arXiv:2603.01477v1 [cs.RO] 2 Mar 2026

SFCo-Nav：慢速大语言模型与快速属性图对齐协作的高效零样本视觉语言导航

Chaoran Xiong[1, †]，*IEEE*研究生会员，Litao Wei[1, 2, †]，Xinhao Hu[1, †]，Kehui Ma[1],Ziyi Xia[1], Zixin Jiang[1], ZhenSun[1],和 Ling Pei[1, 3, *]，*IEEE*高级会员

摘要——近年来，大视觉语言模型（VLMs）和大语言模型（LLMs）的进展推动了视觉语言导航（VLN）零样本方法的发展。在此类方法中，智能体仅依靠自我中心感知和推理来遵循自然语言指令。然而，现有零样本方法通常构建简单的观测图，并在其上逐步执行视觉语言模型与大语言模型推理，导致延迟和计算成本高昂，限制了实时部署。为此，我们提出高效的零样本视觉语言导航框架 SFCo-Nav，其灵感源自慢-快认知协作原则。SFCo-Nav 集成了三个关键模块：1）基于慢速大语言模型的规划器，用于生成策略性的子目标链，每个子目标都与一个想象物体图相关联；2）快速反应导航器，用于实时构建物体图并执行子目标；3）轻量级异步慢–快桥接模块，通过对齐高级结构化的属性想象图与感知图来估计导航置信度，仅在必要时触发慢速大语言模型规划器。据我们所知，SFCo-Nav 是首个依据内部置信度支持异步大语言模型触发的快慢协作零样本视觉语言导航系统。在公开的 R2R 和 REVERIE 基准测试上进行评估后，SFCo-Nav 的零样本视觉语言导航成功率达到或超过此前最先进的方法，同时将每条轨迹的总令牌消耗降低了 50% 以上，运行速度提升超过 3.5×倍。最后，我们在酒店套房环境中使用足式机器人演示了 SFCo-Nav，展现了其在室内环境中的效率与实用性。

一、引言

视觉语言导航（VLN）要求具身智能体通过感知环境、推理指令并执行一系列动作来实现目标 [1]。这是通用具身导航（EN）系统的一项基础任务 [2]。

近年来，大规模视觉语言模型（VLM） [3]和大语言模型（LLM）[4], [5]的成功，推动了零样本视觉语言导航的发展系统 [6]–[10]。这些零样本方法利用模型预训练的推理和落地能力，无需进行任务特定微调。与传统的端到端训练视觉语言导航智能体 [11]–[13]，相比，零样本视觉语言导航具有显著优势，例如无需训练成本、可快速部署到新环境中，以及对未见过的指令和布局具有较强的泛化能力 [14]。

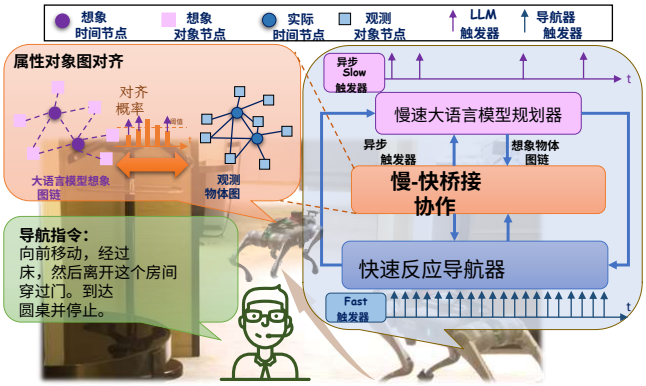


图 1. SFCo-Nav 是一种受慢-快认知协作启发的高效零样本视觉语言导航框架。它由三个模块组成：1）基于大语言模型的慢速脑规划器；2）快速脑反应式导航器；3）轻量级异步慢–快桥接模块，用于对齐想象图与感知图、估计导航置信度，并仅在必要时触发大语言模型。该设计在保持较高导航成功率的同时，最大限度地减少了高成本的大语言模型调用。

零样本视觉语言导航的核心挑战在于，如何将视觉观测和文本指令高效地转换为准确的导航动作。早期工作（如 NavGPT [8],）使用视觉语言模型 BLIP-2[15] 预处理所有可导航视点，以生成场景的文本描述。随后，智能体查询大语言模型，以输出下一个目标视点 ID。尽管 NavGPT 验证了大语言模型的导航能力，但它属于离线系统，需要大量计算。最近，MapGPT [9]采用朴素的结构化拓扑地图表示，并集成 GPT-4V [3] 或 BLIP-2 [15] 等视觉语言模型，以进行在线、逐步的图像到文本转换和直接动作预测。尽管这种方法支持在线处理，但仍需要视觉语言模型在每一步对多个候选视点进行处理。这导致较高的令牌用量和较慢的推理速度。为实现更高的效率，NavCoT [14] 通过微调大语言模型以减少令牌消耗，在一定程度上降低了成本，但在每一步仍保留了同样昂贵的视觉语言模型多点感知过程。

当前的零样本视觉语言导航方法 [6]–[10], [14] 通常采用视觉语言模型—大语言模型范式。这种范式可称为完整的慢脑策略：在每个导航步骤中，

†Chaoran Xiong、Litao Wei 和 Xinhao Hu 对此作出了同等贡献工作。

*C对应 ding Auth或: L ing Pe i.

本研究部分得到了国家自然科学基金委员会（NSFC）项目（批准号：62273229）的资助，部分得到了上海市科学技术委员会项目（批准号：24DZ3101300、24TS1402600 和 24TS1402800）的资助。

[1]作者单位为上海交通大学（SJTU）上海市导航与位置服务重点实验室，中国上海，邮编 200240。 [2]Litao Wei 还任职于上海交通大学致远学院。 [3] Ling Pei 还任职于上海交通大学海洋地质国家重点实验室。（电子邮箱： {sjtu4742986; oscar0731; xinhaohu; khma0929; matcha.latte; zhen- sun; yan.xiang; ling.pei}@sjtu.edu.cn; jiangzixin0214@163.com）。

演示和代码将在 https://anonymous.4open. science/r/GQ-Nav-5EB5/ 发布。

系统会穷举查询视觉语言模型，获取所有可见视点的信息，并使用大语言模型直接决定下一步动作。这会导致词元成本高昂，决策过程缓慢。相反，人类导航者通常会先进行一段慢速思考：解析指令、设定子目标并制定内部计划。随后，快速思考会凭借感知直觉执行这些子目标。只有在陌生或发生变化的情境中置信度下降时，才会启用慢速推理。详情请参阅 [16]。这种慢—快协作机制有望提升导航效率，但在零样本视觉语言导航中却一直被忽视。

