030Cˇ
/TemplateVersion (2026.1)

CorrectNav: 自纠飞轮赋能视觉-语言-动作导航模型

Zhuoyuan Yu[*12], Yuxing Long[*†12], Zihan Yang[12], Chengyan Zeng[2], Hongwei Fan[12], Jiyao Zhang[12], Hao Dong[‡12]

[1]CFCS, School of Computer Science, Peking University, [2]PKU-Agibot Lab

*Equal contribution, † Project Leader, ‡ Corresponding author

https://correctnav.github.io

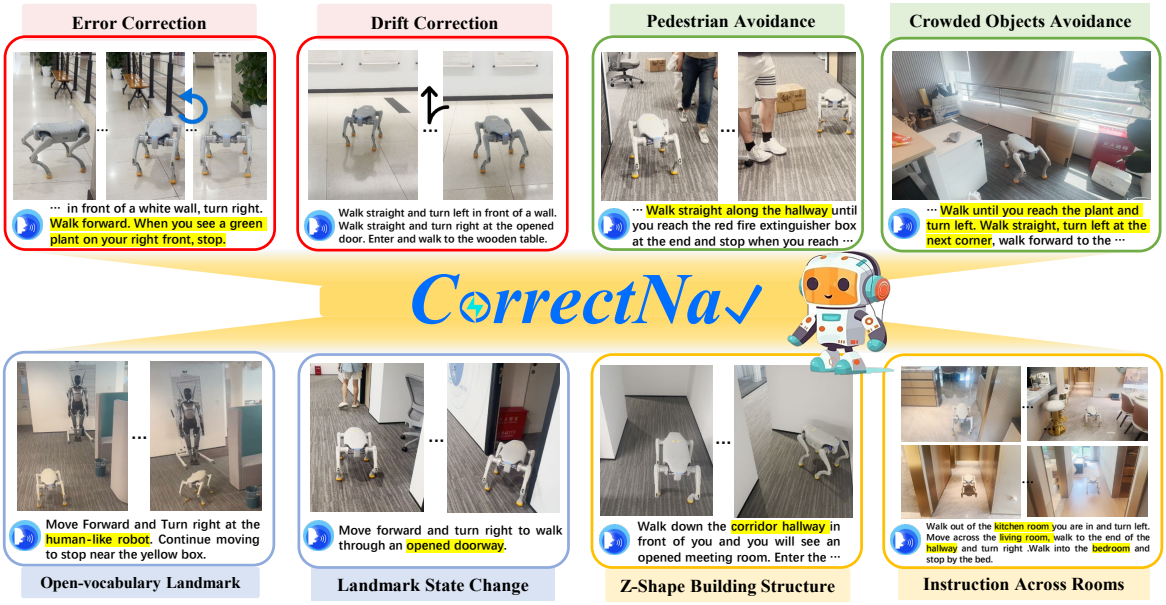

Figure 1: Diverse Capabilities of CorrectNav. The model takes only monocular RGB video and language instructions as inputs, predicting navigation actions. Empowered by the Self-correction Flywheel post-training, CorrectNav not only maintains outstanding multimodal reasoning (Blue), but also displays improved deviation correction (Red), obstacle avoidance (Green), and complex action execution (Yellow).

Abstract

现有的视觉与语言导航模型在执行指令时常常偏离正确的路径。然而，这些模型缺乏有效的错误修正能力，阻碍了其从错误中恢复。为了解决这一挑战，我们提出了自我校正飞轮，这是一种新颖的后训练范式。我们的范式不将模型在训练集上的错误轨迹视为弊端，而是强调其作为宝贵数据源的重要性。我们开发了一种方法来识别这些错误轨迹中的偏差，并设计了创新技术来自动生成用于感知和行动的自我校正数据。这些自我校正数据为模型的持续训练提供了动力。当我们在训练集上重新评估模型时，我们的范式的闪光点显现出来，揭示了新的错误轨迹。此时，自我校正飞轮开始旋转。通过多次飞轮迭代，我们逐步增强了我们基于单目 RGB 的 VLA 导航模型 CorrectNav 。在 R2R-CE 和 RxR-CE 基准测试中的实验表明，CorrectNav 达到了 65.1 % 和 69.3 % 的最新的最先进成功率，比以前最佳的 VLA 导航模型超出了 8.2 % 和 16.4 %。在各种室内和室外环境中的真实机器人测试展示了 CorrectNav 在错误修正、动态障碍物规避和长指令跟随方面的卓越能力。

在视觉与语言导航（VLN）任务中，用户通过自然语言指令控制机器人在未知环境中移动到期望的位置，比如“前进并右转进入客厅，在沙发附近等待。”由于其用户友好的交互特性，VLN 成为体现智能的基本能力，并吸引了广泛的研究兴趣。在导航过程中，模型不可避免地会预测错误的移动动作，导致机器人偏离正确路径。这些偏差通常会导致环境与指令之间的不匹配。以上述指令为例。如果机器人在当前位置直接右转而不是先前进，它将进入厨房，无法定位沙发。这时，机器人很容易因这种不匹配而感到困惑，无法到达目的地。

现有的 VLN 模型主要通过改进特征表示 (An et al. 2024; Hong et al. 2023) 或增加训练数据 (Zhang et al. 2024a, 2025) 来增强视觉感知和多模态推理能力。他们旨在让模型在每一步中尽可能正确地导航。然而，现实证明与期望有所不同。仅仅几个不完善的逐步预测就能累积显著偏离正确路径，最终导致失败。缺乏自我纠正能力使得先前的 VLN 模型在错误发生时难以从错误中恢复并重新回到正轨，从而限制了它们的整体导航性能。这一缺陷提出了一个重要问题 - 我们能否教会机器人在导航过程中自我纠正错误?

