arXiv:2601.00555v1 [cs.RO] 2 Jan 2026

基于大语言模型的智能体探索：通过技能编排实现机器人导航与操作

Abu Hanif Muhammad Syarubany、Farhan Zaki Rahmani、Trio Widianto韩国科学技术院（*KAIST*）大田，韩国{hanif.syarubany、farhanzaki19、trio.widi}@kaist.ac.kr

摘要——**本文提出一种面向室内购物任务的端到端大语言模型智能体探索系统，并在 Gazebo 仿真环境和相应的真实世界走廊布局中进行了评估。机器人通过检测交叉口处的指示牌，逐步构建轻量级语义地图，并存储方向与兴趣点关系以及估计的交叉口位姿；AprilTag 则提供可重复使用的定位基准，用于接近和对齐。对于自然语言购物请求，大语言模型会在每个交叉口生成受约束的离散动作（方向以及是否进入商店），ROS 有限状态主控制器则通过门控模块化运动基元来执行决策，包括基于局部代价地图的避障、AprilTag 接近、进入商店和抓取。定性结果表明，集成系统能够实现从用户指令到多商店导航和物体取回的端到端任务执行，同时借助基于文本的地图和记录的决策历史，保持模块化且易于调试。**

索引词—**智能体探索、大语言模型、语义建图、移动操作、AprilTag、YOLO**

一、引言

视觉感知、语义建图和大语言模型（LLM）的最新进展，使移动机器人能够遵循高层指令，而非固定航路点。关键挑战在于整合：机器人必须感知离散线索、维护语义环境，并将语言层面的目标转化为安全、可执行的实时运动。

在本项目中，我们研究了基于走廊的购物环境中的智能体探索任务。该环境包含多个商店类别和一个取货点，并在 Gazebo [1]和相应的真实世界环境中进行评估。商店位置的信息并非以真实坐标的形式给出，而是分布在交叉口指示牌上，这些指示牌包含方向箭头、商店图标和AprilTag标签 [2]。机器人必须进行探索，根据这些线索构建语义地图，并在每个路口决定行进方向，以满足用户提供的购物订单。

我们实现了一个端到端 ROS 栈 [3] ，将大语言模型决策层与模块化底层控制器相结合。ORB-SLAM3 [4] 提供全局位姿，AprilTag 标签 [2] 作为稳定锚点，用于引导接近行为；感知节点则会生成轻量级 JSON 语义地图，记录交叉口与兴趣点的关系。根据任务、当前路口上下文和历史日志，大语言模型会输出受约束的离散命令<方向>|||<商店_动作>，然后由有限状态*MainController* 通过控制运动基元（避墙、接近标签、进入商店和抓取）来执行。

II. 相关工作

度量定位和建图是室内移动机器人的核心需求。基于特征的视觉同步定位与地图构建系统（如 ORB-SLAM）通过关键帧建图和闭环检测实现准确的单目位姿跟踪 [5]。ORB-SLAM2 将这一流程扩展到立体视觉和 RGB-D 感知，同时保持实时性能 [6]。ORB-SLAM3 进一步提升了鲁棒性，支持视觉惯性感知和多地图运行，这有助于在跟踪丢失后进行重定位 [4]。在我们的系统栈中，我们依靠同步定位与地图构建提供连续的全局位姿估计，供下游模块用于记录运动基元。

视觉标志广泛用于通过强大且可重复的几何锚点来补充同步定位与地图构建。AprilTag 提供了一种稳健的开放式视觉标志系统，能够从单个摄像头视图中实现精确的位姿估计 [2], ，而 AprilTag 2 提升了检测器在实际环境中的效率与稳健性[7]。在我们的场景中，嵌入指示牌的 AprilTag标签充当可靠的路口/入口参考，有助于稳定语义地图并支持精确的接近行为。

除了纯几何地图之外，语义表示还将地点与符号意义（例如商店类别）相关联，从而使任务能够围绕“是什么”以及“在哪里”来解决，而不仅仅依赖度量坐标。关于移动机器人语义建图的综述总结了常见的建模方式，以及度量表示、拓扑表示和混合表示之间的权衡 [8], [9]。我们的方法沿用了这一思路，维护一个轻量级的基于文本的语义地图，其中包含路口、方向和商店类别，既便于调试，也便于使用结构化上下文提示大语言模型。

近期研究也在探索将大语言模型用作具身智能体的高层规划器或策略。一种方向是利用语言模型，根据结构化状态描述推断可行的动作序列或子目标 [10]。其他方法则明确地将语言与机器人可供性相结合（例如，在约束条件下将指令映射为可执行技能）[11]，或生成可调用底层控制器的类代码策略 [12]。与端到端学习导航不同，我们的项目仅将大语言模型用作路口—商店语义图上的离散决策层；低层运动由独立的 ROS 控制器（沿墙行驶、标志接近、进入商店、恢复）执行，从而实现模块化并提升执行安全性。

