

ASENA: 用于具身导航的自进化智能体

An-Chieh Cheng¹, Isabella Liu¹, Edmund Bu², Johan Bjorck¹, Hongxu Yin¹, Zhengyi Luo¹, Jan Kautz¹, Linxi “Jim” Fan¹, Yuke Zhu¹ and Sifei Liu¹

¹NVIDIA, ² 加州大学圣迭戈分校

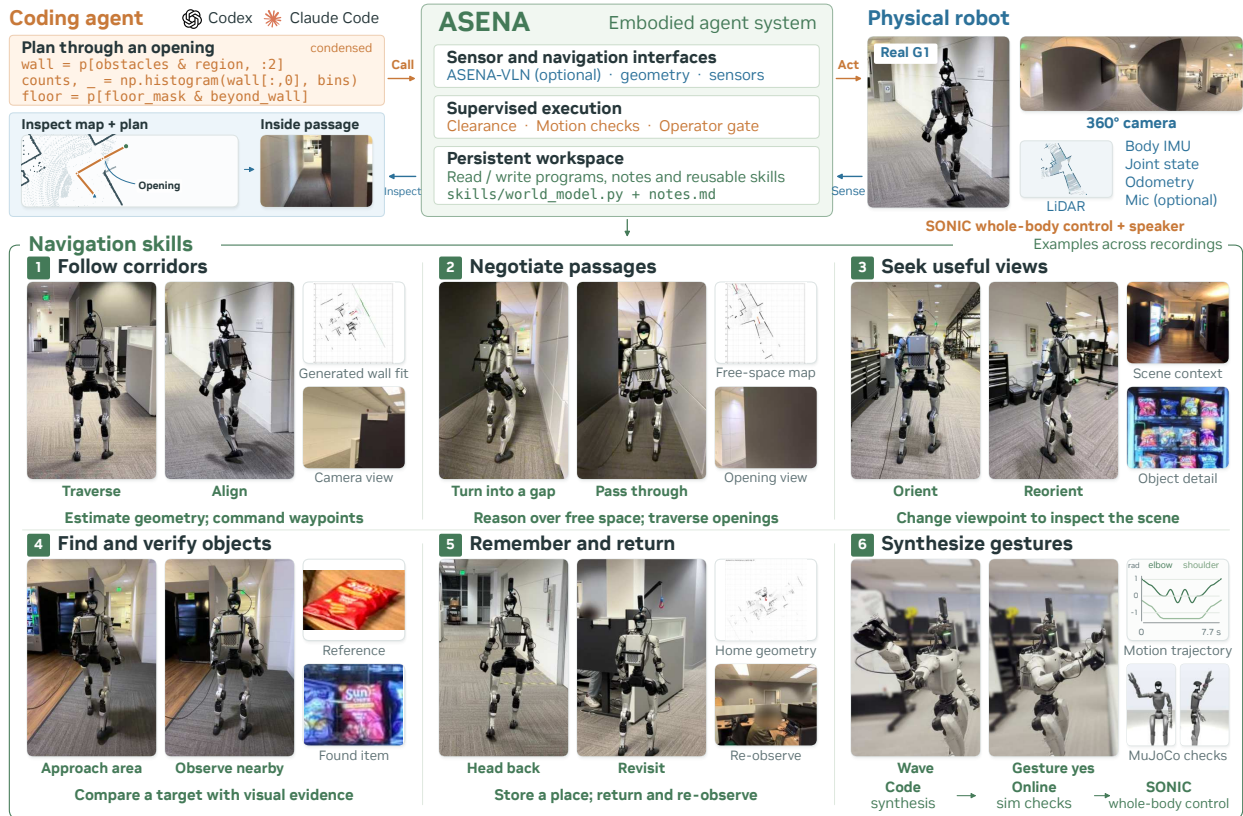


图 1: ASENA 将编程智能体与实体机器人技能相连接。程序结合感知、计算、监督执行与可复用技能，并将学习式导航作为可选工具。图中的编程示例通过分析 LiDAR 几何信息规划一条穿过开口的路线；插图展示了机器人随后在通道内的观测。底部各图展示了系统演进所得的部分技能。 <https://asena-bot.github.io>。

摘要

本文提出 ASENA —— 一个将通用编程智能体与机器人感知、计算、监督执行及持久化经验相连接的具身智能体系统。在模型权重保持固定的情况下，智能体能够编写并执行程序、检查记录的结果、修复故障，以及复用笔记和可执行技能。本文还提出 ASENA-VLN —— 一种作为可选工具接入该可编程系统的 4B 单目导航策略。ASENA-VLN 采用共享视觉-语言解码器预测长距离路线和短时域行为的机体坐标系轨迹；该解码器基于路线指令、视觉问答以及新整理的几何推导原子导航任务数据集进行训练。作为独立策略，ASENA-VLN 在 R2R 和 RxR 上分别取得 68.7% 和 70.2% 的成功率，达到当前最佳水平。与编程智能体集成后，学习式导航使 ASENA 在两个智能体基准上的成功率均提高 11 个百分点，同时缩短执行时间。借助持久化工作区演进和模拟器反馈，在反复出现的 100 个任务子集上迭代十轮后，R2R 的成功率从 72% 进一步提高到 98%，RxR 则从 65% 提高到 89%。在具身问答 (Embodied Question Answering, EQA) 任务上，ASENA 以更少的交互步数达到当前最佳准确率。最后，Unitree G1 上的真实世界演示在无预构建地图的条件下结合了搜索、视觉检查、空间推理与动作合成，表明在线编程能够将机器人行为扩展至路线跟随和预定义技能之外。

1. 引言

实用的导航机器人不能只会抵达目的地。若被要求检查自动售货机中是否有某件商品并返回答案，机器人必须判断搜索位置、识别商品、记住用户位置，并传达搜索结果。因此，导航只是更大任务的一部分，后续步骤取决于机器人的观测。

学习式导航策略提供运动能力，但此类请求还要求机器人判断需要检查、计算和传达的内容。这些流程依赖新的观测，且可能超出预定义技能的范围。编程智能体可在任务执行期间通过程序化接口访问感知、计算、仿真与动作，进而构建并修订这些流程。实体部署还需要与结果关联的反馈、在程序变化后仍然有效的安全防护机制，以及用于保留实用流程的持久化工作区。

本文提出提供上述能力的具身智能体系统 ASENA（图 1）。智能体能够读取传感器数据、推导几何信息、执行代码、检查生成的可视化结果并请求机器人动作。ASENA 通过共享接口、监督执行及相互关联的执行记录连接这些操作。持久化工作区用于存储笔记和可执行技能，使系统能够在模型权重保持固定的情况下，通过修订程序实现改进。

在具备适当训练数据时，学习式策略可进一步扩展这套可编程工具集。将运动委托给专用策略，既能减少智能体的重复决策，也能支持难以用几何规则描述的导航行为。本文开发了 ASENA-VLN ——一种基于路线指令和几何推导导航原子训练的 4B 单目导航策略。在视觉语言导航 (Visual-Language Navigation, VLN) [1–5] 的基础上，ASENA-VLN 能够为长距离路线和短时域目标预测机体坐标系轨迹，例如接近物体或穿过门口。本文整理了新的原子导航数据集，直接监督这些可复用的细粒度行为，并对传统路线级训练形成补充。该策略仍是可选工具：程序可以直接调用它、将其与几何导航结合，或使用其他可用操作完成任务。

本文的主要贡献包括：(1) 提出一个具身智能体系统，使编程智能体能够进行可检查、受监督的任务编程并持续获取技能；(2) 提出达到当前最佳水平的单目导航策略，同时支持路线跟随和可复用的原子技能；(3) 通过评估表明，模型权重保持固定时的工作区演进能够提升任务性能。本文在仿真环境中评估导航与具身问答 (EQA)，并在无预构建地图的真实 Unitree G1 机器人上演示复杂指令跟随与用户交互。

2. 相关工作

以代码为策略. Code as Policies [6] 在程序中组合感知与控制 API。CaP-X [7] 提出 CaP-Gym，并以基准测试研究 API 抽象和执行反馈如何影响面向操作任务的编程智能体。ASPIRE [8] 利用多模态执行轨迹诊断故障、修复机器人程序并保留可复用技能。

智能体自我改进. Voyager [9] 保留并修复可执行技能。Reflexion [10] 在不更新权重的情况下保留文本反馈。ENPIRE [11] 将真实机器人复位与验证同智能体驱动的程序合成及策略训练相结合。相比之下，ASENA 研究如何在所有模型权重保持固定的情况下，围绕预训练导航策略演进持久化工作区。本文在仿真环境中量化工作区演进，并在实体机器人上展示由此获得的监督执行能力。

