

开放世界导航：多模态大语言模型

Mingfeng Yuan¹, Letian Wang¹ and Steven L. Waslander¹

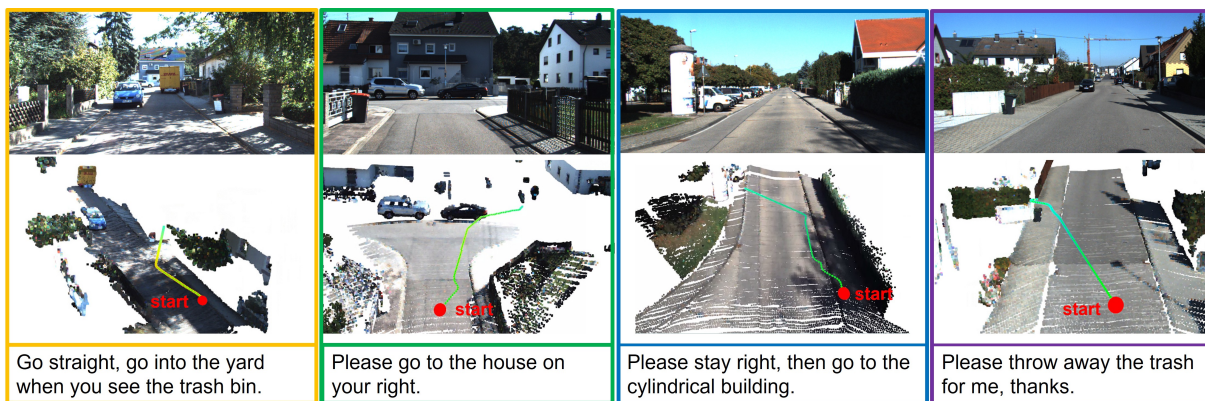


Fig. 1: 给定自由形式的语言指令和传感器观测，OpenNav 能够以零样本的方式为开放世界导航生成一组密集的指导跟随和场景相符的机器人路径点，能够有效处理开放集对象和开放集指令，而不依赖情境示例或预训练技能。

近年来，随着视觉语言模型（VLMs）的快速发展，其将自然语言与视觉信息对齐并执行语义推理的能力显著重塑了机器人感知和人机交互。因此，体感智能作为一个重要的研究领域出现了。然而，大多数这些进展仅限于结构化场景，例如室内固定桌面操作任务或室内目标物体导航任务，这些任务涉及少量预定义的闭合对象和指令。当我们在开放世界环境中部署和释放机器人时，例如户外导航，它们必须应对环境中未结构化布局的开放对象，从而导致显著更大的复杂性和环境不确定性。此外，实际应用中的自由形式开放指令引入了额外的挑战，因为它们需要灵活和分层的理解，不同的指令可能需要不同水平的空间、语义和关系推理——从区分相似对象到识别特定地标，再到理解大型、未结构化户外空间的抽象空间关系。在这样的环境中，现有的方法面临重大挑战，因为它们完全依赖于物体检测器进行环境描述，并使用大型语言模型（LLMs）在语言领域进行推理，导致导航任务所需的重要空间和细微信息的丢失。因此，它们难以处理开放世界应用的丰富性和多样性。

除了开放世界的挑战，另一个根本性挑战是将 MLLM 强大的语言和常识推理能力转换为机器人控制的物理执行。由于在训练过程中，LLMs 和 VLMs 缺乏物理交互数据和几何信息的曝光，现有的努力大多假设这些模型不适合进行依赖硬件的低级控制 [3]，因此，许多工作采用 LLMs 作为高级规划者，忽略低级机器人运动控制或依赖于预定义的运动原语（例如，点到点导航），这牺牲了语言指令在控制机器人运动行为中的灵活性。相比之下，LQR-RRT* 框架在动态行为规划方面表现出显著贡献，强调了将控制可行性整合到轨迹合成中的重要性 [4]。VoxPoser [5] 的出现也证明了 LLMs 在提供抓取任务指令时，擅长推断可操作性和约束。然而，许多零样本框

架，包括 VoxPoser，严重依赖于向 LLM 输入提供上下文中的示例，使其不适合开放世界导航任务。鉴于这些挑战，我们的贡献有三个方面：

- 我们介绍了 OpenNav，一个零样本视觉语言导航框架，据我们所知，这是第一个在户外导航中，根据开放集指令和开放集物体直接使用 MLLM 生成轨迹的框架，而不依赖于预训练的技能、运动原语或上下文示例。
- 我们提出了一个多专家系统，该系统集成了最先进的多模态大语言模型（MLLMs）与一个开放词汇的感知系统，以增强机器人的场景理解能力。通过使用单一的任务无关提示和 MLLM 与感知系统之间的多模态接口，我们的框架显著提高了在未开放集合的对象和语言指令下的鲁棒性。通过结合 MLLMs 的推理、代码生成、函数调用能力与经典规划技术，我们的方法利用了类人推理和几何符合的轨迹合成的优点。
- 我们的 OpenNav 经过评估，适用于 AVD 和地面机器人，突出了其在视觉语言导航和具象智能研究中的有效性。

