：[物体]，

“边”：[机器人状态、空间关系、物体状态]

定义：

J_τ：关节开合状态的阈值。

G_τ：夹爪开合状态的阈值。

$Vol(A)$：物体 A 的体积。

$Vol(B)$：物体 B 的体积。

$Vol(A \cap B)$：A 和 B 的交集体积。

对于 $i \in \mathcal{O}$ 其中 (J, G, P_i) **执行**

  如果 $G > G_\tau$ **then**

    夹爪状态 $\leftarrow$ 抓取

  如果 $J > J_\tau$ **则**

    物体状态 ←打开

  else

    物体状态 $\leftarrow$ 关闭

  如果A 与 B 的相交体积，交并比 $> \tau$ **则**

    A 在 B 内部；B 包含 A

  计算 Z 轴：

  Z_A =姿态 A[2]， Z_B =姿态 B[2]

  如果 $Z_A < Z_B$ **则**

    A 在 B 下方

  else

    A 上方 B

返回场景图

2）仿真中的场景图生成：为了使大语言模型能够理解场景信息，我们引入场景图来表示机器人操作后的空间关系。如

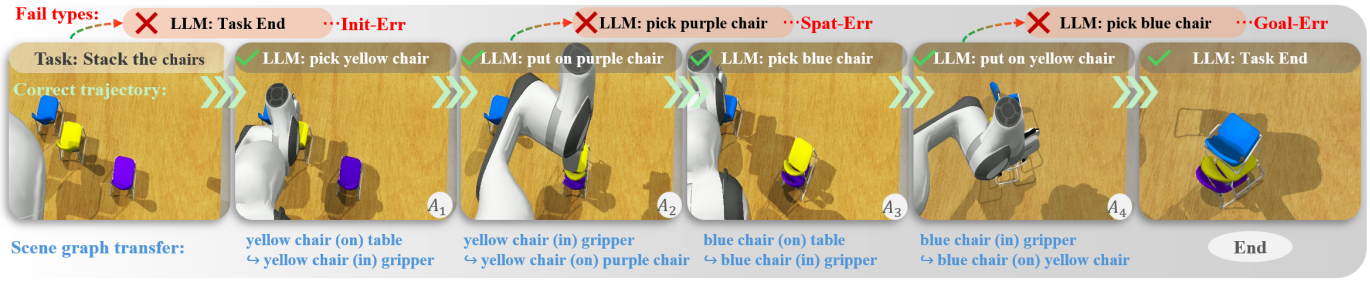

图 12：SA 中的任务案例。叠放椅子。正确的任务完成过程及模型生成的错误指令 ns.

表VI所示，我们使用抓取状态表示机器人夹爪位置，并用打开、关闭、开启和关闭表示物体固有状态。在场景图计算模块中，我们根据交并比（IOU）和预定义的“关系阈值”，确定独立物体之间的空间关系，例如“下方”和“上方”。对于具有关节连接的物体（如柜门、抽屉），我们计算子物体与主体物体之间的关节角度或相对位移。例如，柜门与柜体之间的角度（通过模拟器关节角度接口获取，或根据边界框计算），以及单个抽屉与柜体之间的相对位移（通过模拟器关节值获取，或根据边界框计算）。物体关系和物体状态可通过边界框交并比计算。结合开放世界环境中的六维姿态检测，即可在真实环境中快速部署并实时生成场景图。具体算法流程见算法1

3）高级控制接口：为了在 RLBench 和 LIBERO 环境中根据大语言模型指令进行运动规划，我们构建了一个机器人控制接口。对于环境中的每个物体，我们都定义了多条操作路径。我们以物体坐标系为参考，通过设置一系列不同的路径点，设计了不同的交互方式。例如，对于杯子，我们提供整体抓取、手柄抓取和倾倒等操作。每种操作都在物体坐标系中设计了对应的接近方式。同时，系统会获取目标物体的实时六维姿态，并将目标点和接近轨迹从物体坐标系转换到机器人坐标系。最后，系统采用经典的 RRT*算法，求解机器人当前位置到目标轨迹的完整路径。随后，系统使用逆运动学计算关节值，以生成路径规划和控制命令。底层控制器执行机器人控制任务，确保动作精准执行。

