

NAV-GPT-3：在分层导航运行时中驾驭上下文

Gengze Zhou^{1,2†} Yicong Hong⁴ Jiazhao Zhang⁵ Xunyi Zhao¹ Jian Zhou¹
 Zixing Lei⁶ Zun Wang⁷ Chongyang Zhao⁸ Xionghui Chen⁵
 Stephen Gould^{2,3} Anton van den Hengel^{1,2} Qi Wu^{1,2†}

¹ 阿德莱德大学 AIML; ²Metacognition; ³ 澳大利亚国立大学;

⁴Roblox; ⁵ 北京大学; ⁶ 上海交通大学; ⁷ 北卡罗来纳大学教堂山分校; ⁸ 新南威尔士大学

† 通讯作者: {gengze.zhou, qi.wu01}@adelaide.edu.au

项目主页: <https://metacognitionai.github.io/NavGPT3/>

摘要

经长时程智能体强化学习训练的语言模型能够通过推理泛化知识、表达精确动作，并跨越多步持续追求目标，从而提升具身智能体理解与决策能力的上限。然而，物理交互仍由提供密集、低延迟控制的动作策略承担。本文提出 NavGPT-3，这是连接两个模型的执行框架 (Harness)，其上构建了类操作系统运行时：推理、行动和监控分别作为线程运行，各自拥有独立的上下文、工具与权限；运行时负责调度这些线程，并决定由哪个线程控制机器人运动，使机器人能够通过中断和线程切换应对突发的现实事件。在其下层，视觉-语言-动作 (Vision-Language-Action, VLA) 策略 NavGPT VLA 使用 19.28M 个样本训练，并通过编解码分配 (Codec Allocation) 按场景变化幅度分配视觉词元；仅 8B 模型便在 R2R-CE 上达到 74.51 的成功率 (Success Rate, SR)，并以 78.19 SR 在 RxR-CE 上领先。借助完整执行框架，NavGPT-3 在 R2R-CE 上取得当前最佳结果 (81.51 SR)，并首次使自主智能体达到人类水平：在 RxR-CE 上，其成功率 (90.43 vs. 90.4 SR) 和归一化动态时间规整路径保真度 (normalized Dynamic Time Warping, nDTW; 78.47 vs. 77.7) 均与人类跟随者相当，每个回合耗时 1 min 22 s，而人类约需 3 min。本文全面消融执行框架设计及两个模型间的交互，揭示工具和动作策略如何塑造从语言模型推理到物理控制的路径：当 NavGPT VLA 执行路线时，推理循环得以缩短，系统最短反应时间从每次语言模型决策 3–19 s 降至每个动作策略步 0.5–1 s (1–2 Hz)。这些结果表明，具身接口设计是连接前沿语言模型智能与底层物理控制的关键。我们将发布所有模型、代码和评测记录。

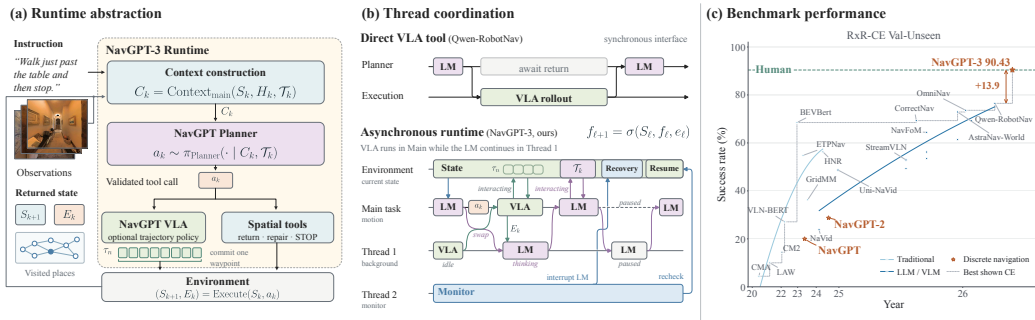


图 1: NavGPT-3 概述: (a) 包含规划器、VLA 和空间工具的运行时抽象; (b) 同步工具调用与异步线程协调; (c) 基准性能。NavGPT-3 在 RxR-CE 上达到人类水平。

1 引言

经长时程智能体强化学习训练的语言模型能够通过推理泛化知识，以工具调用和代码表达动作，并跨越多步持续追求目标 (OpenAI, 2026; Anthropic, 2026b;a; Qwen Team, 2026a)。在视觉-语言导航 (Vision-and-Language Navigation, VLN) 中，新一代语言模型逐步吸收了

导航系统过去以独立模块实现的更多能力,包括常识对齐、空间描述、进度估计和长时程工具使用。执行是其中的例外。执行遵循现实世界而非模型的时钟,控制频率必须达到逐步推理无法维持的水平。模型思考时,现实世界也不会暂停:推理期间仍会有观测输入,运动会持续越过触发它的决策,而危险可能要求系统在任何推理完成前停止。动作策略的规模远小于规划模型,指令遵循能力也更弱,因此在长轨迹展开中容易漂移,必须返回可供推理器核查的证据。现有系统通过分层结合推理与执行解决了部分问题:规划器向学习得到的控制器下达子目标 (Gao et al., 2025; Hu et al., 2026; Chiang et al., 2024; Huang et al., 2026b); 近期智能体还引入图记忆与回溯 (Liu et al., 2026), 或验证与恢复技能 (Wang et al., 2026b)。

要将这些能力连接至物理控制,需要构建完整的交互循环:为决策提供有依据的上下文,为行动提供空间工具,为记忆提供持久状态,并为路线修正提供执行证据。执行框架 (Harness) 通过这一循环塑造模型行为,定义模型接收的上下文、可调用的工具,以及跨调用保留的状态和反馈 (Zhong & Zhu, 2026; DeepSeek-AI, 2026)。真实机器人还带来协调需求:推理、执行和监控必须能够异步推进。执行框架之上的运行时将这些活动作为线程进行管理,并控制可移动机器人的线程。

本文提出 NavGPT-3, 将完整的导航执行框架与类操作系统运行时相结合。该执行框架将上下文管理、空间工具、持久路线状态和模型执行整合到统一的具身接口中。规划器 (Planner) 可以通过这些工具行动,也可以将导航指令的执行委派给 NavGPT VLA。返回的路线图像、执行反馈、观测地图和带编号的已访问位置,使生成的轨迹展开可供检查和修订:规划器能够返回已访问位置、局部修复路线、再次委派,并明确判断任务何时完成。本文通过递归自我改进 (Recursive Self-Improvement, RSI) 开发这些执行框架接口,以自动化研究迭代优化工具定义和视觉提示 (章节 4.1)。运行时基于这些接口调度线程并控制运动权限:路线复核可与执行重叠,监控也可在无需等待推理的情况下中断运动 (图 2)。工具能力和视觉上下文研究在考察执行框架设计的同时,也分析规划器-VLA 交互 (章节 4.2 和 4.3)。在执行框架下层, NavGPT VLA 根据指令和视觉历史预测八个航点。由于视觉上下文容量有限,本文采用视频编码中的编解码分配:每个视角的首次观测作为完整参考 (I 帧), 后续观测 (P 帧) 则按其变化幅度分配视觉词元 (Tang et al., 2026; An et al., 2026)。

在各项基准上,仅 8B NavGPT VLA 便以 78.19 SR 领先 RxR-CE 上已发表的方法;在运行时支持下, NavGPT-3 在 R2R-CE 上取得当前最佳结果 (81.51 SR), 并在 RxR-CE 上以成功率 (90.43 vs. 90.4 SR) 和路径保真度 (78.47 vs. 77.7 nDTW) 同时达到人类跟随者水平。同一运行时还可迁移至物体目标导航和具身问答 (表 6 至 8)。对执行框架工具和 VLA 交互的消融表明,该接口决定了从推理到控制的路径:拥有良好工具时,强语言模型可独立实现出色导航,但每 3-19s 才能决策一次;由 VLA 执行路线则可将系统最短反应时间降至每步 0.5-1s (1-2Hz; 表 3)。

2 相关工作

基于大语言模型的自主视觉-语言导航。VLN 要求智能体在未见环境中遵循自然语言指令,早期研究基于离散导航图 (Anderson et al., 2018; Ku et al., 2020), 随后扩展至连续环境 (Krantz et al., 2020)。NavGPT 开创了一条研究路线,探索大语言模型 (Large Language Model, LLM) 可独立承担多大程度的导航任务 (Zhou et al., 2024b): 给定文本形式的观测,语言模型对指令进度进行推理,并以零样本方式决定每一步移动,无需使用任何动作数据训练。后续导航方法进一步引入地图、讨论和开放词汇定位 (Chen et al., 2024; Pan et al., 2023; Long et al., 2023; Qiao et al., 2025); 近期智能体则允许模型通过代码直接操控具身载体 (Zhou et al., 2026b)。另一条并行路线关注物理执行,将视觉-语言模型 (Vision-Language Model, VLM) 微调为 VLA 策略,在保留 VLM 泛化能力的同时提供密集、低延迟控制 (Zhang et al., 2024; 2025b; Cheng et al., 2025; Wei et al., 2025b;a; InternVLA-N1 Team, 2025; Zhang et al., 2025a; AMAP CV Lab, 2026; Qwen Team, 2026b; Mistral AI, 2026)。NavGPT-2 同时推进两条路线:既强调语言模型推理的作用,又训练行动更快、更准确的策略,并在共享表征之上解耦语言头与动作头 (Zhou et al., 2024a)。然而,这种训练无法使推理与动作完全对齐。NavGPT-3 在系统层面结合二者,通过执行框架连接推理模型与训练后的 VLA 策略,并在其上构建类操作系统运行时。

学习策略之上的语言模型。分层系统在导航 (Chiang et al., 2024; Huang et al., 2026b; Liu et al., 2026) 和操作 (Gao et al., 2025; Hu et al., 2026; Guo et al., 2026; Li et al., 2026; Yang et al., 2026; Wang et al., 2026b) 任务中将规划模型与学习得到的控制器配对;执行框架工程则将上下文、工具和状态视为设计单元 (Zhong & Zhu, 2026; DeepSeek-AI, 2026)。这些系统已涵盖图记忆、回溯、验证和恢复。最接近的系统 NavMCP 跨导航基础模型的多次调

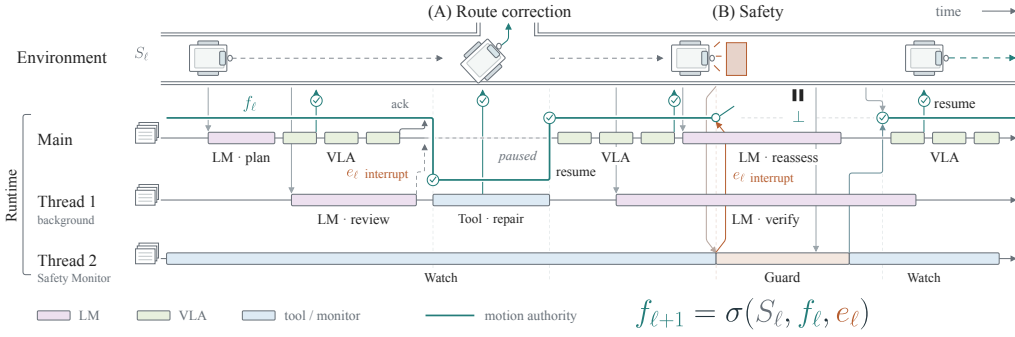


图 2: NavGPT-3 运行时中的前台控制模式示例。细条表示并发活动，青绿色路径表示运动权限。(A) 路线复核中断暂停主执行流程以进行修正，随后恢复执行。(B) 安全中断在推理继续进行时撤销运动权限；恢复执行需要最新状态和运行时授权。

用保留轨迹证据，以回答问题 (Lei et al., 2026)。NavGPT-3 则利用返回的证据修订指令遵循路线：VLA 完成受委派的轨迹展开后，规划器可检查路线、返回任意已访问位置、局部修复并判断任务是否完成。

导航策略中的视觉上下文。导航策略必须在有限视觉预算内容纳长时程、多视角观测历史。现有策略通过合并或剪枝视觉词元压缩历史 (Zhang et al., 2025b; Wei et al., 2025b)，或沿时间与视角分配固定预算 (Qwen Team, 2026b)。NavGPT VLA 则借鉴编解码器对齐的视频编码器，按场景变化幅度分配词元 (Tang et al., 2026; An et al., 2026)。

3 方法：执行框架、运行时与 VLA 协同设计

NAVGP-3 在推理模型与导航策略之间协同设计了完整执行框架，涵盖上下文与状态管理、空间工具、经验证的执行、结构化返回、路线修复，以及显式的任务终止决策。执行框架之上的运行时负责控制其线程。第一部分介绍这两层结构，第二部分介绍其下层 VLA。

3.1 第一部分：NavGPT-3 执行框架与运行时

3.1.1 执行框架与运行时抽象

执行框架通过上下文、状态和工具接口连接规划器、VLA 与环境（模拟器或机器人），从而塑造模型行为。它构建模型可见的观测与历史，维护空间引用和标注，检查工具请求，执行获准调用，并为下一次决策返回证据。规划器决定下一步行动及任务何时完成；VLA 则可选择性地提供高频执行。执行框架之上的类操作系统运行时负责控制逻辑线程、线程生命周期与权限，以及运动权限。图 2 通过路线复核和保护性监控器展示了这一协调过程。

3.1.2 上下文、空间工具与执行返回

执行框架将观测和交互历史转化为决策上下文，并提供空间工具，使该上下文可转化为行动。由于每次感知运动更新都调用规划器并不现实，执行框架还提供 VLA，通过更高频率的感知-动作循环完成长时程导航。VLA 调用并非必需：规划器自行决定使用 VLA，或通过可用工具独立作出全部导航决策。在第 k 次调用时，规划器选择

$$a_k \sim \pi_{\text{Planner}}(\cdot \mid C_k, \mathcal{T}_k), \quad (1)$$

其中， C_k 为执行框架提供的上下文， $\mathcal{T}_k$ 为允许使用的工具集， a_k 为包含参数的调用。执行框架检查选定调用，并在运行时允许时执行；在这些限制内，导航策略由规划器决定。

该接口将操作分为观测、指令委派、空间修订和语义完成四类。观测返回当前视图和已观测几何信息；委派返回路线关键帧与执行状态；修订将已访问位置引用或局部修正转化为运动；标注与终止保留语义，并显式标明任务完成。这些工具及其视觉返回结果通过 RSI 反复迭代（章节 4.1）。共享接口使长时程执行可检查、可修订：VLA 的停止仅结束一项操作，最终任务终止权限仍由规划器持有（章节 A）。

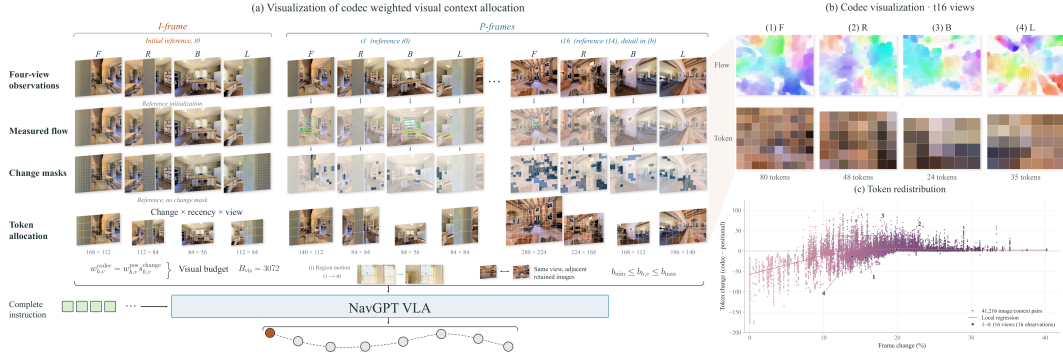


图 3: 基于编解码加权的视觉上下文分配。(a) 对于包含四视图的历史, 变化、新近程度和视图共同决定整幅图像分辨率, 其中初始帧为 I 帧, 后续帧为 P 帧。(b) t_{16} 时刻四个视图的光流诊断结果及对应词元网格。(c) 在 $B_{vis} = 3072$ 下, 来自 13 条记录历史的 41,216 组重放图像/上下文对中, 相较位置加权的分配变化与 RGB 变化 $100\delta_{h,v}$ 的关系。曲线表示局部回归; 标注数值与 (b) 一致。