受人类慢快认知协同的启发 [16],，我们提出了 SFCo-Nav，这是一种零样本视觉语言导航框架，结合基于大语言模型的慢速脑与轻量级反应式快速脑，实现高效的具身导航。慢速脑将指令解析为战略决策链，其中包括目标物体、导航技能和想象物体图。另一方面，快速脑感知实时物体图，并使用物体-技能基元执行计划。异步触发机制会监测结构化归因图 [17] 匹配的置信度，仅在置信度低于阈值时才启动慢速脑，从而尽量减少不必要的推理。在 R2R [1] 和 REVERIE [18],上，SFCo-Nav 的成功率与强大的零样本基线相当或更高，同时显著降低了推理成本。此外，SFCo-Nav 在足式机器人上进行的真实世界酒店套房部署中展现出实用的效率。据我们所知，这是首个基于内部置信度进行异步大语言模型触发的慢快型零样本视觉语言导航系统。我们的主要贡献如下：

1）一种用于具身导航的零样本快慢协作导航框架 SFCo-Nav。我们的慢速组件——大语言模型——会生成由目标物体、所需技能和想象物体图构成的战略决策链。随后，快速组件——反应式导航器——通过高效对齐实时感知的物体图与大语言模型生成的目标物体和想象物体图，执行这些步骤。

2）一种轻量级异步触发机制，用于管理快慢协作，从而提升效率。该机制通过计算先进的结构化属性图匹配概率，并根据感知图与大语言模型想象图链的对齐情况，将其作为导航器置信度。3）在零样本具身导航方面实现了最先进的效率，并在 R2R 和 REVERIE 基准测试上进行了验证。与现有方法相比，SFCo-Nav 达到了相当或更高的任务成功率，同时将每条轨迹的平均总消耗词元数减少了 50% 以上，运行速度提升了 3.5×倍。

本文其余部分安排如下。第二节回顾零样本视觉语言导航和慢快系统的相关工作。第三节形式化阐述了所提出的视觉语言导航慢快协作框架。第四节介绍 SFCo-Nav 的架构和技术细节。第五节报告了在公开视觉语言导航基准测试上的实验结果，包括与最先进零样本方法的比较、消融分析以及一项真实世界案例研究。最后，第六节给出结论。

II. 相关工作

本节回顾零样本视觉语言导航领域的相关文献。第二节A考察近期基于大语言模型的方法，这些方法利用大规模预训练模型来理解导航指令。第二节B讨论感知与决策中的慢快系统原理，以及这些原理提升零样本视觉语言导航效率的潜力。

A. 基于大语言模型的零样本视觉语言导航

基础视觉语言模型（VLM）[3]和大语言模型（LLM） [4], [5] 的进展，使智能体无需特定任务训练即可遵循自然语言导航指令。不同于传统的端到端视觉语言导航 [12], [13], [19], ，零样本视觉语言导航直接利用预训练模型，避免了代价高昂的数据收集和微调，这对现实世界部署而言是一项关键优势 [8]–[10]。

NavGPT [8]率先展示了大语言模型的导航能力：它将所有视点离线转换为 BLIP-2 文本描述，并提示大语言模型选择下一个视点。尽管效果显著，但这种逐步进行的多视点处理方式使其不适用于在线应用。A2Nav 引入了动作感知，使大语言模型能够同时基于场景观测和可用导航技能进行推理 [20], ；而 Console [21] 则利用大语言模型筛选并优化地标，提升了成功率，但仍依赖场景训练的视觉语言导航模型。

为了实现在线执行，MapGPT [9] 使用简单的拓扑地图连接可导航视点，将每一步的多个观察转换为文本，以供大语言模型驱动的动作预测。然而，场景描述的朴素结构以及每一步的多视图视觉语言模型处理导致了高词元成本和延迟。NavCoT [14] 通过大语言模型微调降低了大语言模型的令牌用量并改进了推理，但仍保留了同样昂贵的多视图视觉语言模型计算，从而削弱了零样本优势。这些局限性凸显了对一种实时、低成本、可部署的零样本视觉语言导航方法的需求。

B. 慢快系统

当前的零样本视觉语言导航方法，如 [8]–[10], [14],，通常采用完整的慢脑策略，在每一步调用视觉语言模型–大语言模型推理以输出动作。尽管较为全面，但这会导致冗余感知、高计算成本和低时间效率。受人类快思考与慢思考 [16],的启发，慢快系统在机器人学中受到关注，尤其是在操作领域，其中慢速高层规划器引导快速低层控制器以实现高效执行 [22]。

然而，在零样本视觉语言导航场景中，慢快协作尚未得到充分探索。更长的距离和更大的空间范围要求采用基于置信度的高效触发器来进行重新规划。现有方法 [8], [9], [23]

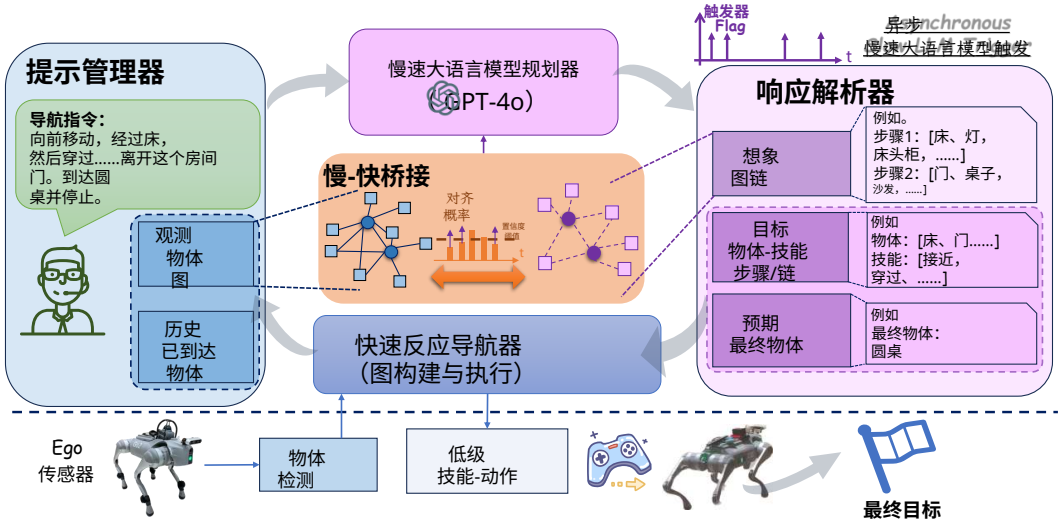


图 2. SFCo-Nav 的系统概述：一种用于高效零样本视觉语言导航的慢快协作框架。慢速大语言模型规划器（Π_{slow}）将导航指令分解为子目标，每个子目标都与一个想象物体图 G_t^i 配对。快速反应导航器（π_{fast}）实时构建感知物体图 G_t^p，并执行低层动作以与 G_t^i 对齐。慢速–快速桥接模块计算图对齐置信度 C_t；高置信度（$C_t > \tau_C$）时由 π_{fast} 保持控制，而低置信度（$C_t \le \tau_C$）则触发 Π_{slow} 进行重新规划。