语言引导导航. DUET 和 ETPNav [12; 13] 等拓扑方法，以及 NaVid、NaVILA 和 StreamVLN [3–5] 等视频到动作策略，均致力于视觉指令跟随。LM-Nav [14] 组合多个预训练模型，InstructNav [15] 则将零样本推理与通用运动基元相结合。DualVLN 和 Qwen-RobotNav [16; 17] 将高层推理与学习式导航策略集成，Uni-LaViRA [18] 研究统一的语言-视觉-动作转换。本文的 VLN 策略与视频到动作导航策略最为相关，但侧重轻量级 4B 模型及其细粒度原子能力；编程智能体可以将这些能力与其他工具和保留程序组合使用。本文既评估该策略的独立导航性能，也评估它在完整具身智能体系统中的效用。

具身问答. Explore-EQA [19] 研究主动探索，直至智能体能够以足够高的置信度作答；3D-Mem 和 GraphEQA [20; 21] 则将观测组织为空间记忆。MemoryEQA [22] 将这一设定扩展到更长时域的多目标问题，FAST-EQA [23] 则侧重高效收集证据。本文使用这些 EQA 设定评估 ASENA 组合感知、导航和记忆工具的能力，以及在模型权重保持固定的情况下，通过修订持久化笔记和可执行技能实现改进的能力。

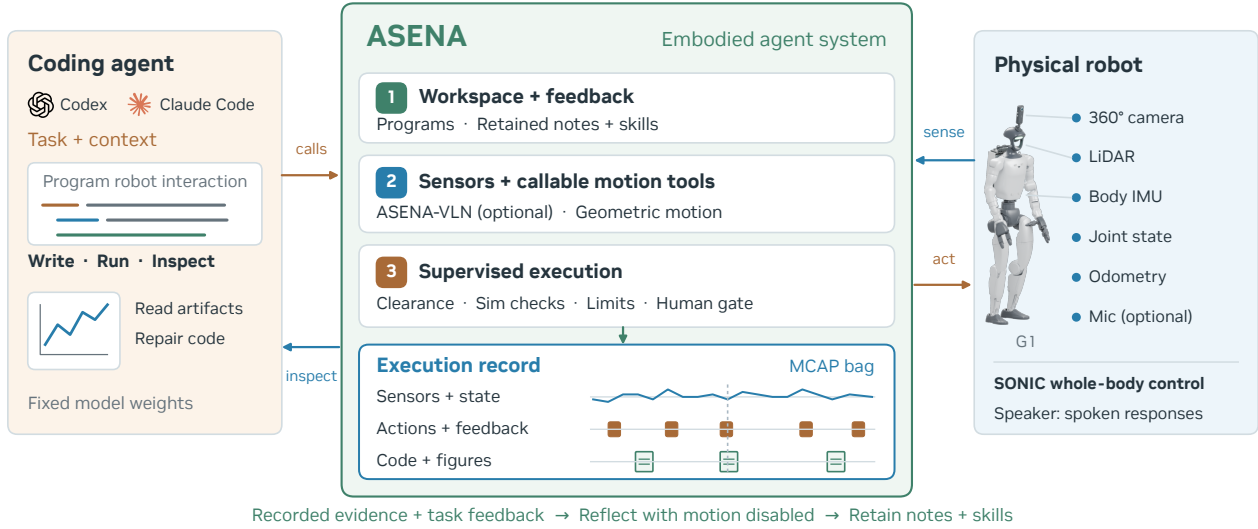


图 2: ASENA 将编程智能体连接至实体机器人。该系统集成机器人感知、持久化工作区、监督执行以及可选的学习式导航策略。每次运行均存储为 MCAP 记录 [24]，将传感器与机器人状态同动作、操作员反馈、代码和可视化结果对齐，以供检查与回放。机器人运动禁用期间，智能体会反思这些记录并更新笔记与技能。

3. 方法

ASENA 为编程智能体提供实体机器人的程序化访问能力（图 2）。该系统将感知、计算、可视化、仿真和动作整合到统一的编写-运行-检查流程中。每次运行均受物理安全保护机制约束，并被记录下来以供后续检查、修复和复用。智能体还可在同一接口中将学习式导航策略作为可选工具调用。

3.1. 以机器人反馈验证程序

程序必须依据物理反馈检验其假设。在 G1 上，ASENA 开放 360 度相机、LiDAR、里程计、机体 IMU 和关节状态。编程智能体处理并可视化这些输入，以估计几何结构、检查物体并监控执行。当动作偏离预期时，智能体可检查记录的证据并修改程序。

生成程序请求的所有物理动作均通过 ASENA 的执行接口，并由 SONIC 全身控制器 [25] 执行，同时受下述验证与审批机制约束。在此接口中，智能体可使用几何导航工具，显式推理航点和安全间距，也可调用学习式策略执行语言条件运动。

ASENA 将感知、控制和安全接口封装为可调用的原子技能，例如接近物体、穿过门口、说话或做手势。智能体将这些操作与计算和交互逻辑相结合，构建规模更大的可复用技能；在保留相同物理执行接口的同时，对任务级程序进行调整。

安全保护机制。与桌面操作相比，人形机器人的全身运动会引入更广泛的安全风险：机器人在共享物理空间中移动整个机体时必须保持平衡。由于合成程序能快速改变这种行为，本文采用独立于程序生成过程的安全保护机制控制物理执行。首先在 MuJoCo 中在线验证合成手势，检查姿态连续性、关节与运动限位以及采样自碰撞。若提案未通过验证，则将违反的约束返回给智能体进行修复。此外，任何已验证动作由 SONIC [25] 执行前，仍须获得操作员批准。

在导航中，本文使用基于 LiDAR 的占据地图进行路径规划和障碍间距检查，并采用独立安全保护机制实施运动限制；当里程计数据过期或电池电量过低时，该机制会拒绝命令。心跳触发停止和人工紧急停止进一步保障执行安全。这些机制贯穿程序合成、仿真验证、人工授权和物理执行，形成分层安全保护。

3.2. 将执行转化为保留经验

ASENA 以 MCAP 格式 [24] 记录运行过程，将传感器与机器人状态同动作、操作员反馈、代码和可视化结果对齐。通过回放这些证据，智能体能够定位执行偏离预期之处，并修复相应程序。

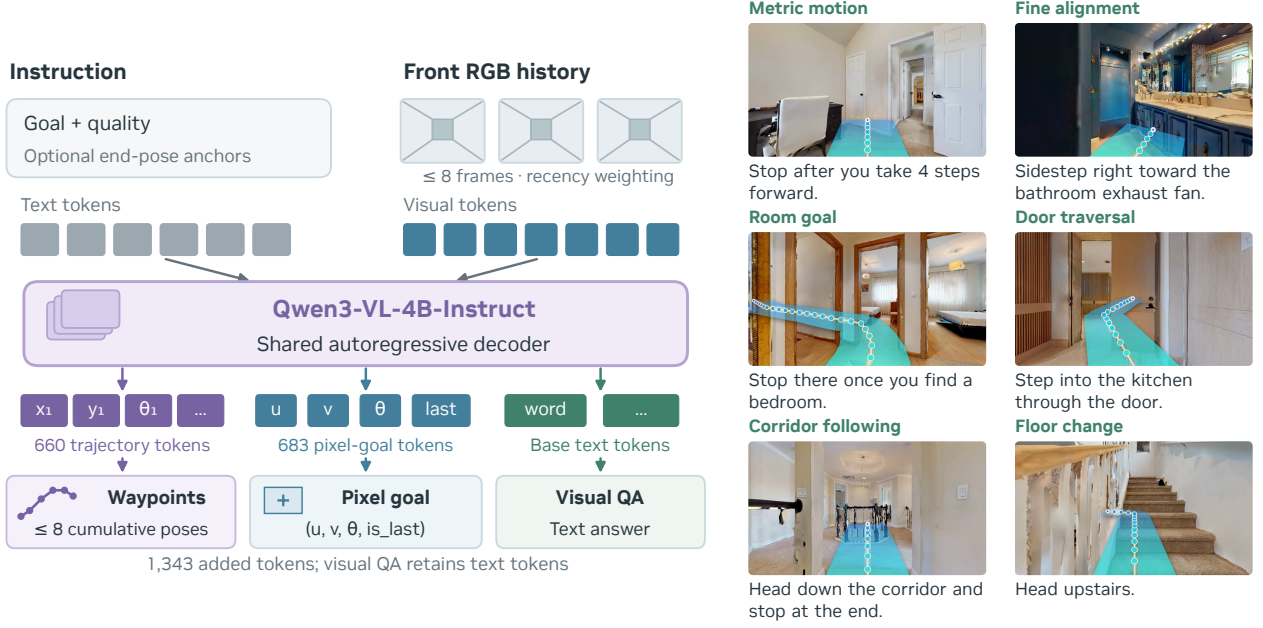


图 3: 导航策略架构与原子训练示例。左: 语言和单目历史观测输入共享自回归解码器。不同词元块分别编码轨迹和像素目标; 视觉问答使用文本词元。运动词元解码为机体坐标系航点。架构图标仅为示意。右: 带有指令和叠加轨迹的训练观测记录; 叠加轨迹未使用度量相机投影。更多示例见 图 6。