4) 命令接口： 为了将运动规划模块与大语言模型集成，我们开发了一套上层动作命令系统，使其能够直接与底层运动控制交互。该系统采用标准化命令格式：“动作类型 +目标物体”（例如，抓取苹果、放到桌子上）。随后，系统通过匹配来确定机械臂操作目标物体所需的姿态

算法2：动作执行引擎

1 输入：动作 $a \in \mathcal{A}$，目标物体 $o \in \mathcal{O}$2 输出：成功标志 σ，最终场景图 $\mathcal{SG}^*$，验证日志 $\mathcal{V}$3 $\mathcal{SG} \leftarrow$ 初始化场景图（），$\mathcal{V} \leftarrow \emptyset$4 如果 $o \notin \mathcal{O}$（$\mathcal{SG}$）则 5 返回 $(0, \mathcal{SG}, \mathcal{V})$6 $\mathcal{W} \leftarrow$ 检索路径点（$o, \mathcal{SG}$）7 $W^* \leftarrow \{W \in$ 按动作筛选（$\mathcal{W}$, a） 可行性判断（$W, \mathcal{SG}$）$\}$8 如果 $W^* = \emptyset$ 则 9 返回 $(0, \mathcal{SG}, \mathcal{V})$10 $W \leftarrow$ 选择（W^*） 11 对每个 $w_i \in W$ 执行12 $d_i \leftarrow \|\mathbf{x}_r - \mathbf{x}_{w_i}\|_2$ 13 $\mathcal{P}_i \leftarrow$ 插值规划$(\mathbf{x}_r, \mathbf{x}_{w_i})$ $d_i \leq \delta$ RRT规划$(\mathbf{x}_r, \mathbf{x}_{w_i})$，否则 14 如果 $\mathcal{P}_i = \emptyset$ 则 15 返回 $(0, \mathcal{SG}, \mathcal{V})$16 $\mathcal{SG} \leftarrow \Phi(\mathcal{SG},$ 执行$(\mathcal{P}_i))$ 17 $\mathcal{SG} \leftarrow \Phi(\mathcal{SG},$ 应用夹爪$(w_i, a))$ 18 $v_i \leftarrow \psi_a(o, \mathcal{SG})$，$\mathcal{V} \leftarrow \mathcal{V} \cup \{(i, v_i)\}$19 如果 $v_i = 0$ 则 20 返回 $(0, \mathcal{SG}, \mathcal{V})$ 21 $\mathcal{SG}^* \leftarrow$FinalizeSceneGraph($\mathcal{SG}$)22 **返回** $(1, \mathcal{SG}^*, \mathcal{V})$

指令中的动作类型，并指定夹爪控制方法。例如，如图 12所示，当前表现最佳的模型能够按正确顺序执行并完成 SA 中的“叠放椅子”任务：“LLM：拿起黄色椅子，LLM：放到紫色椅子上，LLM：拿起蓝色椅子，LLM：放到黄色椅子上”。然而，Claude-4-5 在 SA类任务上的表现不佳，会生成错误的动作序列，例如“LLM：拿起紫色椅子”或“LLM：拿起黄色椅子，LLM：放到蓝色椅子上”。

D. GSR-Bench 的详细信息

1) 仿真基础设施： GSR-Bench 基于一种轻量级的原生 Web 仿真框架构建，专为评估高层次的具身推理、空间理解和决策而设计。与侧重于低层次运动控制的传统模拟器不同，GSR-Bench 着重验证智能体解读复杂场景并规划长时程任务的能力。

为支持这一点，该基准测试提供了丰富的 63 种不同物体类型，范围从简单几何基本体到复杂可动家具。这些物体通过一组空间和逻辑关系（例如，上方、内部、已打开）相互连接，构成场景图的基础。详细资产库和已定义状态空间的概要见VIII。