要么在每一步都进行慢速推理，要么缺乏自适应机制，二者都会造成较高的延迟。这一空白促使我们设计了一个慢速–快速协作零样本视觉语言导航框架，以降低慢速大语言模型的推理开销，便于实际部署。

III. 问题形式化

为解决视觉语言导航中的慢快问题，我们将零样本慢快协作框架形式化如下。我们考虑这样一种零样本导航场景：具身智能体遵循自然语言指令 I，在未知环境中到达目标物体或位置。任务是生成一系列低层动作$\mathbf{A} = (a_0, a1, .., a_T)$，引导智能体到达指令指定的目标，同时尽量减少对计算开销大的推理组件的使用。

在每个时间步t，智能体通过自我中心感知输入感知环境，并维护一个内部状态S_t，用于概括观察结果和任务进度。导航策略采用如下慢速–快速混合架构实现：

1）慢速高层规划器（Π_{slow}）：一种容量强大但成本高昂的推理模块，在被调用时，根据I 和S_t生成或更新高层计划 $\mathscr{P}_t$。2）快速低层控制器（π_{fast}）：一种低延迟策略，利用近期观察结果和当前高层计划 $\mathscr{P}_t$执行局部控制动作。3）协作模块：一种决策机制，根据智能体内部指标决定何时在π_{fast} 和 Π_{slow} 之间切换导航策略，旨在最大限度提高任务成功率，同时尽量减少对 Π_{slow}的依赖。

当智能体在停止距离 d_{stop} 内到达目标时，导航即终止。设计目标是联合优化 Π_{slow}、 π_{fast}和慢-快桥接规则，以最大化任务成功率，同时减少对慢速规划器的依赖，从而提高效率。

IV. 方法

在本节中，为了解决第三节所提出的零样本导航问题，我们介绍所提出的慢-快协作框架 SFCo-Nav。首先介绍 SFCo-Nav 的整体架构，随后详细描述其三个核心组件。

A. 系统概述

如图2所示，SFCo-Nav的架构围绕慢快协作原则设计，以实现高效的具身导航。系统由三个主要模块组成：

- 慢速大语言模型规划器（Π慢）充当“慢速脑”，将用户的指令 I 分解为一系列子目标。每个子目标都包含一个目标物体和所需技能，以及一个想象物体图G_{it}，代表大语言模型对预期子目标场景的语义和空间先验。
- 快速反应导航器（π快）充当“快速脑”，实时运行。它根据自我中心传感器输入（例如 RGB-D）构建局部感知物体图 G_{pt}，并输出低层动作 a_t，以贪心方式最大化G_{pt} 与子目标的G_{it}之间的对齐程度。
- 慢速–快速桥接模块通过计算图对齐置信度分数C_t = Align(G_p 1:$_t$,G_i1:$_i$)，实现异步协作。如果$C_t > \tau_C$，控制权仍由π快速模块掌握。否则，桥接模块会触发 Π慢速模块重新规划。

以下各小节将详细介绍这三个模块各自的设计与实现。

B. 慢速大语言模型规划器：物体-技能-图链生成

将用户的导航指令记为字符串I。智能体对环境的感知由感知物体图G_t^p表示。慢速规划器 Π_{slow} 生成的完整导航计划记为 $\mathscr{P}$。

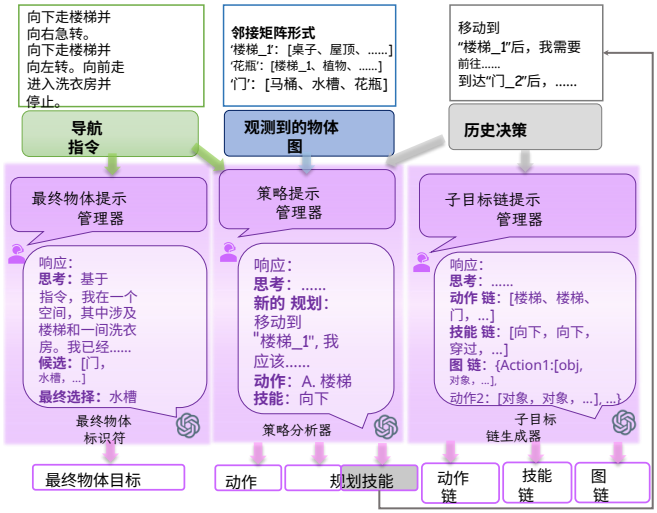


图 3. 慢速大语言模型规划器的提示结构和运行流程。

如图 3 所示，慢速大语言模型规划器由三个功能模块构成，按顺序依次运行：

1) 最终物体标识器 (f_{goal})：此初始化模块解析用户的指令 I，以确定最终导航目标 o_{goal}：

$$o_{\text{goal}} = f_{\text{goal}}(I). \tag{1}$$

大语言模型会被提示提取导航任务的终点物体名称。此输出定义了智能体的全局停止条件。若智能体在最终时间步 s_T 的状态满足 $\text{distance}(s_T, o_{\text{goal}}) \le d^{\text{stop}}$，则任务成功，其中 d^{stop} 是预定义的阈值（例如 3 米）。

2) 策略分析器 (f_{policy})：此模块根据当前上下文生成智能体的即时子目标和长时程战略计划。令 $H_{t-1} = \{o_1, o2, \ldots, o_{t-1}\}$ 表示此前已到达的子目标物体的历史。该模块以指令 I、历史 H_{t-1} 和当前感知物体图 G_t^p 为条件：

$$(R_t, (o_t, sk_t)) = f_{\text{policy}}(I, H_{t-1}, G_t^p). \tag{2}$$

这里，o_t 是即时目标物体，sk_t 是到达该物体所需的技能。推理轨迹 R_t 以自然语言概述高层计划，而子目标 (o_t, sk_t) 则是供下一阶段执行的可执行命令。