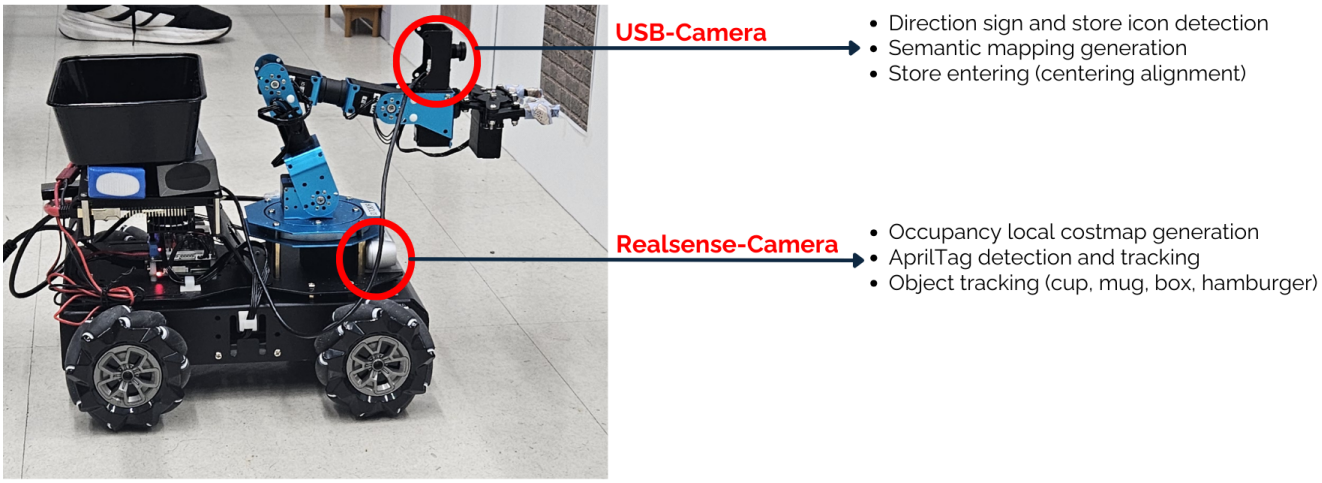


图 1：**所提系统使用的摄像头配置及其职责**。USB摄像头用于指示牌理解和商店入口居中，而RealSense深度相机则用于局部代价地图构建、AprilTag检测和物体跟踪。

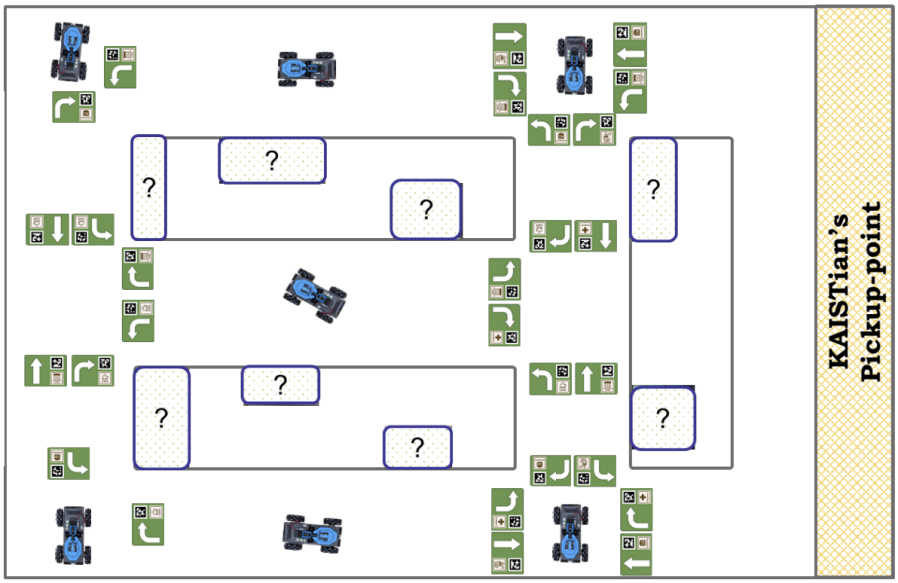
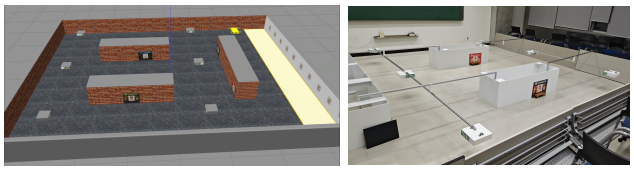


图 2：**基于走廊的购物环境地图布局**。交叉口指示牌（箭头、商店图标和AprilTag标签）支持语义建图和导航，右侧阴影区域标示取货点。

III. 方法



(a) 仿真地图　　(b) 真实地图

图 3：**本研究使用的环境布局**。(a)Gazebo 仿真和(b)相应的真实世界环境。两者具有相同的走廊结构和交叉口指示牌，用于评估。

我们在一个交叉口结构一致、以走廊为基础的购物环境中开发了智能体探索流程，并使用 Gazebo 仿真 [1] 和相应的真实环境布局（图 2 和图 3）。该系统旨在根据标牌线索构建语义地图，并通过与 ROS 运动基元相结合的大语言模型决策层，执行由语言指令驱动的任务。我们的传感器配置采用双摄像头方案（图 1）：RGB USB摄像头用于标牌／商店图标感知和入口居中，而 RealSense RGB-D相机则提供深度信息，用于局部代价地图构建、AprilTag感知 [2]，和抓取目标跟踪。