3.1.3 执行协调与运动权限

各线程在权限约束下以各自节奏进行推理并接收观测, 因此图 2 不同行上的活动在时间上相互重叠。运行时控制哪个线程持有运动权限, 即发出任务运动指令的许可 (蓝绿色路径)。令 $f_\ell \in \mathcal{I}_\ell \cup \{\perp\}$ 表示事件索引 ℓ 处的前台线程, 其中 $\mathcal{I}_\ell$ 为线程集合, $\perp$ 表示无持有者。控制权转移遵循

$$f_{\ell+1} = \sigma(S_\ell, f_\ell, e_\ell), \quad (2)$$

其中, S_ℓ 为共享系统状态, 包括环境 (顶行); e_ℓ 为中断等传入事件 (橙色); σ 为运行时控制权转移规则, 它根据预设事件优先级保留、转移或撤销权限。保护性事件具有最高优先级; 回合限制优先于常规复核, 过期事件不得恢复已失效的操作 (章节 B.6)。交接会保留线程上下文, 且仅在运行时授权和执行确认 (勾号) 后生效。事件顺序 ℓ 不要求所有线程使用统一推理时钟。

对于线程 j 提出的任务运动操作 a , 其前台资格为

$$\text{eligible}_\ell(j, a) = \mathbf{1}[j = f_\ell \wedge a \in \mathcal{P}_j], \quad (3)$$

其中, $\mathcal{P}_j$ 为线程允许执行的操作集。具备资格是必要而非充分条件: 执行还会检查当前状态和权限 (章节 B.7)。运动权限并不赋予独占计算或感知权, 在线程内调用规划器或 VLA 也无需切换线程。该机制同时支持并发推进与同步。例如, 在路线修正中 (图 2A), VLA 持有权限, 后台线程则复核其进度; 复核触发 e_ℓ , VLA 确认后, 权限转交给工具以修复路线, 主线程同时暂停; 随后权限返回, VLA 恢复执行。在安全场景 (B) 中, 监控器守卫无需等待 LLM 推理或常规确认便撤销权限, 使 $f_{\ell+1} = \perp$, 而推理继续进行; 只有获得新状态和新授权后, 主线程才会恢复。

3.2 第二部分: NavGPT VLA

本文将 NavGPT VLA 训练为通用导航基础模型: 以同一视觉-语言-动作 (Vision-Language-Action, VLA) 骨干网络学习多种导航任务、相机配置和具身载体。4B 与 8B 模型使用指令遵循、点目标与物体目标导航、目标跟踪、户外驾驶, 以及具身问答和视觉定位数据进行训练。在 NavGPT-3 中, 该策略负责执行规划器下达的导航指令。

3.2.1 轨迹策略与视觉上下文分配

NavGPT VLA 以 Qwen3-VL-Instruct 为起点, 并添加多层感知器 (Multilayer Perceptron, MLP) 动作头。在 VLA 的第 n 步, 它根据给定指令和最多 16 条保留观测预测有序轨迹 $\tau_n = ((x_{n,r}, y_{n,r}, \theta_{n,r}))_{r=1}^8$; 每条观测均含前、右、后、左四个视图。航点索引 r 遍历当前智能体坐标系中的八个累积目标: x 指向前方, y 指向左侧 (二者单位均为米), θ 为以弧度表示的相对朝向。4B 与 8B 版本共享该输出格式; 每次预测后, 执行仅落实第一个航点。训练结合航点回归与下一词元监督, 详见章节 3.2.3。

在视觉词元预算 B_{vis} 下, VLA 最多接收 N_{hist} 条保留观测, 每条包含 N_{view} 个视图 (图 3a 中的 F、R、B、L); 图像 (h, v) 即观测 h 的视图 v , 获得 $b_{h,v}$ 个词元。本文重新实现的 Qwen-RobotNav 任务自适应分配方法 (Qwen Team, 2026b) 仅使用位置信息: 权重 $w_{h,v}^{\text{pos}}$ 反映时间新近性与视图方向, 分配器将权重转换为满足 $b_{\min} \leq b_{h,v} \leq b_{\max}$ 的整数计数 (章节 C)。位置记录图像拍摄的时间与地点, 却无法表明图像是否包含新内容。

受 I/P 帧视频编码启发, 编解码分配额外引入变化分数 (图 3a)。对于每个视图, 我们比较连续保留图像的 64×64 RGB 缩略图: 首幅图像 (t_0) 为得分取满值的 I 帧, 后续 P 帧则依据其平均绝对像素差 $\delta_{h,v}$ 评分:

$$\delta_{h,v} = \frac{\|\text{thumb}(I_{h,v}) - \text{thumb}(I_{h-1,v})\|_1}{255 \cdot 3 \cdot 64^2}, \quad s_{h,v}^{\text{change}} = \begin{cases} 1, & \text{I-frame,} \\ \text{clip}(5\delta_{h,v}, 0.05, 1), & \text{P-frame,} \end{cases} \quad (4)$$

因此, 编解码权重 $w_{h,v}^{\text{codec}} = w_{h,v}^{\text{pos}} s_{h,v}^{\text{change}}$ 综合了变化程度、时间新近性和视图方向 (图 3a 底行); 分配器与词元总量保持不变, 恒定分数则可还原位置规则。分数能否发挥作用取决于预算容量比 $\rho_{\text{vis}} = B_{\text{vis}} / (N_{\text{hist}} N_{\text{view}} b_{\max})$: 当 $\rho_{\text{vis}} \geq 1$ 时, 每幅图像均可达到 $b_{\max}$, 两种规则等价; 小于 1 时, 词元会向发生变化的视图转移。训练时随机采样 $B_{\text{vis}} \in [1024, 4096]$ 及分配上下限, 使非平凡历史进入该有效区间 (表 13)。

在 t_{16} 时刻 (图 3b), F、R、B、L 分别获得 80、48、24、35 个词元; 光流场和变化掩码仅用于可视化帧间差异, 实际分配使用公式 (4) 中的缩略图分数。在记录的多条历史中 (图 3c), 编解码分配会从低变化图像中移出词元, 并将其分配给发生变化的图像。

3.2.2 训练数据分类体系与混合

基础动作模型从预训练骨干继承广泛的视觉与语言知识, 但其行动可靠性取决于动作数据所覆盖的情形范围; 单纯增加某一任务的示范, 很难覆盖部署策略可能遭遇的边缘情况。因此, NavGPT VLA 语料库围绕覆盖分类体系构建, 而非追求某一基准的数据规模。该体系包含四个维度: 任务要求的能力、执行任务的具身载体与观测接口、任务发生的条件, 以及任务预设的视觉语言理解能力。前三个维度通过动作监督, 第四个维度通过语言监督。无论来源如何, 每个导航数据源均转换为智能体坐标系下相同的八航点目标 (章节 3.2.3), 因此任一维度上的覆盖扩展都会训练同一共享动作空间, 而非相互独立的任务特定策略 (表 1 和图 4)。

能力。 本文覆盖四类对策略提出逐步扩大要求的能力。指令遵循将引导信息落实到运动中, 其形式既包括逐步语言, 也包括仅给出方向或相对坐标的紧凑点目标。由物体目标导航实现的主动探索完全移除了路线: 智能体必须搜索部分可观测场景, 并推断目标可能所在的位置。由目标跟踪实现的动态交互进一步加入运动中的其他智能体; 策略既要随过程预测其运动, 又要在干扰项中保持对目标身份的识别。跨具身导航主要通过 OpenScene (OpenScene Contributors, 2023) 和 nuScenes (Caesar et al., 2020) 的户外驾驶数据实现, 将相同空间规划能力迁移至交通环境中的车辆, 同时面对更远的感知范围、不同的运动尺度和更严格的安全约束。沿这一序列, 环境不确定性、对历史的依赖和具身载体多样性总体递增。这些能力并非彼此割裂: 更困难的设置会复用较简单的技能, 同时引入自身的感知与控制要求。在这些重叠数据上联合训练, 旨在获得可跨任务族迁移的空间规划能力, 而非分别拟合各项任务。

具身载体与观测接口。 服务多个平台的策略必须能够依据各平台搭载的任意相机行动。覆盖这一维度最经济的方法是重新渲染现有回合, 而非采集新回合。因此, 每个 R2R-CE 和 RxR-CE 训练回合 (Krantz et al., 2020; Ku et al., 2020) 均同时呈现为由前、右、后、左相机构成的四视图全景和前向相机序列, 二者使用相同航点目标, 使策略能够从环视输入和仅前向输入中学习同一行为。VLNVerse (Lin et al., 2025) 的前视轨迹、RoboCasa (Nasiriany et al., 2024) 等其他平台的机器人导航示范, 以及驾驶数据 (OpenScene Contributors, 2023; Caesar et al., 2020), 将同一接口扩展至更多模拟器、机体和相机配置。在模拟环境中, 物体目标回合从随机位姿开始, 相机高度和视场也随机变化, 因此单一数据源便涵盖多种传感器布置。

条件与边缘情况。 在单项任务中, 失败集中于训练分布稀疏之处: 陌生的措辞、外观和场景, 偏离专家路径的状态, 以及少见动作。本文尽可能保持监督不变, 同时扩展上述各方面。对于语言和外观, 我们使用 Gemini 3.1 Pro (Google DeepMind, 2026) 润色 R2R-CE 和

RxR-CE 回合中的指令；对于 VLN 和跟踪数据的部分子集，则使用 Qwen-Image-Edit (Wu et al., 2025) 润色 Habitat 渲染的观测，使其更接近真实相机图像。这些变体改变策略读取或看到的内容，但不改变其应遵循的路线。对于场景，SRDF (Wang et al., 2025) 提供 HM3D 环境 (Wang et al., 2023) 中的指令-轨迹对，所涵盖场景远多于 MP3D 中的 R2R 训练划分。本文以两种方式重建这些数据。首先，原始指令通常并不准确，因此使用 Gemini 3.1 Pro (Google DeepMind, 2026) 重新生成全部指令；该模型依据轨迹视频、通往同一目标的最短路径视频和原始指令进行改写。其次，为扩大状态覆盖，我们保留 DAgger 轨迹展开及最短路径路线，但改变其监督方式。原始数据将每个已访问状态标注为轨迹展开自身的下一步，这一伪动作未必使智能体接近目标；本文改为将每个已访问状态标注为从当前位置通往目标的最短路径方向。重新标注的轨迹展开由此覆盖最短路径示范从未访问的偏离路径状态，并监督智能体从各状态恢复。对于动作，几何感知重采样可减轻原始轨迹中的前进动作偏置，提高转向和停止的占比 (章节 3.2.3)。点目标数据还改变智能体的移动速度及周围空间的开阔程度。标准回合每步前进约 0.25 m，低速回合则约为 0.1 m，因此相同位移会跨越两倍以上观测，策略必须根据小得多的帧间变化判断进度。开放空间回合将目标设在场景开阔区域内的整数度量偏移处，例如前方 4 m、左侧 1 m；此时墙壁与走廊不再约束路线，策略必须仅凭坐标到达目标。

视觉语言理解。 动作监督会教授移动方向，却不会教授场景中有什么，或指名物体出现在图像何处。仅用下一词元预测训练的语言监督数据覆盖了这一维度，其中视觉定位数据最为重要。这些数据取自 LocateAnything-Data (Wang et al., 2026a)，该语料库在 12M 幅图像上包含 138M 条语言查询和 785M 个边界框。本文选取的 19 个子集贡献 2.31M 条有效记录，占完整混合数据的 12.0%，规模仅次于四视图指令遵循数据。它们涵盖物体检测 (占这些记录的 40.0%，包括第一人称、部件级、人物和驾驶场景)、指代表达定位 (18.2%)、点与可供性定位 (13.7%)、图形用户界面 (Graphical User Interface, GUI) 元素定位 (12.4%)、文本定位 (11.0%) 和文档布局 (4.6%)。每个样本都将语言查询与其指代的边界框或点配对，从而训练骨干网络在自然、第一人称、机器人、驾驶和屏幕图像中，将词语映射至精确图像位置。每当指令提及地标或目标物体时，导航同样需要这种能力；仅靠航点监督无法获得该能力，因为航点监督将语言与运动而非明确图像位置相关联。其余语言监督数据包括 LLaVA-OneVision-2 (An et al., 2026) 的空间推理数据、导航回合上的问答与对话，以及通用视觉语言指令数据；后者使骨干在学习行动的同时保留更广泛的能力。

混合数据构成。 完整混合数据来自 83 个数据源，共分为 11 个数据族 (表 1)，每个训练轮次包含约 19.28M 条有效记录。如图 4 所示，其中 65.6% 用于监督动作回归，34.4% 仅监督下一词元预测。空间推理和视觉定位数据合计贡献约 18% 的记录，通用视觉语言数据占 8.5%。

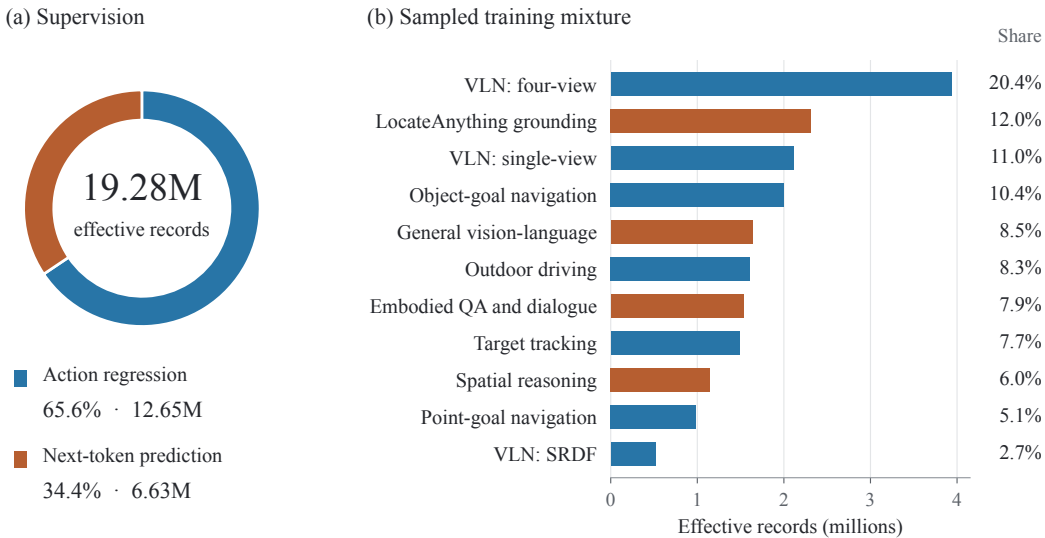


图 4: NavGPT VLA 训练数据混合。按数据类别统计的监督划分与有效记录数。

采样与计数。 多数数据源均完整使用，若干大型数据源则经过子采样（表 1）：SRDF 导航数据源保留 15%，基于 SRDF 的对话数据源保留 10%，两个最大的通用视觉语言数据源保留 30%，最大的空间推理与 LocateAnything 数据源则保留 5% 至 60%。对于包含 n_d 条记录、采样比例 $s_d \in (0, 1]$ 、重复次数 r_d 的数据源 d ，有效规模 $\tilde{N}_d = n_d s_d r_d$ 表示该数据源每轮贡献的样本数；某数据族的占比为其有效规模之和除以 $\sum_d \tilde{N}_d$ 。有效记录统计训练样本数，而非独立轨迹、场景或数据集数量：四视图与单视图 R2R/RxR 数据渲染相同回合，指令润色和观测润色变体复用原有路线，SRDF 也建立在 HM3D 环境之上。

