ByteDance | Seed

北京大学 PEKING UNIVERSITY

VLingNav: Embodied Navigation with Adaptive Reasoning and Visual-Assisted Linguistic Memory

Shaoan Wang[1,2,*], **Yuanfei Luo**[1,*], **Xingyu Chen**[2,3,†], **Aocheng Luo**[2], **Dongyue Li**[2], **Chang Liu**[2], **Sheng Chen**[1,‡], **Yangang Zhang**[1], **Junzhi Yu**[2,†]

[1]ByteDance Seed, [2]Peking University, [3]Zhongguancun Academy

[*]Co-first authors, [†]Corresponding authors, [‡]Project lead

Abstract

Vision-Language-Action (VLA) models have shown promising potential in embodied navigation by unifying perception and planning while inheriting the strong generalization abilities of large Vision-Language Models (VLMs). However, most existing VLA models rely on reactive mappings directly from observations to actions, lacking the explicit reasoning capabilities and persistent memory required for complex, long-horizon navigation tasks. To address these challenges, we propose VLingNav, a VLA model for embodied navigation grounded in linguistic-driven cognition. First, inspired by the dual-process theory of human cognition, we introduce an adaptive chain-of-thought (AdaCoT) mechanism, which dynamically triggers explicit reasoning only when necessary, enabling the agent to fluidly switch between fast, intuitive execution and slow, deliberate planning. Second, to handle long-horizon spatial dependencies, we develop a visual-assisted linguistic memory module (VLingMem) that constructs a persistent, cross-modal semantic memory, enabling the agent to recall past observations to prevent repetitive exploration and infer movement trends for dynamic environments. For training, we construct Nav-AdaCoT-2.9M, the largest embodied navigation dataset with reasoning annotations to date, enriched with adaptive CoT annotations that induce a reasoning paradigm capable of adjusting both when to think and what to think about. Moreover, we incorporate an online expert-guided reinforcement learning stage, enabling the model to surpass pure imitation learning and to acquire more robust, self-explored navigation behaviors. Extensive experiments demonstrate that VLingNav achieves state-of-the-art performance across a wide range of embodied navigation benchmarks. Notably, VLingNav transfers to real-world robotic platforms in a zero-shot manner, successfully executing practical navigation tasks, including previously unseen and untrained tasks, and demonstrating strong cross-domain and cross-task generalization.

Date: January 13, 2026
Correspondence: wangshaoan@stu.pku.edu.cn, luoyuanfei@bytedance.com
Project Page: https://wsakobe.github.io/VLingNav-web/

1 Introduction

Embodied navigation [57] is a fundamental capability for intelligent robots, enabling purposeful movement through previously unseen, structurally complex environments in response to human instructions. As robots are increasingly deployed in open-world settings from household service scenarios to industrial inspection,