针对这个问题，我们分析需要纠正哪些类型的错误

以及如何训练导航模型去纠正这些错误。作为一个视觉-语言-动作（VLA）任务，VLN 要求模型动态地感知环境并依据给定的指令进行导航。错误通常来自两个方面：地标的误解和指令所指定动作的误解。这些错误在决策过程中传播，最终对运动预测产生负面影响。因此，应该关注那些来源于感知和动作的错误。此外，现实世界的应用对模型推理有时间要求，这要求自我纠正能力应通过训练隐式地整合到模型中，而不是通过增加模块或推理过程来实现。

因此，我们提出了一种新的导航训练后范式，称为自我校正飞轮。这种方法源于我们的观察，即使经过良好训练的导航模型在训练集上进行评估时，仍会产生错误轨迹。我们并不将这些错误仅仅视为缺陷，而是将其看作进一步增强模型的宝贵机会。我们的自我校正飞轮通过以下四个步骤进行：（1）在训练集上评估训练好的模型以收集错误轨迹。（2）然后，我们设计了一种能够检测偏差并准确定位错误轨迹中偏差位置的自动方法。（3）识别偏差后，我们从动作和感知的角度创建自我校正数据。对于动作校正，我们收集展示有效纠偏的轨迹。对于感知校正，我们利用大规模多模态模型来分析与导航错误相关的关键帧。（4）利用这些自我校正数据，我们推动导航模型的持续训练以提高其性能。完成上述四个步骤即构成了自我校正飞轮的一个回合。当我们继续在训练集上评估已经历一个回合自我校正训练的模型时，一件显著的事情发生了。我们可以识别新的错误轨迹，从而生成新的自我校正数据并进一步训练模型。这时，自我校正飞轮开始运转，随着多轮训练迭代的进行，导航模型的性能将持续提升。

此外，我们设计了一套导航微调策略，包括观察随机化、指令生成和通用多模态数据回忆。通过我们提出的微调和后训练策略，我们开发了一种新的基于单目 RGB 的 VLA 导航模型 CorrectNav 。在 VLN-CE 基准 R2R-CE 和 RxR-CE 上，CorrectNav 达到了 65.1 % 和 69.3 % 的成功率，超过了之前最新的模型 8.2 % 和 16.4 %。在各种室内和室外环境中进行的真实机器人测试表明，CorrectNav 具有强大的纠错能力、动态避障能力和长指令跟踪能力，优于现有的导航模型。

相关工作

视觉与语言导航

视觉与语言导航（VLN）涉及到一个具体现体代理根据自然语言指令导航到目标位置。像 R2R（Anderson et al. 2018）和 RxR（Ku et al. 2020）这样的数据集在离散的 MP3D（Chang et al. 2017）环境中提供导航指令和轨迹，而 VLN-CE（Krantz et al. 2020）将这些适配到连续环境。当前的 VLN-CE 模型可以分为两类：基于拓扑图的方法，如 BEVbert（An et al. 2023）和 ETPnav（An et al. 2024），依赖于多个传感器来预测路径点；以及基于预训练视觉-语言模型（VLMs）构建的模型，包括 NaVid（Zhang et al. 2024a）、Uni-NaVid（Zhang et al. 2025）和 NAVILA（Cheng et al. 2024），它们根据 RGB 观察端到端推断动作。现有方法常用辅助任务（Zhang et al. 2025, 2024a）、指令增强（Wei et al. 2025b）和数据集扩展（Wei et al. 2025a）等技术来提升性能，但对错误纠正关注较少。为了更容易应用于实际机器人，我们也基于预训练 VLM 构建了我们的 CorrectNav 。然而，我们强调错误纠正的价值，这有助于我们突破当前技术的性能瓶颈。

体现智能中的错误校正

错误在具身智能任务中通常是不可避免的。为增强鲁棒性，纠正错误的能力是必不可少的。在操作任务中，错误纠正方法已经被探索（Ha, Florence, and Song 2023; Ma et al. 2023; Duan et al. 2024; Liu, Bahety, and Song 2023）。然而，在导航任务中的错误纠正则较少被探讨。SmartWay（Shi et al. 2025）利用闭源的大型模型来反思轨迹并决定是否回溯，而 EnvolveNav（Lin et al. 2025）训练模型生成耗时的思维链，改进有限。这些方法通常需要额外的模型或推理步骤，降低了效率并阻碍在现实世界中的部署。相比之下，我们的方法通过自我纠正飞轮训练隐式地教授错误纠正，无需额外的模块或冗长的思考，从而助于真实机器人的部署。

已知语言指令 L_{nav} ，视觉与语言导航任务要求模型在时间步 t 基于观察 $\{O_1, O_2, \ldots, O_t\}$ 预测下一个导航动作 $a_{t+1} \in A$ 。最近，为了克服对多传感器的依赖，研究人员 (Zhang et al. 2024a) 将观察简化为在导航过程中捕获的一系列单目 RGB 图像 $\{I_1, I_2 ... I_t\}$ 。

Algorithm 1: 自我纠正飞轮后训练

Input: oracle trajectories $\{T_g^{(i)}, L_{nav}^{(i)}\}$, dataset D_{nav} , model $\mathcal{M}$, number of flywheel iteration N , distance threshold S , trajectory planner Γ
Output: Model $\mathcal{M}$

$\mathcal{M} \leftarrow \text{Train}(D_{nav}, \mathcal{M})$;
for $c \leftarrow 1$ **to** N :
  $\{T_m^{(i)}\} \leftarrow \mathcal{M}(\{L_{nav}^{(i)}\})$;
  $D_{new} \leftarrow \emptyset$;
  for each sample i in the dataset:
    $K^{(i)}, T_c^{(i)} \leftarrow \text{DeviDetect}(T_g^{(i)}, T_m^{(i)}, S, \Gamma)$;
    $\text{Cap}^{(i)} \leftarrow \text{MLLM\_Description}(K^{(i)})$;
    $\text{Qa}^{(i)} \leftarrow \text{MLLM\_QA}(K^{(i)})$;
    Add $(T_c^{(i)}, \text{Cap}^{(i)}, \text{Qa}^{(i)})$ to D_{new} ;
  $D_{train} \leftarrow \text{Sample}(D_{nav}) \cup \text{Sample}(D_{new})$;
  $\mathcal{M} \leftarrow \text{Train}(D_{train}, \mathcal{M})$;