表 1: **训练数据混合**。规模表示每个训练轮次的有效记录数，单位为百万。

类别	数据源数	采样比例	规模（百万）
动作回归			
VLN: 四视图	7	100%	3.938
VLN: 单视图	5	100%	2.113
VLN: SRDF	2	15%	0.516
点目标导航	5	100%	0.984
目标物体导航	1	100%	2.000
目标跟踪	6	100%	1.486
户外驾驶	5	100%	1.608
下一词元预测			
具身问答与对话	12	10–100%	1.532
通用视觉语言	13	30–100%	1.640
空间推理	8	25–100%	1.149
LocateAnything 视觉定位	19	5–100%	2.311
总计	83		≈ 19.28

3.2.3 数据处理与训练

导航数据源被转换为统一的多模态格式，其中包含一条指令、一个有序图像序列和一个八航点目标。四视图 Habitat 观测按前/右/后/左顺序渲染，单视图数据源则保留前向相机；首条观测与当前观测始终保留，中间历史最多采样 16 步。Habitat 离散动作被模拟为前进 0.25 m、转向 15 度或填充停止，并转换为智能体局部坐标系中的累积坐标，其中 (x, y, θ) 分别按 $(1.0, 0.433, 2.094)$ 归一化。几何感知重采样减轻了原始前进动作偏置，并增加转向和停止。具身导航问答、通用视觉语言、空间推理和视觉定位样本采用相同的图像-对话格式，但不含轨迹目标，仅训练下一词元目标。

两种模型规模均由 Qwen3-VL-Instruct 初始化，并在完整混合数据上训练一个轮次，使用相同的全局批大小、训练计划、编解码分配和动作输出；模型与动作头学习率分别为 5×10^{-6} 和 10^{-3} （表 2）。章节 4.6 和 4.7 中的消融实验还在该混合数据的较小累积范围上训练了额外模型。

表 2: **对齐的 VLA 训练配置**。BS: 全局批量大小；AH LR: 动作头学习率。

骨干模型	GPU 数	BS	步数	序列长度	LR	AH LR
Qwen3-VL-4B	32	256	$\sim 75.3k$	8,192	5×10^{-6}	10^{-3}
Qwen3-VL-8B	32	256	$\sim 75.3k$	8,192	5×10^{-6}	10^{-3}

监督与掩码。 训练采用上述坐标归一化与训练配置，将轨迹回归同视觉语言联合训练相结合（Qwen Team, 2026b）。MLP 动作头预测包含八个 (x, y, θ) 航点的 24 维向量。对于包含导航样本 $\mathcal{B}_{\text{nav}}$ 的小批量，其损失为

$$\mathcal{L}_{\text{nav}} = \frac{1}{24|\mathcal{B}_{\text{nav}}|} \sum_{i \in \mathcal{B}_{\text{nav}}} \sum_{r=1}^8 \|\hat{\mathbf{a}}_{i,r} - \mathbf{a}_{i,r}\|_2^2, \quad \mathcal{L} = \mathcal{L}_{\text{nav}} + \mathcal{L}_{\text{NTP}}. \quad (5)$$

两项损失的权重均为 1。归一化后，位置与朝向具有相同权重，因此坐标尺度决定二者在物理单位下的相对权重。固定八航点目标在轨迹末端附近包含填充停止位置，MLP 损失不额

外应用终止航点掩码。导航样本不参与下一词元监督。相反，问答和视觉定位样本不含动作目标，仅对带标注的回答词元贡献交叉熵；提示词元与填充词元不计入损失。其空轨迹目标会被掩蔽，而不会作为零运动目标学习。某类样本缺失时，其对应损失贡献为零。

4 实验

本节评测 NAVGPT-3 执行框架、其与 VLA 的协作及完整系统。首先，章节 4.1 展示如何在训练回合子集上通过自动化递归自我改进，开发执行框架工具和呈现给模型的视觉上下文。随后，章节 4.2 评测所得工具：针对两个规划器，在基础工具之上依次加入 VLA、地图和回溯工具组，并测量准确率、成本与反应时间。章节 4.3 改变工具返回结果呈现给规划器的方式，章节 4.4 则改变规划器检查 VLA 执行的频率。章节 4.5 在 R2R-CE 和 RxR-CE 上将完整系统与已发表方法进行比较，并将其迁移到物体目标导航与具身问答。章节 4.6 和 4.7 单独研究 NavGPT VLA，包括其编解码分配，以及随数据量和模型规模扩展的规律。最后，章节 4.8 将系统部署到 Unitree Go2 机器人上，章节 4.9 展示定性回合。

对于导航性能，本文在 R2R-CE 和 RxR-CE 的完整 Val-Unseen 划分 (Krantz et al., 2020; Anderson et al., 2018; Ku et al., 2020)、MP3D (Chang et al., 2017) 与 HM3D v2 (Ramakrishnan et al., 2021) 上的物体目标导航 (Batra et al., 2020)，以及具身问答 (Ren et al., 2024; Zhai et al., 2025; Jiang et al., 2025) 上评测 NAVGPT-3；消融研究采用固定 R2R 子集，其中包含来自十个 MP3D 扫描的 100 个回合 (Qiao et al., 2025)。报告指标包括成功率 (Success Rate, SR)、按路径长度加权的成功率 (Success weighted by Path Length, SPL)、归一化动态时间规整 (normalized Dynamic Time Warping, nDTW)、预言机成功率 (Oracle Success Rate, OSR) 和导航误差 (Navigation Error, NE)；问答任务则报告准确率和归一化步数。除非另行改变，执行框架研究均使用 8B NavGPT VLA。每个规划器都运行在其提供商的脚手架中：GPT-6 Astra 使用 Codex，Claude Opus 5 使用 Claude Code；脚手架负责组装消息并管理对话，包括自动压缩长上下文。NAVGP-3 替换系统提示、工具、观测、空间状态和停止规则；两个框架的差异详见 章节 D。

4.1 通过递归自我改进开发执行框架

本文将执行框架的递归自我改进 (RSI) 形式化为 KDLoop 风格的树搜索，在可执行的框架版本空间中探索 (Zhou et al., 2026a)。每个节点规定规划器的系统提示、可用工具、持久空间状态，以及向规划器呈现观测和执行结果的格式；每条边表示一项候选修订。THINK 基于当前执行框架和累积回合证据提出修改；CRITIC 根据接口约束和先前失败对其进行检查；EXPERIMENT 实现并评估通过筛选的分支；DISTILL 记录已确认结论、失败修改和尚未解决的假设。得分最高的候选版本成为下一版执行框架；进展停滞时，REFLECT 会重新审视整类修改。搜索发生在开发阶段：评测期间接口固定，不会在线更新规划器或 VLA 权重。图 5(a) 总结了该研究循环；图 (b) 展示从同一执行框架分叉出的三类修改，即工具接口、视觉上下文和执行结果；这些分支经修订后，选中分支并入下一版最佳执行框架。

搜索使用 Claude Opus 5 在固定的 100 回合子集上评测候选执行框架；该子集采样自 R2R-CE 训练划分，其场景与 Val-Unseen 互不重叠。搜索以成功率为选择信号，并利用回合轨迹诊断失败。图 5(c) 展示在全部 100 个回合上评测且达到 $SR \geq 50\%$ 的 14 种执行框架配置。前沿曲线表示截至当前达到的最佳分数，橙色虚线段表示相邻候选方案间的性能下降。

这些配置勾勒出执行框架的开发过程：前视观测与原始步进达到 66% SR；使用 4B 策略委派 VLA 达到 72%，使用 8B 策略则达到 76%；将 VLA 执行与空间工具结合，取得开发阶段最佳的 81%。早期位置图接口仅达到 54%，整合后的接口达到 77%。一次复核发现，任务提示描述了局部运动，但系统并未提供运动工具；后续修订加入该工具，并将拼接全景图替换为四幅独立的 512 像素视图。所得工具与视觉提示研究见 表 3 和图 6，最终工具接口见 表 4。

4.2 执行框架与 VLA 消融

表 3 衡量执行框架各组成部分的贡献，以及 VLA 在此基础上的增益。两个语言模型规划器 GPT-6 Astra 和 Claude Opus 5 在相同回合限制下，使用同一 8B NavGPT VLA 导航相同的 100 个 R2R-CE 回合。各行之间仅改变向规划器开放的工具，任务提示也只描述这些工具。工具及其视觉返回结果均通过 RSI 开发 (章节 4.1)。

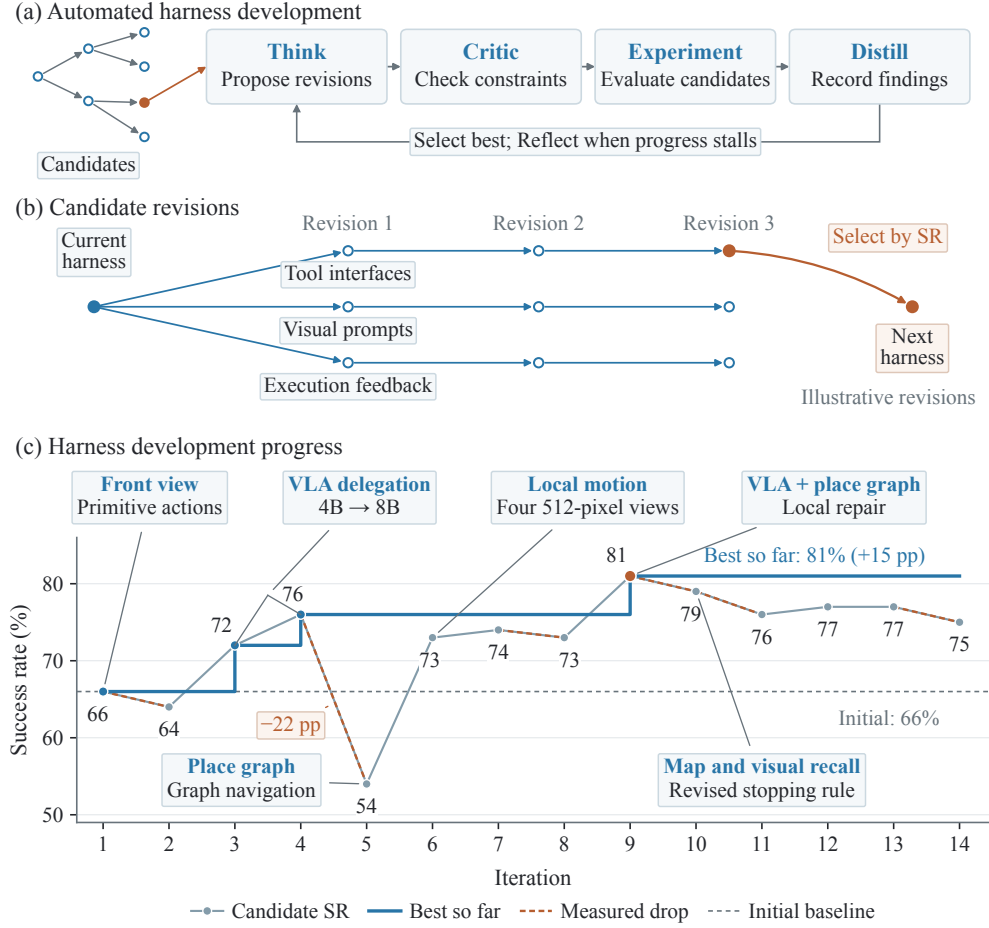


图 5: 通过递归自我改进开发执行框架：研究循环、候选修订和成功率演进。

各配置下规划器可执行的操作。 八个工具分为四组 (表 4)。每种配置都包含四个基础工具：`observe_forward` 和 `observe_panorama` 用于查看当前周围环境, `navigate_relative` 用于按给定角度和距离转向并移动, `terminate_episode` 用于结束回合。仅使用基础工具时, 规划器自行移动智能体, 因此每个环境动作都由规划器决策。还可加入三组工具。VLA 组中的 `navigate_by_instruction` 将完整指令或尚未完成的部分交给 NavGPT VLA; VLA 驾驶智能体, 直至发出停止信号或达到委派上限; 随后, 规划器接收最多八个已执行路线关键帧、终点视图和执行状态。VLA 停止只会结束本次委派, 只有规划器调用 `terminate_episode` 才会结束回合。地图组中的 `observe_map` 返回截至当前已观测区域的俯视图, 已访问位置按访问顺序编号。回溯组中的 `navigate_to_node` 和 `annotate_node` 使规划器能够将智能体送回任意带编号位置, 并用其在该处识别的内容为位置添加标签, 使地图不仅可读, 还可据此行动。

资源使用。 每行均报告单回合平均值。词元数统计全部输入 (包括缓存输入) 与全部输出; 轮次数统计由环境反馈分隔的规划器响应; 成本为提供商收费; 推理时间为模型总推理时间, 不含通信与执行框架开销。反应时间为环境动作之间的平均间隔: 不使用 VLA 时, 取规划器每轮耗时; 使用 VLA 时, 取 VLA 每步耗时 (章节 D)。

执行框架决定语言模型的导航水平。 仅使用基础工具且缺少专用执行框架时, 两个规划器性能较差、开销较高: GPT-6 Astra 达到 72 SR, 每回合使用 325.0k 个词元; Claude Opus 5 达到 66 SR, 每回合使用 721.0k 个词元 (表 3)。更稳健、更高质量的执行框架改变了这一结果。地图工具使规划器能够构建远强于当前视图所能提供的空间上下文, 回溯工具则

表 3: **R2R-CE 子集上的工具与 VLA 消融实验**。资源用量为每回合平均值。Opus 成本采用供应商 SDK 报告值；Astra 成本根据记录的用量和标价估算。推理时间不含通信与执行框架开销。反应时间表示环境动作之间的平均间隔；不使用 VLA 时为规划器每轮耗时，使用 VLA 时为每步 VLA 推理耗时。

规划器	工具			导航				资源用量				
	VLA	地图	回溯	NE↓	OSR↑	SR↑	SPL↑	词元数 (k) ↓	轮数 ↓	成本 (\$) ↓	推理时间 (s) ↓	反应时间 (s) ↓
仅 VLA	✓	–	–	4.23	78	71	66.14	–	–	–	44.1	0.78
GPT-6 规划器	–	–	–	4.67	74	72	62.26	325.0	32.14	0.593	105.7	3.3
GPT-6 规划器	–	✓	✓	2.55	87	83	69.04	190.8	10.71	0.469	66.8	6.2
GPT-6 规划器	✓	–	–	4.58	79	75	67.83	227.9	3.37	0.469	84.4	0.78
GPT-6 规划器	✓	✓	–	4.01	79	76	67.35	201.4	3.42	0.443	80.4	0.78
GPT-6 规划器	✓	✓	✓	3.34	84	82	69.12	155.0	3.88	0.391	79.1	0.78
Opus 5 规划器	–	–	–	5.43	70	66	55.24	721.0	47.93	0.69	425.9	8.9
Opus 5 规划器	–	✓	✓	3.45	85	73	44.82	483.4	16.75	0.990	323.4	19.3
Opus 5 规划器	✓	–	–	3.78	80	76	61.85	199.8	11.19	0.353	201.5	0.78
Opus 5 规划器	✓	✓	–	4.04	85	81	69.29	211.8	8.96	0.393	199.4	0.78
Opus 5 规划器	✓	✓	✓	3.69	83	79	66.70	185.6	8.50	0.397	191.8	0.78

组别	工具	对状态的影响	返回给规划器的内容
基础	observe_forward() observe_panorama() navigate_relative() terminate_episode()	无 更新已观测局部覆盖范围，不改变位姿 执行经验证的局部转向和平移 可选择返回指定节点；随后停止	用于精细语义检查的当前前向 RGB 图像。 四幅带标签的视图，包含方位、净空、位姿变化和节点信息。 实际运动状态、终点全景图及附近候选节点。 停止或返回状态；该调用结束回合且不可撤销。
VLA	navigate_by_instruction()	执行一次 VLA 轨迹展开；VLA 保留其历史	最多八个采样路线关键帧、终点全景图、可选的已观测地图及执行状态。
地图	observe_map()	更新已观测覆盖范围，不改变位姿	仅含已观测区域的俯视图、地图状态及按路线排序的节点表。
回溯	navigate_to_node() annotate_node()	返回已访问节点或最近一次轨迹展开中的航点 为现有节点附加标签	终点全景图、已观测地图、返回状态及更新后的节点表。 标注状态及更新后的节点表。