修复故障后, ASENA 可保留解决方案以供后续任务使用。将修订版本 k 的工作区表示为 $W_k = (N_k, S_k)$, 其中 N_k 包含笔记, S_k 包含可执行技能。固定的编程模型在解决任务时查阅该工作区。两次修订之间, 独立且权重固定的改进器使用执行轨迹 $\mathcal{T}_k$ 和反馈 f_k 更新工作区:

$$W_{k+1} = I_\phi(W_k, \mathcal{T}_k, f_k). \quad (1)$$

模型和控制器权重保持固定。在仿真中, 私有工作进程可在各轮之间编辑工作区, 而冻结评估期间禁用写入。在 G1 上, 执行记录和可选的操作员反馈支持技能修订。每个修订后的程序在物理执行前仍须通过独立安全检查, 并获得操作员批准。

3.3. 学习可复用的视觉导航策略

ASENA-VLN 减少智能体在复杂路线上的重复推理。该可选工具根据语言和近期单目观测预测有界的机体坐标系轨迹。编程智能体决定何时调用该工具、检查进度, 并将其与其他工具结合使用。策略训练在同一编程接口内扩展了导航能力。

单一解码器, 不同输出表示. 本文为轻量级 Qwen3-VL-4B-Instruct [39] 扩展了轨迹与像素目标专用词元词表 (图 3)。模型接收最多八帧前视图和一条结构化 JSON 指令, 使该策略易于作为可调工具使用。输入指定简洁的导航命令, 还可选择包含期望终止姿态、锚定物体或运动约束。按新近程度加权的视觉词元在保留时间上下文的同时突出最新观测。共享自回归解码器预测最多八个累积机体坐标系航点, 每个航点表示为 (x, y, θ) 。各轴上的固定分箱将轨迹词元转换为显式运动目标, 无需单独连续动作头。像素目标词元通过引导模型在图像中定位预期目的地提供辅助监督, 而视觉问答保留模型原有的文本词表。由于像素目标无法直接执行, 还需额外的运动规划器, 或在仿真中使用最短路径跟随器, 因此本文所有评估均采用轨迹词元预测而非像素目标。

精选的原子导航数据. 可调导航策略既须遵循较长路线, 也须执行精确的局部命令。因此, 本文在 MP3D 和 HM3D 中构建九种原子导航任务, 涵盖移动指定距离、旋转至目标航向、从给定方向接近物体、与目标对齐、进入房间、穿过门口、沿走廊行进以及楼层间移动等行为。每项任务均依据测地距离、可见性和场景几

表 1: 独立单目 VLN-CE 策略在 R2R 和 RxR 的 val_unseen 划分上的结果。与已有工作相比, ASENSA-VLN-4B 达到当前最佳性能, 导航误差显著更低, nDTW 更高。

方法	R2R				RxR			
	NE ↓	OSR ↑	SR ↑	SPL ↑	NE ↓	nDTW ↑	SR ↑	SPL ↑
NaVid [3]	5.72	49.2	41.9	36.5	5.72	–	45.7	38.2
Uni-NaVid [26]	5.58	53.3	47.0	42.7	6.24	–	48.7	40.9
NaVILA [4]	5.22	62.5	54.0	49.0	6.77	58.8	49.3	44.0
StreamVLN [5]	4.98	64.2	56.9	51.9	6.22	61.9	52.9	46.0
NaVIDA [27]	4.32	69.5	61.4	54.7	5.23	67.0	57.4	49.6
JanusVLN [28]	4.78	65.2	60.5	56.8	6.06	62.1	56.2	47.5
DecoVLN [29]	5.01	63.5	56.3	50.5	5.73	63.5	54.2	46.3
ActiveVLN [30]	5.31	58.3	50.1	43.7	5.84	58.1	50.7	41.2
Qwen-VLA-Instruct [31]	5.10	69.0	57.5	51.2	5.80	57.1	59.6	47.8
InternVLA-N1 (S2, SPF) [32]	4.25	68.3	60.9	55.2	5.71	46.8	63.5	55.0
RynnBrain-Nav-8B (SPF) [33]	4.92	71.6	58.6	49.6	6.20	59.6	56.1	49.6
DualVLN [16]	4.05	70.7	64.3	58.5	4.58	70.0	61.4	51.8
InternVLA-N1 [32]	4.83	63.3	58.2	54.0	5.91	65.3	53.5	46.1
Qwen-RobotNav-4B [17]	4.22	73.6	66.9	60.5	4.15	68.6	71.3	61.5
ASENSA-VLN-4B	3.52	72.6	68.7	64.2	3.90	73.1	70.2	59.7

表 2: 物体目标导航: 成对报告 SR/SPL (% , 两者均为 ↑)。† 表示 HM3D v1 结果, 突出显示时不计入这些结果。

方法	HM3D	HM3D-OVON		
	v2 验证集	已见	合成	未见
VLFM	63.6/32.5	35.2/18.6	32.4/17.3	35.2/19.6
OpenFMNav†	52.5/24.1	–	–	–
SG-Nav	49.6/25.5	–	–	–
TriHelper†	56.5/25.3	–	–	–
WMNav†	58.1/31.2	–	–	–
CogNav†	72.5/26.2	–	–	–
Uni-NaVid†	73.7/37.1	41.3/21.1	43.9/21.8	39.5/19.8
DAGRL+OD	–	38.5/21.1	39.0/21.4	37.1/19.8
MTU3D	–	55.0/23.6	45.0/14.7	40.8/12.1
NavFoM	–	40.1/27.1	45.4/32.6	45.2/31.9
ABot-N0	–	55.3/32.1	55.4/33.2	54.0/30.5
ApexNav	76.2/38.0	–	–	–
Qwen-RobotNav-4B	75.6/30.6	57.7/24.4	60.1/25.1	53.1/20.9
Qwen-RobotNav-8B	71.2/33.0	56.1/28.5	57.8/28.8	51.2/24.0
ASENSA	80.8/38.1	60.5/33.5	65.8/37.9	64.9/38.4

何结构生成并验证, 确保每条指令描述可行且可测量的运动。每条指令均配有航点轨迹和辅助像素目标监督。这些数据直接教授任务程序可调用和组合的可复用细粒度操作。

本文使用这些原子任务、短时域 ObjectNav, 以及来自 R2R、RxR、ScaleVLN 和 SRDF [1; 2; 40; 41] 的视觉语言导航数据混合训练 ASENSA-VLN。为适配可调用策略接口, 本文使用 GPT-5.5 [42] 将这些数据集的原始指令重新描述为结构化 JSON 格式, 其中包含简洁导航命令, 并在适用时包含终止姿态、锚定物体和运动约束。训练数据还纳入多视图 VQA 数据 [43], 以增强模型的空间理解能力。训练从数据中抽取 430 万个样本, 轨迹预测与空间 VQA 的混合比例为 3:1。

4. 评估与应用

首先将 ASENSA-VLN 作为独立的指令跟随策略进行评估, 随后在目标物体导航任务上评估完整的 ASENSA 系统。接着研究模型权重固定时, 学习式导航策略与工作区中的保留经验如何影响编程智能体的性能。最后, 在仿真环境中评估具身问答, 并在 Unitree G1 机器人上展示该系统。

表 3: 在 R2R 和 RxR Agentic Split 上进行自进化前的导航结果。NE 的单位为米，OSR/SR/SPL 以百分比表示。调用数/回合为 100 个回合的平均工具调用次数。绿色百分比表示相对于同一后端不使用 VLN 时的降幅。