我们的 CorrectNav 由三个模块组成：视觉编码器 $v(\cdot)$ 、投影器 $p(\cdot)$ 和大型语言模型（LLM）$f(\cdot)$ 。具体来说，我们使用 SigLIP（Zhai et al. 2023）、一个两层的 MLP（Liu et al. 2024）和 Qwen2（Yang et al. 2024）。给定一个 RGB 视频，视觉编码器从采样帧中提取视觉特征，生成 $Z_v = v(\{I_1, I_2 ... I_t\})$ 。MLP 投影器将这些视觉特征映射到 LLM 的语义空间，得到一系列视觉标记 $H_v = p(Z_v)$ 。使用视觉标记 H_v 和从任务指令 L 编码的文本标记 X ，LLM $f(\cdot)$ 以自回归方式进行预测。在导航微调之前，CorrectNav 从 LLaVA-Video 7B（Zhang et al. 2024b）初始化。

导航微调

导航动作预测 我们从 VLN-CE R2R 和 RxR 训练样本中收集 MP3D 室内场景的 oracle 导航轨迹。每条 oracle 轨迹包含一个导航指令、逐步的 RGB 观察和导航动作 $\tau = (L_{nav}, \{(I_t, a_t)\}_{t=1}^{T})$ 。为了增强视觉多样性，我们实施了一套域随机化策略。这些策略包括随

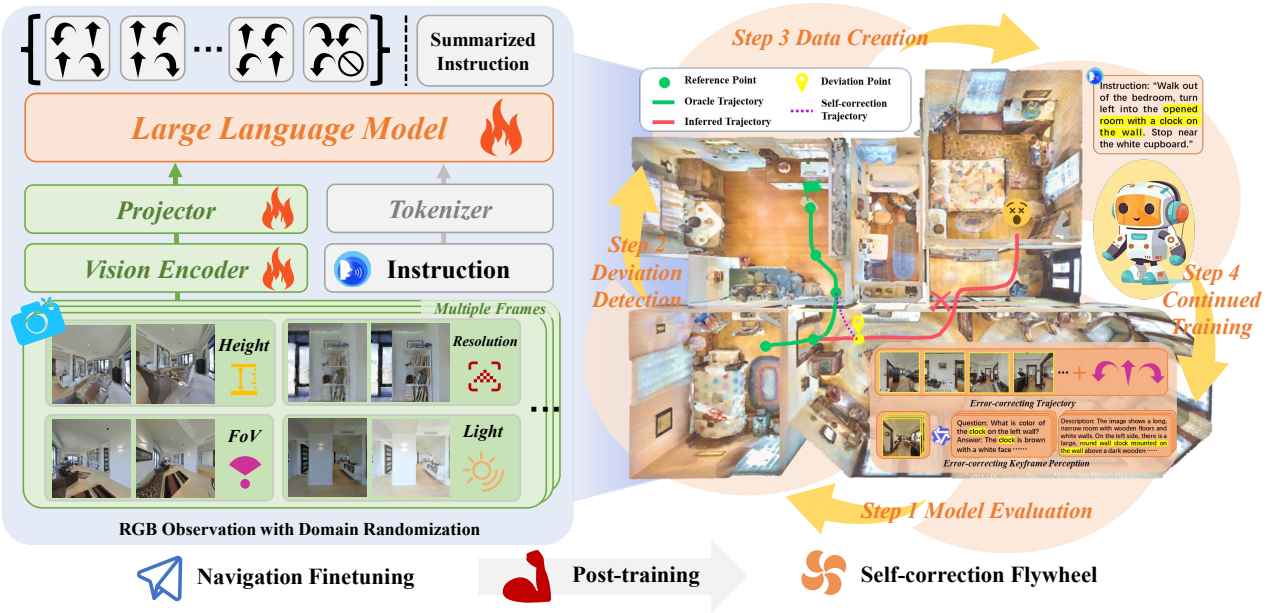

Figure 2: CorrectNav 训练的概述。首先在导航任务（左侧）上微调 CorrectNav ，包括动作预测和指令生成。为了增强视觉多样性，我们实施了一套领域随机化策略。随后，使用我们提出的自我纠正飞轮范式对 CorrectNav 进行后训练（右侧）。这个范式在模型评估、偏差检测、数据创建和持续训练的循环中运行。具体而言，数据创建部分可以自动收集纠错轨迹和关键帧感知数据。通过多次训练迭代，CorrectNav 可以学习如何从偏差中恢复。

机化相机高度、调整视野、改变观察分辨率和改变光照条件，如图所示 2 。通过这些策略，我们收集了超过 210 万步的导航动作预测数据 D_{nav} ，包括 527K 个来自 R2R 的样本和 158 万来自 RxR 的样本。在这个任务中，我们采取导航指令 L_{nav} 和逐步的 RGB 观察 $\{I_1, I_2...I_t\}$ 作为 CorrectNav 的输入，并要求模型预测一个包含 m 步的动作干 $\{a_{t+1}, a_{t+2}...a_{t+m}\}$ 。

基于轨迹的指令生成 在这个任务中，我们从 VLN-CE R2R 和 RxR 数据集中收集完整的 oracle 导航轨迹。在这些轨迹中，10K 来自 R2R 而 20K 来自 RxR。CorrectNav 需要根据单目 RGB 观察历史生成语言格式的导航指令。在训练期间，我们输入整个 oracle 轨迹 $\{I_1, I_2...I_T\}$ 的 RGB 观察，并将对应的指令 L_{nav} 作为目标。