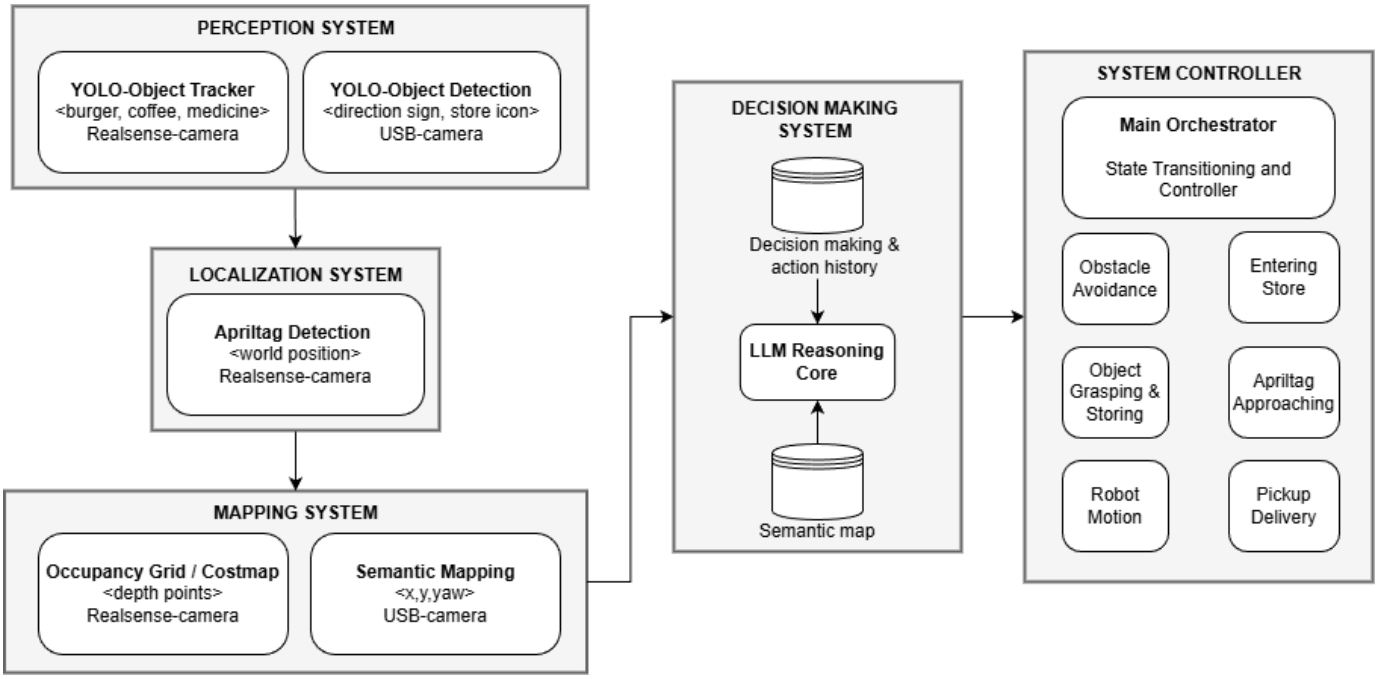


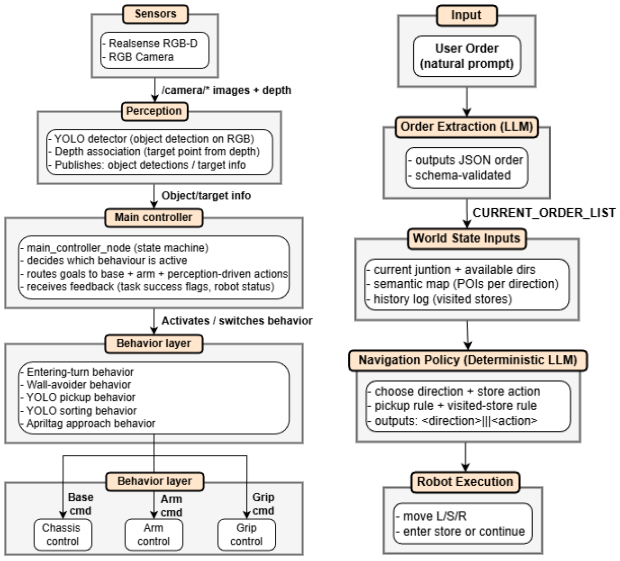
图 4：**智能体探索栈的系统级流水线，包括感知、定位、建图和大语言模型推理核心**。控制器层利用语义地图和决策历史，通过有限状态协调器触发底层技能。

表 I：YOLO 数据集中各类别的训练图像和验证图像分布。

类别	训练	验证	总计
负样本	1335 (4.6%)	340 (4.6%)	1675 (4.6%)
Cup	3258 (11.3%)	838 (11.2%)	4096 (11.3%)
Mug	3287 (11.4%)	853 (11.4%)	4140 (11.4%)
汉堡包	4851 (16.8%)	1258 (16.9%)	6109 (16.8%)
Box	3283 (11.4%)	850 (11.4%)	4133 (11.4%)
左转方向	1081（3.7%）	278（3.7%）	1359（3.7%）
右转方向	1105（3.8%）	286（3.8%）	1391（3.8%）
直行方向	1142（4.0%）	292（3.9%）	1434（3.9%）
取货点	1581（5.5%）	409（5.5%）	1990（5.5%）
Cafe	1644（5.7%）	428（5.7%）	2072（5.7%）
便利店	1574（5.5%）	406（5.4%）	1980（5.5%）
汉堡店	1575（5.5%）	408（5.5%）	1983（5.5%）
药学	1602（5.5%）	413（5.5%）	2015年（5.5%）
AprilTag	1557 (5.4%)	395 (5.3%)	1952 (5.4%)
总计	28875 (100%)	7454（100%）	36329（100%）

A. 目标检测开发

为同时支持标牌理解和抓取目标感知，我们通过记录仿真和真实环境运行过程中不同视角、光照条件及杂乱程度下的相机帧，筛选出具有代表性的正样本和难负样本，从而构建了自定义 YOLO [13] 数据集。为减少人工标注工作量，同时保留强监督信号，我们采用弱蒸馏流程：通过生成——



（a）架构流程　　（b）大语言模型智能体流程

图 5：**所提系统的软件流程**。(a) 从传感器和感知开始的 ROS 架构流程。(b) 大语言模型智能体流程：将自然语言指令转化为受约束动作 <方向>|||<存储_动作>，供机器人执行。

生成伪标签（类别标签和边界框），然后通过简单的合理性检查（例如尺寸/宽高比约束和重复项去除）对其进行优化。由此得到的数据集涵盖

算法 1 带门控的 AprilTag 接近与最终前进前进。

输入： 标签 检测结果 $(x,\ z,\ \psi)$，门控 g，目标 $(x^*,\ d^*)$，容差 $(\tau_x, \tau_z, \tau_\psi)$
参数： 增益 $(k_x,\ k_z,\ k_\psi)$，限值 $(v_{x,\max},\ v_{y,\max},\ \omega_{\max})$，超时 T，前进 $(d_{\text{adv}},\ v_{\text{adv}})$
启用←False；前进完成←False；$t_{\text{det}} \leftarrow -\infty$
如果 $g = 1$ **则**
    启用←True；前进完成←False
结束如果
如果 $g = 0$ **则**
    活动←假；发布停止指令；发布成功$= 0$
结束如果
当ROS 正在运行**执行**
    如果非活动**则**
        继续
    结束条件判断
    如果 新标签 **则**
        更新 $(x,\ z,\ \psi)$；$t_{\text{det}} \leftarrow$ 现在
    结束条件判断
    如果 现在 $- t_{\text{det}} > T$ **则**
        发布停止指令；发布成功$= 0$；**继续**
    结束条件判断
    $e_x \leftarrow x - x^*$，$e_z \leftarrow z - d^*$，$e_\psi \leftarrow \psi$
    如果 $|e_x| > \tau_x$ **或** $|e_z| > \tau_z$ **则**
        $v_y \leftarrow \text{sat}(-k_x e_x,\ \pm v_{y,\text{最大值}})$，$v_x \leftarrow$ 满足$(k_z e_z,\ \pm v_{x,\max})$
        $\omega \leftarrow$ 满足$(k_\psi e_\psi,\ \pm\omega_{\max})$；发布 $(v_x,\ v_y,\ \omega)$；成功$= 0$
    否则如果 $|e_\psi| > \tau_\psi$，**则**
        $\omega \leftarrow$ 满足$(k_\psi e_\psi,\ \pm\omega_{\max})$；发布 $(0,\ 0,\ \omega)$；成功$= 0$
    否则如果未前进完成，**则**
        向前行驶 $v_x = \min(|v_{\text{adv}}|,\ v_{x,\max})$，持续 d_{adv}/v_x 秒；然后停止
        高级任务完成←True；发布成功$= 1$
    结束 if
结束 while