I. 相关工作

A. 开放集感知与规划

早期关于物体检测和语义 3D 映射的工作主要依赖于深度学习的方法和标注数据集 [6] 来获得环境中物体的空间表示（例如占据空间和点云）。然而，它们的性能在很大程度上受到训练数据丰富程度的限制，因为这些模型在监督学习模式下是在一个封闭的对象类别和特定场景数据集上训练的。这一限制使得这些模型在复杂和开放环境中识别未见过的对象类别变得具有挑战性。最近在互联网上规模化数据上训练的 VLMs 能够进行开放词汇的 2D 和 3D 理解。在机器人应用中，最近的方法通常利用诸如 SAM 和 CLIP 的模型来生成带有检测标签和特征嵌入的分割物体点云 [7]。此外，一些工作结构化了从点云数据中获取的空间关联文本场景描述 [2]。虽然

Authors are with the University of Toronto Institute for Aerospace Studies and the University of Toronto Robotics Institute, Toronto, Canada { mingfeng.yuan, letian.wang, steven.waslander } @robotics.utoronto.ca

有效用于简单和小规模环境（例如台面场景），但这种方法在复杂和大规模开放环境中变得不足以处理关键的细粒度空间关系。这种方法类似于一个人在一个环境中试图基于另一人提供的口头描述来计划行动，同时闭上双眼。与现有的将开放词汇物体检测与基于 LLM 的计划相分离的工作不同，我们利用 MLLMs 的最新进展，并采用一个多专家系统框架，其中一个单一模型负责场景理解和决策。在我们的方法中，采用开放词汇感知系统 (OVPS) 来获取分割物体的详细说明以及 3D 语义-几何图，从而增强 MLLMs 在导航任务中解释模糊语言指令和执行理性决策的能力。

B. 视觉-语言导航

视觉-语言导航 (VLN) 是具身智能的核心任务，要求代理使用视觉线索和自然语言指令在复杂环境中导航。该领域在近年来取得了显著进展。早期对如 Matterport3D 上的离散导航研究集中于基于传送的预定义节点之间的移动 [8]。随着预训练大型语言模型 (LLM) 的兴起，许多研究使用大规模模型 [9] 和预训练技术 [10] 来增强导航，但主要解决的是高层次决策，忽视了低层次运动控制。最近的工作 [11] 转向连续环境，在这些环境中，代理使用中层动作进行导航（例如，向前移动、在原地旋转），如在 VLN-CE 中使用 Habitat [1]。为了弥合离散和连续导航之间的差距，一些方法使用航点模型来预测候选位置 [12]，尽管提高了性能，但在泛化能力有限以及缺乏运动规划或障碍物规避方面存在不足。本研究探讨了一种新的范式，利用多模态大模型 (MLLM) 进行轨迹生成，在轨迹层面统一了高层次决策和运动规划，并结合障碍物规避，以实现更有效的导航。

C. 代码即政策

先前的研究利用大型语言模型 (LLM) 作为机器人操作的规划器，尤其是 SayCan [13]，其在推理时严重依赖于上下文中的示例 [5]。虽然我们的方法与 Kwon 等人的研究有相似之处 [3]，但他们主要关注操作任务。相反，我们使用一个单一的、与任务无关的提示——不包含上下文中的示例或预定义的运动原语——以实现通用化的导航。我们的模型能够自主调用 API 来检索与任务相关的信息，生成可执行的轨迹代码，并能够在失败时检测、调试和重新执行，从而提高适应性和稳健性。

在这项研究中，我们研究了预训练的 MLLM 能够在零样本情况下处理导航任务的程度，任务中具有开放式指令和开放式对象，而无需依赖场景内示例或预定义的运动原语。为了解决这个问题，我们提出了 OpenNav，一个零样本 VLN 框架，该框架集成了开放词汇感知系统、由 MLLM 赋能的高级计划以及中级轨迹优化器。如图 3 所示，我们的方法利用了 MLLM 固有的多模态理解、代码生成、推理和计划能力，以预测密集轨迹，从而成功执行任务。该算法的流程可在算法 1 中找到。以下各部分详细描述了我们的框架，我们将在其中定义关键挑战并介绍我们提出的解决方案。我们首先介绍系统如何整体运作，在第 I-D 节，然后在第 I-E 节详细阐述开发的开放词汇感知系统。

D. 开放集零样本视觉-语言导航

1) 开放集指令：导航任务通过自由形式的语言命令 l 指定（例如，“去你右边的房子”）。为了成功执行这样

的任务，机器人必须具备以下能力：(a) 理解和推理指令；(b) 将高层次任务分解为子任务，如物体检测、轨迹形状规划以及通过代码合成生成轨迹；(c) 顺序执行每个子任务；以及 (d) 检测任务失败并相应地重新规划。处理开放集指令的主要挑战在于无法通过上下文示例或基于规则的方法预定义任务执行，因此需要强大的零样本能力。为此，我们设计了一个单一的任务无关提示， p ，使机器人能够在各种户外导航任务中泛化，包括从具体（例如，“去红色的车”），到抽象（例如，“扔掉垃圾”），到方向性（例如，“朝黑色的车行驶，然后前往房子”）。这个系统提示包含五个关键组件：