表 VIII：GSR-Bench 资产库

类别	数量	典型示例
基本几何体	11	盒子（红色，黄色，蓝色），马克杯（红色，黄色，蓝色），立方体（红色，黄色，蓝色，绿色、白色）
可活动资产	10	可移动家具和电器，包括带有多个抽屉的橱柜和冰箱-箱，以及带有可用盒盖的盒子。
稳定 HOPE	14	标准化食品和调味品：牛奶纸盒、番茄酱和芥末酱瓶。
稳定扫描物品	11	高保真扫描厨房用具：盘子，碗（白色/红色）和托盘。
Turbosquid	17	多样化的商用3D模型：咖啡machines以及专用厨房电器。
总资产	**63**	一个多样化的集合，用于评估高-层任务规划与决策。

GSR-Bench 的技术架构旨在通过以下三个组件，连接高层语义规划与物理环境状态：

- **物理与渲染**：模拟器采用 cannon-es 进行刚体动力学模拟（重力、摩擦力和碰撞），以确保物理合理性。Three.js 提供 WebGL 渲染，为大规模数据合成提供一种基于浏览器、高吞吐量的替代方案，无需使用 Isaac Gym 等大型模拟器。
- **通信与动作接口**：通信与语义接口：ROS2 通过 rosbridge suite 充当通信中间件。ROSManager 负责协调系统，以 10Hz 的频率广播状态摘要，并监听高层语义命令（例如，将 [A] 移动到 [B]）。为专注于推理，模拟器在收到命令后直接更新物体状态，从而有效地将任务规划与机器人运动学解耦。

- **实时场景图提取**：场景图根据真实物理状态动态生成。空间关系（例如上方、内部）通过轴对齐包围盒（AABB）计算，而逻辑状态（例如，*is opened*）直接从可动关节物体的属性中提取。