表 4: 实现导航能力的工具，分组方式与表 3 中的消融实验一致。

使其可以更高效地依据该上下文行动。二者共同将成功率提高至 83 和 73 SR，同时将词元量分别减少 41% 和 33%；Astra 的纯工具配置也是表中准确率最高的一行。

VLA 是否有益取决于语言模型。 VLA 总能缩短推理循环，但缺少良好执行框架时并不一定提升性能。对 Opus 而言，VLA 确有帮助：轮次数从 16.75 降至 8.50–11.19，成功率从 73 提升至 76–81 SR，每回合成本从 \$0.99 降至 \$0.35–0.40。对 Astra 则不然：轮次数从 10.71 降至 3.37，但成功率降至 75 SR，成本维持在 \$0.469，因为导航节省的词元又用于监督和修正 VLA 执行。因此，语言模型越强，越能自行吸收原本由 VLA 提供的收益；成功率上限由语言模型本身决定：当模型已能充分理解环境并独立作出准确决策时，成功率受限于其自身导航能力，而非接口。为 VLA 配备专属执行框架，并针对其轨迹展开提供地图与回溯能力后，两个模型的词元使用量均降至最低（Astra 为 155.0k，Opus 为 185.6k），同时成功率接近各自最佳值。

VLA 决定最短反应时间。 不使用 VLA 时，系统只能按语言模型的决策频率行动；根据模型和工具不同，平均每 3–19s 执行一次。使用 VLA 后，控制循环改为按 VLA 推理频率运行；8B 模型平均每步耗时 0.78s，从而为整个系统提供基本的实时响应保证（表 3）。章节 4.4 通过逐步提高 VLA 执行检查频率，考察规划器能在多大程度上利用这一更高交互频率。

4.3 视觉提示组织方式

执行框架还决定如何向规划器呈现工具返回结果：将相机视图与已观测地图构成视觉提示，并标注视图标签、节点 ID、朝向和转向提示。保持工具不变的情况下，图 6 以这一带定位标注的参考方案为基准，每次改变一种呈现因素（章节 D.5）：说明文字标签将标注移至相邻文本；紧凑字形使用更小的数字字体；朝向向上方案使地图随智能体旋转；绝对方位则用固定坐标系方向替换相对转向提示。两个语言模型对视觉提示的组织方式均具有稳健性：在全部五种版本中，Astra 的 SR 保持在 76–78%，Opus 5 则保持在 73–77%。

变化的是不同版本所需的推理量，而这一需求与人类可读性并不一致。对人而言最难读取的绝对方位版本，反而使两个模型使用的词元最少；成本最高的版本则因模型而异：Astra 为紧凑字形，Opus 为带定位标注的参考方案；二者每回合平均累计词元量分别落在 160–210k 和 139–173k 范围内。因此，强语言模型对视觉上下文组织方式较为稳健，其易读内容并不遵循人类可读性规律。

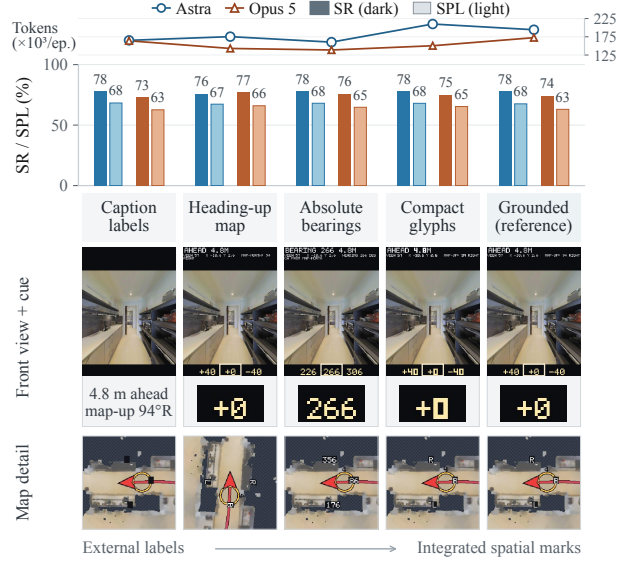


图 6: 视觉提示组织。Astra 和 Opus 5 的 SR/SPL 与词元用量。

4.4 通信频率与成本

表 5 改变规划器检查 VLA 执行的频率，以每次检查间的 VLA 前向传播次数衡量；每次前向传播预测八个航点，并执行第一个（章节 3.2.1）。不进行周期性检查时，只有当 VLA 停止或达到委派上限，规划器才会重新获得控制权；此时步数中位数为 51，90 分位数为 123。每次检查时，规划器会复核截至当前的路线，而 VLA 继续执行；仅当需要修订或终止路线时，规划器才会中断 VLA。实验使用工具和指令保持一致的 GPT-6 Astra。

更频繁的检查能够改善路线，但代价高昂。每 32 步检查一次时，多数回合至多检查一次，结果几乎不变（78 vs. 77 SR）。每 16 步检查一次，即每回合约三次，可将 SR 提升至 83，SPL 从 67.51 提升至 77.36，并将 NE 从 3.82 降至 2.01m。由于每次检查都会重新读取累积路线上下文，词元量的增长远快于准确率：每回合词元量从 179.2k 增至 32 步检查时的 1,529.9k，并在 16 步检查时达到 2,319.5k。因此，周期性检查以高成本监督换取准确率，这也说明现实部署需要事件触发式复核。

表 5: 检查频率。每次检查间隔的 VLA 步数。

频率	NE↓	OSR↑	nDTW↑	SR↑	SPL↑	词元数 (k) ↓
无	3.82	81	16.74	77	67.51	179.2
32	3.56	81	16.66	78	68.88	1,529.9
16	2.01	85	18.72	83	77.36	2,319.5

表 6: 视觉语言导航。在完整 R2R-CE 和 RxR-CE Val-Unseen 划分上的评测结果。[†]RxR Val-Unseen 的英语人类跟随者 (Ku et al., 2020), 在 DE 中测量。两个 NavGPT-3 系统均使用 8B VLA。

方法	模型	R2R-CE Val-Unseen				RxR-CE Val-Unseen			
		NE↓	OSR↑	SR↑	SPL↑	NE↓	nDTW↑	SR↑	SPL↑
人类 [†] (Ku et al., 2020)	—	—	—	—	—	1.32	77.7	90.4	—
NavFoM (Zhang et al., 2025a)	7B	4.61	72.1	61.7	55.3	4.74	65.8	64.4	56.2
ABot-N0 (AMAP CV Lab, 2026)	4B	3.78	70.8	66.4	63.9	3.83	—	69.3	60.0
AstraNav-World (Chen et al., 2026)	3B+5B	3.86	73.9	67.9	65.4	3.82	—	72.9	61.5
OmniNav (Xue et al., 2025)	3B	3.74	74.6	69.5	66.1	3.77	—	73.6	62.0
Qwen-RobotNav (Qwen Team, 2026b)	4B	3.80	77.2	69.5	63.6	3.80	71.9	75.2	65.0
Qwen-RobotNav (Qwen Team, 2026b)	8B	3.53	78.5	72.1	66.6	3.58	72.5	76.5	65.7
Robostrat Navigate 导航模型 (Mistral AI, 2026)	8B	3.20	81.3	77.4	74.2	3.47	—	75.1	68.7
本文: 可执行导航策略									
NavGPT VLA	4B	3.41	79.91	72.54	67.19	3.35	73.33	76.77	67.77
NavGPT VLA	8B	3.29	80.19	74.51	68.54	3.05	74.85	78.19	68.98
本文: 完整导航系统									
NavGPT-3	Opus 5	2.77	84.07	79.01	70.16	2.27	75.09	84.80	69.85
NavGPT-3	GPT-6 Astra	2.18	86.99	81.51	70.42	1.56	78.47	90.43	74.31

表 7: 目标物体导航。以往方法使用 HM3D v1, 其余方法使用 HM3D v2。[†]: 500 回合子集, 采用更严格的停止判据评分。[‡]: 深度或里程计。

模型	MP3D		HM3D	
	SR↑	SPL↑	SR↑	SPL↑
以往方法: HM3D v1				
VLFM [†]	36.4	17.5	52.5	30.4
SG-Nav [†]	40.2	16.0	54.0	24.9
WMNav [†]	45.4	17.2	58.1	31.5
CogNav [†]	46.6	16.1	72.5	26.2
Uni-NaVid	—	—	73.7	37.4

表 8: 具身问答。在 HM-EQA (Ren et al., 2024) 和 MT-HM3D (Zhai et al., 2025) 上的准确率与归一化步数 (Zhai et al., 2025); 在 EXPRESS-Bench (Jiang et al., 2025) 上的 LLM 评分与路径效率。^{*}: 数据来自以往工作。NavGPT-3 采用 GPT-6 Astra 评判器。

模型	HM-EQA		MT-HM3D		EXPRESS	
	准确率↑	步数↓	准确率↑	步数↓	评分↑	路径效率↑
以往系统						
Explore-EQA (Ren et al., 2024)	58.4	0.52	36.2 [*]	0.64	—	—
Graph-EQA (Saxena et al., 2024)	63.5	0.20	45.6 [*]	0.45	—	—
Memory-EQA (Zhai et al., 2025)	61.4	0.40	43.1	0.41	—	—
Fine-EQA (Jiang et al., 2025)	56.0	0.54	—	—	63.95	25.58
3D-Mem (Yang et al., 2025)	50.4	0.63	—	—	—	—
FAST-EQA (Zhang et al., 2026)	69.2	0.65	50.5	0.52	68.7	29.25
Qwen-RobotNav 4B 模型	52.2	16.0	75.6	30.6	—	—
Qwen-RobotNav 8B 模型	48.8	17.7	71.2	33.9	—	—
NavMCP (Lei et al., 2026)	76.7	0.15	54.4	0.19	79.27	33.96
NAVGP-3 [†]	59.2	25.2	75.3	40.8	80.43	34.26

4.5 整体导航性能

表 6 在 R2R-CE 和 RxR-CE 的完整 Val-Unseen 划分上, 将完整 NavGPT-3 系统与已发表方法进行比较。即使不使用任何执行框架, NavGPT VLA 在 RxR-CE 上也已达到当前最佳水平。引入运行时后, 使用 GPT-6 Astra 可将 R2R-CE 成功率从 74.51 提升至 81.51 SR; 在 RxR-CE 上增益更大: 相较于专用策略单独运行时的 78.19 SR, 完整系统分别达到 84.80 和 90.43 SR, nDTW 也提高至 78.47。RxR 指令更长, 对路线的描述更为详细, 因此返回的路线证据、返回已记录位置和局部修正具有更大发挥空间。NavGPT-3 在成功率 (90.43 vs. 90.4 SR) 和路径保真度 (78.47 vs. 77.7 nDTW) 上均达到 RxR 人类跟随者水平, 成为该基准上首个达到人类水平的自主智能体 (Ku et al., 2020)。

物体导航与具身问答。表 7 在 MP3D 与 HM3D v2 验证划分的 500 回合子集上评测 NavGPT-3, 并与已发表的完整划分结果比较; 只有智能体在距离阈值内面向目标停止时才计为成功。两项跨任务研究均在 Claude Opus 5 规划器下使用 8B NavGPT VLA。两个基准体现运行时贡献的方式不同: MP3D 包含 21 类目标, 其中许多尺寸较小或粒度较细

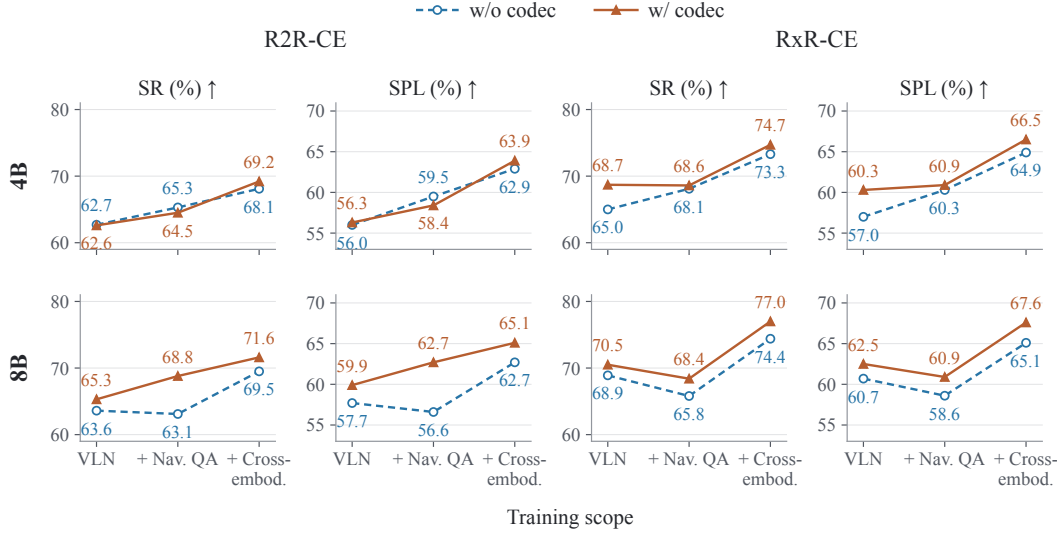


图 7: 不同训练范围的编解码器对比。

(靠垫、毛巾、图片), 增益体现在成功率上; HM3D v2 则采用六个尺寸较大、视觉特征鲜明的类别, 已发表系统约四分之三的回合都能成功, 因此增益主要体现在路径效率上。在具身问答中 (表 8), NAVGPT-3 与此前最强系统 Qwen-RobotNav (Lei et al., 2026) 相当: 在归一化步数相近时, 其在两个数据集上的准确率均领先 (章节 D.3)。

4.6 不同训练范围下的编解码分配

图 7 对比了不使用编解码分配与使用编解码分配两种方案。前者是本文重新实现 Qwen-RobotNav 观测接口 (Qwen Team, 2026b) 时采用的位置分配, 后者是本文提出的变化感知分配。4B 与 8B 策略分别在三个累积数据范围上训练, 标记为 VLN、+ Nav. QA (加入具身导航问答) 和 + Cross-embod. (进一步加入跨具身导航)。所有策略均在完整 R2R-CE 和 RxR-CE Val-Unseen 划分上独立评测。

编解码分配在所有 8B 设置中均有帮助, 在 RxR-CE 上对两种模型规模也均有效。对于 8B 模型, 它在三个数据范围上分别将 R2R-CE SR 提高 1.7、5.7 和 2.1 个百分点, 将 RxR-CE SR 提高 1.6、2.6 和 2.6 个百分点; 加入导航问答后的大幅增益, 部分源于该数据范围下较弱的位置分配基线。对于 4B 模型, 它将 RxR-CE SR 分别提高 3.7、0.5 和 1.4 个百分点; 在 R2R-CE 上, 前两个数据范围内的 SR 变化不超过 1 个百分点 (−0.1 和 −0.8), 只有加入跨具身数据后才有所提升 (+1.1)。使用跨具身数据时, 编解码分配在两种模型规模和两个基准上均能同时改善 SR 与 SPL。