- 1) 可用功能，包括可调用的 API，如 `det_object()`、`A_star_plan(start, end)` 用于开放集感知和基于价值映射的轨迹规划，以及 `visual_3D(traj)` 用于轨迹可视化。
- 2) 环境描述，详细说明坐标系、机器人姿态和传感器配置。
- 3) 避碰导航指导，指示 MLLM 在执行之前识别潜在障碍物和可通行区域。
- 4) 初始规划，提示模型进行逐步推理和跨模态理解感觉观察。在这个阶段，需要 MLLM 根据系统提示和给定任务完成前述的 (a) 和 (b) 任务。
- 5) 代码生成，要求 MLLM 要么调用现有 API，要么生成代码以分段生成轨迹，然后将这些段连接成完整的轨迹。该部分旨在完成之前提到的任务 (c) 和 (d)。

这种任务无关的提示以及自由形式语言的导航任务，被作为输入提供给 MLLM，如图 3 所示，我们使用的是 ChatGPT-4o，这是最先进的专有模型之一。

在现实世界的导航任务中，可能会遇到各种各样的物体，而目标物体可以通过多种方式来描述：(1) 根据其固有类别，例如房子；(2) 通过基于特征的描述，例如圆柱形建筑，引用形状、外观或其他属性；或 (3) 通过功能描述，例如扔垃圾，其中最相关的物体将是垃圾桶。为了处理这种变异性，我们采用了一个多专家对象感知系统。第一个组件是一个 OVPS，由一组开源视觉-语言感知模型组成，负责物体检测、分割、生成图像说明，并提取物体属性，如位置和大小，其实现细节将在接下来的第 I-E 节介绍。系统中的第二个专家模型利用了 MLLM 的多模态理解和推理能力。OVPS 的输出是一个带注释的 RGB 图像，以及每个物体属性的文本描述，包括图像说明、中心位置、尺寸和每个可驾区域到目标物体的最近可达点 (NRP)（格式为字典 $\{Reg.: Point, \dots\}$ ）。这些信息，结合前一节的提示，作为输入提供给 MLLM，并构成 MLLM 的环境感知。解释和执行自由形式语言指令是体现智能中的一个基本而具有挑战性的问题。根据前几小节中描述的环境感知，MLLM 首先理解和推理任务。例如，如图 1 所示，给定指令“驾驶向黑色汽车，然后朝向你右边的房子”，MLLM 利用其语义理解和任务规划能力来识别相关目标，如房子、可驾驶的表面和路线上的障碍物。具体来说，当环境中存在多个相似对象时，模型必须进一步使用指令中提到的空间关系（例如“在你右边”）来确定相关和目标对象。在识别目标和相关对象后，它通过首先选择靠近相关对象的中间路径点，然后选择目标对象附近最近检测到的可驾驶表面点来推断轨迹形状。例如，对于上述指令，模型将首先朝黑色

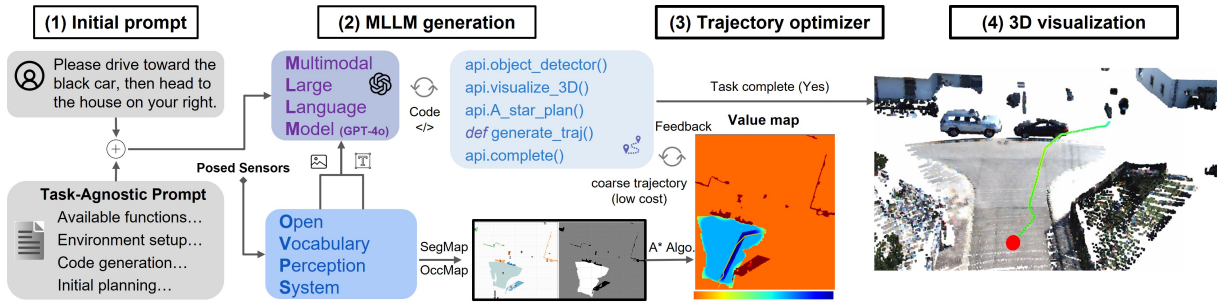


Fig. 3: OpenNav 概述。给定环境的 RGB-Lidar 观测和一个开放集的自由形式语言指令，1) 我们利用与任务无关的提示来实现零样本的泛化能力和对各种指令的适应性；2) MLLM 生成代码，与 OVPS 交互，以产生开放集多模态场景感知输出和基于操作环境的 2D 鸟瞰视角（BEV）值图（由语义图和占用图组成）。3) MLLM 根据指令、场景理解及其推理能力，合成类似人类的粗略轨迹。生成的 BEV 值图随后成为运动规划器的目标函数，运动规划器精确轨迹以确保符合几何要求的导航。请参阅图 4，以获取 OVPS 的详细流程、输入和输出。