方法/编程智能体	VLN 工具	R2R Agentic Split					RxR Agentic Split			
		NE ↓	OSR ↑	SR ↑	SPL ↑	调用数/回合 ↓	NE ↓	SR ↑	SPL ↑	调用数/回合 ↓
已发表的零样本方法										
InstructNav [15]	—	6.89	47.0	31.0	24.0	—	—	—	—	—
Open-Nav [34]	—	6.70	23.0	19.0	16.1	—	—	—	—	—
CA-Nav [35]	—	7.58	48.0	25.3	10.8	—	10.4	19.0	6.0	—
GC-VLN [36]	—	7.30	41.8	33.6	16.3	—	8.80	33.8	13.8	—
Three-Step Nav [37]	—	5.87	39.0	34.0	29.1	—	9.21	22.0	16.1	—
Uni-LaViRA [18]	—	3.66	73.7	60.7	47.7	—	6.48	51.3	34.0	—
Minimal (Claude Sonnet 5, SDK) [38]	—	5.80	61.3	51.3	37.8	—	—	—	—	—
ASENA 策略与新工作区智能体										
ASENA-VLN-4B	—	3.16	82.0	78.0	70.7	—	4.54	62.0	52.9	—
ASENA (Claude Sonnet 5)	×	<u>5.23</u>	<u>70.0</u>	<u>61.0</u>	<u>34.2</u>	<u>83.32</u>	<u>5.80</u>	<u>50.0</u>	<u>31.9</u>	<u>103.83</u>
ASENA (Claude Sonnet 5)	✓	3.31	83.0	72.0	54.9	33.96 ↓59.2%	4.94	61.0	45.5	38.91 ↓62.5%
ASENA (GPT-6 Astra)	×	1.84	90.0	<u>87.0</u>	<u>72.4</u>	<u>26.37</u>	0.82	93.0	78.0	<u>40.80</u>
ASENA (GPT-6 Astra)	✓	<u>2.18</u>	<u>89.0</u>	88.0	72.8	14.00 ↓46.9%	<u>1.65</u>	<u>86.0</u>	<u>64.0</u>	19.98 ↓51.0%

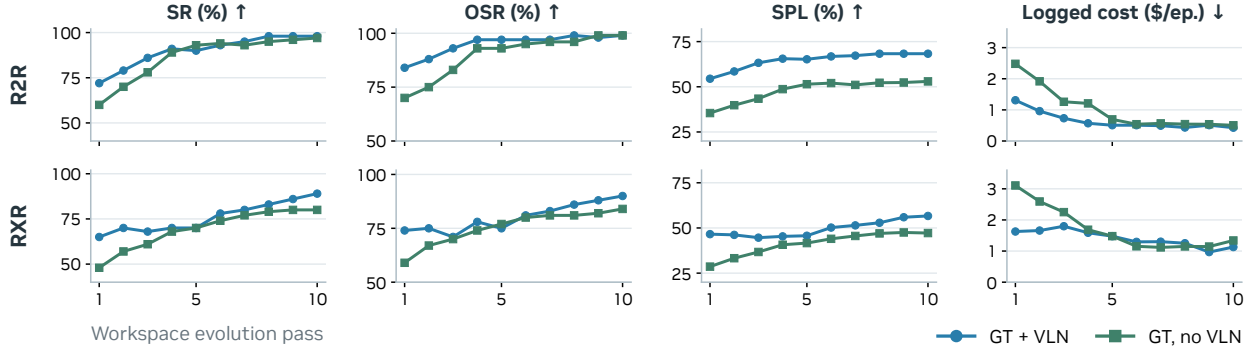


图 4: 导航工作区的自进化。随着进化轮次增加，成功相关指标（SR、OSR 和 SPL）均有所提升，而每轮成本总体下降。使用 VLN 工具能持续提升 SPL，使智能体更高效地到达目的地。

4.1. 独立 VLN 策略

表 1 在不采用编程智能体规划的情况下，分别使用 R2R 的较短指令以及 RxR 更长、更详细的路线独立评估 ASENA-VLN。

前进步长设为 0.25 m，旋转步长设为 15 度。导航误差（Navigation Error, NE）表示以米为单位的最终目标距离。预言成功率（Oracle Success Rate, OSR）衡量智能体是否进入目标区域，而成功率（Success Rate, SR）要求智能体在目标区域内停止。路径长度加权成功率（Success Weighted by Path Length, SPL）衡量路径效率，归一化动态时间规整（Normalized Dynamic Time Warping, nDTW）衡量轨迹与参考路线的相似度。这四项指标均以百分比表示。仿真器最短路径跟随器（Simulator Shortest-Path Follower, SPF）供像素目标方法使用。

4.2. 目标物体导航

表 2 将评估从沿给定路线行进扩展到搜索初始可能不可见的物体。由于 ASENA-VLN 主要针对原子导航行为而非全局探索进行训练，我们将其与独立的视觉语言模型（Vision-Language Model, VLM）[44] 结合，由后者规划搜索位置，并利用新观测到的证据更新计划。在 HM3D-v2 和 HM3D-OVON 上，这种全局搜索推理与学习式局部导航相结合的方案，相较于文献 [17; 45] 中已发表的方法取得了当前最佳结果。对于已见、同义和未见的 OVON 物体类别，其性能均保持稳定，表明该系统能够应对物体描述以及寻找物体所需路线的变化。

表 4: 具身问答。不同覆盖范围下的准确率 (%) 与平均归一化步数。

方法	HM-EQA		MT-HM3D	
	准确率 ↑	步数 ↓	准确率 ↑	步数 ↓
Explore-EQA [19]	58.4	.52	35.1	.64
3D-Mem [20]	50.4	.63	—	—
Fine-EQA [46]	56.0	.54	—	—
GraphEQA [21]	63.5	.20	45.6	.45
MemoryEQA (Qwen2VL-7B) [22]	—	—	51.2	.40
MemoryEQA (GPT-4o) [22]	63.4	.40	55.1	.41
FAST-EQA [23]	69.2	.65	50.5	.52
Qwen-RobotNav [17]	76.7	.15	54.4	.19
ASENA base (low effort)	70.9	.15	64.8	.10
ASENA hand-designed (efficient)	72.2	.11	63.8	.11
ASENA hand-designed (quality)	78.8	.22	67.1	.24
ASENA self-evolved 17 (low)	74.8	.13	64.9	.08
ASENA self-evolved 17 (high)	81.2	.25	65.5	.22

表 5: 冻结的 EQA 工作区：准确率 (%) 与归一化探索步数。阴影表示最终工作区。

轮次	推理强度	HM-EQA		MT-HM3D	
		准确率 ↑	步数 ↓	准确率 ↑	步数 ↓
基线	低	70.9	.153	64.8	.101
4	低	73.4	.136	66.5	.091
9	低	75.8	.147	65.5	.116
11	低	75.4	.152	65.3	.114
17	低	74.8	.127	64.9	.076
17	高	81.2	.253	65.5	.224

4.3. 智能体与策略的互补能力

表 3 研究编程智能体与 ASENA-VLN 各自对导航性能的贡献。我们比较已发表的零样本方法、独立的 ASENA-VLN，以及采用两种编程后端实例化的 ASENA。沿用 Uni-LaViRA 和 Open-Nav [18; 34] 的缩减集评估方式，R2R 与 RxR 的智能体划分 (*Agentic Splits*) 各包含 100 个任务。在每个基准的十个场景中，所有策略与智能体变体均使用相同的任务 ID，且每个编程智能体均从全新的工作区开始。Claude Sonnet 5 采用高推理强度运行，GPT-6 Astra 采用最高推理强度运行。四个 RxR 回合产生了非有限的 NE 或 SPL 值；计算 SR 时将其视为失败，NE 与 SPL 则基于其余 96 个回合计算。

强编程智能体无需学习式导航策略也能有效导航。 仅使用几何与编程工具，GPT-6 Astra 便取得当前最佳性能，并超过独立的 ASENA-VLN。在 R2R 上，其成功率也接近已报告的人类表现，即 90% SR。凭借强大的推理能力，该智能体无需依赖学习式导航策略，即可理解指令、构建路线、监控进度并从错误中恢复。这些结果表明，编程智能体本身就是强大的零样本导航系统。

学习式导航提升较弱编程后端的性能。 ASENA-VLN 对 Claude Sonnet 5 的帮助更为明显。加入该策略后，成功率与路径效率均有提升，R2R 和 RxR 上的 SR 均提高 11 个百分点。因此，专用策略能够弥补路线执行方面的不足，使较弱的编程后端完成更多任务。相比之下，GPT-6 Astra 即使不使用该策略也有很强的表现，因而从中获益较少；在 RxR 上，启用该策略甚至略微降低其成功率。因此，学习式导航的价值取决于编程后端，以及后端将策略预测融入规划的有效程度。