这一现象可能三个原因。第一, 当图像竞争有约束力的预算时 ($\rho_{\text{vis}} < 1$, 章节 3.2.1), 编解码分配最为重要。RxR-CE 路线更长, 指令描述的沿途地标更多, 因此更多步骤会使用完整历史; 保留变化视图中的细节能够产生收益, 这与两种规模模型在 RxR-CE 上的增益一致。第二, 变化感知规则的价值随训练数据视觉多样性的增加而上升。在 VLN 数据中, 每个回合都来自同一模拟器, 使用四个固定相机和一种运动尺度, 因此可根据时间和相机基本预测帧变化幅度, 位置规则已接近编解码规则; 加入导航问答只是在相同回合上添加问答, 不会引入新的视觉条件。跨具身训练则加入单相机视图、慢速与快速运动、移动目标、随机相机高度与视场, 以及户外驾驶; 不同数据源的帧间变化幅度差异显著。固定位置规则无法适应这些差异, 编解码分配则会跟随每个样本中的变化。第三, 编解码分配以不同分辨率呈现图像, 模型必须学会读取。8B 模型在每种数据范围上均能利用这一特点, 而 4B 模型只有在跨具身训练提供足够多样的视觉数据后, 才能在 R2R-CE 上获益。

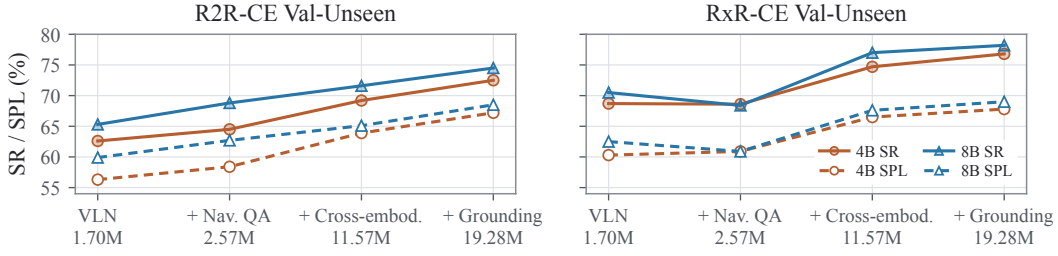


图 8: 模型与数据规模扩展。

4.7 模型与数据扩展

图 8 绘制了使用编解码分配训练的 4B 和 8B 策略，在四个累积训练范围上的完整 Val-Unseen 结果。前三个数据范围对应图 7 中的使用编解码分配曲线；第四个范围加入视觉定位数据，即主要比较中使用的约 19.28M 条记录的完整混合数据。

随着训练数据和模型规模增长，NavGPT VLA 的性能稳步提高。数据贡献最大：将混合数据扩大十一倍，从 1.70M 条 VLN 记录增至完整的 19.28M 条，使 4B 模型的 R2R-CE SR 从 62.6 提高至 72.54，8B 模型从 65.3 提高至 74.51；RxR-CE SR 则分别从 68.7 提高至 76.77、从 70.5 提高至 78.19。R2R-CE SR 在每一步均有改善；在 RxR-CE 上，加入导航问答后，4B 模型的 SR 保持不变，8B 模型反而下降（从 70.5 降至 68.4），而大幅增益出现在加入跨具身数据后。增大模型规模带来的提升较小，但稳定一致：在每个数据范围上，8B 模型的 R2R-CE SR 均比 4B 模型高 2.0 至 4.3 个百分点。因此，数据带来的效果超过参数量翻倍。加入跨具身数据后，4B 模型（69.2 SR）超过仅用 VLN 训练的 8B 模型（65.3）；完整 4B 模型（72.54）也超过未进行最终视觉定位训练的 8B 模型（71.6）。数据曲线尚未饱和：最后一步加入视觉定位数据，将混合数据从 11.57M 条扩大至 19.28M 条，仍使 4B 和 8B 模型的 R2R-CE SR 分别提高 3.3 和 2.9 个百分点；8B 模型的优势在最大规模下依然存在，表明增加数据和模型规模仍可能带来进一步增益。由于每一步同时改变数据量和数据源构成，这些曲线衡量的是累积配方，而非任一单独数据源的价值。

4.8 现实世界部署

本文将 NavGPT-3 部署到 Unitree Go2 上，使用 RGB 输入、远程或机载 VLA 推理，以及运行时 LiDAR 危险监控器。在配对比较中（表 10），相较于顺序排队处理，将推理重叠与优先级监控相结合可使 SR 从 73.3 提高至 83.3，并将停止延迟的第 95 百分位数 (p95) 从 1840 降至 180ms。

4.8.1 部署设置

机器人与感知。 本文使用 Unitree Go2 平台和机载 VLA 推理。NavGPT VLA 接收机器人的前向 RGB 流，并在保留的单视图历史上分配视觉上下文。这与章节 3.2.1 中模拟器的四视图配置不同。机载 LiDAR 为运行时监控器提供几何危险检测，但不作为 VLA 的额外输入。

远程与机载推理。 在远程配置中，机器人压缩并传输观测及规划器选定的导航指令至服务器；服务器预测航点并返回给机器人执行。机载配置遵循 Qwen Team (2026b)，在 NVIDIA Jetson Thor 上通过 FP8 量化和 TensorRT 加速运行 4B NavGPT VLA。两种配置均向执行框架提供相同的航点接口。机载推理省去了 VLA 预测的网络往返；托管规划器仍作为独立服务运行。模型推理、通信和物理执行分别影响端到端响应时间，下文将单独测量。

执行框架与运行时集成。 执行框架管理观测、空间状态、工具调用和执行返回；其上层运行时则协调节奏相互独立的推理、执行与监控。无论 VLA 位于服务器还是机器人上，这一职责划分均不改变。在机器人上，每条任务运动指令都会通过最终指令过滤器；只有发出指令的操作持有运行时运动许可时，过滤器才会转发指令。机器人评测检验 LiDAR 事件能否在不等待规划器的情况下撤销该许可，以及系统是否只在获得新感知与新许可后恢复执行（章节 B.6）。

4.8.2 玻璃门执行可视化



图 9: 记录的玻璃门执行过程。带边框的第四幅图突出显示机器人在门附近暂停前进，随后横向修正并继续通过。

接近与暂停。 图 9 展示一条进入商店的已记录指令。图 1–3 显示机器人从室内大厅向已打开的玻璃门口接近，门旁立有独立标牌。图 4 (5.48s) 突出显示机器人在玻璃附近中断前进。传感器位姿表明，暂停与修正发生在 5.48 至 7.36s 之间；横向运动仍在继续，因此机器人并非完全静止。

修正与继续。 图 5 (6.85s) 展示横向调整。图 6 (7.71s) 中机器人恢复前进，图 7 中通过入口，图 8 则显示机器人已进入商店，未与玻璃门发生可见接触。相连点展示了接近过程被中断后，机器人沿修订路线通过开口：进入商店的任务保持不变，但即时路线发生改变。

与运行时控制的联系。 该序列展示了运行时所针对的“暂停–修订–继续”模式：挂起当前运动操作，检查新观测，并在重新授权后沿更新路径恢复执行。

4.8.3 运行时评测

受控输入。 各条件共享前向 RGB 流、LiDAR 检测器及带时间戳事件、速度限制、路线和制造商安全防护。每组运行时比较均固定 VLA 部署位置与精度，同时固定单相机输入配置、VLA 检查点、规划器提示、检测阈值和固件。独立的操作员停止功能始终保持启用；操作员介入计为试验失败。

表 9: Go2 场景与测量指标。

场景	受控事件	主要测量指标
A: 路线修订	在长时间执行期间需要绕行或修正终点；改变事件相对于规划器推理的触发时机。	完成 SR；从事件触发到获准修复的时间；路线效率；规划器调用次数；执行空闲时间。
B: LiDAR 防护	运动过程中，软质障碍物进入已标定的保护区域，包括规划器繁忙时；以匹配的无害通行过程作为负样本。	漏停与误停；从触发到停止的 p50/p95；触发后移动距离；最小间隙；操作员干预次数。
C: 优先级与恢复	在修复期间或 VLA 即将完成时触发危险；延迟模型回复，向机器人的命令过滤器重发过期命令，随后解除危险。	被接受的过期命令；事件顺序；停止延迟；未经授权的恢复；正确恢复与任务完成。

比较设置。 本文比较纯 VLA 与带工具规划器基线，并进行配对 2×2 运行时比较：规划器–VLA 顺序执行与重叠执行交叉搭配排队监控处理与优先级监控处理。四种运行时配置共享相同模型、工具定义、路线证据和 LiDAR 检测结果。排队处理在下一常规控制步处理检测到的事件；优先级处理则在指令过滤器处立即撤销当前许可。在顺序条件下保留优先级监控器，可检验保护作用能否仅由简单、独立的停止门解释；重叠对比则检验推理期间的任务推进情况。最终运行时配置将重叠执行与优先级处理相结合。这些控制变量将检测器质量、调度与权限转移的影响分离开来。

结果。 重叠执行与优先级处理分别带来独立贡献（表 10）。由于规划器复核路线时执行仍在继续，规划器推理与 VLA 执行重叠可将排队处理下的修复时间从 12.8 缩短至 9.6s，并将优先级处理下的修复时间从 11.5 缩短至 7.9s。优先级处理将顺序运行时中的漏停次数从 7/30 降至 2/30，在重叠执行时从 5/30 降至 0/30；相应地，p95 停止延迟分别从 1840 降至 420ms、从 1310 降至 180ms。将二者结合，可取得最高修订 SR (83.3；顺序排队处理为 73.3)、最少的已接受过期指令 (1/30)，以及最高恢复 SR (80.0)。

表 10: Go2 运行时结果。

设置	A: 路线修订		B: 安全防护		C: 恢复	
	SR	修复时间 (s)	漏停/ N	停止 p95 (ms)	过期指令/ N	恢复 SR
仅 VLA	60.0	—	—	—	—	—
规划器 + 工具	70.0	14.2	9/30	2150	11/30	53.3
顺序执行 + 队列	73.3	12.8	7/30	1840	8/30	60.0
顺序执行 + 优先级	76.7	11.5	2/30	420	3/30	70.0
重叠执行 + 队列	76.7	9.6	5/30	1310	6/30	66.7
重叠执行 + 优先级	83.3	7.9	0/30	180	1/30	80.0

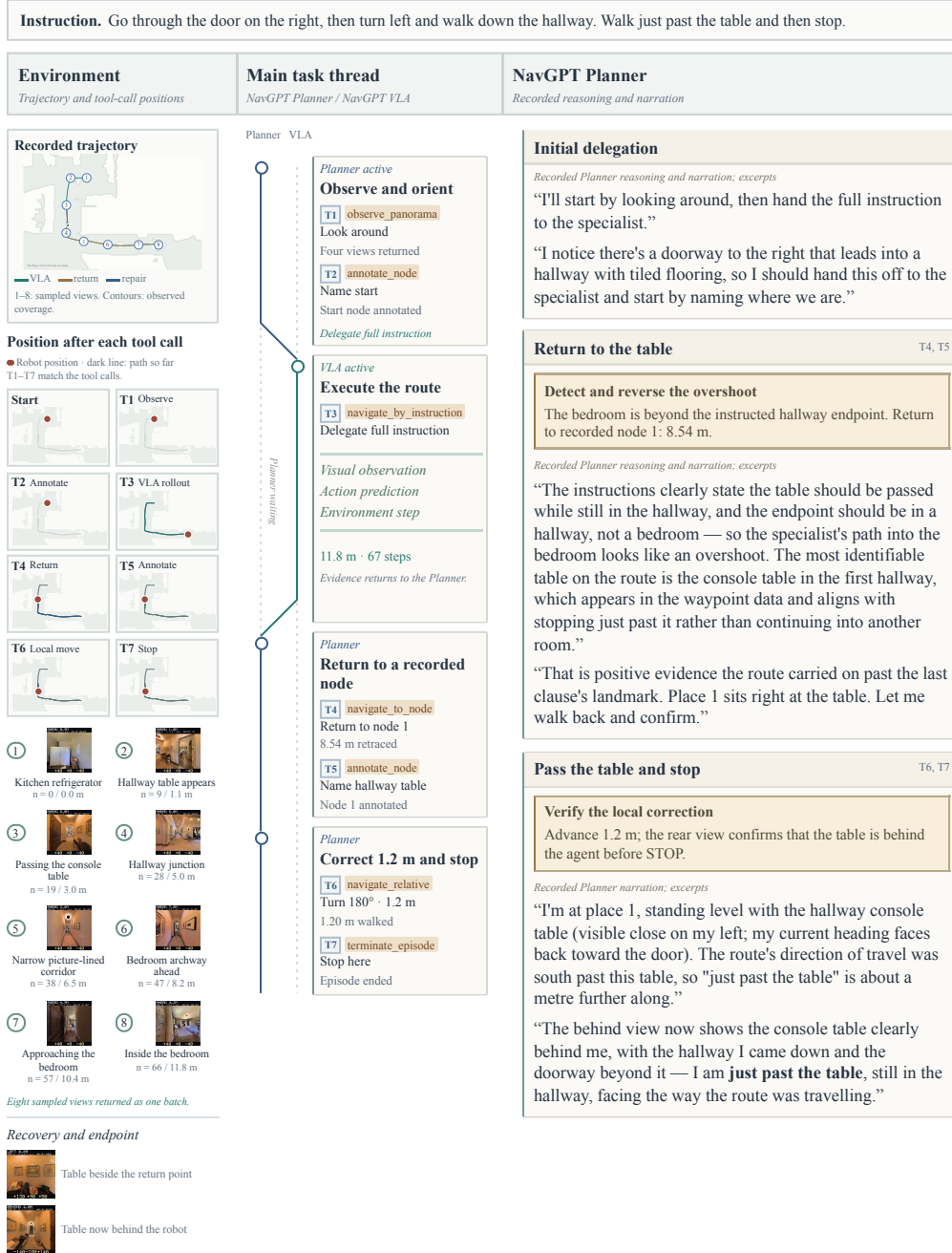
事件计时与结果。 本文分别记录观测压缩、网络传输和 VLA 推理耗时，并在同一同步时钟上记录传感器采集、检测器触发、复核开始与结束、许可撤销、最后一条被接受的非零指令、停止指令下发、物理停止、许可更新及运动恢复。机器人软件接口接受停止指令本身并不计为停止：只有独立测得的速度在固定时间内持续低于试验前设定的阈值，才认为机器人已停止。过期指令指在已撤销许可版本下发出的指令；Stale/ N 统计这类指令被接受的试验数。恢复成功要求获得新感知、新许可，并在无操作员介入的情况下完成剩余任务。

试验分配与不确定性。 每种条件和场景先进行五次试验性测试以验证日志记录，这些测试不计入比较。正式评测使用五条路线、三种事件时机，并对每种条件和场景重复两次（共 30 次试验），以配对区组随机化。检测假阴性和假阳性独立于检测器输出进行标注。

4.9 定性结果

四个精选成功回合展示了长时程 VLA 执行后，返回观测如何引导规划器。Claude Opus 5 识别出 VLA 已走过走廊桌子，回溯 8.54m 至节点 1，并验证 1.2m 的终点修正（图 10）。在一条更长的路线中，它从死胡同角落返回，并直接完成被遗漏的橱柜和门口指令子句（图 11）。GPT-6 Astra 从执行停滞中恢复：重复调用 VLA 且局部移动受阻后，它退回节点 3，改走另一条过道（图 12）。最后，当局部楼梯尝试仍未完成指令时，Opus 再次进行委派：下一次轨迹展开到达楼上浴室（图 13）。

每个案例均包含完整指令。地图 T1 及后续地图显示每次工具调用返回后的位置；带编号视图采样自首次 VLA 轨迹展开。实线表示持有运动权限的线程，虚线表示等待。词元计数包含缓存输入和输出。



Planner VLA

Planner active

Observe and orient

T1 observe_panorama
Look around
Four views returned
T2 annotate_node
Name start
Start node annotated
Delegate full instruction

VLA active

Execute the route

T3 navigate_by_instruction
Delegate full instruction

Visual observation
Action prediction
Environment step

11.8 m · 67 steps
Evidence returns to the Planner.