Algorithm 1 开放世界导航管道

Require: task-agnostic prompt p , task instruction l , and observations $\{ \text{text: (cap, pos, dim, nrp)}; \text{visal: I} \}$

```

1:  $\text{prompt} \leftarrow p \oplus l \oplus I$  ▷ concatenate
2:  $\text{task\_completed} \leftarrow \text{False}$ 
3: while  $\text{task\_completed} = \text{False}$  do
4:    $\text{output} \leftarrow \text{MLLM}(\text{prompt})$ 
5:    $\text{prompt} \leftarrow \text{None}$ 
6:   if  $\text{output}$  contains code then
7:      $\text{try exec}(\text{output})$  ▷ extract code
8:     except Exception then
9:        $\text{prompt} \leftarrow \text{error message}$ 
10:  else
11:    if  $\text{detect\_object}()$  is called then
12:       $\text{prompt} \leftarrow (\text{cap}, \text{pos}, \text{dim}, \text{nrp}) \oplus I$ 
13:    else if  $(\text{reg}, \text{traj})$  are available then
14:       $\text{val\_map} \leftarrow \text{occ\_map} \oplus \text{val\_map}(\text{traj}, \text{reg})$ 
15:       $\text{final\_t} \leftarrow \text{a\_star\_plan}(\text{val\_map})$ 
16:       $\text{visual\_3D}(\text{final\_t})$ 
17:      ▷ final_t : final t trajectory;
18:      ▷ nrp : n earest r eachable p oints
19:      ▷ traj : tra jectory generated by code;
20:      ▷ reg : identified drivable reg ion index;
21:    end if
22:  end if
23: end while
```

汽车行驶，然后向右转向最终目标房子。具体而言，为了生成与语言指令一致的轨迹，我们利用 MLLM 强大的代码生成能力来避免其在直接生成连续轨迹方面的不足。根据环境观测，MLLM 首先通过推理任务来推断轨迹形状，然后合成一个 Python 脚本来生成轨迹，指定起点、中间路径点和终点。该脚本将在 Python 解释器中执行，如果出现任何错误，MLLM 会根据终端错误消息自动调试脚本并重新执行它，直到生成有效的轨迹。此过程最终产生一个密集的 3D 轨迹序列。

2) 符合几何的轨迹优化：尽管 MLLM 具备推理能力，我们观察到即使是最先进的专有模型在生成精确、顺畅

且无碰撞的轨迹方面也存在困难。为了解决这个问题，我们引入传统的路径规划算法，例如 A^* ，这种算法可以高效地基于几何地图生成无碰撞的轨迹，但本质上强调最短轨迹的优先级，且缺乏对环境的语义理解。为了使轨迹生成与语言指令保持一致，我们提出了一个将 MLLM 与 A^* 规划相结合的框架，作为轨迹优化器。具体来说，MLLM 生成的轨迹被视为与任务描述对齐的粗略轨迹段，然后将其投射到二维鸟瞰图视角（BEV）价值地图上，在该地图上，已穿越的区域（考虑到每个点周围的预定义半径）被分配比周围可驾驶表面区域更低的成本，从而确保基于语言的意图与运动规划之间的对齐。请注意，初始价值地图是通过占用地图和语义地图联合初始化的。在占用地图中值为 1 的区域在价值地图中被分配一个高成本，以确保避免碰撞。此外，为了控制可导航区域（例如，区分铺设的道路和人行道），价值地图为可导航区域分配略高于粗略轨迹覆盖区域的成本，而其他地面区域则被分配与占用区域相同的高成本。请参阅第 I-E 节详细信息。然后， A^* 基于更新的 BEV 价值地图生成一个优化的无碰撞轨迹，确保与语言指导的一致性。这种方法还缓解了传统规划者在理解环境语义方面的局限性。例如，仅靠 A^* 很难避开一个被水淹没的区域，因为几何地图缺乏此类信息，而 MLLM 可以从语义地图中推断出占用情况以指导轨迹生成。最后，我们对轨迹应用 B 样条平滑，并在重建的 3D 地图中进行可视化以进行验证和成功率评估。

E. 用于 VLN 的开放词汇感知系统

我们的开放词汇感知系统（OVPS）由两个组件组成：一个开放词汇对象感知模块和一个基于 OpenGraph 框架 [14] 的地图重建模块。该系统接收 RGB 图像、对应的带位姿的三维 LiDAR 点云作为输入，并输出两种类型的信息：1) 开放词汇对象感知模块为 MLLMs 提供多模态输入（文本和视觉环境信息），用于场景理解和任务推理；2) 地图重建模块生成语义和占用地图，用于轨迹规划和三维可视化。我们的感知系统的整体框架如图 4 所示，接下来，我们将介绍这两个模块和输出的详细信息。

开放词汇物体感知。与之前的方法不同，这些方法主要依赖从 OVPS 提取的文本提示来描述场景信息，作为在下游计划任务中供大型语言模型使用的输入。我们认为，这种方法丢失了许多重要的视觉细节，这些细节可能对机器人任务成功率产生关键影响，尤其是在任务无关和