算法 2 带门控有限状态机的局部代价地图墙壁避让。

输入： 局部占用栅格 M，门控 $g \in \{0, 1\}$，参数 $(d_{\text{safe}}, v_f, \omega, \theta_{90}, \theta_{180})$
状态： $s \in$ {前进，旋转 90 度 旋转180}，上次标签时间 t
$s \leftarrow$ 前进；发布 /wall_avoider_state$\leftarrow s$
当 ROS 运行时 **执行**
    如果 $g = 0$ **则**
        发布停止指令一次；$s \leftarrow$ 前进；**继续**
    结束如果
    $d \leftarrow$ 前方距离(M) ▷ 前向走廊中的最小障碍物距离切片
    如果 $s =$ 前进 **则**
        如果 $d \neq \emptyset$ **且** $d \leq d_{\text{safe}}$ **则**
            $s \leftarrow$ 旋转 90 度；$t \leftarrow$ 现在
        else
            发布 $(v_f,\ 0)$
        结束如果
    否则如果 $s =$ 旋转 90 度**则**
        如果 现在 $- t < \theta_{90}/|\omega|$ **则**
            发布 $(0,\ \pm\omega)$
        else
            发布停止指令；$d' \leftarrow$DistAhead(M)
            如果 $d' \neq \emptyset$ **则**
                $s \leftarrow$ 旋转 180 度；$t \leftarrow$现在
            else
                $s \leftarrow$ 前进
            结束 if
        结束 if
    否则如果 $s =$ 转向180 **则**
        如果 现在 $- t < \theta_{180}/|\omega|$ **则**
            发布 $(0,\ \pm\omega)$
        else
            $s \leftarrow$ 前进；发布 $(v_f, 0)$
        结束 if
    结束 if
    发布 /wall_避墙_状态$\leftarrow s$
结束 while

抓取目标（杯子、马克杯、汉堡包、盒子）、方向箭头（左/右/直行）、语义线索（商店图标和取货点）以及AprilTag标签（表 I）。其中，各类别的数量存在一定程度的不平衡（例如，在抓取物体中汉堡包占主导地位），反映了我们环境中的任务分布。重要的是，我们纳入了大量负样本，使检测器能够接触到视觉上相似但不可执行的场景，从而减少误报，并提高触发下游行为（例如商店入口对齐和抓取）时的稳定性。我们训练并部署了 YOLO11-small，作为贯穿整个系统使用的实时检测器。

B. 系统架构概述

我们的系统是一个端到端 ROS 栈，在模拟购物中心环境中连接感知、语义建图、基于大语言模型的决策和低层执行。如图 4 所示，原始传感器流经感知和定位模块处理后，生成 (i) 用于安全移动的局部度量表示，以及 (ii) 用于高层推理、记录路口与商店关系的符号语义地图。

前端由两条感知流水线组成：用于抓取目标的目标检测与跟踪，以及用于导航提示的标牌理解；同时采用基于 AprilTag 的定位，提供可靠的路口 / 入口锚点（图 4）。此外，建图层会维护基于占用栅格的局部代价地图以进行避障，并维护一个轻量级语义地图（以 JSON 格式存储），记录每个已发现路口的方向到兴趣点的映射对，以及估计的二维位姿。

控制后端遵循图 5(a) 所示的分层架构。*MainController* 有限状态机通过锁存的门控话题协调模块化行为（避墙、AprilTag 接近、进入商店、抓取和恢复），确保同一时刻只有一个技能能够控制 /cmd_vel。这种设计支持在不同行为之间稳健切换，而无需使用端到端学习控制器。

最后，大语言模型智能体流程如图 5(b) 所示。首先将自然语言用户指令转换为经过模式验证的结构化指令列表，然后将其与当前交叉口上下文、语义地图和动作历史日志相结合，生成受约束的离散命令，其形式为 <方向>|||<商店_动作>。主控制器使用该命令触发适当的导航和操作低层行为，同时确保安全性和运动执行保持在本地且具有确定性。

C. 感知与语义建图

1）指示牌与 *AprilTag* 感知：Gazebo 世界在主要路口设有指示牌。每块指示牌通过其位姿编码一个交叉路口标识符，并包含一个或多个方向箭头、商店类别图标，以及一个用作可靠基准标记锚点的 AprilTag。专用感知节点（由小组提供）订阅 RGB-D 相机话题和

算法 3 MainController 有限状态机（运动门控锁存）。

禁用所有门控； $S \leftarrow$ 避墙
当ROS 正在运行**时**
  如果 $g_m = 0$ **则**
    全部禁用；发布停止指令； $S \leftarrow$ 避墙；**继续**
  结束如果
  如果 $S =$避墙门控 **则**
    启用避墙门控
    如果标签近期 $\wedge$ 未转向 $\wedge$ 不在冷却期内 **则**
      $S \leftarrow$ 标签接近状态
    结束如果
  否则如果 $S =$标签接近状态 **则**
    启用 AprilTag 接近门控
    如果接近成功 **则**
      启动映射计时器； $S \leftarrow$ 映射
    结束如果
    如果 标签丢失 $\wedge$ 上次距离 $> d_{\text{far}}$ **则**
      $S \leftarrow$ 避墙
    结束 if
  否则，如果 $S =$映射 **则**
    启用语义映射门控
    如果 已过去 $\geq T_{\text{map}}$ **则**
      停止建图； $S \leftarrow$ WAITACTION
    结束 if
  否则如果 $S =$等待动作 **则**
    如果 收到动作 $a = (dir,\ target)$ **则**
      触发进入前(dir)； $S \leftarrow$ 进入前
    结束如果
  否则如果 $S =$进入 **则**
    如果 $r_p \neq$成功 **则**
      $S \leftarrow$ 避墙
    结束条件判断
    如果 $r_p =$成功 **则**
      如果 $target =$ 取货 **则**
        启用取货门； $S \leftarrow$ 取货
      结束条件判断
      如果 $target =$ 继续**那么**
        $S \leftarrow$ 避墙
      结束如果
      启用进入商店$(target)$; $S \leftarrow$ 进入商店
    结束如果
  否则如果 $S =$进入商店 **则**
    if $r_e =$成功 **则**
      启用抓取门控； $S \leftarrow$ 抓取
    结束条件判断
    如果 $r_e \neq$成功 **则**
      $S \leftarrow$ 避墙
    结束条件判断
  否则，如果 $S =$抓取 **则**
    如果 $s_g = 1$ **则**
      抓取后转向（方向与 dir_{prev} 相反），然后定时后退； $S \leftarrow$ 避墙
    结束如果
  否则，如果 $S =$拾取 **则**
    永久开启取货门；忽略新操作
  结束如果
结束循环