3) 子目标链生成器 (f_{chain})：此模块将人类可读的推理轨迹 R_t 转换为结构化、可由机器执行的子目标链 $\mathscr{P}_t$：

$$\mathscr{P}_t = f_{\text{chain}}(R_t). \tag{3}$$

该计划 $\mathscr{P}_t$ 包含 N 个未来子目标：

$$\mathscr{P}_t = [(o_{t+j}, sk_{t+j}, G_{t+j}^i)]_{j=0}^{N-1}, \tag{4}$$

其中，o_{t+j} 表示目标物体，sk_{t+j} 表示所需技能，G_{t+j}^i 表示由大语言模型生成的想象物体图，代表子目标 o_{t+j} 对应预期场景的结构化语义—空间先验。第一个元组 (o_t, sk_t, G_t^i) 对应要由 π_{fast} 执行的即时子目标，除非慢速–快速桥接模块 Π_{slow} 因置信度较低而触发。

C. 快速反应导航器：物体图构建与技能规划器

快速反应导航器负责将慢速规划器（Π_{slow}）的抽象指令落地为低层动作。它通过紧密的感知—动作循环运行，主要承担两项功能：1）将环境构建为实时感知物体图；2）基于该表示执行当前由大语言模型定义的子目标。

1）感知物体图构建：在每个时间步 t，导航器都会构建一个动态属性图 $G_t^p = (V_t, E_t)$，以表示其对场景的局部理解，如图 4 所示。该图将智能体自身包含为一个特殊的参考节点。节点集合 V_t 由一个表示智能体的时间步节点 v_{timestep} 以及当前视野中检测到的每个物体对应的节点 v_j 组成。每个对象节点 v_j 都带有其语义标签（例如椅子）、相对于智能体估计的三维位置，以及其最近一次观测的时间戳。另一方面，边 E_t 表示节点之间的空间关系。为简便起见并提高效率，我们构建一个星形拓扑图，其中的边将时间步节点 v_{timestep} 与每个观测到的对象节点 v_j 相连。每条边 $e_{\text{timestep},\ j} \in E_t$ 都带有从智能体到物体的相对距离和方位角。

与其他使用大型视觉语言模型为所有可导航视点生成描述的零样本视觉语言导航方法不同，我们的物体图仅通过对智能体当前视图进行一次轻量级目标检测来构建，从而减少视觉感知过程中的 token 消耗和时间开销。

2）子目标执行规划器：该规划器的目标是执行慢速规划器 Π_{slow} 提供的当前子目标 (o_t, sk_t, G_t^i)。它实现了一种反应式策略 π_{fast}，将当前感知图 G_t^p、目标物体 o_t 及其对应的技能 sk_t 映射为低级控制动作 a_t。例如，子目标执行规划器可以采用接近、穿过、向上或向下前往目标物体等技能，从而提高导航器执行准确率和效率。

D. 慢-快桥接：属性对象图对齐

慢速–快速桥接模块采用轻量级的异步触发机制，统筹慢速与快速协作，从而提升效率。具体而言，它通过计算先进的结构化归因图 [17] 匹配概率，将其作为导航器置信度 C_t。该置信度分数基于感知物体图 G_t^p 与想象物体图链 G_t^i 之间的对齐结果。

结构化属性图中的顶点表示视点时间步（用户）$\mathscr{V}_{\text{u}}$ 和作为属性的物体 $\mathscr{V}_{\text{a}}$。为描述边的概率模型，我们首先考虑对象–对象顶点对集合 $\mathscr{E}_{\text{u}} \triangleq \mathscr{V}_{\text{u}} \times \mathscr{V}_{\text{u}}$ 和对象–视点顶点对集合 $\mathscr{E}_{\text{a}} \triangleq \mathscr{V}_{\text{u}} \times \mathscr{V}_{\text{a}}$。对于任意顶点对 $e \in \mathscr{E} \triangleq \mathscr{E}_{\text{u}} \cup \mathscr{E}_{\text{a}}$，若存在边，则记为 $G_t^p(e) = 1$；否则记为 $G_t^p(e) = 0$（对于 $G_t^i(e)$，记法类似）。边对的联合观察记为 $(G_t^p, G_t^i)(e)$。边根据相关模型生成。

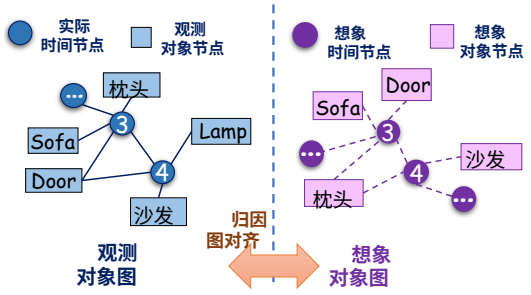


图4. 观测到的与想象的归因图结构 ure.

Algorithm 1: P, Q Computation for Graph Alignment

Input : Perceived Graph $G^p_{1:t}$, Imagined Graph $G^i_{1:t}$
Output: Probability matrices P, Q

$\mathscr{T} \leftarrow \text{time_node}(G^p_{1:t}) \cup \text{time_node}(G^i_{1:t})$;
$\mathscr{O} \leftarrow \text{objects}(G^p_{1:t}) \cup \text{objects}(G^i_{1:t})$;
$Q = \begin{bmatrix} q_{00} & q_{01} \\ q_{10} & q_{11} \end{bmatrix} \leftarrow \mathbf{0}_{2\times 2} \quad P = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} \leftarrow \mathbf{0}_{2\times 2}$;
for $(t,o) \in \mathscr{T} \times \mathscr{O}$ **do**
  $i \leftarrow (t \in G^p_{1:t} \wedge o \in G^p_t)$;
  $j \leftarrow (t \in G^i_{1:t} \wedge o \in G^i_t)$;
  $q_{ij} \leftarrow q_{ij} + 1$;
for $t_m, t_n \in \mathscr{T}, m < n$ **do**
  $i \leftarrow \left(\exists o' \in \mathscr{O} : o' \in G^p_{t_m} \wedge o' \in G^p_{t_n}\right)$;
  $j \leftarrow \left(\exists o' \in \mathscr{O} : o' \in G^i_{t_m} \wedge o' \in G^i_{t_n}\right)$;
  $p_{ij} \leftarrow p_{ij} + 1$;
$P \leftarrow Norm_1(P)$, $Q \leftarrow Norm_1(Q)$;
return P, Q;