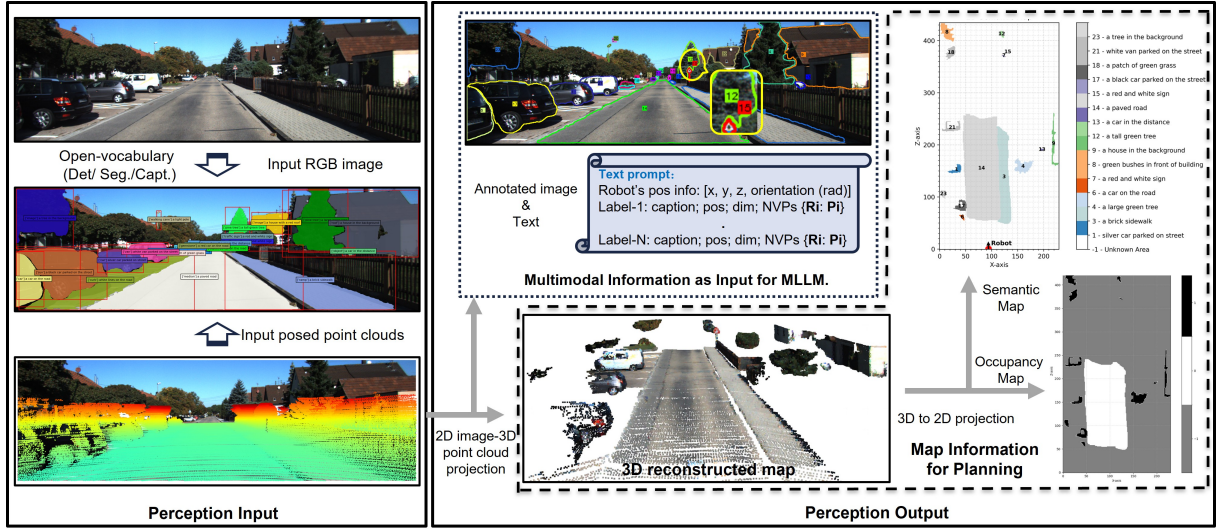


Fig. 4: 我们的开放词汇感知系统依次进行检测、分割和物体标题生成。结合 3D 点云，该系统将生成 1) VLN 的多模态观测，包括用于 MLLMs 的文本提示和图像提示，2) 3D 重建地图，以及用于轨迹优化的 2D 占用和语义地图。

开放世界的环境中。具体而言，人们指定的任务通常涉及到细致入微的物体描述，例如颜色、形状、尺寸、功能和空间关系，这些都是难以单纯用格式化文本描述来完整表达的。相比之下，视觉信息天生就可以捕捉到这些细节。得益于多模式大型模型（MLLMs）的出现，我们现在可以更有效地弥合感知模块和计划模块之间的差距，在这里我们利用感知系统提供文本和视觉信息。具体而言，我们生成包含检测到的物体标题、全局位置和尺寸的文本提示，同时给每个检测物体分配一个唯一的数字标识符。此外，我们提供注释的 RGB 图像，其中物体轮廓用不同的颜色标出，并标记上在文本描述中找到的相应数字标识符，以帮助 MLLMs 区分检测到的物体。

具体而言，在时间 t ，给定输入的 RGB 图像 $I^{(t)}$ ，我们首先应用识别任何东西模型（RAM）[15]，记作 $\text{RAM}(\cdot)$ ，用于识别图像中存在的类别。凭借其开放集识别能力，RAM 能够检测出各种常见的物体类别。接下来，识别出的开放词汇类别名称连同原始图像被输入到 Grounding DINO 模型 [16]，表示为 $\text{GD}(\cdot, \cdot)$ ，用于开放集目标检测。此步骤生成目标检测边界框，这些边界框作为 TAP（通过提示标记任何东西）模型 [17] 的基本输入，记作 $\text{TAP}(\cdot, \cdot)$ 。在检测到的边界框的指导下，TAP 模型对图像中的主要物体进行分割和描述，产生一组分割掩码 $\{m_i^{(t)}\}_{i=1, \dots, m}$ 以及与帧 $I^{(t)}$ 对应的文本描述 $\{c_i^{(t)}\}_{i=1, \dots, m}$ 。整个过程可以形式化为

$$\{m_i^{(t)}, c_i^{(t)}\} = \text{TAP} \left(I^{(t)}, \text{GD} \left(I^{(t)}, \text{RAM} \left(I^{(t)} \right) \right) \right) \quad (1)$$

，其中该流程中使用的具体模型可以替换为提供类似功能的替代模型。

BEV 值图构建。最近的研究表明，由于训练中缺乏深度和几何信息，当前的预训练 MLLM 在空间推理方面表现不佳。然而，我们认为现代机器人平台通常配备有 RGB-D 相机或 3D 激光雷达，因此获取物体几何信息是很简单的。因此，为了生成符合场景几何的轨迹，我们的感知系统生成一个 2D BEV 值图，包括用于避免碰撞的 2D 占用图和用于类别信息的 2D 语义图。为了初始化值图，我们采用多传感器校准和融合方法将 3D 激光雷达点云 $C^{(t)}$ 投影到二维图像平面上。投影过程提取对齐到