我们下游导航任务的格式与一般多模态训练任务有显著不同。仅在导航任务上训练会导致训练过程中一般多模态能力的遗忘。为了解决这个问题，我们包含了来自 LLaVA-Video 178K 数据集中一部分的视频数据。我们专注于 Activitynet-QA 和 NextQA，这些任务强调时间和空间场景理解，与我们的目标一致。因此，我们从 ActivityQA 和 NextQA 中随机抽取 240K 训练实例，以保持模型的一般多模态能力。

为了教导航模型如何从偏差中恢复，我们提出了一种新的后训练范式，自我纠正飞轮。训练的一个迭代包括模型评估、偏差检测、自我纠正数据创建和持续训练。这四个步骤可以在两个端形成一个闭环，创造一个自我纠正的飞轮。通过多次训练迭代，可以特别地提高自我纠正能力。概述在算法 1 中介绍。每个步骤将在下面详细说明。

R2R-CE 和 RxR-CE 的数据集的训练集提供了大量的指令和 oracle 轨迹对。在数据集中，每个 oracle 轨迹由一系列顺序的参考点定义，记为 $T_g = (G_1, \ldots, G_n)$ 。在导航微调过程中，我们已经使用这些数据为 CorrectNav 训练提供逐步的监督信号。虽然模型已经在这些数据上进行了训练，但我们发现在训练集上进行评估时它仍然会出错。我们意识到这是收集纠正数据的绝佳来源。训练数据集不仅包含丰富的数据，还包括真实的参考点。因此，我们收集了模型在训练集上评估时产生的错误轨迹。可以将这些轨迹表示为 $T_m = (M_1, \ldots, M_m)$ ，其中 M_i 表示机器人在 i 时刻的位置。

由于收集的误差轨迹缺乏标注来指示偏差发生的位置，我们开发了一种方法来检测这些偏差。关键原则是通过测量误差轨迹与基准轨迹之间的距离来评估偏差。为了计算机器人位置 M_i 到基准轨迹 T_g 的距离，我们首先在参考点之间进行均匀插值，形成一个均匀间隔的序列 T'_g 。对于每个机器人位置 $M_i \in T_m$ ，我们定义从 M_i 到 T_g 的距离为

$$h_i = \min_{x \in T'_g} \|M_i - x\|_2$$

我们进一步定义 M_i 在 T_g 上的正交落脚点为

$$P_i = \arg\min_{P \in T'_g} \|M_i - P\|_2.$$

设 S 为预定义阈值。如果存在一个时间步 t 使得

$$h_t > S \quad \text{and} \quad h_i \leq S, \quad \forall i < t, i \in N^*$$

那么我们宣称模型在时间步 M_t 开始偏离基准轨迹。在时间步 t 附近的观测可以被标记为误差修正的关键帧。

通过分析误差轨迹的偏差，我们确定导航错误主要源于感知和行动。因此，我们提出了纠正这两个方面的自我修正任务和数据创建方法。

纠错轨迹为了教模型如何从偏差中恢复，我们基于检测到的偏差收集纠错轨迹。给定一个 oracle 轨迹 T_g 和带有偏差的模型轨迹 T_m ，我们已经在步骤 2 中检测出

	Observation				R2R-CE Val-Unseen				RxR-CE Val-Unseen			
	S.RGB	Pano.	Depth	Odo.	NE ↓	OS ↑	SR ↑	SPL ↑	NE ↓	SR ↑	SPL ↑	nDTW ↑
HPN+DN * (Krantz et al. 2021)		✓	✓	✓	6.31	40.0	36.0	34.0	-	-	-	-
CMA * (Hong et al. 2022)		✓	✓	✓	6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0
VLN ↻ BERT * (Hong et al. 2022)		✓	✓	✓	5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7
Sim2Sim * (Krantz and Lee 2022)		✓	✓	✓	6.07	52.0	43.0	36.0	-	-	-	-
GridMM * (Wang et al. 2023c)		✓	✓	✓	5.11	61.0	49.0	41.0	-	-	-	-
Ego ^{2} -Map * (Hong et al. 2023)		✓	✓	✓	5.54	56.0	47.0	41.0	-	-	-	-
DreamWalker * (Wang et al. 2023a)		✓	✓	✓	5.53	59.0	49.0	44.0	-	-	-	-
Reborn * (An et al. 2022)		✓	✓	✓	5.40	57.0	50.0	46.0	5.98	48.6	42.0	63.3
ETPNav * (An et al. 2024)		✓	✓	✓	4.71	65.0	57.0	49.0	5.64	54.7	44.8	61.9
HNR * (Wang et al. 2024)		✓	✓	✓	4.42	67.0	61.0	51.0	5.50	56.3	46.7	63.5
BEVBert * (An et al. 2023)		✓	✓	✓	4.57	67.0	59.0	50.0	-	-	-	-
HAMT+ScaleVLN * (Wang et al. 2023b)		✓	✓	✓	4.80	-	55.0	51.0	-	-	-	-
AG-CMTP (Chen et al. 2021)		✓	✓	✓	7.90	39.0	23.0	19.0	-	-	-	-
R2R-CMTP (Chen et al. 2021)		✓	✓	✓	7.90	38.0	26.0	22.0	-	-	-	-
InstructNav (Long et al. 2024)		✓	✓	✓	6.89	-	31.0	24.0	-	-	-	-
LAW (Raychaudhuri et al. 2021)	✓		✓	✓	6.83	44.0	35.0	31.0	10.90	8.0	8.0	38.0
CM2 (Georgakis et al. 2022)	✓		✓	✓	7.02	41.0	34.0	27.0	-	-	-	-
WS-MGMap (Chen et al. 2022)	✓		✓	✓	6.28	47.0	38.0	34.0	-	-	-	-
AO-Planner (Chen et al. 2024)		✓	✓		5.55	59.0	47.0	33.0	7.06	43.3	30.5	50.1
Seq2Seq (Krantz et al. 2020)	✓		✓		7.77	37.0	25.0	22.0	12.10	13.9	11.9	30.8
CMA (Krantz et al. 2020)	✓		✓		7.37	40.0	32.0	30.0	-	-	-	-
RGB-Seq2Seq (Krantz et al. 2020)	✓				10.10	8.0	0.0	0.0	-	-	-	-
RGB-CMA (Krantz et al. 2020)	✓				9.55	10.0	5.0	4.0	-	-	-	-
NaVid (Zhang et al. 2024a)	✓				5.47	49.0	37.0	35.0	-	-	-	-
Uni-NaVid (Zhang et al. 2025)	✓				5.58	53.5	47.0	42.7	6.24	48.7	40.9	-
NaVILA (Cheng et al. 2024)	✓				5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8
StreamVLN (Wei et al. 2025a)	✓				4.98	64.2	56.9	51.9	6.22	52.9	46.0	61.9
CorrectNav (Ours)	✓				4.24	67.5	65.1	62.3	4.09	69.3	63.3	75.2