Planner

Return to a recorded node

T4 navigate_to_node
Return to node 1
8.54 m retraced
T5 annotate_node
Name hallway table
Node 1 annotated

Planner

Correct 1.2 m and stop

T6 navigate_relative
Turn 180° · 1.2 m
1.20 m walked
T7 terminate_episode
Stop here
Episode ended

Initial delegation

Recorded Planner reasoning and narration; excerpts

“I’ll start by looking around, then hand the full instruction to the specialist.”

“I notice there’s a doorway to the right that leads into a hallway with tiled flooring, so I should hand this off to the specialist and start by naming where we are.”

Return to the table T4, T5

Detect and reverse the overshoot

The bedroom is beyond the instructed hallway endpoint. Return to recorded node 1: 8.54 m.

Recorded Planner reasoning and narration; excerpts

“The instructions clearly state the table should be passed while still in the hallway, and the endpoint should be in a hallway, not a bedroom — so the specialist’s path into the bedroom looks like an overshoot. The most identifiable table on the route is the console table in the first hallway, which appears in the waypoint data and aligns with stopping just past it rather than continuing into another room.”

“That is positive evidence the route carried on past the last clause’s landmark. Place 1 sits right at the table. Let me walk back and confirm.”

Pass the table and stop T6, T7

Verify the local correction

Advance 1.2 m; the rear view confirms that the table is behind the agent before STOP.

Recorded Planner narration; excerpts

“I’m at place 1, standing level with the hallway console table (visible close on my left; my current heading faces back toward the door). The route’s direction of travel was south past this table, so “just past the table” is about a metre further along.”

“The behind view now shows the console table clearly behind me, with the hallway I came down and the doorway beyond it — I am **just past the table**, still in the hallway, facing the way the route was travelling.”

图 10: 回溯至走过的终点。Claude Opus 5 在 R2R-CE 上的第 144 个回合。规划器识别出 VLA 在卧室停止前经过的走廊桌子，回溯 8.54 m 至节点 1，并在停止前验证 1.2 m 的局部修正。

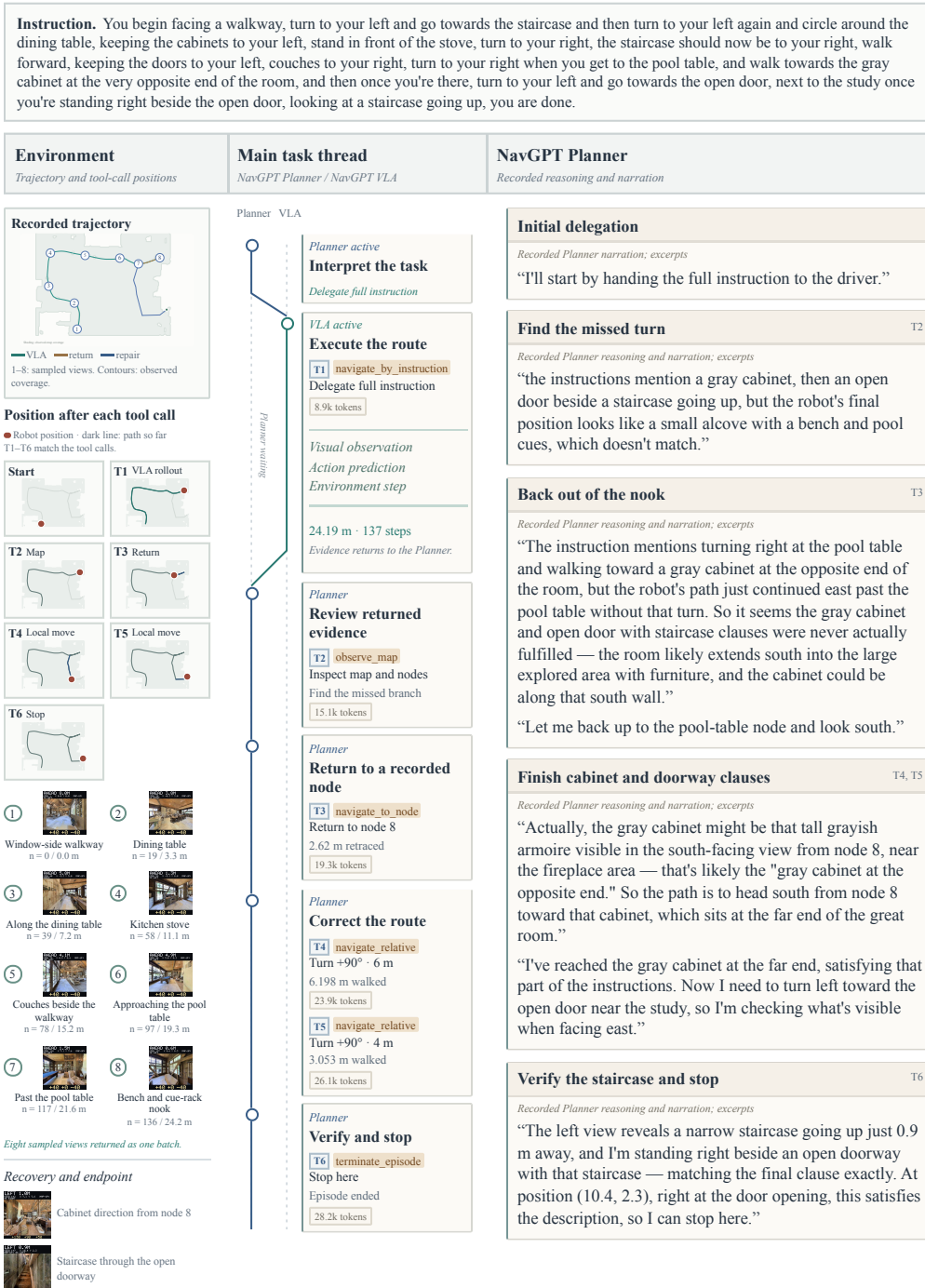


图 11: 补全遗漏的路线子句。Claude Opus 5 在 RxR-CE 上的第 2523 个回合。规划器从死胡同角落返回节点 8，随后通过相对运动到达灰色橱柜和带上行楼梯的开放门口。

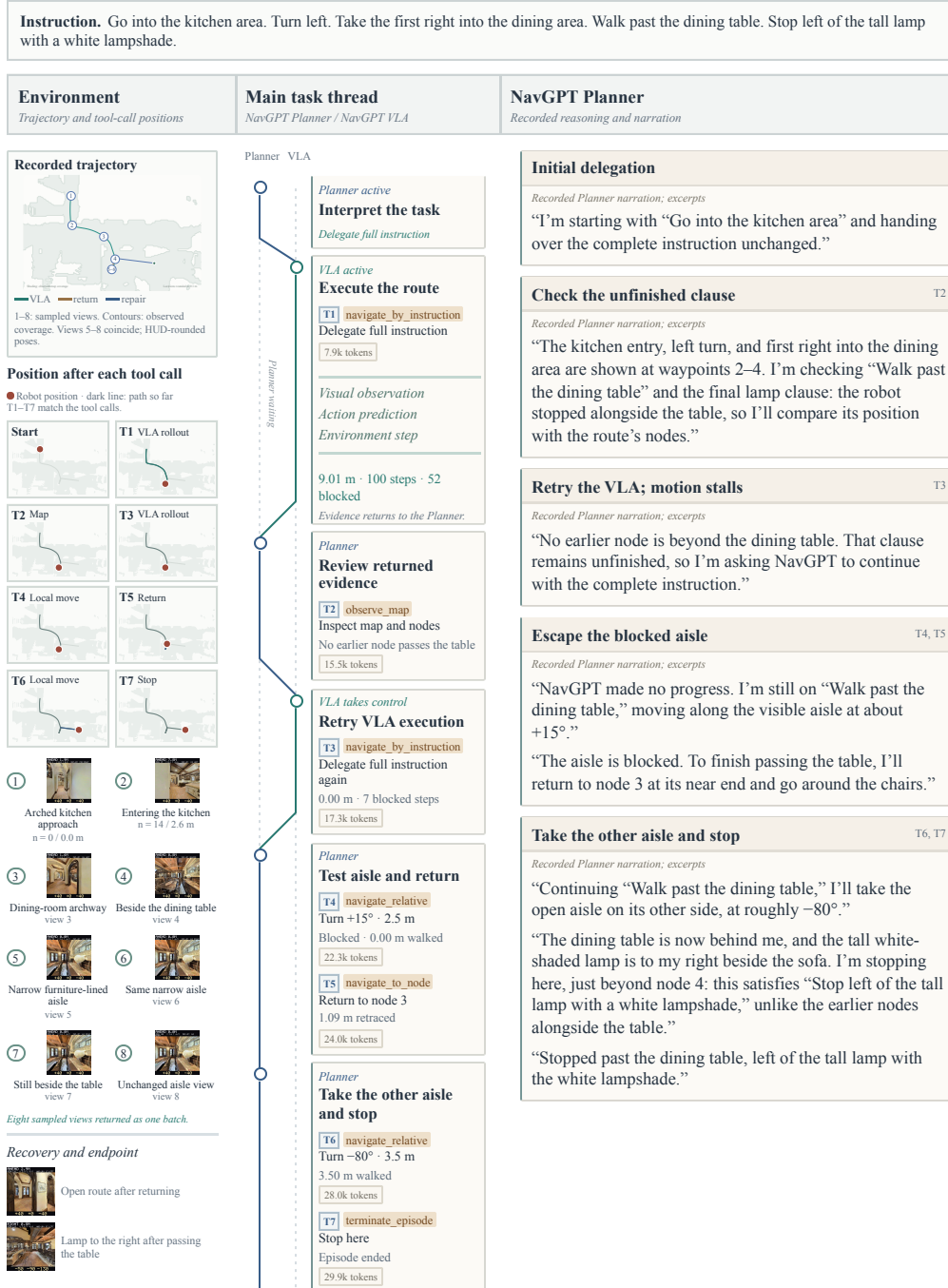


图 12: 从过道受阻中恢复。GPT-6 Astra 在 R2R-CE 上的第 43 个回合。重复执行 VLA 且直接运动尝试停滞，规划器返回节点 3，穿过另一条过道，并停在灯旁桌子的另一侧。



图 13: 局部修正失败后恢复委派。Claude Opus 5 在 RxR-CE 上的第 5782 个回合。楼梯子句仍未完成，规划器再次委派完整指令。第二次轨迹展开到达楼上浴室，规划器验证后停止。

5 结论

本文提出 NavGPT-3: 一个完整的导航执行框架, 其上设有运行时层。执行框架组织上下文和导航状态, 提供空间工具, 并返回执行证据, 以支持路线修复和任务完成判断; 运行时则通过线程控制与运动权限协调这些能力。独立 VLA 和完整系统在完整划分上均取得了出色的导航结果, 后者在连续控制而非基于图的控制条件下, 于 RxR-CE 上达到人类跟随者水平。执行框架研究表明, 空间工具能够支持规划器 (Planner) 导航, 视觉呈现会影响处理成本, 而 VLA 委派可以用更少的推理轮次提供高频动作更新。这些组件共同构成完整的具身接口, 使推理能够检查、修正并控制长时程物理执行。随着更强推理模型和动作模型的出现, 该执行框架可为其集成提供基础结构。

AI 使用声明

大语言模型是本文所研究系统的组成部分：如 章节 4 和 章节 D 所述，GPT-6 Astra 与 Claude Opus 5 担任规划器，GPT-6 Astra 评审模型则为 EXPRESS-Bench 的回答评分。此外，我们使用生成式 AI 编程助手（Claude Code 和 Codex）协助实现实验与评测基础设施、创建和修订科学图表，并起草、编辑和排版本文部分内容。我们已审查所有 AI 辅助工作：AI 辅助生成的代码在使用前均经过测试；所有文本、图表和报告数值均由作者依据实验记录输出进行核对。我们对本文最终内容承担责任，包括借助生成式 AI 产生的文本、论断或工件。

参考文献

- AMAP CV Lab. ABot-N0: Technical report on the vla foundation model for versatile embodied navigation. *arXiv preprint arXiv:2602.11598*, 2026. 2, 12
- Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Kaichen Zhang, Wenkang Zhang, Zheng Cheng, Nansen Zhang, Chunsheng Wu, Chunjiang Ge, Zimin Ran, Dehua Song, Chunyuan Li, Shikun Feng, Ming Hu, Zhangquan Chen, Junbo Niu, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. LLaVA-OneVision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026. 2, 3, 6
- Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pp. 3674–3683, 2018. 2, 8
- Anthropic. Claude Fable 5.1 and Claude Mythos 5.1 system card. <https://www.anthropic.com/claude-fable-and-mythos-5-1>, 2026a. System card released 2026-09-01. 1
- Anthropic. Claude Opus 5 system card. <https://www-cdn.anthropic.com/b514064af1408018e64b1ad24e7d5e75850b4ffd/Claude%20Opus%20System%20Card.pdf>, 2026b. System card released 2026-07-24. 1
- Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Motlaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects. In *arXiv:2006.13171*, 2020. 8
- Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11621–11631, 2020. 5
- Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pp. 667–676. IEEE, 2017. 8
- Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K Wong. Mapgpt: Map-guided prompting for unified vision-and-language navigation. *arXiv preprint arXiv:2401.07314*, 2024. 2
- Jintao Chen, Junjun Hu, Haochen Bai, Minghua Luo, Xinda Xue, Botao Ren, Chengyu Bai, Shichao Xie, Ziyi Chen, Fei Liu, Zedong Chu, Xiaolong Wu, Mu Xu, and Shanghang Zhang. AstraNav-World: World model for foresight control and consistency. *arXiv preprint arXiv:2512.21714*, 2026. 12
- An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. In *Robotics: Science and Systems (RSS)*, 2025. 2
- Hao-Tien Lewis Chiang, Zhuo Xu, Zipeng Fu, Mithun George Jacob, Tingnan Zhang, Tsang-Wei Edward Lee, Wenhao Yu, Connor Schenck, David Rendleman, Dhruv Shah, Fei Xia, Jasmine Hsu, Jonathan Hoech, Pete Florence, Sean Kirmani, Sumeet Singh, Vikas Sindhwani, Carolina Parada, Chelsea Finn, Peng Xu, Sergey Levine, and Jie Tan. Mobility VLA: Multimodal instruction navigation with long-context VLMs and topological graphs. *arXiv preprint arXiv:2407.07775*, 2024. 2
- DeepSeek-AI. DeepSeek Harness: Everything is a plugin. <https://github.com/deepseek-ai/deepseek-harness>, 2026. Accessed 2026-08-29. 2