对应遮罩 $m_i^{(t)}$ 的特定物体的 3D 点云 $p_i^{(t)}$ ：

其中， l_k 是一个 LiDAR 点， M_K 是内参相机矩阵， M_T 是用于 LiDAR-相机对齐的外参转换矩阵。值映射中的成本值随后使用由 MLLM 生成的轨迹进行更新，以与语言指令对齐，如之前在第 I-D 节中讨论的那样。

由于缺乏用于评估地面机器人在 VLN 任务中性能的户外基准，我们利用了现有的 AVD，如 SemanticKITTI。该数据集提供了丰富的真实世界户外场景，以及 2D RGB 图像、3D LiDAR 点云和相应的位置信息，使其成为我们实验的理想选择。我们提出的框架直接生成格式为 $[x, y, z, \text{heading}]$ 的轨迹。为了验证我们方法的有效性，我们进行了实验，重点关注两个关键方面：1) 方法在处理开放集指令和开放集对象时的鲁棒性。2) 生成轨迹与给定指令的一致性。

F. 在开放集合指令和对象下的导航

任务描述。为了评估我们所提出的零样本框架在现实世界户外导航任务中面对自由形式语言指令的鲁棒性，对于 SemanticKITTI 序列 05 中的每个场景，我们考虑两个类别的导航任务。

1) 移动到物体任务。在此类别中，机器人必须从当前位置导航到目标物体。我们逐步放宽指令中物体描述的限制，以评估算法在开放集物体上的性能：

- 具体对象导航：指令明确指定目标对象，且环境中只有一个这样的对象，例如，“去红色的车那里。”
- 模糊物体导航：场景中包含多个相似的物体，要求机器人使用多模态信息来解决歧义。指令可能包括空间参照、外观或其他细微提示，例如，“到你左边的第二辆车那里”或“在停车标志附近的树旁等我。”

2) 高级语言指令任务。该类别评估人机交互的灵活性，在这种情况下，指令没有明确提到目标对象或无法通过单个对象说明来解决。模型必须理解任务，进行推理，并整合环境观测来生成轨迹。示例包括：“扔掉垃圾”，此时模型必须推断出垃圾桶是最相关的目标；“先直走，然后右转进入院子。”，此时模型必须将指令分解为连续的导航步骤，并选择最相关的环境信息以生成轨迹。

Task-Type	Subcategories	LLM-TG [3]		OpenNav	
		NE	SR	NE	SR
Move to obj.	Specificity	1.44	43/50	1.01	45/50
	Ambiguity	7.40	13/50	1.63	42/50
High-level	Reasoning	5.90	23/50	2.40	40/50
Total		4.91	53 %	1.68	84 %

TABLE I: LLM-TG 和 OpenNav 的性能比较。SR: 成功率, NE: 导航误差。

评估指标。为了评估我们系统的有效性,我们在各种场景下进行实验,并报告以下的定量指标。

- 成功率 (SR): 当轨迹的最终位置在真实位置 1 米以内时,它被视为成功。
- 导航误差 (NE): 估计的终点与真实终点之间的平均欧几里得距离。

对比基线。我们将我们的方法与由 Kwon 等人提出的框架 LLM-TG 进行比较。在 LLM-TG 中,观察信息完全依赖于目标检测器提供的数据,包括物体的标题、坐标、大小以及每个可行驶区域到目标的最近可达点。相比之下,我们的方法利用多模态输入。为了确保公平比较,两个框架都使用 ChatGPT-4o 作为决策模型,仅在提供的输入模式上有所不同。

结果与分析。表 I 中的实验结果表明,当指令中不存在歧义时,LLM-TG 和我们的多模态框架的性能保持相似,都达到 100 % 的成功率。然而,在指令存在歧义的情况下,纯文本方法的性能显著下降,而我们的方法则保持了较高的成功率。LLM-TG 的失败案例揭示了其失败主要归因于两个关键因素: 1) 依赖于物体检测器的准确性——物体检测中的错误直接影响导航的成功; 2) 描述信息的细粒度——标题中有限的描述细节降低了解释能力。例如,指令“前往圆柱形建筑”导致 LLM-TG 失败,因为: 物体检测器错误地将圆柱形结构分类为垃圾桶(见图 ??)。该方法仅依靠标题信息,无法识别正确的目标。相比之下,我们的方法首先在标注的图像中识别圆柱形建筑的数字标签,然后验证相应的物体标题(垃圾桶),并且意识到来自 OVPS 的错误分类。最终的轨迹是基于与修正的物体标签相关联的地面上最近的可达点计算得出,成功地完成任务,如图 1 所示。LLM-TG 的另一个失败案例见图 1, 其中给出了“前往你右边的房子”的指令。环境中存在多座房屋。机器人必须正确解释空间参考以确定最终目标。我们的方法通过利用标注图像的场景理解,优于仅文本方法。此外,与 LLM-TG 相比,我们的方法在处理高级语言指令方面表现出显著优势,如表 I 所示的测试结果所示。这验证了我们最初的假设,即统一的感知和规划框架对于机器人导航至关重要。机器人应该根据所见进行轨迹规划,而不仅仅依赖于文本描述。