Table 1: 在 R2R-CE (Anderson et al. 2018) 和 RxR-CE (Ku et al. 2020) 的 Val-Unseen 分割上与最先进方法的比较。* 表示使用自 Hong et al. (2022) 的航路点预测器的方法。即使在这些方法利用额外输入如深度、全景视图和里程计的情况下，CorrectNav 仍优于所有不依赖模拟器预训练航路点预测器的方法。

偏差点 M_t 和对应的正交脚点 P_t 。如果 P_t 位于线段 $\overline{G_kG_{k+1}}$ ($G_k, G_{k+1} \in T_g$) 上，我们可以知道模型已经正确通过了 G_k 和所有先前的参考点，但在向 G_{k+1} 前进时发生了轻微偏离。然后我们利用轨迹规划器 Γ 生成一条新轨迹

$$T_e = (M_t, G_{k+1}, \ldots, G_n)$$

。该轨迹始于 M_t ，经过随后的参考点，并在目的地 G_n 结束。这样，我们得到了一条纠错轨迹，可以作为动作校正的训练数据。训练类似于导航动作预测。为了确保模型专注于学习校正行为，动作学习仅在纠错轨迹 T_e 上进行，而 M_t 之前的轨迹仅提供观察历史。

关键帧感知为了真正赋予 CorrectNav 错误纠正能力，我们不仅要教会它如何行动，还要教会它为什么这样做。在视觉-语言导航的过程中，错误通常源于导航模型在偏差位置 M_t 附近的多模态感知错误。为了在纠正训练期间增强 CorrectNav 的多模态感知能力，我们选择在 M_t 的观察帧，以及 M_t 前后的帧，作为纠正关键帧 $\{K_1, K_2, K_3\}$ 。我们然后利用多模态大语言模型 Qwen-VL-Plus 基于这些关键纠正帧创建视觉分析数据，如图 2 的右侧所示。第一种视觉分析是描述潜在的导航地标，例如家具、装饰品或建筑结构，这些地标出现在给定的帧中。

$$C_i = \text{MLLM}(K_i, L_{cap})$$

第二种视觉分析是生成关于这些帧的视觉问答对。这些问题集中在导航中的重要视觉元素上，包括对象的相对位置、对象颜色和当前机器人方向。

$$\{(Q_j, A_j)\}_{j=1}^{x} = \text{MLLM}(K_i, L_{qa})$$

在训练过程中，我们输入观察视频 $\{I_1, I_2 ... K_i\}$ ，并将字幕 C 作为目标，训练 CorrectNav 理解当前观察；在相同的视频中，对于任意的 (Q_i, A_i) ，我们指导 CorrectNav 基于当前观察 (K_i) 回答 Q_i ，激活 CorrectNav 理解错误纠正行为。

使用收集到的自我修正数据，我们继续训练 CorrectNav 。为提高效率，我们随机抽取一半的纠错轨迹及其相应的关键帧感知数据用于训练。此外，我们加入了原始训练数据中的 oracle 轨迹以保持训练的稳定性。这些 oracle 轨迹的数量被设置为抽样纠错轨迹数量的一半。通过利用这些自动生成的数据，我们可以进一步训练 CorrectNav 以增强其自我修正能力。此时，我们已完成一次自我修正飞轮训练的循环。

当我们再次在训练集上测试自我纠正的 CorrectNav 时，出现了新的错误轨迹。这些新的错误使我们能够生成新的纠正任务数据，以继续训练 CorrectNav 。这开启了自我纠正飞轮，允许进行多轮自我纠正训练迭代。

在配备 8 块 NVIDIA A100 GPU 的服务器上训练 CorrectNav 。导航微调需要 80 小时，而一次自我校正飞轮迭代需要 20 小时。在推理时，CorrectNav 以 16 个采样的 RGB 帧作为输入，并预测一个包含 4 个有效动作的动作段。

实验

我们进行实验以回答以下问题：（1）CorrectNav 在 VLN-CE 基准测试中与最新模型相比如何？（2）通过自我纠正飞轮迭代，CorrectNav 取得了哪些改进？（3）每种自我纠正训练技术对 CorrectNav 性能的个体影响是什么？（4）CorrectNav 在现实世界中的有效性如何？