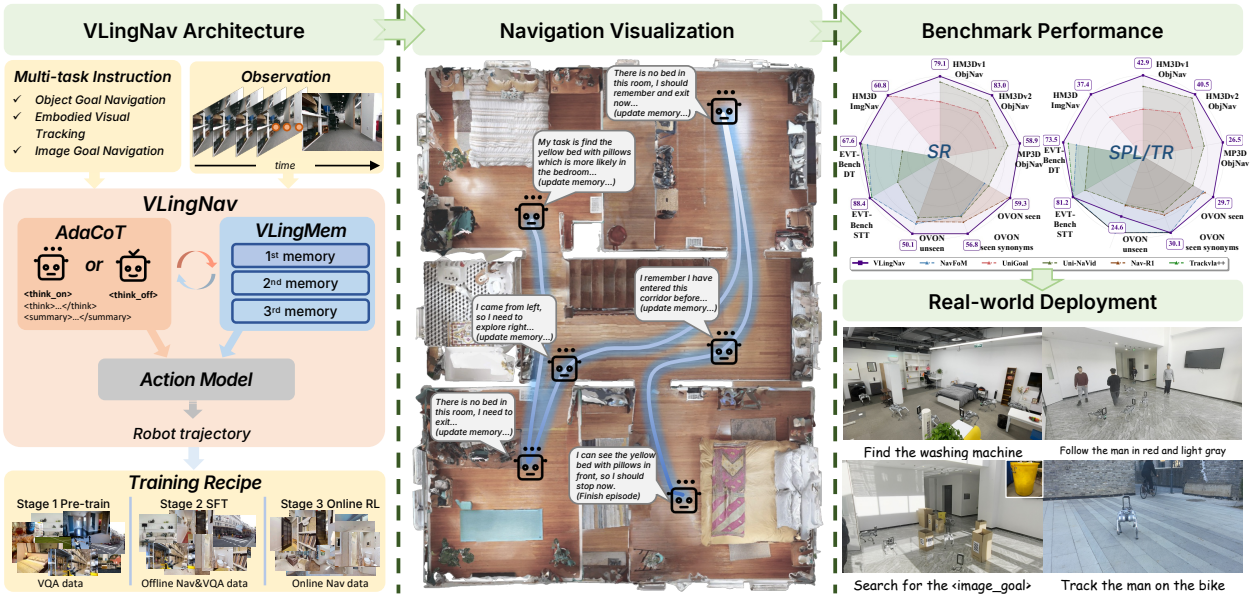

Figure 1 Overview of VLingNav. VLingNav is a VLA model enhanced with adaptive CoT reasoning and visual-assisted linguistic memory. This architecture allows the model to leverage historical visual and linguistic memory, achieving SOTA results on several embodied navigation benchmarks. Furthermore, VLingNav can be deployed zero-shot on real-world robots to perform diverse and complex navigation tasks.

navigation systems must deliver accurate perception and decision-making while robustly generalizing to novel scenes and tasks. Traditional modular approaches [8, 9, 65] decompose navigation into submodules (e.g., perception, mapping, planning) by leveraging mature techniques such as visual foundation models [32, 44], SLAM [38, 58], and path-planning algorithms [21]. However, these pipelines require manually defined module interfaces; over-reliance on hand-crafted rules compromises robustness and induces error accumulation, limiting adaptability in dynamically complex environments. Recent advances in large-scale vision–language–action (VLA) models have made compelling progress toward this goal. By unifying multimodal scene understanding with language-conditioned action generation, VLA-based agents substantially improve the adaptability and expressiveness of embodied navigation systems.

Despite this progress, current VLA models are reactive systems, often lacking the explicit reasoning mechanisms, memory structures, and interpretability that are important for reliable real-world deployment. Most existing models operate under a fixed inference budget, producing actions with a predetermined amount of computation, and therefore cannot increase deliberation when faced with ambiguity. In addition, these models often lack persistent semantic memory, relying solely on limited context windows. Without a mechanism to retain historical context, agents struggle to track their progress over extended trajectories, resulting in redundant exploration, looping behaviors, and poor adaptation to dynamic changes in the environment.

Addressing these limitations requires rethinking VLA architectures from a linguistic perspective, moving beyond passive perception-action mapping toward active reasoning, memory construction, and interpretable decision-making. Motivated by principles from cognitive science and human problem solving, we argue that effective embodied navigation demands two missing capabilities: 1) adaptive reasoning, enabling the agent to adjust the granularity of its internal deliberation according to task complexity; and 2) linguistically grounded long-term memory, providing stable cross-modal semantics that support consistent and context-aware navigation behavior.

Furthermore, most current VLA training paradigms rely on supervised fine-tuning (SFT) via imitation learning. However, this approach often limits generalization, preventing models from performing beyond expert demonstrations. While post-training paradigms rooted in reinforcement learning (RL) have proven effective for enhancing LLMs and VLMs on complex tasks [13, 16, 47], their application in embodied navigation

remains preliminary [14, 31, 82]. Notably, existing efforts typically focus on autoregressive RL in discrete space, leaving the exploration of RL for refining continuous control policies an open area for further investigation.