AprilTag检测话题，并将每条路标观测转换为一条符号化的路口记录。记录包含观测到指示牌时的机器人位姿（位于里程计坐标系中）、路口名称以及“兴趣点对”列表。这些记录通过语义地图话题发布，并追加到持久化 JSON 文件中。

2）语义地图表示：语义地图以路口条目列表的形式存储。每个条目都包含一个唯一标识符（例如，“junction_3”）、方向–兴趣点字符串列表，以及发现该路口时估计的二维位姿（x、y、yaw）

算法 4 大语言模型导航规则。

要求： 语义地图 $\mathcal{M}$、当前路口 J、剩余指令 R、历史日志 H
确保： 仅输出 <方向>|||<动作>，使用 $dir \in \{1,2,3\}$ (L/S/R) 和 $act \in \{1,2,3,4,5\}$ (商店/继续)
从 H 中提取已访问的商店实例（路口，方向，兴趣点）和已尝试的分支（路口，方向）
如果 $R = 0$ **那么** ▷购物完成
  前往取货点（如果位于 J，则朝其方向前进）；输出 <方向>|||5
否则 ▷购物进行中
  通过将商店类型与 $\mathcal{M}$中的 R中的剩余物品
  如果该实例位于 J **则**
    现在输入：输出 <方向>|||<商店_编号>
  结束 if
  否则移动：朝目标前进（或者选择一个尚未尝试的分支，如果没有已知的有用商店）
  如果有其他选择，则避免取货方向 $R > 0$；输出 <方向>|||5
结束 if

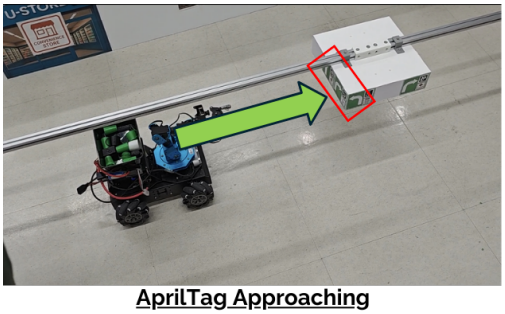
。这种表示方式有意将符号知识（“左转通往一家药学商店”）与度量知识（路口的大致位姿）分开。它足够简单，便于检查或手动编辑，同时又足够丰富，使大语言模型导航策略能够围绕交叉口、商店类型和取货点进行规划，而无需处理原始几何信息或连续轨迹。

3) 历史轨迹记录：为了支持智能体行为和事后分析，历史建图节点会订阅语义事件和高层动作决策。当机器人停在路口或执行转向指令时，该节点会将一条文本日志写入历史 JSON 文件。随着时间推移，该文件会记录已访问过的交叉口、采取过的方向以及进入过的商店实例。导航大语言模型会利用该日志，避免不必要地再次访问同一商店实例、推理尚未尝试的分支，并重建任务的完成进度。

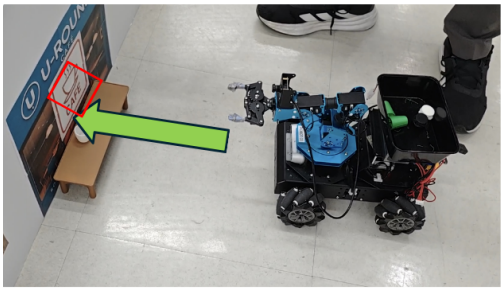
D. 基于大语言模型的导航与任务解析 g

1) 从自然语言解析订单：运行开始时，用户会通过专用的 ROS 话题以自然语言指定购物任务（例如，“带两个汉堡和一份急救药品”）。Python 节点会订阅该话题，并将原始文本转发给一个受 JSON 模式约束的大语言模型提示。模型必须输出一个包含四个非负整数的字典，分别表示所需的汉堡、冰咖啡、热咖啡和药品数量。该节点会对结果进行后处理，执行基本的合理性检查，并将规范化后的订单列表保存到 JSON 文件中。此文件被视为真实购物清单，并由导航策略和物体抓取模块共同读取。

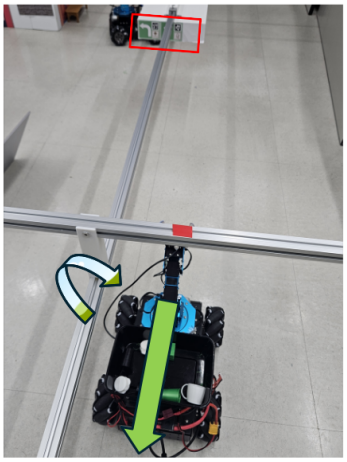
2）基于语义地图的导航策略：决策层的核心是 DecisionMakers 节点。每当机器人在某个路口完成语义建图后，该节点会收集：(i) 语义地图中当前路口的条目，(ii) 已发现的所有交叉口及其兴趣点对的完整列表，(iii) 当前订单列表，以及 (iv) 轨迹日志中此前决策的文字记录



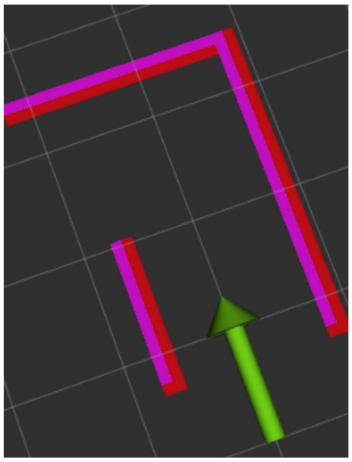
AprilTag Approaching
PID control-based x,z,yaw alignment and positioning