模拟实验

环境与指标 我们在 VLN-CE 基准上评估我们的 VLA，这些基准提供连续的环境以在重建的逼真室内场

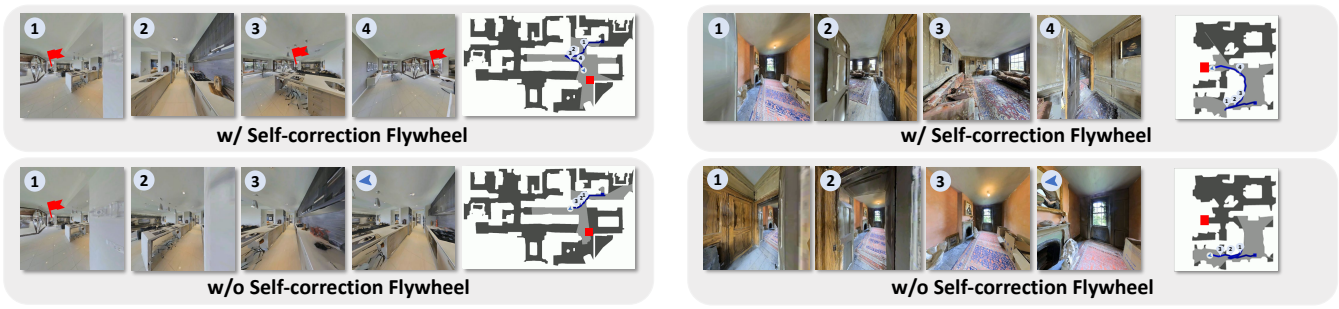

Figure 3: 关于有无自校正飞轮训练后处理机制的 CorrectNav 案例研究。左上角：CorrectNav 错误地进入错误路径，丢失目标，然后迅速转回正确路径。右上角：CorrectNav 首先进入前门，意识到没有目标（台阶）后，离开并直接进入正确的侧门。普通的 CorrectNav 在两种情况下都失败。

景中执行导航动作。我们专注于 R2R（Room-to-Room）和 RxR（Room-across-Room）数据集中与 VLN-CE 相关的 Val-Unseen 分割，因为这是 VLN 中两个最受认可的基准。按照 VLN-CE（Krantz et al. 2020）基准的设置，我们采用 Habitat 3.0（Puig et al. 2023）作为模拟器来进行评估。此外，我们采用以下广泛使用的评估指标：导航误差（NE）、Oracle 成功率（OS）、成功率（SR）、成功加权路径长度（SPL）和归一化动态时间扭曲（nDTW）。导航误差表示代理的最终位置与目标之间的平均距离（以米为单位）；成功率指代 NE 小于 3 米的路径比例；Oracle 成功率表示给定 Oracle 停止策略的 SR。nDTW（Ilharco et al. 2019）涉及时间扭曲以测量模型轨迹与真实路径之间的距离。

与其他 VLN-CE 模型的比较 我们将我们的 VLA 与现有的 VLN-CE 模型在 R2R-CE 和 RxR-CE 基准上进行比较。这些基线包括基于航路点预测器的模型和导航大模型。从表 1 可以看出，尽管我们的 CorrectNav 仅以单目 RGB 观察作为输入，但它在 R2R-CE 和 RxR-CE 基准上优于所有现有的模型。与最先进的导航大模型 StreamVLN 相比，CorrectNav 在 R2R-CE 和 RxR-CE 上的成功率分别提高了 8.2 % 和 16.4 %。此外，CorrectNav 超过了顶尖的基于航路点预测器的模型，在 R2R-CE 上超过了 HNR 4.1 %，在 RxR-CE 上超过了 13.0 %。

为了研究不同自我纠正训练技术对模型性能提升的贡献，我们通过在首次迭代的自我纠正飞轮中分别去除每项技术来进行消融研究。如表 2 所示，去除任何这些技术都会导致 CorrectNav 在 R2R-CE 和 RxR-CE Val-Unseen 数据集上的性能下降。值得注意的是，移除导航轨迹纠正策略引起了最大的性能下降。这些消融结果证实了我们提出的每种自我纠正训练技术都对提升模型的整体有效性有积极的贡献。

自我修正飞轮迭代的力量 为了研究自我纠正飞轮提高导航性能的能力，我们在每次自我纠正飞轮训练迭代后评估 CorrectNav 在 R2R-CE 和 RxR-CE Val-Unseen 基准测试上的成功率和导航误差。实验结果绘制成如图 4 所示的折线图。从图中可以观察到，随着自我纠正飞轮训练迭代的进展，CorrectNav 在前三次迭代中在两个基准测试上表现出持续的性能提升。这定量地证明了多次自我纠正飞轮迭代可以有效增强导航 VLA 模型的能力。当 CorrectNav 在第四次迭代中性能下降时，我们停

	R2R-CE Val-Unseen			RxR-CE Val-Unseen		
	NE ↓	SR ↑	SPL ↑	NE ↓	SR ↑	SPL ↑
CorrectNav	4.50	63.0	59.0	4.40	63.1	57.0
w/o Navigation Trajectory Correction	4.92	59.2	57.2	4.55	60.7	55.1
w/o Error-correcting Keyframe Perception	4.70	60.1	56.5	4.47	61.0	56.3
w/o Data Sampling Strategy	4.71	60.0	57.5	4.47	62.2	56.2

Table 2: 基于 R2R-CE（Anderson et al. 2018）和 RxR-CE（Ku et al. 2020）的 Val-Unseen 划分，对自校正飞轮训练后技术的消融研究。实验是基于 1_{st} 自校正飞轮训练后进行的。

止训练。此外，我们还进行了自我纠正飞轮效果的定性分析，如图 3 所示。从图中可以看到，与原始 CorrectNav 相比，经过自我纠正飞轮后训练的 CorrectNav 实现了误差纠正能力。