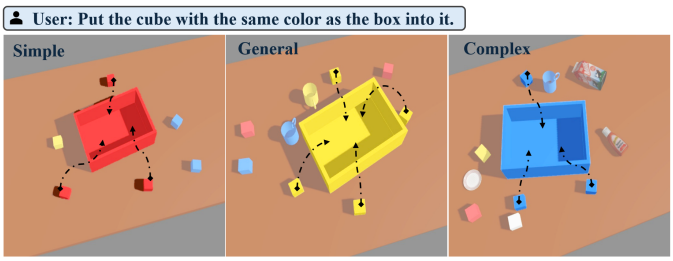

图 13：语义物体消歧任务示例

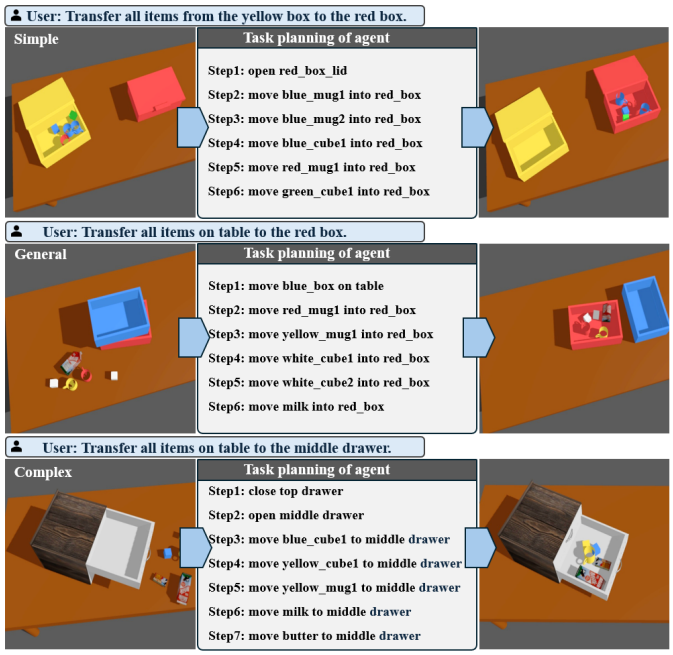

图 14：空间感知排序任务示例

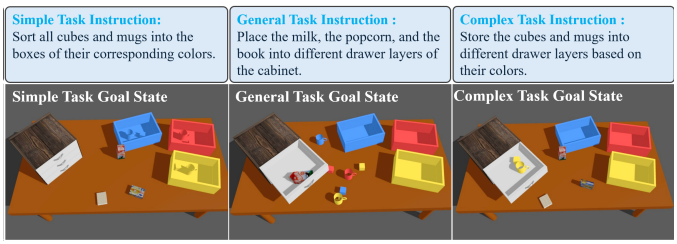

图15：目标条件泛化任务示例

2) 任务套件： GSR-Bench 提出了一套全面的评估方案，旨在评估具身智能体在三个基本认知领域的表现：语义物体消歧、空间感知序列规划和目标条件泛化。每个套件包含三个难度递增的级别，具体通过增加障碍物数量、施加更多物理约束或扩大指令范围来实现。各项测试的具体自然语言指令总结见表

IX。以下将介绍各套件的详细环境配置：

a) 语义物体消歧：该套件评估智能体在包含多种物体类别的场景地图中识别目标节点的能力。随着干扰元素密度增加、物体类别更加多样，场景的复杂度也随之提升。

- **简单**：场景中有一个盒子（随机为红色、黄色或蓝色）和 7–10 个颜色各异的立方体，放在桌子上。指令："拿起所有与盒子颜色相同的立方体，并将它们放进盒子里。"
- **一般**：场景中增加了 9–12 个颜色各异的立方体和马克杯，干扰项密度随之增加。指令："拿起所有与盒子颜色相同的立方体，并将它们放进盒子里。"
- **复杂**：除一般级别中的物品外，还会随机加入三件不同类别的物品（例如盘子、碗、牛奶）。指令："拿起所有与盒子颜色相同的立方体，并将它们放进盒子里。"

表 IX：GSR-Bench 各任务套件的指令规范

任务套件	级别	自然语言指令
语义物体消歧	简单 一般 复杂	"拿起所有与盒子颜色相同的立方体，并将它们放进盒子里。"
空间感知序列规划	简单	"将所有物品从黄色盒子转移到红色盒子。"
	一般	"将桌子上的所有物体移入红色盒子。"
	复杂	"将盒子里的所有物体转移到中间抽屉里。"
目标条件泛化	简单	"将所有立方体和马克杯分类放入与其颜色对应的盒子中颜色。"
	一般	"将牛奶、爆米花和书放入不同的抽屉橱柜的层。"
	复杂	"将立方体和马克杯存入不同的抽屉层，基于它们的颜色。"

b) 空间感知序列规划：该测试套件旨在评估智能体对物理因果链和目标指令的理解能力，重点在于识别并排除物理障碍，从而生成连贯的动作序列。

- **简单**：包含两个关闭的盒子（红色和黄色）。黄色盒子里有 5–8 件随机物品。智能体必须推断出明确的操作顺序：打开两个箱盖，以便取放物品。指令："将黄色盒子中的所有物品转移到红色盒子中。"
- **一般**：包含红色、黄色和蓝色盒子的堆叠配置，其中红色盒子始终位于底部。桌子上随机放置了 5–8 个物体。智能体必须分析 $OnTop$ 关系以清理堆叠然后才能访问目标箱子。指令："将桌子上的所有物体移入红色盒子。"
- **复杂**：场景中桌子上有 5–8 件随机物品，旁边有一个三层柜子，顶层抽屉处于打开状态。智能体必须识别到，打开的顶层抽屉会在物理上阻碍进入中间层。指令："将盒子中的所有物体转移到中间抽屉里。"

c) 目标条件泛化：该测试套件用于测试智能体将抽象的长时程目标转化为状态对齐动作的灵活性。环境布局保持不变，包含 3 个彩色盒子、6–10 个立方体/马克杯、一盒牛奶、爆米花和一本书，而指令的复杂程度则逐步提高。

- **简单**：专注于将特定的独特物品映射到不同的功能位置。指令："将牛奶、爆米花和书放入橱柜的不同抽屉层中。"
- **一般**：专注于基于属性的多物体分组和分类。指令："将所有立方体和马克杯按其对应的颜色分类放入盒子中。"
- **复杂**：需要进行跨物体逻辑映射，将语义属性与环境层级相匹配。指令："根据颜色将立方体和马克杯放入不同的抽屉层中。"