Store Entering
Center alignment



Robot re-positioning
Rotate and backward motion to approach apriltag



Local Costmap-based
Obstacle Avoidance, Wall centering alignment

图 6：**探索和语义地图构建期间采用的代表性运动策略**，包括基于 AprilTag 的接近、底盘重新定位、局部代价地图避障和商店入口对齐。主控制器会触发这些基元，以便可靠地到达指示牌和入口，同时确保在走廊中安全导航。

。随后，它会构建结构化提示，并调用导航大语言模型。

系统提示（算法 4）定义了确定性导航策略。大语言模型必须输出形如 <DIRECTION>|||<STORE_ACTION> 的离散命令，其中第一段代码用于选择左转、直行或右转，第二段代码用于决定现在是否进入某种特定商店类型，还是继续搜索。提示还规定了如何处理不同的商店功能、如何处理各个商店实例（通过路口名称、方向和兴趣点名称识别），以及在收集完所有所需物品后，何时优先前往取货点。节点会解析这两个整数代码，验证所选方向是否存在于当前路口，最后发布文本命令。重要的是，大语言模型不会生成速度指令；它只决定采取哪个语义分支，以及是否进入该分支处的商店。

E. 低层运动执行与恢复

1）运动基元与局部代价地图控制器：低层运动由传统 ROS 控制器处理，它们作用于 /cmd_vel，并使用局部代价地图确保安全。局部代价地图节点会将深度信息或超声波读数融合为机器人周围的二维占据栅格。附近障碍物所在的栅格单元具有较高代价，而机器人前方的空闲区域则显示为较低代价。避墙控制器（算法 2）将该局部代价地图与期望方向（左／直行／右）结合起来，生成平滑的速度指令，使机器人沿相应分支行进，同时与障碍物保持安全距离。

其他运动基元包括 AprilTag 接近控制器（算法 1），用于使机器人对准标牌或商店入口并在其前方停下；“进入前”控制器，用于根据选定的门口方向将机器人居中；以及入店控制器，用于驱动机器人进入商店并在稳定的位姿下停下。这些基元以小型状态机实现，以固定频率发送速度指令，并使用里程计和距离测量，而不依赖任何大语言模型反馈。

2）主控制器与恢复状态：MainController 节点按照算法 3 中的描述，使用有限状态机来协调所有运动基元。主门控话题激活后，控制器从墙壁避障状态开始：它启用墙壁避障节点（因此也启用局部代价地图），使机器人能够安全地探索走廊，直到检测到 AprilTag。检测到附近的标签后，控制器切换至 AprilTag接近状态（算法 1），随后进入语义建图状态，让感知节点更新语义地图。建图完成后，控制器进入等待下一步动作状态，并阻塞等待大语言模型通过 /next_controller_action 发送新的高级命令。根据该命令（如图 6 所示），控制器会触发预进入、进入商店、多物体抓取、抓取后

转向或取货行为，每种行为都有各自的成功主题。

为提高鲁棒性，控制器会监控简单的进度指标，例如各基元的成功标志和超时情况。如果某个基元报告失败，或机器人未能取得进展，控制器便会退回到基于局部代价地图的避墙控制，或激活特殊的拾取状态，将控制权交给专用的拾取控制器。无论哪种情况，恢复逻辑都保持局部且具有反应性；机器人到达新的路口且语义建图完成后，控制权便会通过下一动作主题交还给大语言模型导航策略。

IV. 实验与结果

A. 端到端任务执行

我们在 Gazebo 仿真环境和与之对应的真实环境中评估了完整流程，这两种环境均包含多家商店和一个取货点（图 7）。每个回合开始时，机器人从取货区域附近出发，此时语义地图为空。在初始探索阶段，墙壁避障控制器利用局部代价地图驱动机器人穿过走廊，同时与障碍物保持安全距离；感知节点则在交叉口检测指示牌，并将路口位姿以及方向与商店类别之间的符号关系记录在共享的 JSON 地图中。

提供用户目标后，系统会从自然语言指令到物体拾取，完成一项端到端的完整任务。基于大语言模型的导航策略会读取当前语义地图和已访问位置的历史记录，并在每个路口输出下一步高层动作。随后，系统会将生成的指令转发给主控制器，由其触发相应的运动基元：转入选定的岔路、使用局部代价地图沿走廊行进、接近下一个 AprilTag，以及在需要时运行入店前流程和入店流程。当机器人到达目标货架附近时，系统会触发独立的抓取控制器，使末端执行器对准目标并执行物体拾取。

从定性结果来看，该集成系统能够在仿真和真实场景中完成多商店任务：经过短暂的探索阶段后，机器人可以按合理顺序访问所需的商店类别，然后返回起始区域附近。高层语义决策（大语言模型）与低层运动执行（基于局部代价地图的导航和抓取节点）之间的明确分工，使我们能够更改或改进策略逻辑，而无需修改底层控制器。

B. 语义建图

语义建图模块会将从路口检测到的指示牌信息汇总为紧凑的 JSON 表示。每个条目都记录观察时的机器人位姿以及方向关系列表。由于这种表示形式不依赖任何特定控制器，因此既可供大语言模型导航策略使用，也可用于调试工具。