对于每个对象–对象对$e \in \mathscr{E}_{\text{u}}$，边概率$P = (p_{11}, p_{10}; p_{01}, p_{00})$定义如下：

$$(G^p_t, G^i_t)(e) = \begin{cases} (1,1) & \text{w.p. } p_{11} \text{ (shared edge)} \\ (1,0) & \text{w.p. } p_{10} \text{ (disagreement)} \\ (0,1) & \text{w.p. } p_{01} \text{ (disagreement)} \\ (0,0) & \text{w.p. } p_{00} \text{ (shared non-edge)} \end{cases} \tag{5}$$

其中 $p_{11} + p_{10} + p_{01} + p_{00} = 1$。对于每个对象–视点对 $e \in \mathscr{E}_{\text{a}}$，我们以概率$Q = (q_{11}, q_{10}; q_{01}, q_{00})$定义一个类似的分布。给定感知图 G^p_t和想象图 G^i_t，可以通过算法1近似求得P，Q 。然后我们定义：

$$\psi_{\text{u}} \triangleq \left(\sqrt{p_{11}p_{00}} - \sqrt{p_{10}p_{01}}\right)^2, \tag{6}$$

$$\psi_{\text{a}} \triangleq \left(\sqrt{q_{11}q_{00}} - \sqrt{q_{10}q_{01}}\right)^2. \tag{7}$$

为了使导航器有效，我们需要衡量感知图 G^p_t与想象图G^i_t之间对齐的置信度。我们定义：

- 对齐是n 个物体在G_{pt} 与m 个物体在G_{it}之间的一种映射或置换。我们用 π表示一个特定的对齐。
- $\mathscr{S}_n$ 是所有 $n!$ 种可能对齐的集合。
- πid 是恒等对齐，表示真实且正确的对齐，其中 G_{pt} 中的每个物体都与G_{it}中的对应物体相匹配。
- $\delta_\pi(G_{pt}, G_{it})$是对齐差异。该分数衡量真实对齐$\pi$id相较于错误对齐 π更有可能成立的程度。如果 $\delta_\pi > 0$, 则之前的对齐更为匹配。

如果存在任何看起来与真实对齐同样好或更好的错误对齐 $\pi \in \mathscr{S}_n \setminus \{\pi_{\text{id}}\}$，就会发生匹配错误。当 $\delta_\pi(G^p_t, G^i_t) \le 0$时，就会出现这种情况。

因此，模糊对齐的概率定义为

$$\text{P(A)} = \text{P}\left(\exists \pi \in \mathscr{S}_n \setminus \{\pi_{\text{id}}\}, \delta_\pi\left(G^p_t, G^i_t\right) \le 0\right). \tag{8}$$

精确计算这一概率通常难以实现。相反，我们依据最新的图对齐理论 [17],给出的上界，该上界限定了这一错误概率可能达到的最大值。给定 $\left(G^p_t, G^i_t\right)$，其节点数分别为 n 和 m，上界可通过以下方式计算：

$$\text{P(A)} \le e^{-2\log n + 2m\psi_{\text{a}} + 2np_{11}}. \tag{9}$$

根据这一错误概率上界，导航器的置信度C_t 为

$$C_t = 1 - \text{P(A)} \ge 1 - e^{-2\log n + 2m\psi_{\text{a}} + 2np_{11}}. \tag{10}$$

如果$C_t > \tau_C$，则认为G^p_t与G^i_t 之间的对齐具有C_t的可靠概率，智能体无需调用代价高昂的慢速系统 Π_{slow}即可继续执行，从而实现更高的效率。我们的 SFCo-Nav 的操作流程如算法 2 所示。

V. 实验

本节将 SFCo-Nav 与一些先进的零样本视觉语言导航系统进行比较。首先，我们介绍实验设置、评估指标以及用于测试的视觉语言导航环境。随后，展示在开放数据集上开展的实验，重点关注两个方面：1）整体任务成功表现；2）SFCo-Nav 的令牌/时间计算效率。

A. 实验设置

1）评估指标：视觉语言导航系统从三个层面进行评估：任务有效性、令牌效率和时间效率。

任务层面指标使用标准视觉语言导航指标 [1]评估导航有效性和路径效率。成功率（SR）衡量任务完成情况。按路径长度加权的成功率（SPL）衡量成功率和效率。导航误差（NE）衡量与目标的最终距离。预言机成功率（OSR）衡量轨迹质量。轨迹长度（TL）衡量路径长度。

令牌层面指标量化计算成本，以验证效率提升。

- 每条路径的语言令牌数 (L-Tok)：每回合处理的大语言模型令牌的平均数量，用于衡量慢速系统的成本。
- 每条路径的视觉令牌数[1] (V-Tok)：平均数量

[1]视觉令牌数根据 GPT-4o 分词器测量。

算法 2： SFCo-Nav：慢速–快速协作
零样本视觉语言导航

输入： 导航指令 I
输出： 动作序列 $\mathbf{A} = (a_0, a1, .., a_T)$
初始化： $G_0^p \leftarrow /0$，$H_0 \leftarrow /0$;
调用 Π_{slow} 以通过公式 (1) 确定最终目标o^{goal}；
生成初始计划 $\mathscr{P}_0 \leftarrow f_{\text{chain}}(f^{\text{policy}}(I, H_0, G_0^p))$;
对于 $t \leftarrow 1$ **到** T **执行**
  1. 感知与局部图更新：
    构建感知物体图G_t^p从机载传感器;
  2. 置信度评估：
    从 $\mathscr{P}_{t-1}$检索当前想象图G_t^i；
    通过公式（10）计算C_t，比较G_t^p与G_t^i;
  如果$C_t \le \tau_C$，**则**
    3a. 慢速大语言模型触发：
      将已到达子目标追加至H_{t-1};
      $(R_t, (o_t, sk_t)) \leftarrow f^{\text{policy}}(I, H_{t-1}, G_t^p)$;
      $\mathscr{P}_t \leftarrow f_{\text{chain}}(R_t)$；
  else
    3b. 快速导航器执行：
      保留 $\mathscr{P}_{t-1}$，而不触发大语言模型;
  4. 反应式导航：
    $a_t \leftarrow \pi_{\text{fast}}(G_t^p, o_t, sk_t)$; 执行 a_t
    弹出 $\mathscr{P}_{t-1}$中第一个已执行的项目;
  如果已到达o^{goal} 以内d^{stop} **则**
    终止并返回 $\{a_k\}_{k=0}^t$; **跳出**;