学习式导航减少重复的智能体交互。 单次 ASENA-VLN 调用可执行多个运动决策，从而减少重复的观测与动作请求。对于每种后端和基准，平均工具调用次数均有所下降（表 3）：GPT-6 Astra 在 R2R/RxR 上分别下降 46.9%/51.0%，Claude Sonnet 5 则分别下降 59.2%/62.5%。因此，即使成功率变化不大，该策略也能承担低层导航。这些调用次数的下降表明，学习式导航在提升较弱后端性能的同时，也可减少智能体交互，与强智能体的直接编程控制形成互补。

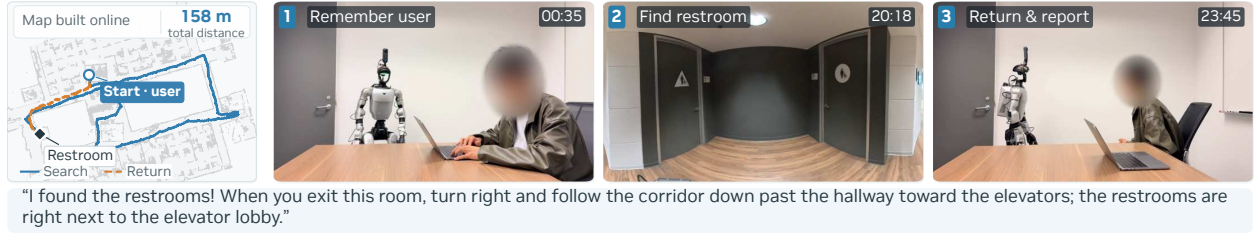
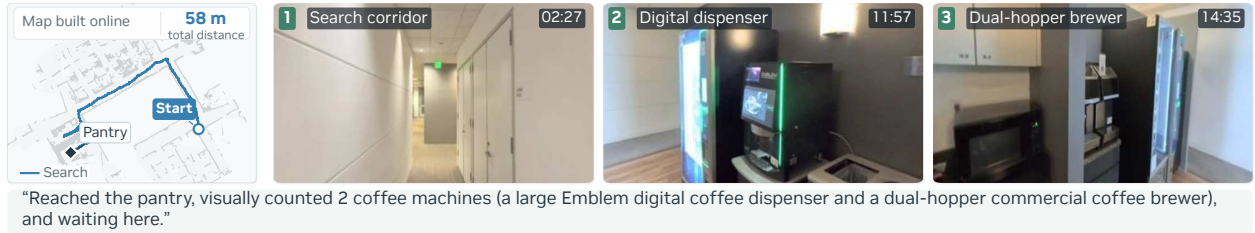
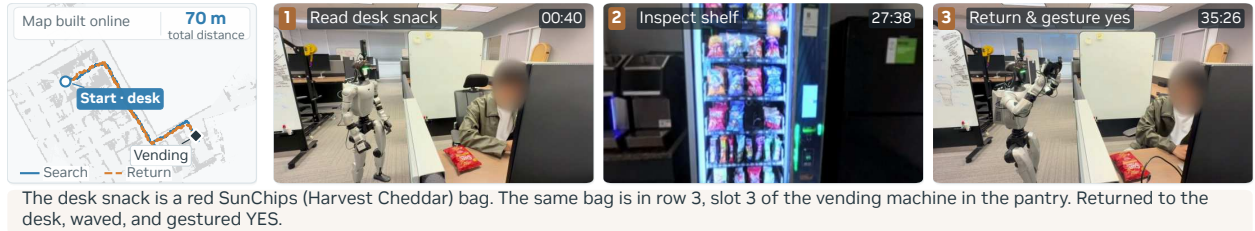
A “Could you find the restroom and then come back and tell me where it is?”**B “Go to the pantry and count the coffee machines. Wait there.”****C “I’m in a meeting, so don’t speak. Find the vending machine and check if it has the snack on my desk. Come back and answer with body language.”**

图 5: G1 机器人上的真实世界导航与交互。三项监督任务将在线地图和记录的轨迹与观测及响应对应起来。蓝色实线与橙色虚线分别表示搜索和返回。距离根据行驶里程计估算。提示与响应经精简后展示。

4.4. 保留经验提升重复任务的性能

通过重复提供相同的任务、场景、指令与初始位姿来评估工作区演化。在每一轮中，16 个采用高推理强度的 Claude Sonnet 5 工作智能体执行任务，并可在各自的私有工作区中保留程序、笔记与先前的动作序列。两轮之间，一个改进智能体利用这些智能体的轨迹与反馈更新规范工作区。在 R2R 和 RxR 上，分别开展使用和不使用 ASENA-VLN 的实验，并采用真实反馈，同时固定所有模型、策略与控制器权重。

如图 4 所示，四组实验的成功率均随轮次提升。在 R2R 上，使用与不使用 ASENA-VLN 的变体最终成功率相近；对于更长的 RxR 指令，该策略仍保持更明显的优势。在两个基准上，配备工具的工作区也保持更高的 SPL，表明即使最终成功率趋于一致，ASENA-VLN 仍能提高路径效率。随着工作区演化，记录的工作智能体与改进智能体成本也随之下降。这些结果表明，无需更新模型权重，保留经验即可提升重复任务的性能并降低成本，而学习式导航仍能带来效率优势。

4.5. 具身问答

具身问答 (Embodied Question Answering, EQA) 要求智能体探索环境、收集相关视觉证据，并以尽可能少的运动回答问题。使用 HM-EQA 和 MT-HM3D 发布的初始位姿评估 ASENA [19; 22]。按照 Explore-EQA 的瞬移协议，对于面积为 A 的场景，每个回合的名义步数预算为 $3\sqrt{A}$ 。报告答案准确率与归一化探索步数，后者定义为智能体消耗步数占该名义预算的比例。

手工设计的参考配置. EQA 智能体组合使用导航、视觉检查、物体搜索、地点记忆与答案生成工具。首先构建两个手工设计的配置作为参考点。质量配置优先收集足够证据以正确回答问题，且不接收任何明确要求减少探索的指令。高效配置则通过提示将探索步数视为有限资源，更有选择地收集证据。在 HM-EQA 上，为高

效智能体设置相当于名义预算 30% 的目标预算，同时为两种配置保留相同的仿真器硬上限。这些配置给出了人工设计的准确率与效率运行点，并非针对单一组件的受控消融实验。

工作区演化协议. 接着研究智能体能否通过修改工作区改善这一权衡。构建固定演化池，其中包含训练场景中的 398 个 HM-EQA 问题和 1,183 个 MT-HM3D 问题。每一轮中，16 个工作智能体从每个基准抽取 50 个问题，使用私有工作区执行任务并反思结果。独立的改进智能体汇总其轨迹与修改，以更新规范工作区；所有模型权重始终保持固定。未被工作智能体抽取的问题用于监控，其汇总结果也会提供给改进智能体。

工作区演化改善成本-准确率权衡. 冻结工作区后，第 17 轮在 HM-EQA 上取得最高准确率 (81.2%)，第 4 轮则在 MT-HM3D 上表现最佳 (66.5%) (表 5)。在低推理强度下，演化后的工作区以与手工设计的高效配置相近的步数取得了更高准确率 (表 4)。随着工作区获得新流程并完善现有指导，不同修订版本的改进幅度有所变化，形成不同的准确率与探索效率权衡。

学习式导航减少探索. 在 132 个 MT-HM3D 问题的配对消融实验中，ASENA-VLN 使归一化探索步数的中位数降低约 19%，而准确率几乎不变。与前述 R2R 和 RxR 结果一致，学习式导航通过减少重复的智能体交互来提高效率。

4.6. 真实世界应用

使用 Unitree G1 在三项监督任务上评估 ASENA (图 5)。每项任务中，机器人都需要在此前未见且没有预建地图的环境中搜索初始不可见的目标。机器人在线收集视觉与几何证据，并利用记忆和多模态交互完成用户请求。

在洗手间任务中，机器人定位电梯厅附近的门，基于在线地图推理出从用户位置出发的更短路线，并返回提供路线指引。在茶水间任务中，机器人找到房间，检查饮水机与咖啡冲煮机，并报告有两台咖啡机。零食比较任务要求机器人检查参考零食、找到自动售货机、判断是否有相同商品、返回用户身边，并生成全身姿态动作作为回答。这些任务展示了开放世界搜索、证据收集、空间推理、记忆与多模态交互。总体而言，它们说明在线程序修订如何突破预定义技能库，将导航扩展到自适应搜索与用户交互。

图 1 展示了精简后的记录代码，该代码将 LiDAR 几何信息转换为地图，并提出航路点 (橙色)。通过安全检查并获得操作员批准后，执行过程返回相机观测，供后续推理使用。生成的姿态动作在由 SONIC 执行前须通过在线仿真检查。