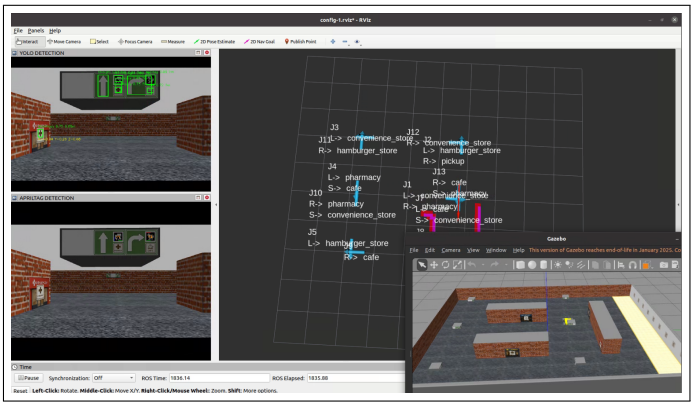


图 7：**探索过程中的系统概览**。展示用于语义地图和轨迹可视化的 RViz，以及 Gazebo 仿真。

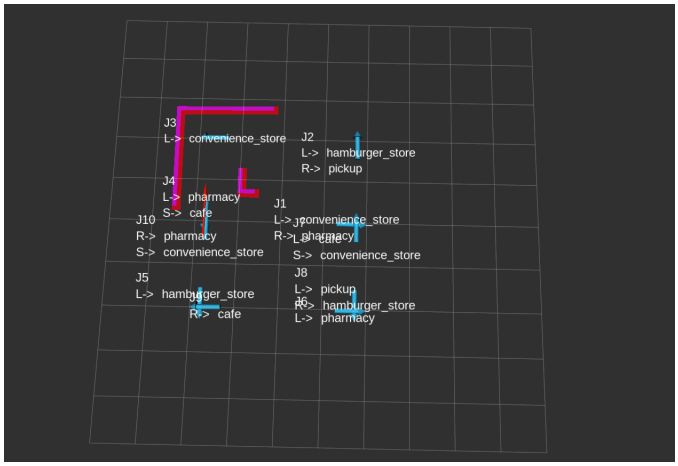


图 8：**探索过程中在线构建的语义地图**。每个路口都存储与商店类别相关的方向关系（左/直行/右）。

实际上，机器人在避墙控制下探索新走廊时，地图会逐步扩展（图 8）。指示牌只要被看到过一次，其信息就会在本回合余下时间内保留，使大语言模型能够推理当前视野之外、但先前观察到的商店。将感知（指示牌解析）与推理（大语言模型）解耦后，无需重新设计建图组件，就能在同一环境中执行多个任务。

C. 大语言模型策略与历史日志

大语言模型导航策略接收两个主要输入：当前语义地图，以及一份历史日志，用于描述已经访问过的交叉口和商店实例及其访问顺序。给定用户请求（例如按商品类别列出的购物清单），导航节点根据这些输入构建结构化提示，并要求大语言模型以受限格式<方向>|||<商店_动作>输出离散高层动作。

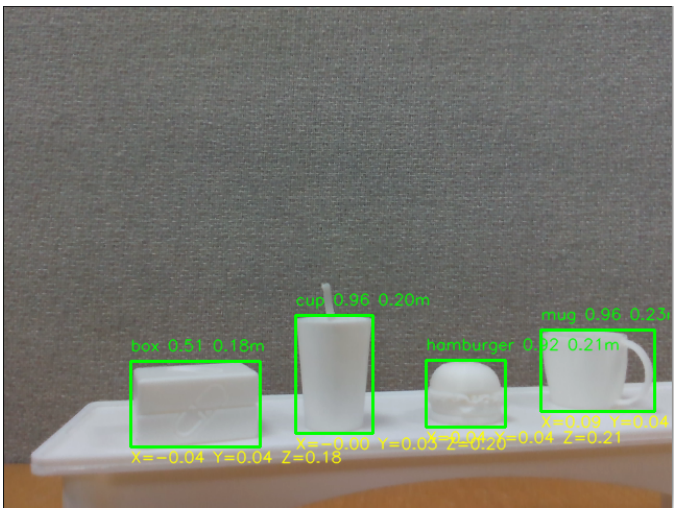


图 9：**多物体抓取的感知输出示例**。叠加信息显示了类别标签、检测置信度和基于深度的位置估计，为抓取控制器提供指引。

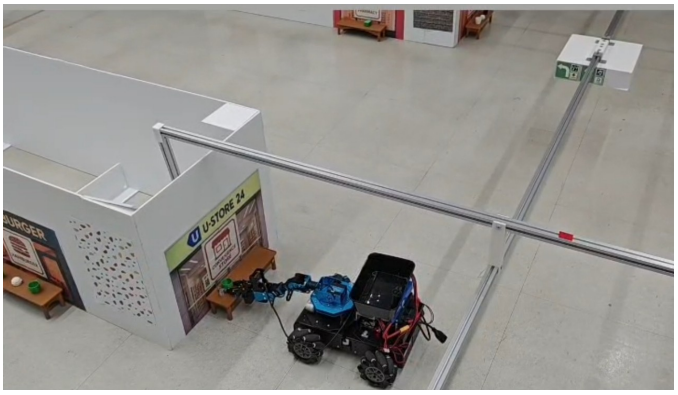


图 10：**机器人在商店环境中执行物体取回的快照**。通过导航基元完成底座定位后，机械臂控制器会执行预定义抓取序列，拾取所请求的物品。

我们将每次大语言模型查询及其响应记录到一个 JSON 文件中，从而能够重建回合期间所作决策的序列。通过这些日志，我们观察到，在运行初期，策略倾向于选择能够扩大未探索路口覆盖范围的动作；而在后期，决策则集中于完成购物清单上的剩余物品，并在所有计数归零后引导机器人返回取货区域。尽管没有明确施加最优性目标，但最终生成的序列通常是合理的，并且除非确有必要，否则会避免重新进入已经完成任务的商店实例。

D. 抓取与物体取回

机器人导航到目标商店或货架附近后，即会触发抓取。在我们的设置中，低层导航控制器负责将底盘利用基于 AprilTag 的对齐和局部代价地图，将底盘引导至合适的接近位姿，随后由专用的抓取节点（在之前的作业中开发）接管。如图 9 所示，抓取流水线首先生成目标物体的类别标签、置信度分数和基于深度的位置估计，然后据此对齐末端执行器，再执行预定义的取物动作。

在成功的试验中，机器人能够靠近货架，将夹爪合拢在目标物体周围，并在不发生碰撞的情况下后退。图 10 展示了机器人进入商店后执行物体取回的一个真实场景示例。从大语言模型和导航策略的角度来看，抓取是一项原子动作：到达商店后，抓取节点会运行至完成并报告成功或失败，随后由高层策略决定接下来要前往哪家商店或哪个地点。

E. 失败案例与局限性

尽管集成系统总体上表现成功，但仍观察到一些局限性，主要包括接近过程中标签丢失，以及物体拥挤或排列过于紧密。

1）接近过程中的标签丢失：AprilTag 检测结果有时会因观察距离、角度，或被货架和附近物体部分遮挡而丢失。当这种情况发生在商店入口附近时，接近控制器可能无法获得稳定的相对位姿，导致接近动作中止或出现偏差，尽管局部代价地图仍能防止严重碰撞。