每回合处理的视觉令牌数，用于衡量视觉感知负载。

- 每路径统一令牌数（U-Tok）：表示整体令牌吞吐量的综合指标，计算方式为

$$\text{U-Tok} = \text{L-Tok} + \lambda \times \text{V-Tok}, \tag{11}$$

其中，λ 是成本系数，用于衡量视觉令牌相对于语言令牌的权重。根据 OpenAI 报告的推理成本[2]，，我们将 $\lambda = 4$ 设定为相应数值，以反映每个视觉令牌的处理成本约为文本令牌的四倍。

时序级指标用于衡量实际部署中的墙钟延迟。

- 每路径大语言模型耗时（L-Time）：每个回合所有大语言模型推理调用的墙钟总时间（秒），用于捕捉慢系统延迟。
- 每路径视觉处理耗时[3]（V-Time）：每个回合所有视觉处理的总墙钟时间（秒），用于衡量视觉感知延迟。
- 每路径总耗时（T-Time）：用于衡量整体回合延迟，计算方式为

$$\text{T-Time} = \text{L-Time} + \text{V-Time}. \tag{12}$$

[2]OpenAI 的令牌价格可参考 https://platform. openai.com/docs/pricing#image-tokens。[3]我们的方法使用 Grounding-DINOv2 [24],测量视觉感知时间，而基于视觉语言模型的基线方法则使用 BLIP-2。

2）实验数据集的开发：为展示 SFCo-Nav 的有效性，我们在公开的视觉语言导航数据集和真实世界套件环境中开展了实验。

R2R [1]：视觉语言导航的标准基准，要求智能体在逼真的室内环境中遵循详细的基于路径的指令。

REVERIE [18]：更具挑战性的目标定位基准，智能体需遵循高层次目标导向指令，找到特定的目标物体。

真实世界套件：在室内酒店套房环境中部署于实体足式机器人上，以测试系统在真实世界条件下的稳健性和实际性能。

3）对比算法：SFCo-Nav 与最新的（部分）基于零样本视觉语言模型与大语言模型的视觉语言导航方法进行比较，具体如下：

- NavGPT [8]：一种基础方法，将大语言模型用作零样本规划器，完全依据视觉语言模型从视觉观察中获得的文本描述进行决策。
- MapGPT [9]：该方法通过构建实时拓扑地图，并将视觉语言模型生成的地图文本描述输入大语言模型进行导航，为大语言模型规划器增添了显式的空间记忆。
- NavCoT [14]：该方法采用思维链（CoT）提示来提升推理能力，指示大语言模型在选择动作前生成明确的思考过程；该方法基于零样本视觉语言模型和经过微调的大语言模型。
- SF-Nav：我们提出的慢速–快速框架变体，不包含用于异步大语言模型触发的慢速–快速桥接模块。在此设置下，每个导航步骤都会调用慢速大语言模型规划器。

B. 实验结果与讨论

*1) R2R*数据集：如表 I 所示，在 R2R基准上，SFCo-Nav 实现了准确性与效率之间的良好平衡。其任务成功率指标具有竞争力，与以往的零样本视觉语言导航基线相当或更优，同时大幅降低了计算成本。与 NavGPT、MapGPT、NavCoT 和 SF-Nav 相比，SFCo-Nav 凭借异步慢速–快速触发机制，实现了最低的统一令牌用量。这一减少直接带来了最快的总推理时间：比 MapGPT 快 4×，比 SF-Nav 快 33%，且不牺牲导航质量。这些结果证实，SFCo-Nav 在达到接近最先进水平的准确性的同时，实现了无与伦比的令牌和时间效率。

2）*REVERIE*数据集：在 REVERIE 数据集上，SFCo-Nav 的导航误差为 7.16 米，成功率为 31.33%，排名仅次于 SF-Nav，优于所有其他基线方法，如表 II 所示。在效率方面，SFCo-Nav 将大语言模型令牌消耗降至 9.44k，

表 I R2R数据集上的比较。

方法	任务成功指标					令牌效率度量			时间效率指标[s]		
	TL	NE	OSR	SR	SPL	V-Tok	L-Tok	U-Tok	V-Time	L-Time	T-Time
NavGPT (GPT-4)	11.45	6.46	42.0	34.0	29.0	108k	64.62k	496.62k	1440	48.19	1488.19
MapGPT（GPT-4）	–	6.29	**57.6**	38.8	25.8	4.95k	9.46k	29.26k	66	41.01	107.01
NavCoT（微调* Llama）	**9.83**	6.67	44.0	36.4	33.17	4.50k	**1.42k**	19.42k	60	**16.42**	76.42
SF-Nav（GPT-4o）	10.89	**5.70**	53.8	**41.3**	**36.35**	**0.9k**	13.10k	16.10k	**3.36**	36.0	39.36
SFCo-Nav（GPT-4o）	10.80	6.04	50.5	38.2	32.54	**0.9k**	9.94k	**13.54k**	**3.36**	22.8	**26.16**

表 II REVERIE数据集上的对比。

方法	任务成功指标					令牌效率度量			时间效率指标[s]		
	TL	NE	OSR	SR	SPL	V-Tok	L-Tok	U-Tok	V-Time	L-Time	T-Time
NavGPT (GPT-4)	–	–	28.3	19.2	14.6	108.00k	57.35k	489.35k	1440	49.74	1489.74
MapGPT (GPT-4)	–	–	42.6	28.4	14.5	4.95k	7.56k	27.36k	66	26.57	92.57
NavCoT（微调* Llama）	12.36	–	14.20	9.20	7.18	4.50k	**1.44k**	19.44k	60	**11.40**	71.40
SF-Nav (GPT-4o)	**10.05**	7.87	**46.63**	**35.87**	**30.62**	**0.9k**	14.04k	17.64k	**3.36**	31.4	34.76
SFCo-Nav (GPT-4o)	10.32	**7.16**	42.36	31.33	27.01	**0.9k**	9.44k	**13.04k**	**3.36**	21.6	**24.96**