5. 讨论与结论

ASENA 统一了编程智能体编程、监督执行与持久化技能，在模型权重固定的情况下提升重复任务的性能。可选的 4B 单目 ASENA-VLN 策略基于精选的原子导航数据训练，以更少的智能体交互在 R2R/RxR 上取得领先的成功率。ASENA 也取得了领先的 EQA 准确率。在 G1 上，在线编程将导航扩展至搜索、检查与交互，其中包括在线合成并验证的全身响应动作。

真实世界任务需要 10–20 分钟，因此仍需加快推理与执行速度。未来工作将强化面向动态环境的安全防护，并将工作区演化扩展到学习式导航策略的更新。

致谢

感谢 Jarred Travers、Amanpreet Singh、Peter Pham、Jinhyung (David) Park、Xiangchen Tian 和 Tingwu Wang 对真实机器人基础设施提供的帮助。

参考文献

- [1] P. Anderson, Q. Wu, D. Teney *et al.*, “Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments,” in *CVPR*, 2018. 2, 5, 13
- [2] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, “Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding,” in *EMNLP*, 2020. 5, 13

-
- [3] J. Zhang, K. Wang, R. Xu *et al.*, “NaVid: Video-based VLM plans the next step for vision-and-language navigation,” in *RSS*, 2024. [2](#), [5](#)
 - [4] A.-C. Cheng, Y. Ji, Z. Yang *et al.*, “NaVILA: Legged robot vision-language-action model for navigation,” in *RSS*, 2025. [5](#)
 - [5] M. Wei, C. Wan, X. Yu *et al.*, “StreamVLN: Streaming vision-and-language navigation via slowfast context modeling,” in *ICRA*, 2026. [2](#), [5](#)
 - [6] J. Liang, W. Huang, F. Xia *et al.*, “Code as policies: Language model programs for embodied control,” in *ICRA*, 2023, pp. 9493–9500. [2](#)
 - [7] L. Fu, J. Yu, K. El-Refai *et al.*, “CaP-X: A framework for benchmarking and improving coding agents for robot manipulation,” *arXiv preprint arXiv:2603.22435*, 2026. [2](#)
 - [8] R. Lu, Y. Wu, E. Kou *et al.*, “ASPIRE: Agentic /skills discovery for robotics,” *arXiv preprint arXiv:2607.00272*, 2026. [2](#)
 - [9] G. Wang, Y. Xie, Y. Jiang *et al.*, “Voyager: An open-ended embodied agent with large language models,” *Transactions on Machine Learning Research*, 2024. [2](#)
 - [10] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, “Reflexion: Language agents with verbal reinforcement learning,” in *NeurIPS*, 2023. [2](#)
 - [11] W. Xiao, J. Xie, T. Zhang *et al.*, “ENPIRE: Agentic robot policy self-improvement in the real world,” *arXiv preprint arXiv:2606.19980*, 2026. [2](#)
 - [12] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, “Think global, act local: Dual-scale graph transformer for vision-and-language navigation,” in *CVPR*, 2022. [2](#)
 - [13] D. An, H. Wang, W. Wang *et al.*, “ETPNav: Evolving topological planning for vision-language navigation in continuous environments,” *TPAMI*, vol. 47, no. 7, pp. 5130–5145, 2025. [2](#)
 - [14] D. Shah, B. Osiński, B. Ichter, and S. Levine, “LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action,” in *CoRL*, 2023. [2](#)
 - [15] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, “InstructNav: Zero-shot system for generic instruction navigation in unexplored environment,” in *CoRL*, 2025. [2](#), [6](#)
 - [16] M. Wei, C. Wan, J. Peng *et al.*, “Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation,” in *ICLR*, 2026. [2](#), [5](#)
 - [17] Qwen Team, “Qwen-RobotNav technical report: A scalable navigation model designed for an agentic navigation system,” 2026, *arXiv:2606.18112*. [2](#), [5](#), [6](#), [7](#)
 - [18] H. Ding, S. Zhang, Z. Xu *et al.*, “Uni-LaViRA: Language-vision-robot actions translation for unified embodied navigation,” *arXiv preprint arXiv:2605.27582*, 2026. [2](#), [6](#), [7](#)
 - [19] A. Z. Ren, J. Clark, A. Dixit, M. Itkina, A. Majumdar, and D. Sadigh, “Explore until confident: Efficient exploration for embodied question answering,” in *RSS*, 2024. [2](#), [7](#), [8](#)
 - [20] Y. Yang, H. Yang, J. Zhou *et al.*, “3D-Mem: 3D scene memory for embodied exploration and reasoning,” in *CVPR*, 2025. [2](#), [7](#)
 - [21] S. Saxena, B. Buchanan, C. Paxton *et al.*, “GraphEQA: Using 3D semantic scene graphs for real-time embodied question answering,” in *CoRL*, 2025. [2](#), [7](#)
-

-
- [22] M. Zhai, Z. Gao, Y. Wu, and Y. Jia, “Memory-centric embodied question answering,” 2025, arXiv:2505.13948. 2, 7, 8
 - [23] H. Zhang, N. Savaliya, F. Siddiqui, and E. Sachdeva, “FAST-EQA: Efficient embodied question answering with global and local region relevancy,” in WACV, 2026. 2, 7
 - [24] Foxglove Developers, “MCAP: Serialization-agnostic log container file format,” <https://github.com/foxglove/mcap>, 2024, accessed: September 15, 2026. 3
 - [25] Z. Luo, Y. Yuan, T. Wang *et al.*, “SONIC: Supersizing motion tracking for natural humanoid whole-body control,” *Science Robotics*, vol. 11, no. 117, p. eaed4592, 2026. 3
 - [26] J. Zhang, K. Wang, S. Wang *et al.*, “Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks,” in RSS, 2025. 5
 - [27] W. Zhu, Z. Zhang, X. Wang *et al.*, “NaVIDA: Vision-language navigation with inverse dynamics augmentation,” *arXiv preprint arXiv:2601.18188*, 2026. 5
 - [28] S. Zeng, D. Qi, X. Chang *et al.*, “JanusVLN: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation,” in ICLR, 2026. 5
 - [29] Z. Xin, W. Li, Y. Jiang *et al.*, “DecoVLN: Decoupling observation, reasoning, and correction for vision-and-language navigation,” in CVPR, 2026. 5
 - [30] Z. Zhang, W. Zhu, H. Pan *et al.*, “ActiveVLN: Towards active exploration via multi-turn RL in vision-and-language navigation,” in ICRA, 2026. 5
 - [31] Q. Wang, M. Li, J. Guan *et al.*, “Qwen-VLA: Unifying vision-language-action modeling across tasks, environments, and robot embodiments,” *arXiv preprint arXiv:2605.30280*, 2026. 5
 - [32] Intern Robotics, “InternVLA-N1: An open dual-system vision-language navigation foundation model with learned latent plans,” 2025, technical report, Shanghai AI Laboratory. 5
 - [33] R. Dang, J. Guo, B. Hou *et al.*, “RynnBrain: Open embodied foundation models,” *arXiv preprint arXiv:2602.14979*, 2026. 5
 - [34] Y. Qiao, W. Lyu, H. Wang *et al.*, “Open-Nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source LLMs,” in ICRA, 2025. 6, 7
 - [35] K. Chen, D. An, Y. Huang *et al.*, “Constraint-aware zero-shot vision-language navigation in continuous environments,” *TPAMI*, vol. 47, no. 11, pp. 10 441–10 456, 2025. 6
 - [36] H. Yin, H. Wei, X. Xu, W. Guo, J. Zhou, and J. Lu, “GC-VLN: Instruction as graph constraints for training-free vision-and-language navigation,” in CoRL, 2025. 6
 - [37] W. Zheng, Y. Ge, and L. Itti, “Three-Step Nav: A hierarchical global-local planner for zero-shot vision-and-language navigation,” in AISTATS, 2026. 6
 - [38] J. Zhou, X. Zhao, G. Zhou *et al.*, “Embodied agents take control: Minimal-interface zero-shot agents rival industrial-scale policies in vision-and-language navigation,” *arXiv preprint arXiv:2607.26148*, 2026. 6
 - [39] S. Bai, Y. Cai, R. Chen *et al.*, “Qwen3-VL technical report,” 2025, arXiv:2511.21631. 4
 - [40] Z. Wang, J. Li, Y. Hong *et al.*, “Scaling data generation in vision-and-language navigation,” in ICCV, 2023. 5, 13
 - [41] Z. Wang, J. Li, Y. Hong *et al.*, “Bootstrapping language-guided navigation learning with self-refining data flywheel,” in ICLR, 2025. 5, 13
-