G. 语言引导的零样本轨迹合成

本节评估 OpenNav 生成的轨迹与基于语言指令的人类标注轨迹的契合度。为了评估这种契合度,我们使用轨迹形状相似性度量,其中更高的相似性表示在语言引导的轨迹合成中表现更好。这个实验作为一项消融研究,比较了三种轨迹生成方法:

- A*: 仅基于占用地图生成无碰撞轨迹,只使用起点和终点。

TABLE II: 在轨迹合成中的性能

Methods	Fréchet Distance ↓	NDTW ↑	Collision ↓
A*	24.20	0.08	0/30
VLT-Code	17.53	0.16	16/30
OpenNav	12.60	0.38	2/30

- VLT-Code: 一个基线方法,其中 MLLM 通过代码直接生成轨迹,选择起点、中间点(IP)和终点,不依赖于价值映射,仅依靠其空间推理(参见第 I-D.1 节)。
- OpenNav: 我们提出的完整版本的 VLN 框架,结合了 MLLM 和符合几何的轨迹优化,如第 ?? 节所述。

在评估中,用户提供语言指令,然后在地图上手动绘制他们偏好的轨迹,这作为评价生成轨迹质量的真实值。为了全面评估生成轨迹和用户定义的轨迹之间的相似性,我们使用 Fréchet 距离 [18] 和规范化动态时间规整 (NDTW) [19]。Fréchet 距离严格保持轨迹形状,因此对偏差非常敏感,而 NDTW 通过允许灵活的时间对齐来考虑速度变化。较低的 Fréchet 距离表明更高的空间相似性,而较高的 NDTW 分数反映更好的整体对齐。

实验结果如表 II 所示。A* 算法在避碰方面表现出色,在十个选定的任务场景中未记录任何碰撞。然而,其轨迹相似性性能不佳,这是可以预期的,因为 A* 专注于寻找起点和目标之间的最短轨迹,而缺乏根据任务需求塑造轨迹的灵活性。相比之下,VLT-Code 方法在轨迹相似性方面比 A* 有所改进,但由于依赖于在训练时缺乏深度和几何信息的预训练 MLLMs,导致其在空间避碰性能不佳。我们提出的 OpenNav 在轨迹相似性和避碰方面都表现出明显的改进。图 5a 展示了三个具有代表性的任务。在任务 1 中,指令是:“沿着阴影下开往黑色围栏,然后朝高大的树木行驶。”这需要模型将任务分解为两个步骤:首先,根据当前位置和方向确定围栏附近的中间点,然后从该点导航到最终目标。用户定义的轨迹在图 5b 中以黑线显示,作为地面真值。任务 1 中的价值地图显示,A* (绿色)忽略了中间点,直接朝高大的树木前进,而 VLT-Code (黄色)和 OpenNav (红色)都成功地结合了中间点,使得轨迹更符合用户定义的路线。同样地,在任务 2 中,指令为:“先直行,然后向车门上有蓝色图案的白色车行驶。”VLT-Code 和 OpenNav 都有效地遵循了预期的轨迹。然而,VLT-Code 有时会生成与任务意图一致但由于其有限的空间推理能力可能导致碰撞的轨迹。如图 5b 中任务 3 所示,VLT-Code 轨迹穿过了一个有车辆存在的被占用区域。对于 OpenNav,结果清楚地显示它成功整合了 MLLM 与基于几何的空间约束的优势,使其能够在轨迹遵从与碰撞避免之间取得平衡,确保低碰撞率的同时紧密跟随用户喜欢的路径。由于空间限制,我们在项目网站上提供了更全面的实验结果和真实机器人演示。

II. 结论

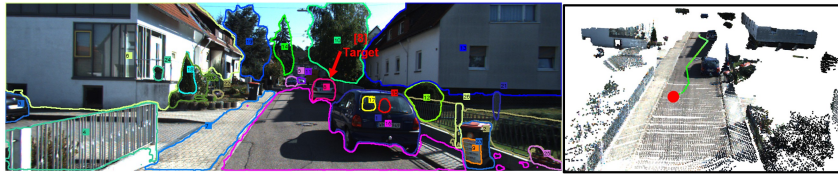
本研究引入了 OpenNav,这是一种零样本 VLN 框架,通过生成导航轨迹,成功地将 MLLMs 强大的语言推理、高级规划和代码生成能力转化为可执行的机器人动作。我们的方法无需任何训练数据或特定场景的提示工程,在复杂的非结构化场景中展示了对开放集指令和开放集对象的强泛化。此外,我们提出了一种任务无关的提示策略,以及感知系统与 MLLM 之间的多模态接口设计,以



Task 1: First, go to the **black fence** straight ahead **under the shadow** while maintaining a safe distance. Then, from there, move toward **tall tree**.