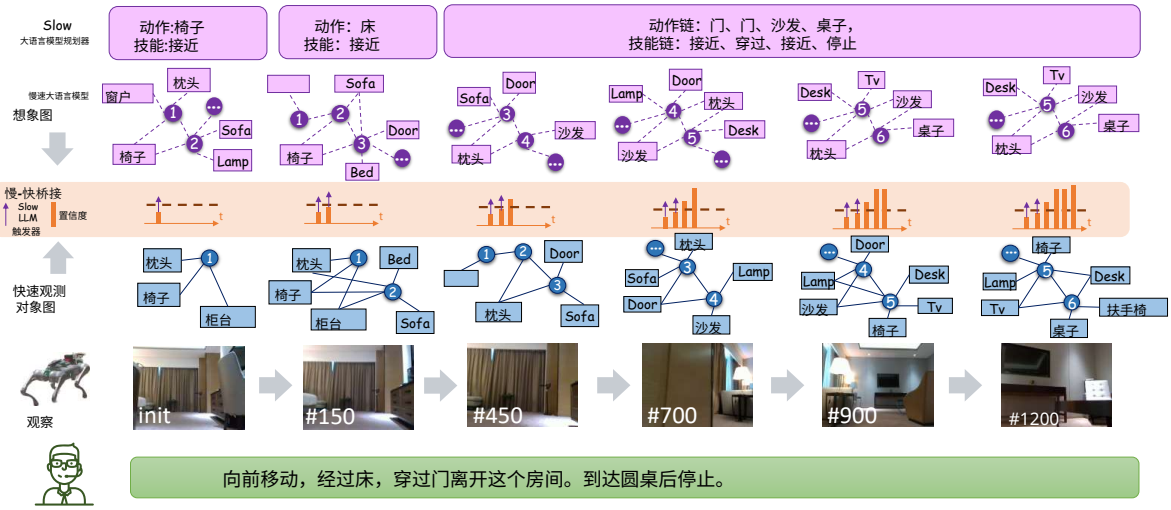


图5. SFCo-Nav在真实世界酒店套房中的部署。导航初期，稀疏观测会产生较低的匹配概率，从而触发慢速规划器。随着观察增多，置信度超过阈值，进而实现快速、无需大语言模型的执行。这种快慢协作在保持成功率的同时，提高了时间效率并减少了令牌用量。

表III R2R子集上不同置信度水平的比较。

SFCo-Nav- 置信度阈值	任务成功指标			效率指标	
	SR	OSR	SPL	U-Tok	T-Time[s]
SFCo-Nav-1.0	**39.67**	49	**37.08**	12.58k	28.08
SFCo-Nav-0.95	36.67	**50.33**	34.10	10.55k	22.5
SFCo-Nav-0.85	36.67	**50.33**	34.03	10.42k	22.2
SFCo-Nav-0.6	33.33	48.33	31.22	9.77k	20.34
SFCo-Nav-0.4	29.67	45	27.39	**9.51k**	**19.64**

比 SF-Nav 减少了 33%，在所有方法中实现了最低的统一令牌用量。如此精简的令牌占用使其总推理时间仅为 24.96 秒，速度比 MapGPT 快 3.7× 倍以上，比 NavGPT 快 59× 倍以上。这些结果表明，SFCo-Nav 在保持竞争力的任务性能的同时，显著提升了计算效率和时间效率。

3) 消融研究：消融研究从两个主要方面展开：a) 置信度阈值的影响。b) 使用不同预训练大语言模型对想象链和物体-技能对模块进行消融。为降低评估成本，使用了包含300条指令轨迹的R2R子集

表 IV SFCo-Nav在R2R子集上的消融研究。

LLMs	链 决策	物体-技能 Pair	SR	U-Tok	T-Time[s]
GPT-4o	×	×	32	13.29k	32.90
	✓	×	31.33	10.13k	28.73
	×	✓	45.67	13.06k	31.05
	✓	✓	37	9.9k	27.47
Deepseek-V3.1	×	×	32.67	15.07k	99
	✓	×	31	12.87k	96.84
	×	✓	41	15.91k	102.24
	✓	✓	34	12.52k	97.56

针对本次消融研究。

a) 置信度阈值的影响：我们通过将调用慢速大语言模型规划器的触发阈值从 1.0 调至 0.4，研究置信度阈值对 SFCo-Nav 性能的影响。如表III所示，较高的阈值通常能带来更优的任务成功指标，但也会增加 token 数量和时间开销。具体而言，SFCo-Nav-1.0 的 SR 和 SPL 最高，但每条轨迹需要消耗 12.58k 个 token，运行 28.08 秒。相比之下，SFCo-Nav-0.4 的计算量最低，总 token

数最少，运行时间最快，但 SR 降至 29.67%，SPL 降至 27.39%。中间阈值 0.85–0.95 则实现了更好的平衡：运行时间减少 20%，同时 SR 与最高值的差距保持在 3% 以内。这些结果凸显了由置信度阈值决定的准确率与效率之间的权衡。

b) 不同大语言模型下想象链与物体–技能对模块的消融实验：我们使用 GPT-4o 和 Deepseek-V3.1 作为骨干模型，评估了 SFCo-Nav 中想象链和物体–技能对模块的影响，如表 IV 所示。在两种大语言模型下，物体–技能对都能稳定提升成功率，而想象链主要减少令牌用量和运行时间，对成功率的影响较小。两者结合后实现了较为均衡的权衡：在 GPT-4o 下，令牌用量最低、运行时间最快；在 Deepseek 下，令牌用量从 15.07k 降至 12.52k，而成功率仅略有下降。这些结果表明，物体–技能对能够提升导航质量，而想象链则能提高效率。

4) 真实世界套房案例研究：为评估 SFCo-Nav 在真实世界中的部署能力，我们将其部署在一台足式机器人上，让机器人在配有家具的酒店套房中导航，如图5所示。在早期步骤中，观测到的物体较少，因此匹配概率和置信度较低，触发慢速大语言模型规划器进行规划和多步想象。随着观察增多，置信度超过阈值。此时，快速模块无需额外的大语言模型调用即可执行任务。这种异步的慢速–快速触发机制在保持导航成功率的同时，提高了时间效率并减少了令牌用量。这表明 SFCo-Nav 能够在真实世界中实现高效、准确的导航。

VI. 结论

本文提出了 SFCo-Nav，这是一种用于零样本视觉语言导航的慢快协作框架，通过异步置信度触发桥接机制，将基于慢速大语言模型的规划器与快速反应导航器相结合。在 R2R 和 REVERIE 上进行的实验表明，SFCo-Nav 达到了具有竞争力或更高的成功率，同时显著降低了推理成本。在配备家具的酒店套房中，将该方法部署于足式机器人进行现实世界测试，展示了其仅在必要时自适应调用大语言模型推理的能力，在保持导航准确性的同时提高了时间效率并减少了令牌用量。这些结果凸显了 SFCo-Nav 在现实世界环境中实现实用、高效具身人工智能的潜力。