-
- Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024. URL <https://arxiv.org/abs/2410.00037>. 30
- Chongkai Gao, Zixuan Liu, Zhenghao Chi, Junshan Huang, Xin Fei, Yiwen Hou, Yuxuan Zhang, Yudi Lin, Zhirui Fang, Zeyu Jiang, and Lin Shao. VLA-OS: Structuring and dissecting planning representations and paradigms in vision-language-action models. *arXiv preprint arXiv:2506.17561*, 2025. 2
- Google DeepMind. Gemini 3.1 Pro model card. <https://deepmind.google/models/model-cards/gemini-3-1-pro/>, 2026. 5, 6
- Pengyuan Guo, Zhonghao Mai, Zhengtong Xu, Kaidi Zhang, Quan Khanh Luu, Heng Zhang, Zichen Miao, Arash Ajoudani, Zachary Kingston, Qiang Qiu, and Yu She. PPlanAR: Planning-language-grounded agentic reasoning for robot manipulation. *arXiv preprint arXiv:2602.01662*, 2026. 2
- Jiaheng Hu, Mohit Shridhar, Caden Lu, Dhruv Shah, Hao-Tien Lewis Chiang, Jie Tan, and Annie Xie. What matters in orchestrating robot policies: A systematic study of hierarchical VLA agents. *arXiv preprint arXiv:2606.10267*, 2026. 2
- Muye Huang, Lingling Zhang, Xingyu Yu, Lei Shi, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Jun Liu. DuplexOmni: Real-time listening, seeing, thinking, and speaking for full-duplex interaction. *arXiv preprint arXiv:2606.09186*, 2026a. URL <https://arxiv.org/abs/2606.09186>. 30
- Xiangyun Huang, Xiangchen Wang, Runfeng Lin, Yihao Xu, Kangyu Huang, Hengchen Jiang, Xiwang Dong, and Jiarong Lin. From routes to steps: Separating semantic progress from local execution in vision-and-language navigation. *arXiv preprint arXiv:2608.03143*, 2026b. 2
- InternVLA-N1 Team. InternVLA-N1: An open dual-system navigation foundation model with learned latent plans. Technical report, Shanghai AI Laboratory, 2025. URL https://internrobotics.github.io/internvla-n1.github.io/static/pdfs/InternVLA_N1.pdf. 2
- Kaixuan Jiang, Yang Liu, Weixing Chen, Jingzhou Luo, Ziliang Chen, Ling Pan, Guanbin Li, and Liang Lin. Beyond the destination: A novel benchmark for exploration-aware embodied question answering. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9091–9101, 2025. 8, 12
- Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, pp. 104–120. Springer, 2020. 2, 5, 8, 32
- Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4392–4412, 2020. 2, 5, 8, 12, 32
- Zixing Lei, Gengze Zhou, Xiong-Hui Chen, Jiazhao Zhang, Yiyang Huang, Hang Yin, Haoqi Yuan, Qi Wu, Weixin Li, and Siheng Chen. Scaffolding foundation models into physical-world agents pushes the frontier of long-horizon navigation. *arXiv preprint arXiv:2608.30396*, 2026. 3, 12, 13
- Ruiying Li, Yunlang Zhou, YuYao Zhu, Kylin Chen, Jingyuan Wang, Sukai Wang, Kongtao Hu, Minhui Yu, Bowen Jiang, Zhan Su, Jiayao Ma, Xin He, Yongjian Shen, Yang Yang, Guanghui Ren, Maoqing Yao, Wenhao Wang, and Yao Mu. RoboClaw: An agentic framework for scalable long-horizon robotic tasks. *arXiv preprint arXiv:2603.11558*, 2026. 2

-
- Sihao Lin, Zerui Li, Xunyi Zhao, Gengze Zhou, Liuyi Wang, Rong Wei, Rui Tang, Juncheng Li, Hanqing Wang, Jiangmiao Pang, Anton van den Hengel, Jiajun Liu, and Qi Wu. VLNVerse: A benchmark for vision-language navigation with versatile, embodied, realistic simulation and evaluation. *arXiv preprint arXiv:2512.19021*, 2025. 5
- An Liu, Bingxi Liu, Hongyu Ding, Yixuan Jiang, Yaran Chen, Fulin Tang, Cong Leng, Hong Zhang, and Jian Cheng. HAM-VLN: Harnessing hierarchical agentic memory for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2607.29600*, 2026. 2
- Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. Discuss before moving: Visual language navigation via multi-expert discussions. *arXiv preprint arXiv:2309.11382*, 2023. 2
- Mistral AI. Robostrat Navigate. *arXiv preprint arXiv:2607.20785*, 2026. 2, 12
- Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024. 5
- OpenAI. GPT-6 Astra system card. <https://deploymentsafety.openai.com/gpt-6-astra>, 2026. Released 2026-09-03. 1
- OpenScene Contributors. OpenScene: The largest up-to-date 3D occupancy prediction benchmark in autonomous driving. <https://github.com/OpenDriveLab/OpenScene>, 2023. 5
- Bowen Pan, Rameswar Panda, SouYoung Jin, Rogerio Feris, Aude Oliva, Phillip Isola, and Yoon Kim. Langnav: Language as a perceptual representation for navigation. *arXiv preprint arXiv:2310.07889*, 2023. 2
- Yanyuan Qiao, Wenqi Lyu, Hui Wang, Zixu Wang, Zerui Li, Yuan Zhang, Minghui Tan, and Qi Wu. Open-Nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source LLMs. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. URL <https://arxiv.org/abs/2409.18794>. 2, 8, 32
- Qwen Team. Qwen3.8-Max: A new bar for coding and cowork. <https://qwen.ai/blog?id=qwen3.8>, 2026a. Released 2026-08-03. 1
- Qwen Team. Qwen-RobotNav technical report: A scalable navigation model designed for an agentic navigation system. *arXiv preprint arXiv:2606.18112*, 2026b. 2, 3, 5, 7, 12, 13, 14, 30, 32
- Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 8
- Allen Z. Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until confident: Efficient exploration for embodied question answering. In *Robotics: Science and Systems (RSS)*, 2024. 8, 12
- Saumya Saxena, Blake Buchanan, Chris Paxton, Bingqing Chen, Narunas Vaskevicius, Luigi Palmieri, Jonathan Francis, and Oliver Kroemer. GraphEQA: Using 3D semantic scene graphs for real-time embodied question answering. *arXiv preprint arXiv:2412.14480*, 2024. 12
- Feilong Tang, Xiang An, Yunyao Yan, Yin Xie, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Chunyuan Li, Shikun Feng, Changrui Chen, Huajie Tan, Ming Hu, Manyuan Zhang, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. OneVision-Encoder: Codec-aligned sparsity as a foundational principle for multimodal intelligence. *arXiv preprint arXiv:2602.08683*, 2026. 2, 3

-
- Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, Jan Kautz, Lei Zhang, and Zhiding Yu. LocateAnything: Fast and high-quality vision-language grounding with parallel box decoding. *arXiv preprint arXiv:2605.27365*, 2026a. 6
- Wei Wang, Wenqiao Zhang, Yutong Lin, Yuqian Yuan, Tianwei Lin, Jinhao Mao, Zhenxuan Fan, Mingjian Gao, Yang Dai, Wentong Li, Zheqi Lv, Zheng Dong, Yingjie Niu, Jiaqi Zhu, Jun Xiao, Chao Li, and Yueting Zhuang. EmbodiedSkills: A unified framework for orchestrating, training, and deploying VLA agents. *arXiv preprint arXiv:2609.01281*, 2026b. 2
- Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12009–12020, 2023. 6
- Zun Wang, Jialu Li, Yicong Hong, Songze Li, Kunchang Li, Shoubin Yu, Yi Wang, Yu Qiao, Yali Wang, Mohit Bansal, and Limin Wang. Bootstrapping language-guided navigation learning with self-refining data flywheel. In *International Conference on Learning Representations (ICLR)*, 2025. 6
- Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. *arXiv preprint arXiv:2512.08186*, 2025a. 2
- Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025b. 2, 3
- Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 6
- Xinda Xue, Junjun Hu, Minghua Luo, Shichao Xie, Jintao Chen, Zixun Xie, Kuichen Quan, Wei Guo, Mu Xu, and Zedong Chu. OmniNav: A unified framework for prospective exploration and visual-language navigation. *arXiv preprint arXiv:2509.25687*, 2025. 12
- Tianshuo Yang, Guanyu Chen, Yutian Chen, Zhixuan Liang, Yitian Liu, Zanzin Chen, Chunpu Xu, Haotian Liang, Jiangmiao Pang, Yao Mu, and Ping Luo. HiVLA: A visual-grounded-centric hierarchical embodied manipulation system. *arXiv preprint arXiv:2604.14125*, 2026. 2
- Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3D-Mem: 3D scene memory for embodied exploration and reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17294–17303, 2025. 12
- Mingliang Zhai, Zhi Gao, Yuwei Wu, and Yunde Jia. Memory-centric embodied question answering. *arXiv preprint arXiv:2505.13948*, 2025. 8, 12
- Haochen Zhang, Nirav Savaliya, Faizan Siddiqui, and Enna Sachdeva. FAST-EQA: Efficient embodied question answering with global and local region relevancy. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1664–1673, 2026. 12
- Jiazhaio Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024. 2
- Jiazhaio Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025a. 2, 12

-
- Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks. *Robotics: Science and Systems*, 2025b. [2](#), [3](#)
- Hailin Zhong and Shengxin Zhu. AI Harness Engineering: A runtime substrate for foundation-model software agents. *arXiv preprint arXiv:2605.13357*, 2026. [2](#)
- Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. NavGPT-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision (ECCV)*, pp. 260–278. Springer, 2024a. [2](#)
- Gengze Zhou, Yicong Hong, and Qi Wu. NavGPT: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024b. doi: 10.1609/aaai.v38i7.28597. [2](#)
- Jian Zhou, Sihao Lin, Jin Li, Shuai Fu, Gengze Zhou, and Qi Wu. Automating the design of embodied agent architectures. *arXiv preprint arXiv:2606.30111*, 2026a. [8](#)
- Jian Zhou, Xunyi Zhao, Gengze Zhou, Zerui Li, Sihao Lin, Jiajun Liu, and Qi Wu. Embodied agents take control: Minimal-interface zero-shot agents rival industrial-scale policies in vision-and-language navigation. *arXiv preprint arXiv:2607.26148*, 2026b. [2](#)

A 导航接口实现

章节 3.1.2 在不涉及具体工具名称的情况下描述了导航能力。完整的 NAVGPT-3 配置通过八个工具提供这些能力（表 4）。执行框架（Harness）在执行每次调用前检查工具名称和参数，环境执行相应运动，执行框架则保存路线及已访问地点（节点）的记录。规划器（Planner）无法通过指令添加工具，也不能直接更改导航状态。

A.1 职责划分

图 14 列出了 NAVGPT-3 系统各部分所拥有的状态及其负责的决策。这一划分不依赖运行时中的线程，也不依赖各模型的部署位置。主任务线程将规划器推理、VLA 执行或其他工具调用作为当前操作。规划器上下文由多个来源组装而成：捕获的观测、持续维护的路线记录、格式化的工具结果，以及模型提供方保存的对话历史。

职责	组件与作用
线程协调	运行时：调度线程、管理权限并控制运动权限。
导航决策	规划器：解释证据、选择工具并判断是否到达目标。
上下文构建	执行框架：选择并渲染展示给模型的图像与状态。
持久化导航状态	执行框架：保存路线参照、节点与标注。
工具检查	执行框架：验证请求并执行获准的工具调用。
低层导航	VLA：根据视觉历史预测航点。
物理执行	环境：执行运动并报告感知信息与碰撞。

图 14: NAVGPT-3 系统中的职责划分。

A.2 能力与执行

能力	导航用途	输入与返回内容
本地观测	定位地标并检查当前环境	将前向或全景外观与当前观察方向和位姿关联起来。
指令委派	使用学习得到的策略执行较长路线	将请求的指令传给 VLA，跨调用保留其历史，并返回已执行路线的图像及状态。
路线检查	将当前进展与此前到访的地点关联起来	显示已观测地图和按序排列的已访问地点列表，不将未观测空间标为空闲空间。
空间修正	重访记录地点或纠正局部终点误差	将已访问地点的参照或相对运动请求转换为经检查的运动，并报告实际执行内容。
语义标注	保留对已访问地点的解释	为记录地点附加标签，但不改变其记录位置。
任务终止	接受最终导航状态	明确标识回合结束，并将其与委派轨迹展开（Rollout）的结束区分开。

表 11: 导航能力及其接口。

执行与返回结果。 指令委派作为主任务线程的当前操作运行：它将规划器选择的指令传给 VLA；VLA 跨步骤保留历史，并持续执行预测，直至 VLA 发出停止信号、达到单次委派的步数上限或达到回合上限。VLA 停止只会结束此次委派，而不会结束回合。指令可以是完整任务指令，也可以是子任务指令。已观测地图区分未观测空间和已观测空闲空间，带编号的节点为返回操作提供稳定目标。标注可为现有节点附加名称、说明或指令子句，而不改变记录地图。由于结束回合时可能需要返回所选节点，因此任何返回限制既适用于这一选项，也适用于专用返回工具。完整配置不含运动基元工具、用于调取任意历史图像的工具或目标检测器。

B 运行时状态与时序

本附录扩展了 公式 (1) 至 (3)，将模型可见的执行框架接口与上层运行时执行的线程控制区分开。

B.1 逻辑线程与共享状态

以 j 表示逻辑线程，以 k 表示调用，以 t 表示墙钟时间。共享系统状态 $S(t)$ 由执行框架保存的指令与空间记录、环境观测，以及运行时维护的线程和正在运行的操作记录共同组成。每个线程包含

$$S_j^{\text{thread}}(t) = (\text{role}_j(t), H_j(t), \text{status}_j(t), \mathcal{R}_j(t), \mathcal{W}_j(t)), \quad (6)$$

其中， role_j 表示线程职责， H_j 表示工作历史， status_j 表示生命周期状态， $\mathcal{R}_j, \mathcal{W}_j$ 分别表示获准读取的共享状态和提议写入内容。共享空间历史记录路线与地点，而工作历史记录线程自身的交互。运行时分别跟踪每个工具调用的状态，因此在 VLA 轨迹展开继续或暂停时，主线程可以请求规划器重新评估；工具调用结束并不意味着任务完成。

控制转移规则 σ 响应事件而保留、转移或撤销运动权限 (Motion Authority)。常规转移会撤销前一操作的运动许可，并等待其当前运动步骤完成后才允许新的运动。线程上下文在移交 (Handoff) 后仍会保留，但恢复执行时，过时命令不会重新生效。

B.2 同步实现

在顺序循环中，移交时刻为 t_k ，令 $S_k = S(t_k)$ ，主线程历史为 H_k ，可用工具为 $\mathcal{T}_k$ ，则每个工具调用周期为

$$\begin{aligned} C_k &= \text{Context}_{\text{main}}(S_k, H_k, \mathcal{T}_k), \\ a_k &\sim \pi_{\text{Planner}}(\cdot | C_k, \mathcal{T}_k), \quad (S_{k+1}, E_k) = \text{Execute}(S_k, a_k). \end{aligned} \quad (7)$$

执行框架构建上下文 C_k 、验证调用 a_k 、在获准时执行调用，并返回更新后的状态与执行证据 E_k 。运行时控制线程执行和运动权限；环境负责执行物理运动。VLA 轨迹展开等长时操作在两次规划器调用之间可能跨越多个观测和运动步骤。VLA 执行与规划器直接选择的运动使用相同接口。

B.3 上下文构建与状态更新

对于线程 j 的第 k 次调用，通用上下文与决策接口为

$$C_{j,k} = \text{Context}_j(S|_{\mathcal{R}_j}, H_j, \mathcal{E}_{j,k}, \mathcal{T}_j), \quad a_{j,k} \sim \pi_{j,k}(\cdot | C_{j,k}, \mathcal{T}_j), \quad (8)$$

该接口在调用开始时求值，其中 $\mathcal{E}_{j,k}$ 包含传递给线程的事件， $S|_{\mathcal{R}_j}$ 则将可见状态限制为获准读取的部分。执行框架决定收集和保留哪些信息，并记录每项信息的来源； Context_j 根据线程 j 获准使用的状态与工具组装一次模型请求。由于托管模型提供方保存对话历史，若脱离提供方精确重放请求，还需完整记录每次请求。执行框架维护导航记录，环境执行物理运动。仅当建议更新位于 $\mathcal{W}_j$ 及当前任务和委派权限范围内时，运行时才会接受；拥有目标状态的组件还会检查更新格式和目标。

B.4 运动权限与保护控制

令 $G_o(t) \in \{0, 1\}$ 表示物理操作 o 是否持有运动许可， $g(t)$ 表示权限版本 (Permission Version) 号；每当权限被撤销或替换时，该版本号都会递增。只有前台线程中获准的操作可以持有运动许可 (公式 (3))，每条任务命令 u_o 都携带其下发时对应的版本号。命令仅在以下条件下有效：

$$\sum_o G_o(t) \leq 1, \quad \text{eligible}(u_o, t) = G_o(t) \mathbf{1}[\text{version}(u_o) = g(t)] \mathbf{1}[\text{preconditions}(u_o, S(t))], \quad (9)$$