- [42] OpenAI, “GPT-5.5,” 2026. 5
- [43] Z. Cai, R. Wang, C. Gu *et al.*, “Scaling spatial intelligence with multimodal foundation models,” in *CVPR*, 2026. 5, 13
- [44] Google DeepMind, “Gemini 3.1 Flash-Lite model card,” Google DeepMind, Tech. Rep., 2026. 6
- [45] H. Zhu, Z. Li, Z. Liu *et al.*, “SysNav: Multi-level systematic cooperation enables real-world, cross-embodiment object navigation,” *arXiv preprint arXiv:2603.06914*, 2026. 6
- [46] K. Jiang, Y. Liu, W. Chen *et al.*, “Beyond the destination: A novel benchmark for exploration-aware embodied question answering,” in *ICCV*, 2025. 7
- [47] Y. Hong, C. Rodriguez, Q. Wu, and S. Gould, “Sub-instruction aware vision-and-language navigation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 3360–3376. 13
- [48] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans *et al.*, “Habitat-matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI,” in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 1, 2021. 13
- [49] Insta360, “X5: Webcam live streaming tutorial using OBS,” <https://onlinemanual.insta360.com/x5/en-us/camera/appuse/obs>, accessed September 9, 2026. 17
- [50] Livox, “MID-360 specifications,” <https://www.livoxtech.com/mid-360/specs>, accessed September 9, 2026. 17

A. 补充材料

A.1. 原子指令格式与训练数据

存储字段	类型	含义
goal_type	category; text	目标类别与文本。重写标题的记录可以仅存储目的地短语，而原始指令和生成的原子任务保留完整指令。类别包括物体、区域、自身、范围和点。
goal_value		
route_steps	有序列表	单独标注时使用的有序方向和地标提示。
constraints	列表	避障、穿越或相对位置要求。
end_pose	锚点列表	根据场景几何推导的目标、空间关系和距离，以物体、房间或机器人起始位置为参照。
quality	1-5	用于策略条件控制的质量标注或指定设置。R2R/RxR 评估取值为 5。

表 6: 存储的导航标注。策略接收目标文本、质量以及可选的终止位姿锚点（表 10）。基于坐标的点目标采用独立输入格式。

数据集	监督信号	加权数据池（百万）	环境
R2R [1]	轨迹, 像素, CoT	1.502	MP3D
R2R (增强)	轨迹, 像素	3.290	MP3D
FGR2R [47]	轨迹, 像素	0.704	MP3D
RxR [2]	轨迹, 像素, CoT	4.191	MP3D
RxR (增强)	轨迹, 像素	4.836	MP3D
ScaleVLN [40]	轨迹, 像素, CoT	3.120	HM3D
SRDF [41]	轨迹, 像素	3.916	HM3D
ObjectNav MP3D v3	轨迹, 像素	0.707	MP3D
ObjectNav HM3D v3 [48]	轨迹, 像素	2.888	HM3D
Nav-atoms (合成, MP3D)	轨迹, 像素	1.443	MP3D
Nav-atoms (合成, HM3D)	轨迹, 像素	1.825	HM3D
VLN-MME 及其他通用 VQA	VQA	0.964	多种
SenseNova-SI [43]	VQA	8.164	多种
轨迹数据池		28.422	
VQA 数据池		9.128	

表 7: ASENA-VLN 的训练数据池。数据池大小计入质量与后退动作过滤、替代视角、思维链（CoT）变体、重复的轨迹结束窗口以及采样权重。各项计数独立取整，且包含重复样本；其表示可用数据池规模，而非训练期间实际使用的样本数。采样细节见节 A.1。

训练采样。 训练使用 64 台设备，有效批量大小为 512。每台设备每步处理三个轨迹样本和一个视觉问答（Visual Question Answering, VQA）样本，并跨两步累积梯度。训练至第 8,400 步时共抽取 4,300,800 个样本（含重复样本），达到第 0.1178 轮。VQA 数据池将一轮定义为 71,309 次更新。由于轨迹数据池更大，每轮仅采样其中一部分。为强化停止行为，对每个符合条件的轨迹结束窗口额外复制五份。数据池还包含替代视角、思维链（Chain-of-Thought, CoT）变体和重复观测。

A.2. Jetson Thor 量化与运行时性能

为在 G1 的 Jetson AGX Thor 上运行该策略，使用 NVIDIA ModelOpt 进行训练后 FP8 量化，并用 TensorRT-Edge-LLM 构建推理引擎。这些测量采用训练至第 8,400 步的早期 ASENA-VLN 检查点，与表 1 中的基准策略不同。所有 FP8 变体均量化语言模型；标记为“BF16 视觉”的变体将含 4.15 亿参数的视觉编码器保持为 BF16。所有变体中的嵌入层和输出投影层均不量化。

仅用于导航的引擎需要预测运动词元，而非通用文本。因此，将输出词表从 152,596 项缩减至 768 项，同时保留全部 660 个轨迹词元。由此从每个解码步计算的输出投影层中移除 3.887 亿个参数。

变体	导出大小 (GiB)	视觉编码	预填充	解码 (32 步) 延迟 (ms)	总计	加速比
PyTorch BF16	9.01	167.3	363.4	2,275.5	2,806	1.00×
FP8, BF16 视觉编码器	5.63	73.3	128.7	847.4	1,050	2.67×
FP8, BF16 视觉编码器, 缩减词表	4.91	78.3	125.5	694.1	898	3.13×
FP8, 缩减词表	4.53	64.9	129.1	683.4	877	3.20×

表 8: Unitree G1 的 Jetson AGX Thor 上的策略推理。这些测量采用与基准策略不同的早期检查点。所有变体的批量大小均为 1，使用相同的八帧混合分辨率历史，并执行 32 步贪心解码。总计为三个组件之和；加速比根据未取整的总和计算。BF16 端到端实测延迟为 2,735 ms。导出大小指对应 ONNX 文件的大小。BF16 模型加载后占用 8.28 GiB 分配内存，此数值并非峰值运行内存。BF16 结果为五次预热运行的中位数：预填充时间等于首词元延迟减去视觉编码时间，解码则将余下 31 个词元的耗时缩放至 32 步。TensorRT-Edge-LLM 的视觉编码和预填充数值为 10 次运行均值，解码数值为五次运行的中位数。

变体	首航点一致率	完整轨迹精确一致率
FP8	90.23%	75.39%
FP8, BF16 视觉编码器	91.41%	82.81%

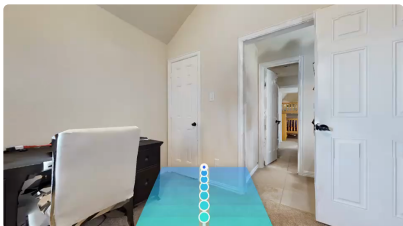
表 9: 早期量化模型及其 BF16 版本在 256 个留出输入上的预测一致率。首航点一致要求第一个预测的 (x, y, θ) 元组匹配；完整轨迹精确一致要求整条预测轨迹匹配。这些指标衡量输出一致性，而非导航成功率。

表 8 展示了各推理阶段的耗时。采用缩减词表的 FP8 后，各组件延迟之和从 2,806 ms 降至 877 ms；在相同输入设置下，相比 PyTorch BF16 加速 3.20×。解码仍是开销最大的阶段，耗时 683.4 ms，占 FP8 总时延的 78%。无需改变视觉编码或预填充即可降低该开销：在采用 BF16 视觉编码器的变体中，缩减词表使解码耗时从 847.4 降至 694.1 ms（下降 18.1%），其余两个阶段基本不变。

表 9 检验量化预测在留出输入上与 BF16 模型的吻合程度。视觉编码器保持 BF16 可提高首个航点和完整轨迹的一致率。这些检验衡量输出一致性，导航成功率仍需单独进行滚动评估。设备端检验还确认了导航词元 ID 有效，且八航点轨迹格式正确。