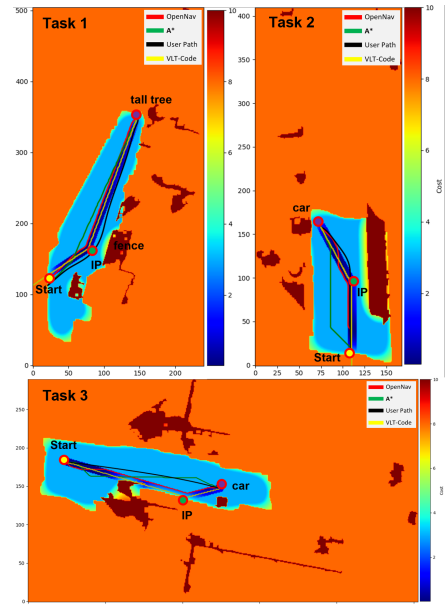


Task 2: Drive **straight** first, then steer towards the **car** ahead with a **blue image on the door** as you get closer to it.



Task 3: Go to the **car on your right** that is **farther away**.

(a) 任务和使用 OpenNav 生成的轨迹



(b) 价值图

Fig. 5: 选定的示例展示了 OpenNav 如何利用价值地图生成与任务对齐且符合几何的导航轨迹: (a) 用户指定的任务和由 OpenNav 生成的轨迹; (b) 价值地图展示了基于给定任务由不同算法生成的轨迹。

成功解决环境感知的粒度问题, 尤其对于自由形式的语言指令, 同时表现出对检测器错误检测的鲁棒性。我们的工作结合了 MLLMs 的优势和经典规划算法, 确保生成的轨迹与语言指令一致, 同时遵循环境的几何约束以实现避碰。我们在 AVDs 和真实世界的轮式机器人部署中, 在不同的场景和任务类型下验证了我们的方法, 始终表现出强劲的性能。在未来, 我们计划将这种方法扩展到更复杂和动态的环境中。

REFERENCES

- [1] R. Ramrakhya, E. Undersander, D. Batra, and A. Das, "Habitat-web: Learning embodied object-search strategies from human demonstrations at scale," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5173–5183.
- [2] M. Parakh, A. Fong, A. Simeonov, T. Chen, A. Gupta, and P. Agrawal, "Lifelong robot learning with human assisted language planners," in IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 523–529.
- [3] T. Kwon, N. Di Palo, and E. Johns, "Language models as zero-shot trajectory generators," IEEE Robotics and Automation Letters, 2024.
- [4] X. Zhong, Z. Wei, and T. Chen, "Motion planning and pose control for flexible spacecraft using enhanced lqr-rrt," IEEE Transactions on Aerospace and Electronic Systems, vol. 59, no. 6, pp. 8743–8751, 2023.
- [5] 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," arXiv preprint arXiv:2307.05973, 2023.
- [6] Y. Shi, R. Yang, Z. Wu, P. Li, C. Liu, H. Zhao, and G. Zhou, "City-scale continual neural semantic mapping with three-layer sampling and panoptic representation," Knowledge-Based Systems, vol. 284, p. 111145, 2024.
- [7] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa et al., "ConceptGraphs: Open-vocabulary 3d scene graphs for perception and planning," in IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 5021–5028.
- [8] G. Zhou, Y. Hong, Z. Wang, X. E. Wang, and Q. Wu, "NavGPT-2: Unleashing navigational reasoning capability for large vision-language models," in European Conference on Computer Vision. Springer, 2024, pp. 260–278.
- [9] A. Majumdar, A. Shrivastava, S. Lee, P. Anderson, D. Parikh, and D. Batra, "Improving vision-and-language navigation with image-text pairs from the web," in European Conference on Computer Vision. Springer, 2020, pp. 259–274.
- [10] P.-L. Guhur, M. Tapaswi, S. Chen, I. Laptev, and C. Schmid, "Airbert: In-domain pretraining for vision-and-language navigation," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1634–1643.
- [11] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "NaVid: Video-based vlm plans the next step for vision-and-language navigation," arXiv preprint arXiv:2402.15852, 2024.
- [12] Y. Hong, Z. Wang, Q. Wu, and S. Gould, "Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 15 439–15 449.
- [13] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman et al., "Do As I Can, Not As I Say: Grounding language in robotic affordances," arXiv preprint arXiv:2204.01691, 2022.
- [14] Y. Deng, J. Wang, J. Zhao, X. Tian, G. Chen, Y. Yang, and Y. Yue, "OpenGraph: Open-vocabulary hierarchical 3d graph representation in large-scale outdoor environments," IEEE Robotics and Automation Letters, 2024.
- [15] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu et al., "Recognize Anything: A strong image tagging model," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 1724–1732.
- [16] 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 European Conference on Computer Vision. Springer, 2024, pp. 38–55.
- [17] T. Pan, L. Tang, X. Wang, and S. Shan, "Tokenize Anything via Prompting," in European Conference on Computer Vision. Springer, 2024, pp. 330–348.
- [18] H. Alt and M. Godau, "Computing the fr chet distance between two polygonal curves," International Journal of Com-

putational Geometry & Applications , vol. 5, no. 01n02, pp. 75–91, 1995.

- [19] G. Ilharco, V. Jain, A. Ku, E. Ie, and J. Baldridge, “General evaluation for instruction conditioned navigation using dynamic time warping,” arXiv preprint arXiv:1907.05446 , 2019.