2）拥挤或紧密排列物体：当目标物体彼此之间，或与周围几何结构之间的距离过近时，预定义的抓取动作可能导致指尖碰撞或不稳定抓取。在某些情况下，机器人仍能抓起物体，但抓持较为勉强，容易受到后续运动或底座产生的轻微位姿误差的干扰。

这些失败模式表明了若干改进方向，例如加入用于重新获取标签的简单启发式方法、使底座在货架附近减速，或引入短程局部搜索，在触发抓取动作前调整最终底座位姿。

五、结论

我们提出了一套端到端的 ROS 流水线，用于在基于走廊的购物任务中实现大语言模型驱动的智能体探索。该系统集成了：(i) 感知与 AprilTag 辅助定位，(ii) 以紧凑的 JSON 格式记录路口到商店关系的在线语义地图，(iii) 输出受约束高层动作的大语言模型决策层，以及 (iv) 用于导航、进入商店和抓取的有限状态协调器，通过门控协调模块化低层技能。在 Gazebo 和与之匹配的真实世界布局中开展的定性实验表明，机器人能够根据自然语言指令执行完整任务，涵盖多商店导航和物体取回。

我们将高层语义决策与低层确定性控制器模块化分离，简化了开发并提高了执行稳定性，因为策略变更无需重新设计运动控制。尽管如此，我们

仍观察到一些实际故障模式，例如入口附近偶发的标签丢失，以及目标拥挤或间距过小时抓取鲁棒性下降。未来的改进方向包括提高标签重新获取和入口对齐的可靠性、在货架附近采用自适应减速和局部搜索，以及加强不确定性感知与恢复行为之间的协同。

参考文献

[1] N. Koenig 和 A. Howard，《Gazebo：开源多机器人模拟器的设计与使用范式》，载于*IEEE/RSJ*智能机器人与系统国际会议（*IROS*），2004年。[2] E. Olson，《AprilTag：一种稳健灵活的视觉基准标记系统》，载于*2011*年*IEEE*国际机器人与自动化会议（*ICRA*），2011年，第3400–3407页。[3] M. Quigley、K. Conley、B. Gerkey、J. Faust、T. Foote、J. Leibs、R. Wheeler 和 A. Y. Ng，《ROS：一种开源机器人操作系统》，载于*ICRA*开源软件研讨会，2009年。[4] C. Campos、R. Elvira、J. J. G´omez Rodr´ıguez、J. M. M. Montiel 和 J. D. Tard´os，《ORB-SLAM3：用于视觉、视觉惯性和多地图SLAM的高精度开源库》，*IEEE*机器人学汇刊，第37卷，第6期，第1874–1890页，2021年。[5] R. Mur-Artal、J. M. M. Montiel 和 J. D. Tard´os，《ORB-SLAM：一种多用途、高精度的单目SLAM系统》，*IEEE*机器人学汇刊，第31卷，第5期，第1147–1163页，2015年。[6] R. Mur-Artal 和 J. D. Tard´os，《ORB-SLAM2：一种适用于单目、立体和RGB-D相机的开源SLAM系统》，*IEEE*机器人学汇刊，第33卷，第5期，第1255–1262页，2017年。[7] J. Wang 和 E. Olson，《AprilTag 2：高效且稳健的基准标记检测》，载于*2016*年*IEEE/RSJ*智能机器人与系统国际会议（*IROS*），2016年。[8] I. Kostavelis 和 A. Gasteratos，《面向移动机器人任务的语义建图：综述》，《机器人与自主系统》，第66卷，第86–103页，2015年。[9] S. Garg、N. S¨underhauf、F. Dayoub、D. Morrison、A. Cosgun、G. Carneiro、Q. Wu、T. Chin、I. D. Reid、S. Gould、P. Corke 和 M. Milford，《机器人地图构建、感知与交互中的语义：综述》，*CoRR*，第abs/2101.00443卷，2021年。 [在线]。可访问：https://arxiv.org/abs/2101.00443 。W. Huang、P. Abbeel、D. Pathak 和 I. Mordatch，《将语言模型用作零样本规划器：为具身智能体提取可执行知识》，载于国际机器学习会议（*ICML*），《机器学习研究论文集》系列，第162卷。PMLR，2022年，第9118–9147页。 [在线]。可访问：https://proceedings.mlr.press/v162/huang22a.html [11] M. Ahn、A. Brohan、N. Brown、Y. Chebotar、O. Cortes、B. David、C. Finn、C. Fu、K. Gopalakrishnan、K. Hausman、A. Herzog、D. Ho、J. Hsu、B. Ichter、A. Irpan、E. Jang、S. Jesmonth、N. Joshi、R. Julian、D. Kalashnikov、Y. Kuang、Q. V. Le、L. Lee、S. Levine、Y. Lu、N. Mourad、R. Omoindrot、K. Rao、K. Reymann、L. Riano、M. S. Ryoo、R. Salakhutdinov、N. Sievers、T. Tan、G. Tucker、V. Vanhoucke、F. Xia、A. Zeng、P. Zhang 和 B. Zhu，《做我能做的，而非我所说的：以机器人可供性为语言奠定基础》，*arXiv*预印本，2022年。

[10] [12] 2204 01691 在线。可访问：https://arxiv.org/abs/。J. Liang、W. Huang、F. Xia、P. Xu、K. Hausman、B. Ichter、P. Florence 和 A. Zeng，《代码即策略：用于具身控制的语言模型程序》，载于*2023*年*IEEE*国际机器人与自动化会议（*ICRA*），2023年，第9493–9500页。[13] G. Jocher 和 J. Qiu，《Ultralytics YOLO11》，2024年。 [在线]。可访问：https://github.com/ultralytics/ultralytics [14] S. Bai、K. Chen、X. Liu、J. Wang、W. Ge、S. Song、K. Dang、P. Wang、S. Wang、J. Tang、H. Zhong、Y. Zhu、M. Yang、Z. Li、J. Wan、P. Wang、W. Ding、Z. Fu、Y. Xu、J. Ye、X. Zhang、T. Xie、Z. Cheng、H. Zhang、Z. Yang、H. Xu 和 J. Lin，《Qwen2.5-VL技术报告》，2025年。[在线]。可访问：https://arxiv.org/abs/2502.13923