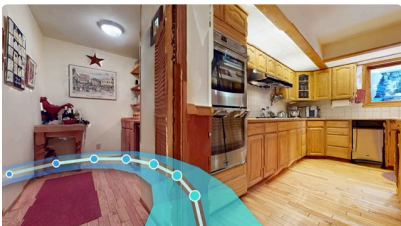
A.3. 原子导航数据集样本

度量运动



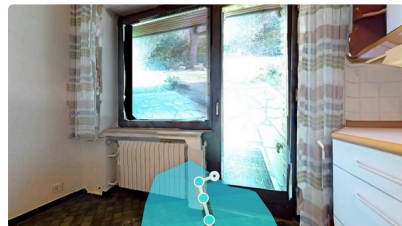
向前走 4 步后停下。

原地旋转



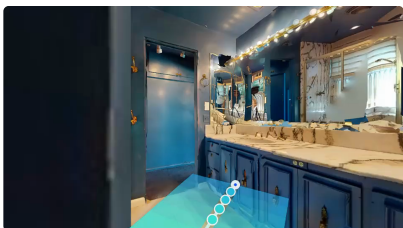
原地转身，面向正后方。

航向相对目标



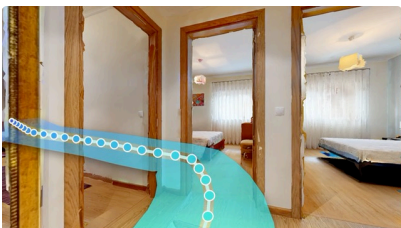
走到左侧的嵌入式洗碗机旁。

精细对齐



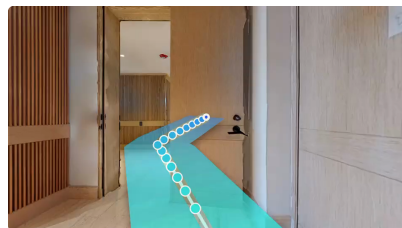
向右横移，靠近浴室排气扇。

房间目标



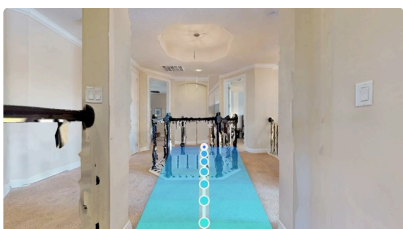
找到卧室后在其中停下。

穿门



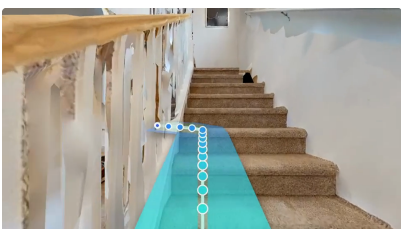
穿过门进入厨房。

沿走廊行进



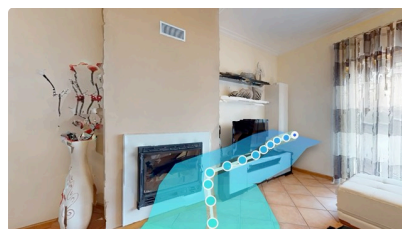
沿走廊前进，在尽头停下。

楼层变换



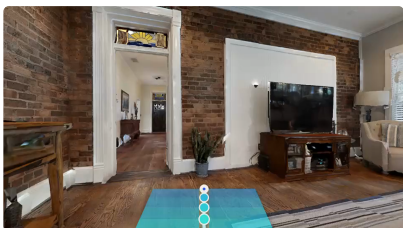
上楼。

航向相对目标



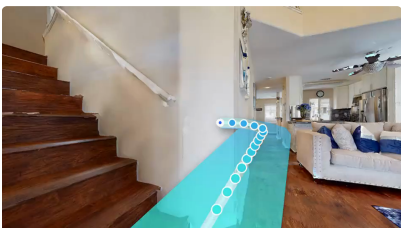
转身，走到身后白色壁柜旁。

度量运动



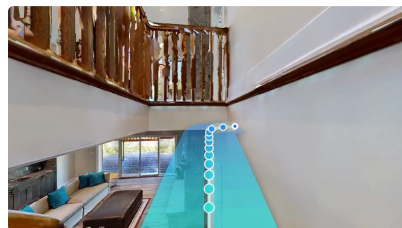
向前移动 0.5 米。

穿门



进入走廊。

楼层变换



下楼。

图 6: 原子导航训练样本。每个观测都与一条指令和叠加轨迹配对。叠加轨迹未采用度量相机投影。样本涵盖度量运动、旋转、相对目标、对齐、房间目标、门、走廊和楼层变换。

A.4. 导航指令与策略输入

表 10 展示了导航指令如何表示为策略输入。每个输入包含目标文本和质量设置。可选的终止位姿锚点描述机器人应在物体、房间或起始位置的何种相对位置停下。

表 10: 导航指令与策略输入。每个 JSON 输入包含目标文本和质量设置，还可包含指定机器人终止位置的终止位姿锚点。

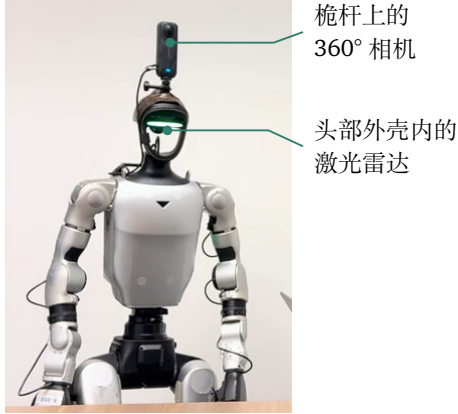
导航指令	策略输入 (JSON)
Route following Descend down the stairs. Turn left and stop in the room.	<pre>{ "goal": "Descend down the stairs. Turn left and stop in the room. ", "quality": 5 }</pre>
Metric motion Could you move forward 1 meters for me?	<pre>{ "goal": "Could you move forward 1 meters for me?", "quality": 5, "end_pose": [{ "target": "agent_start", "relation": "forward", "distance_m": 1.0 }] }</pre>
Relative object goal Walk over to the built-in dishwasher off to your left.	<pre>{ "goal": "Walk over to the built-in dishwasher off to your left.", "quality": 5, "end_pose": [{ "target": "built-in dishwasher", "relation": "near", "distance_m": 0.55 }] }</pre>
Room goal Stop there once you find a bedroom.	<pre>{ "goal": "Stop there once you find a bedroom.", "quality": 5, "end_pose": [{ "target": "bedroom", "relation": "inside", "distance_m": 0.0 }] }</pre>
Floor change Make your way downstairs to the hallway.	<pre>{ "goal": "Make your way downstairs to the hallway.", "quality": 5, "end_pose": [{ "target": "hallway", "relation": "inside", "distance_m": 0.0 }] }</pre>

方向、路线提示和约束保留在指令文本中。终止位姿距离以米表示，并四舍五入至小数点后两位。房间目标使用关系 `inside`，距离为 0.0。五个样本的质量均设为 5。策略同时接收该指令输入与视觉历史。

A.5. G1 硬件与导航设置

G1 配备安装在桅杆上的全景相机和倒置的 Livox Mid-360 激光雷达（图 7）。Jetson AGX Thor 运行本地传感器与运动服务。相机接口同时支持 Insta360 X5 和 Ricoh Theta X。表 11 列出传感器与状态输出，表 12 汇总导航设置和指令限制。

A 正视图



B 侧视图

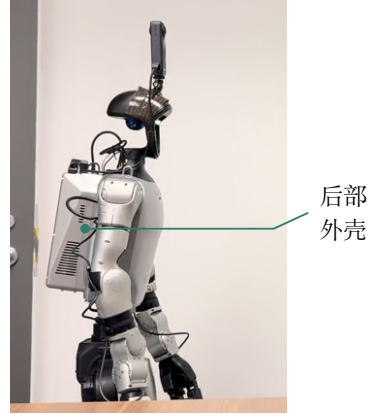


图 7: G1 传感器安装位置。正视图和侧视图展示传感器在机器人上的位置。

表 11: 传感器与状态接口。标称规格与配置的发布频率。激光雷达角度采用制造商定义的方向。

组件	原生输入/规格	配置输出
Insta360 X5	USB panorama: 2880×1440; 30 fps [49]	6 Hz JPEG; 180° 偏航校正
Livox Mid-360	200,000 点/s; 视场: 水平 360°, 垂直 -7° 至 52° [50]	SLAM 桥接: 10 Hz; 每批最多 20,000 点
机体状态	29 个关节的位置/速度; IMU、里程计和模式	20 Hz

表 12: 导航设置与安全保护机制。慢速行走控制器限制单条指令和路径段的速度、持续时间与幅度。

参数	数值	参数	数值
指令发布频率	10 Hz	路径前视距离	0.40 m
平移速度范围	0.12–0.35 m/s	制动提前时间	0.60 s
近目标速度 (≤ 0.5 m)	0.25 m/s	速度指令有效期	2.0 s
到达容差	0.10 m; 4°	活跃任务心跳超时	3.0 s
移动/转向指令限制	2.0 m; 180°	路径分段限制	6.0 m; 16 个点
移动任务超时	25 s	路径任务超时	45 s

慢速行走设置控制运动速度、停止容差、指令幅度和超时，可在运行时调整；系统还提供独立的行走步态。相机偏航校正设定全景图的水平方向。机器人头部和身体会部分遮挡倒置激光雷达的视场。