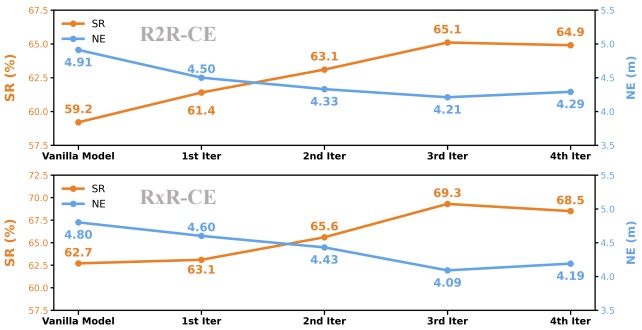

Figure 4: CorrectNav 在 R2R-CE 和 RxR-CE Val-Unseen 分割上的表现随着自我校正飞轮迭代的变化。

真实机器人实验

对于实际实验，我们使用 AgiBot Lingxi D1 四足机器人作为我们的平台。每个 Lingxi D1 机器人都配备有一个单目 RGB 摄像头和可靠的运动 API。机器人接收到导航指令后，会将 RGB 观测图像上传到部署在配备 NVIDIA A100 GPU 的远程服务器上的 CorrectNav 。CorrectNav 将预测包含四个动作的导航动作块，并调用 D1 运动 API 执行。

为了全面评估我们方法的有效性，我们在办公室、家庭和校园环境中与两个最先进的导航大模型，NaVID

Figure 5: Qualitative results from the real-world deployment of CorrectNav. (c)(d) The robot dynamically avoids pedestrians and obstacles, correctly passing through cluttered environments to reach the destination. (e)(f) The robot successfully recovers from a navigation error to complete a long-horizon instruction. (g) The robot completes outdoor long-distance navigation. Videos are shown on our project website.

	Office				Home				Campus			
	Simple		Complex		Simple		Complex		Simple		Complex	
	NE ↓	SR ↑	NE ↓	SR ↑	NE ↓	SR ↑	NE ↓	SR ↑	NE ↓	SR ↑	NE ↓	SR ↑
NaVid (Zhang et al. 2024a)	1.88	0.55	4.89	0.30	2.22	0.50	5.27	0.15	1.94	0.55	5.02	0.25
NaVILA (Cheng et al. 2024)	2.06	0.45	5.25	0.20	2.21	0.45	5.49	0.10	1.97	0.50	5.15	0.20
CorrectNav (Ours)	1.52	0.80	1.81	0.75	1.33	0.95	1.54	0.80	1.47	0.85	1.86	0.75

Table 3: 不同环境中的真实世界实验。Simple 和 Complex 分别指简单和复杂的指令执行任务。

和 NaVILA 进行比较。在每种场景中，我们分别对每个模型测试 20 个简单指令和 20 个复杂指令。复杂指令涉及长路径、复杂的建筑结构、拥挤的障碍物以及动态场景变化。在成功率（SR）和导航误差（NE）指标方面的实际定量表现如表 3 所示。图 5 展示了 CorrectNav 在室内和室外环境中的实际定性表现。

从表格 3 中可以观察到，与现有导航大型模型相比，CorrectNav 在执行现实世界导航指令方面表现出更强的能力。如图 5 所示，这种改进主要源于 CorrectNav 在自纠正飞轮后训练中获得的偏差校正能力。这些能力提高了 CorrectNav 在复杂指令上的鲁棒性，使其能够快速纠正自身错误或适应环境变化。

尽管基于单目 RGB 观察的 VLA 导航模型（包括我们的 CorrectNav ）节省了额外传感器的成本，但它们面临共同的限制：在感知机器人体与周围环境之间的相对位置关系时精度不足。这种缺陷带来的潜在风险是，当机器人（如四足机器人）靠近障碍物通过时，它的后腿可能会刮擦到障碍物。未来研究的一个有前途的方向是探讨如何将机器人身体尺寸和状态信息作为先验知识纳入导航模型推断中，从而进一步改进基于单目 RGB 的 VLA 导航模型。

References

An, D.; Qi, Y.; Li, Y.; Huang, Y.; Wang, L.; Tan, T.; and Shao, J. 2023. Bevbert: Multimodal map pre-training for language-guided navigation. In ICCV.

An, D.; Wang, H.; Wang, W.; Wang, Z.; Huang, Y.; He, K.; and Wang, L. 2024. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. IEEE TPAMI.

An, D.; Wang, Z.; Li, Y.; Wang, Y.; Hong, Y.; Huang, Y.; Wang, L.; and Shao, J. 2022. 1st place solutions for rxr-habitat vision-and-language navigation competition. In CVPRW.

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and van den Hengel, A. 2018. Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niessner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. International Conference on 3D Vision (3DV).

Chen, J.; Lin, B.; Liu, X.; Liang, X.; and Wong, K.-Y. K. 2024. Affordances-Oriented Planning using Foundation Models for Continuous Vision-Language Navigation. arXiv preprint.

Chen, K.; Chen, J. K.; Chuang, J.; Vázquez, M.; and Savarese, S. 2021. Topological planning with transformers for vision-and-language navigation. In CVPR.

Chen, P.; Ji, D.; Lin, K.; Zeng, R.; Li, T.; Tan, M.; and Gan, C. 2022. Weakly-supervised multi-granularity map learning for vision-and-language navigation. In NeurIPS.

Cheng, A.-C.; Ji, Y.; Yang, Z.; Gongye, Z.; Zou, X.; Kautz, J.; Bıyık, E.; Yin, H.; Liu, S.; and Wang, X. 2024. Navila: Legged robot vision-language-action model for navigation. arXiv preprint arXiv:2412.04453.

Duan, J.; Pumacay, W.; Kumar, N.; Wang, Y. R.; Tian, S.; Yuan, W.; Krishna, R.; Fox, D.; Mandlekar, A.; and Guo, Y. 2024. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. arXiv preprint arXiv:2410.00371.

Georgakis, G.; Schmeckpeper, K.; Wanchoo, K.; Dan, S.; Miltsakaki, E.; Roth, D.; and Daniilidis, K. 2022. Cross-modal map learning for vision and language navigation. In CVPR.