因此，延迟到达的模型输出或暂停前排队航点无法重新获得权限。保护性停止 (Protective Stop) 标志 $p_{\text{stop}}(t)$ 一经置位便会保持，直至显式清除；低延迟过滤器 Protect 处理最新传感器观测 $O_{\text{phys}}(t)$ 。机器人端的最终命令过滤器输出为

$$u_{\text{robot}}(t) = \begin{cases} u_{\text{stop}}, & p_{\text{stop}}(t) = 1 \text{ or no task command is eligible,} \\ \text{Protect}(O_{\text{phys}}(t), u_{\text{task}}(t)), & \text{otherwise.} \end{cases} \quad (10)$$

检测到障碍物或控制环路超时，无需等待规划器即可置位 p_{stop} 并撤销当前权限；恢复则要求清除停止状态、刷新状态并授予新权限。只有规划器能够判定已到达目标；无论规划器如何判定，保护路径都可以阻止运动。

B.5 调度与交互频率

不同活动可按不同更新周期并由不同事件触发运行，类似全双工和异步交互-思考模型 (Défossez et al., 2024; Huang et al., 2026a)。该运行时设计允许规划器在 VLA 持续执行期间审查当前状态的固定快照，仅当审查要求修正或终止时才中断。每个 VLA 步骤预测八个航点并执行第一个；丢弃的预测和失败的重试均不计入步骤数。在 Go2 上测量并发审查的收益 (章节 4.8)。若单步时长恒为 P_{VLA} ，每隔 F 步检查一次，则 FP_{VLA} 为两次检查之间的执行时间；总耗时还包括为审查而暂停的时间。

B.6 优先级与移交策略

运行时在创建活动为其分配优先级，由高到低依次为：保护性停止与控制环路超时、回合限制、操作完成以及路线审查请求。每个事件都携带对应操作、权限版本、时间戳和序列号；优先级相同的事件按到达顺序处理。过时的审查或完成事件会被丢弃，针对同一操作的重复审查请求会合并为一个待处理请求。保护性停止会立即撤销运动权限并递增版本号；推理可以继续，但延迟到达的模型响应和排队中的航点都无法凭旧权限驱动机器人。仅清除警报不会恢复运动，还需要最新观测和新权限。回合限制同样会撤销运动权限；在其他情况下，自行完成的操作优先于同时到达的定时审查。若审查允许继续当前路线，则保留 VLA 历史；若需要修正，则先等待当前运动步骤完成，再允许新操作。

该策略描述了完整的运行时设计。用于消融实验的模拟器实现更为简化：每次只执行一个运动操作，并实现版本检查、自行完成操作的优先级、单个待处理审查请求，以及保留 VLA 历史的继续执行。独立安全监视器的抢占及其物理时序在 Go2 上测量 (章节 4.8)。

B.7 交互频率评估

检查调度由 章节 B.5 定义。所有频率设置均使用相同的规划器、VLA、提示词、工具、路线摘要格式及恢复流程，并保留 VLA 历史；检查可以允许当前路线继续，而不选择恢复动作。执行循环还会在连续两次移动受阻后记录审查请求；此类请求会在下一次定时检查时或 VLA 自行停止时交付，不会触发额外检查，因此频率对比不含事件触发的审查。请求不能更改导航状态或结束回合；在允许替代运动前，运行时会撤销 VLA 的运动许可。成本包括所有模型调用和等待时间；周期性检查与 VLA 自行停止时发生的返回分别计数。表 12 总结了各组件在移交时可以执行的操作。

表 12: 执行中断时各组件可执行的操作。

操作	规划器	VLA	强制执行方
中断正在执行的 VLA 轨迹	获准时可提出请求	完成当前步骤后停止	运行时：撤销运动权限
请求规划器提前复核	获准时接收请求	继续执行	运行时：将请求排入队列或丢弃
恢复或修订路线	选择可用工具	执行有效委派	运行时：向一项操作授予运动权限
更改物理状态或路线状态	无法直接更改	提议航点	环境：执行运动；执行框架：更新路线记录
结束回合	明确决定停止	其停止操作仅结束当前委派	运行时：强制执行停止与回合限制
在规划器与 VLA 间共享信息	仅发送结构化指令	仅返回结构化状态与路线证据	运行时：控制可见性；执行框架：构建上下文

C 视觉上下文分配细节

基线与编解码分配规则。 位置基线重新实现了 Qwen-RobotNav 的观测接口 (Qwen Team, 2026b)：时间权重和相机权重在保留图像之间分配共享预算，每幅图像再缩放至分配到的像素预算。按从最早观测开始计数， T 个保留观测中第 t 个观测的时间权重为 $\exp(\gamma t / (T - 1))$ ，因此 $\gamma \geq 0$ 控制偏向近期观测的程度。编解码分配 (Codec Allocation) 加入了 公式 (4) 定义的 RGB 变化乘数，同时沿用相同的分配器与图像缩放路径。该分数衡量视觉变化，包括相机运动和光照引起的变化；它既不是学习得到的语义重要性估计，也不是光流估计器。

表 13: 分配饱和。确定性合成变化分数诊断。

B_{vis}	N_{hist}	N_{view}	b_{max}	ρ_{vis}	发生变化的 图像数	平均 $ \Delta b $ (词元)
容量饱和区						
3072	2	1	196	7.84	0 / 2	0.0 饱和
3072	8	1	196	1.96	0 / 8	0.0
3072	15	1	196	1.04	0 / 15	0.0
4096	4	4	256	1.00	0 / 16	0.0
3072	16	1	196	0.98	3 / 16	8.0
3072	8	4	196	0.49	30 / 32	47.3
1024	16	1	196	0.33	14 / 16	31.4
2048	8	4	196	0.33	31 / 32	18.9
1024	16	4	196	0.08	57 / 64	7.9

整数分配。 将保留历史按时间优先、前/右/后/左的顺序展平。对于 $M = N_{\text{hist}}N_{\text{view}}$ 幅图像，将每幅图像的初始分配设为 b_{min} ，并要求 $Mb_{\text{min}} \leq B_{\text{vis}}$ 。剩余预算按正联合权重成比例分配，对各份额向下取整，将追加量限制在 b_{max} ，再把溢出部分重新分配给尚未达到上限的图像。最后的整数余量按权重降序分配；先填满一幅图像的剩余容量，再移至下一幅。权重相同时，按固定排序顺序打破平局。分配的词元数不会超出图像容量：当 $B_{\text{vis}} \geq Mb_{\text{max}}$ 时，所有图像均获得 b_{max} ，多余预算不再使用。图像缩放保持宽高比，并将尺寸舍入至编码器的图像块/合并网格，因此实际图像词元数可能低于分配上限。

评估配置。 评估时，四视角 VLA 使用 $B_{\text{vis}} = 3072$ 、 $\gamma = 2$ ，前/右/后/左相机权重依次为 (2, 1, 0.5, 1)，且 $b_{\text{min}} = 4$ 、 $b_{\text{max}} = 196$ 。系统最多保留 16 个观测，始终包括首个和当前观测；其余索引从中间历史中均匀无放回采样，并按时间顺序排列。在这些设置下，不超过三个观测的短历史会达到单图像上限；从四个观测开始，分配受总预算限制。这些评估参数不同于随机化训练预算，也不同于下述合成检查的设置。

表 13 是对分配器的合成检查，采用 $b_{\text{min}} = 4$ 、 $\gamma = 2$ 、上述相机权重（单视角时权重为一），以及各行给定的预算和上限。变化分数在固定随机种子下从 $[0.05, 1)$ 均匀采样，每个视角的首幅保留图像分数设为一。表中报告了两种规则下分配结果不同的图像数量，以及单幅图像的平均 $|\Delta b|$ ，其中 $\Delta b = b^{\text{codec}} - b^{\text{pos}}$ 。所有整数边界和分配总量均满足约束；在两种规则理应一致的四个饱和行中，分配结果完全相同。我们将发布分数数组、分配结果和分配器代码。这些是合成计算，并非导航结果；导航性能在图 7 中评估。

D 补充实验细节

D.1 评估设置

表 14: 执行框架评测设置。

设置	取值
RGB 分辨率	512 像素
相机顺序	前、右、后、左
返回的路线关键帧	最多 8 帧，沿执行轨迹均匀采样
回合/委派上限	500/200 个环境步
规划器调用上限	每回合 200 轮
回合超时	2,400 s

基准目标距离和成功标签仅供离线评估器使用。

D.2 RxR-CE 上的人类表现

RxR 要求智能体沿路线指令前往终点。人类演示使用离散图边，抽象掉低层运动和避障；RxR-CE 则通过连续运动实现导航 (Ku et al., 2020; Krantz et al., 2020)。尽管在相同指令下承担了额外控制负担，本文达到的 90.43 成功率 (Success Rate, SR) 和 78.47 nDTW 仍与英语人类跟随者的表现相当 (90.4 SR 和 77.7 nDTW)。

D.3 跨任务评估

目标物体子集按目标类别和场景比例抽取。参照值采用 Qwen Team (2026b) 的表 4。由于比例采样使最稀有类别各自仅有约两个回合，因此不报告 MP3D 的分类结果。问答指标采用 Qwen Team (2026b) 的表 7；以往 EXPRESS-Bench 分数使用了现已弃用的 GPT-4o-mini 评判器。

D.4 规划器脚手架与统计口径

本文将各脚手架的默认任务提示词替换为 NAVGPT-3 导航提示词，后者描述机器人、指令、工具、运动预算和停止规则，并提供相应导航工具。脚手架自身的任务提示词不再使用，所有报告的规划器结果都由在脚手架内运行的 NAVGPT-3 执行框架生成。脚手架仍负责组装对话和管理上下文窗口，包括提供方执行的任何自动压缩。执行框架依据运行时的可见性权限，决定哪些观测、路线证据和空间状态进入这些消息。因此，替换任务提示词并不会使 Codex 与 Claude Code 完全相同：二者的系统指令、内置工具以及工具结果格式仍可能不同。单个规划器内部的消融实验固定脚手架和提示词；跨规划器对比描述的是完整的“规划器-脚手架”组合，而非孤立地比较底层模型。

词元数 (Tokens, k) 统计所有请求中已处理的输入词元 (包括缓存输入) 与生成的输出词元之和，再除以 1,000。轮次数统计由环境反馈分隔的规划器响应，包括初始响应和最终确认；一次响应中的文本、推理及并行工具调用合计为一轮。成本指每个尝试回合的托管推理总费用，以美元计，包括重试费和缓存费。VLA 延迟单独报告，不计入价格。Opus 成本采用提供方 SDK 报告的数值；Astra 成本则按记录的词元用量和标价计算。推理计时不含通信与执行框架开销。表 3 中的动作更新间隔，在不使用 VLA 时指规划器的单轮时间，使用 VLA 时指每步 VLA 推理时间；其中不含中断信号到达机器人或机器人制动所需的时间。

D.5 视觉提示词变体

所有变体均保持相同的八个工具、相机帧、路线关键帧、位姿读数、节点 ID、标注、已观测几何结构和工具行为。具身参照方案 (Grounded Reference) 以 5×7 像素字体在图像上绘制标签，显示地图时令地图坐标系的北向朝上，并给出航向箭头及相对于智能体航向的转向提示；这里的北向指地图坐标系，而非地理北方。紧凑字形采用高度和对比例相同的 3×5 像素数字字体；航向朝上的地图保持相同裁剪范围和比例；绝对方位角提供足以推导相同方向的方位与位姿信息。标签始终位于场景内容之外，因此所有变体的遮挡情况均不改变。

E 评估覆盖范围与验证

E.1 R2R 消融子集

包含 100 个回合的消融集采用公开的 OpenNav 子集 (Qiao et al., 2025)。本文运行的全部 100 个回合在回合标识符、指令和场景方面均与已发布子集一致；每个回合的指令文本、场景、起始位置、参考路径和测地距离也都与 R2R-CE v1.3 Val-Unseen 版本一致。表 15 将这些回合属性与包含 1,839 个回合的完整划分进行了比较。数值项给出均值 $\pm$ 标准差。绝对标准化均值差异 (Standardized Mean Difference, SMD) 比较该子集与其互不重叠、包含 1,739 个回合的补集，分母采用二者等权方差的平方根； D 表示该子集与完整划分的经验累积分布之间的最大差异。方向词提及次数统计 left、right 和 turn 的各种形式；参考路径转向指至少 30° 的水平方向变化，而非实际执行的运动基元转向。这些指标是可解释的任务复杂度代理，并非完整的语义复杂度度量。

路径距离和方向词提及次数的均值差异较小；子集中的指令更短，长度分布也更集中。该子集覆盖 11 个场景中的 10 个，遗漏的场景在完整划分中包含 18 个回合；子集与完整划分之间的场景频率总变差距离为 0.097。具体而言，某个场景占子集的 3%，而在完整划分中

表 15: R2R 消融子集的覆盖情况与分布。

统计量	完整集	子集	SMD	D
回合数	1,839	100	—	—
场景数	11	10	—	—
唯一参考轨迹数	613	97	—	—
测地距离 (m)	8.90 ± 2.67	8.99 ± 3.06	0.033	0.072
指令长度 (词)	26.65 ± 11.48	24.68 ± 8.28	0.207	0.103
方向词出现次数	2.14 ± 1.80	2.20 ± 1.75	0.034	0.040
参考路径转向次数	1.56 ± 1.08	1.46 ± 1.21	0.095	0.100

占 9.63%。子集场景频率的 Pearson χ^2 统计量为 8.14；在无放回抽取的 10,000 个随机 100 回合子集中，56.7% 的统计量不低于该值。因此，观测到的不均衡程度在同等规模的随机子集中较为典型，但这并不能证明该子集具有代表性，特别是该子集经过了人工筛选。公开 OpenNav 文件与 v1.3 文件在所选全部 100 个回合的初始航向上也存在差异，本文在评估中予以修正。因此，该对比只能说明所列属性的覆盖范围和相似性，不能证明完整执行协议或性能等价。基准对比仍以完整划分结果为依据。

E.2 匹配的 OBJECTNAV 评估

表 16 中的验证在每个数据集上使用完全相同的 500 个回合运行 Qwen-RobotNav 8B 和 NAVGPT-3，二者采用相同的场景版本、感知设置、运动预算、碰撞处理以及停止/成功规则。这是系统层面的对比；完整系统还额外使用托管规划器。在观察任一系统的结果前，所选回合即已固定。SR 和 SPL 均以百分比表示。 $\Delta\text{SR vs. full}$ 指 500 回合 SR 减去 Qwen-RobotNav 8B 已发布的完整划分 SR，以百分点计（MP3D 上为 -1.0 ，HM3D v2 上为 $+1.2$ ）；该指标反映所选回合与完整划分的接近程度。NAVGPT-3 没有完整划分运行结果，因此对应项以短横线表示。

表 16: 相同回合上的 ObjectNav 对比。

数据集	系统	N	SR	SPL	相对完整集的 ΔSR
MP3D	Qwen-RobotNav 8B	500	47.8	17.5	-1.0
MP3D	NAVGPT-3	500	59.2	25.2	—
HM3D v2	Qwen-RobotNav 8B	500	72.4	33.1	$+1.2$
HM3D v2	NAVGPT-3	500	75.3	40.8	—

E.3 EXPRESS 评判器验证

本文在相同的已保存答案和最终观测上，将 GPT-6 Astra 评判器与以往 EXPRESS-Bench 结果所用的 GPT-4o-mini 评判器进行比较，并以人工判断为准评分。表 17 报告了二者在 50 个带人工标签答案上的一致情况：Astra 有 46 个与人工标签一致，GPT-4o-mini 有 36 个一致，相差 20.0 个百分点。若缺少逐项配对信息，精确 McNemar p 值取决于两个评判器错误项的重叠情况；按上述总数，其范围为 0.002 至 0.031。该问题适用于 EXPRESS 的开放式答案评分；HM-EQA 和 MT-HM3D 的选择题准确率采用参考选项判定。

表 17: EXPRESS 评判器与人工标注的一致性。

评判器	标注数 N	一致率 (%)
GPT-4o-mini	50	72.0
GPT-6 Astra	50	92.0
Astra — 4o-mini		$+20.0$