参考文献

[1] P. Anderson、Q. Wu、D. Teney、J. Bruce、M. Johnson、N. S¨ underhauf、I. Reid、S. Gould 和 A. V. D. Hengel，“视觉与语言导航：在真实环境中理解基于视觉定位的导航指令”，载于*IEEE* 计算机视觉与模式识别会议论文集（*CVPR*），2018 年，第 3674–3683 页。[2] Y. Liu、L. Liu、Y. Zheng、Y. Liu、F. Dang、N. Li 和 K. Ma，“具身导航”，《中国科学：信息科学》，第 68 卷，第 4 期，第 141101 页，2025 年。[3] Z. Yang、L. Li、K. Lin、J. Wang、C.-C. Lin、Z. Liu 和 L. Wang，“大语言多模态模型的黎明：使用 GPT-4V（ision）进行初步探索”，*arXiv* 预印本 *arXiv:2309.17421*，2023 年。[4] DeepSeek-AI，“Deepseek-v3 技术报告”，*arXiv* 预印本 *arXiv:2412.19437*，2025 年。

[5] OpenAI，“Gpt-4 08774, 2024 2303 技术报告”，arXiv 预印本 arXiv: . .[6] Y. Qiao、W. Lyu、H. Wang、Z. Wang、Z. Li、Y. Zhang、M. Tan 和 Q. Wu，“Open-nav：利用开源大语言模型在连续环境中探索零样本视觉与语言导航”，载于 IEEE 国际机器人与自动化会议论文集（ICRA），2025 年。[7] C. Huang、O. Mees、A. Zeng 和 W. Burgard，“用于机器人导航的视觉语言地图”，载于 IEEE 国际机器人与自动化会议论文集（ICRA），2023 年。[8] G. Zhou、Y. Hong 和 Q. Wu，“NavGPT：利用大语言模型对视觉与语言导航进行显式推理”，载于 AAAI 人工智能会议论文集（AAAI）。AAAI Press，2024 年，第 849–857 页。[9] J. Chen、B. Lin、R. Xu、Z. Chai、X. Liang 和 K.-Y. Wong，“MapGPT：面向视觉与语言导航的地图引导提示与自适应路径规划”，载于计算语言学协会年会论文集（ACL）。泰国曼谷：计算语言学协会，2024 年，第 9796–9810 页。[10] W. Zhang、C. Gao、S. Yu、R. Peng、B. Zhao、Q. Zhang、J. Cui、X. Chen 和 Y. Li，“CityNavAgent：结合分层语义规划与全局记忆的空中视觉语言导航”，载于计算语言学协会年会论文集（ACL）。奥地利维也纳：计算语言学协会，2025 年，第 31 292–31 309 页。[11] A.-C. Cheng、Y. Ji、Z. Yang、X. Zou、J. Kautz、E. Biyik、H. Yin、S. Liu 和 X. Wang，“NaVILA：用于导航的足式机器人视觉语言动作模型”，载于机器人科学与系统会议论文集（RSS），2025 年。[12] S. Chen、P.-L. Guhur、M. Tapaswi、C. Schmid 和 I. Laptev，“着眼全局，立足局部：用于视觉与语言导航的双尺度图变换器”，载于 IEEE/CVF 计算机视觉与模式识别会议论文集（CVPR），2022 年，第 16 537–16 547 页。[13] S. Chen、P.-L. Guhur、C. Schmid 和 I. Laptev，“用于视觉与语言导航的历史感知多模态变换器”，载于神经信息处理系统进展（NeurIPS），2021 年。[14] B. Lin、Y. Nie、Z. Wei、J. Chen、S. Ma、J. Han、H. Xu、X. Chang 和 X. Liang，“Navcot：通过学习解耦推理提升基于大语言模型的视觉与语言导航”，IEEE《模式分析与机器智能汇刊》，第 47 卷，第 7 期，第 5945–5957 页，2025 年。[15] J. Li、D. Li、S. Savarese 和 S. Hoi，“BLIP-2：利用冻结图像编码器和大语言模型引导语言图像预训练”，arXiv 预印本 arXiv:2301.12597，2023 年。[16] N. Zhang、Z. Wang、W. Wang 和 L. Wang，“属性图对齐”，IEEE《信息论汇刊》，第 70 卷，5934，2024 8，第 5910– 页。[17] Y. Qi、Q. Wu、P. Anderson、X. Wang、W. Y. Wang、C. Shen 和 A. van den Hengel，“REVERIE：真实室内环境中的远程具身视觉指代表达”，载于 IEEE/CVF 计算机视觉与模式识别会议论文集（CVPR），2020 年，第 9982–9991 页。[18] Z. Wang、J. Li、Y. Hong、Y. Wang、Q. Wu、M. Bansal、S. Gould、H. Tan 和 Y. Qiao，“扩展视觉与语言导航的数据生成”，载于 IEEE/CVF 国际计算机视觉会议论文集（ICCV），2023 年。[19] P. Chen、X. Sun、H. Zhi、R. Zeng、T. H. Li、G. Liu、M. Tan 和 C. Gan，“A^2 nav：利用基础模型的视觉与语言能力实现动作感知零样本机器人导航”，载于 NeurIPS 研讨会论文集，2023 年，arXiv:2308.07997。[21] B. Lin、Y. Nie、Z. Wei、Y. Zhu、H. Xu、S. Ma、J. Liu 和 X. Liang，“利用大模型实现视觉语言导航中的可纠正地标发现”，IEEE《模式分析与机器智能汇刊》，第 46 卷，第 12 期，第 8534–8548 页，2024 年。[22] H. Xue、J. Ren、W. Chen、G. Zhang、Y. Fang、G. Gu、H. Xu 和 C. Lu，“反应式扩散策略：面向高接触操作的慢快视觉触觉策略学习”，载于机器人科学与系统会议论文集（RSS），2025 年。[23] Z. Zhan、L. Yu、S. Yu 和 G. Tan，“MC-GPT：通过记忆地图和推理链赋能视觉与语言导航”，arXiv 预印本 arXiv:2405.10620，2024 年。[24] S. Liu、Z. Zeng、T. Ren、F. Li、H. Zhang、J. Yang、Q. Jiang、C. Li、J. Yang、H. Su、J. Zhu 和 L. Zhang，“Grounding DINO：将 DINO 与基于定位的预训练相结合，实现开放集目标检测”，载于欧洲计算机视觉会议论文集（ECCV）。Springer Nature Switzerland，2024 年，第 38–55 页。