Ha, H.; Florence, P.; and Song, S. 2023. Scaling Up and Distilling Down: Language-Guided Robot Skill Acquisition. arXiv:2307.14535.

Hong, Y.; Wang, Z.; Wu, Q.; and Gould, S. 2022. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In CVPR.

Hong, Y.; Zhou, Y.; Zhang, R.; Dernoncourt, F.; Bui, T.; Gould, S.; and Tan, H. 2023. Learning navigational visual representations with semantic map supervision. In ICCV.

Ilharco, G.; Jain, V.; Ku, A.; Ie, E.; and Baldridge, J. 2019. General Evaluation for Instruction Conditioned Navigation using Dynamic Time Warping. arXiv:1907.05446.

Krantz, J.; Gokaslan, A.; Batra, D.; Lee, S.; and Maksymets, O. 2021. Waypoint models for instruction-guided navigation in continuous environments. In CVPR.

Krantz, J.; and Lee, S. 2022. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In ECCV.

Krantz, J.; Wijmans, E.; Majundar, A.; Batra, D.; and Lee, S. 2020. Beyond the Nav-Graph: Vision and Language Navigation in Continuous Environments. In ECCV.

Ku, A.; Anderson, P.; Patel, R.; Ie, E.; and Baldridge, J. 2020. Room-Across-Room: Multilingual Vision-and-Language Navigation with Dense Spatiotemporal Grounding. In EMNLP.

Lin, B.; Nie, Y.; Zai, K. L.; Wei, Z.; Han, M.; Xu, R.; Niu, M.; Han, J.; Lin, L.; Lu, C.; and Liang, X. 2025. EvolveNav: Self-Improving Embodied Reasoning for LLM-Based Vision-Language Navigation. arXiv:2506.01551.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved baselines with visual instruction tuning. In CVPR.

Liu, Z.; Bahety, A.; and Song, S. 2023. REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction. arXiv preprint arXiv:2306.15724.

Long, Y.; Cai, W.; Wang, H.; Zhan, G.; and Dong, H. 2024. InstructNav: Zero-shot System for Generic Instruction Navigation in Unexplored Environment. arXiv:2406.04882.

Ma, Y. J.; Sodhani, S.; Jayaraman, D.; Bastani, O.; Kumar, V.; and Zhang, A. 2023. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. arXiv:2210.00030.

Puig, X.; Undersander, E.; Szot, A.; Cote, M. D.; Partsey, R.; Yang, J.; Desai, R.; Clegg, A. W.; Hlavac, M.; Min, T.; Gervet, T.; Vondruš, V.; Berges, V.-P.; Turner, J.; Maksymets, O.; Kira, Z.; Kalakrishnan, M.; Malik, J.; Chaplot, D. S.; Jain, U.; Batra, D.; Rai, A.; and Mottaghi, R. 2023. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots.

Raychaudhuri, S.; Wani, S.; Patel, S.; Jain, U.; and Chang, A. 2021. Language-Aligned Waypoint (LAW) Supervision for Vision-and-Language Navigation in Continuous Environments. In EMNLP.

Shi, X.; Li, Z.; Lyu, W.; Xia, J.; Dayoub, F.; Qiao, Y.; and Wu, Q. 2025. SmartWay: Enhanced Waypoint Prediction and Backtracking for Zero-Shot Vision-and-Language Navigation. arXiv:2503.10069.

Wang, H.; Liang, W.; Van Gool, L.; and Wang, W. 2023a. Dreamwalker: Mental planning for continuous vision-language navigation. In ICCV.

Wang, Z.; Li, J.; Hong, Y.; Wang, Y.; Wu, Q.; Bansal, M.; Gould, S.; Tan, H.; and Qiao, Y. 2023b. Scaling data generation in vision-and-language navigation. In ICCV.

Wang, Z.; Li, X.; Yang, J.; Liu, Y.; Hu, J.; Jiang, M.; and Jiang, S. 2024. Lookahead Exploration with Neural Radiance Representation for Continuous Vision-Language Navigation. In CVPR.

Wang, Z.; Li, X.; Yang, J.; Liu, Y.; and Jiang, S. 2023c. Gridmm: Grid memory map for vision-and-language navigation. In ICCV.

Wei, M.; Wan, C.; Yu, X.; Wang, T.; Yang, Y.; Mao, X.; Zhu, C.; Cai, W.; Wang, H.; Chen, Y.; Liu, X.; and Pang, J. 2025a. StreamVLN: Streaming Vision-and-Language Navigation via SlowFast Context Modeling. arXiv:2507.05240.

Wei, Z.; Lin, B.; Nie, Y.; Chen, J.; Ma, S.; Xu, H.; and Liang, X. 2025b. Unseen from Seen: Rewriting Observation-Instruction Using Foundation Models for Augmenting Vision-Language Navigation. arXiv:2503.18065.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Yang, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Liu, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; Guo, Z.; and Fan, Z. 2024. Qwen2 Technical Report. arXiv:2407.10671.

Zhai, X.; Mustafa, B.; Kolesnikov, A.; and Beyer, L. 2023. Sigmoid loss for language image pre-training. In ICCV.

Zhang, J.; Wang, K.; Wang, S.; Li, M.; Liu, H.; Wei, S.; Wang, Z.; Zhang, Z.; and Wang, H. 2025. Uni-NaVid: A Video-based Vision-Language-Action Model for Unifying Embodied Navigation Tasks. arXiv:2412.06224.

Zhang, J.; Wang, K.; Xu, R.; Zhou, G.; Hong, Y.; Fang, X.; Wu, Q.; Zhang, Z.; and He, W. 2024a. NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation. In RSS.

Zhang, Y.; Wu, J.; Li, W.; Li, B.; Ma, Z.; Liu, Z.; and Li, C. 2024b. Video Instruction Tuning With Synthetic Data. arXiv:2410.02713.