In this work, we present VLingNav, a linguistic-driven VLA framework designed to endow embodied agents with cognitive abilities through two core components. First, inspired by the fast-and-slow thinking paradigm, we introduce an Adaptive Chain-of-Thought (AdaCoT) mechanism. AdaCoT dynamically triggers explicit reasoning only when necessary, allowing the agent to efficiently switch between fast reactive execution and deliberate planning depending on the situation. Second, to handle long-term spatial dependencies, we develop a Visual-assisted Linguistic Memory module (VLingMem). By constructing a persistent cross-modal memory, VLingMem enables the agent to recall past observations to prevent repetitive exploration and infer movement trends for dynamic tasks, thereby ensuring coherent decision-making over extended interactions.

To support the training of such cognitively enriched VLA models, we construct Nav-AdaCoT-2.9M, the largest embodied navigation dataset with reasoning annotations to date, incorporating adaptive CoT annotations that teach the model when to think and what to think about. Beyond imitation learning, we further employ online expert-guided RL for post-training, enabling VLingNav to acquire self-improving navigation behaviors that surpass the limitations of supervised demonstrations.

Extensive experiments across diverse embodied navigation benchmarks show that VLingNav achieves state-of-the-art performance, outperforming existing VLA-based agents in both success rate and efficiency metrics. Notably, VLingNav transfers to real-world robots in a zero-shot manner, successfully executing novel navigation tasks in the real world without any additional fine-tuning. These results highlight the strong generalization ability of linguistic-driven cognition and demonstrate the promise of integrating adaptive reasoning and persistent memory into VLA models for embodied navigation. The contributions are as follows:

- We propose VLingNav, a novel framework integrating Adaptive Chain-of-Thought (AdaCoT) and Visual-Assisted Linguistic Memory (VLingMem). AdaCoT enables the agent to dynamically switch between fast execution and slow deliberation based on task complexity, while VLingMem eliminates redundant exploration and infers movement trends through persistent cross-modal storage.
- We construct Nav-AdaCoT-2.9M, the largest embodied navigation dataset with reasoning annotations to date, enriched with adaptive CoT annotations that induce flexible reasoning patterns. We further introduce an online expert-guided RL post-training stage, empowering the model to surpass the limitations of imitation learning and acquire more robust, self-optimized navigation behaviors.
- We conduct extensive experiments across standard embodied navigation benchmarks, demonstrating that VLingNav achieves state-of-the-art performance, with significant gains in long-horizon reasoning and success rate. Moreover, VLingNav exhibits remarkable zero-shot transfer to real-world robot platforms, successfully executing unseen tasks and illustrating strong cross-domain and cross-task generalization.

2 Related Works

2.1 Embodied Navigation Models

As a core task in robotics, navigation has long attracted significant attention from robotics researchers [11]. With the rise of embodied AI in recent years, robot navigation has gradually shifted from classical point-to-point navigation [22] to more intelligent embodied navigation. Embodied navigation includes subtasks such as vision-language navigation (VLN) [10, 56, 70, 73, 75], object goal navigation (ObjectNav) [42, 60, 62], image goal navigation (ImageNav) [25, 59, 63], and embodied visual tracking (EVT) [30, 52, 84], emphasizing that robots follow natural language instructions to perceive, reason, and plan in unseen environments.

Embodied navigation methods can be broadly categorized into modular and end-to-end approaches. The modular paradigm relies on well-established components such as off-the-shelf large models [85, 86], SLAM [5, 38], vision foundation models [39, 71], and planning algorithms [21]. It decomposes the navigation task into distinct modules (*e.g.*, perception, localization, planning) and aligns them via manually defined interfaces. This design yields high interpretability and strong zero-shot transfer [63]. However, integrating multiple modules inevitably incurs information loss [34]; moreover, tight coupling across modules increases system fragility [34]. End-to-end

approaches leverage data-driven learning to directly map sensor inputs to robot actions [59, 66, 69]. By removing manually designed interfaces and mitigating information loss, these methods have achieved notable progress [42, 43]. However, they exhibit limited generalization and can produce abnormal actions under out-of-distribution conditions. With the rapid advancement of large models in recent years, an increasing number of studies have adopted pre-trained VLMs as the backbone to enhance generalization, environmental perception, and spatial understanding.

NaVid [73] represents the first embodied navigation VLA model. It designs a video-based VLM and finetunes on VLN datasets, demonstrating robust generalization capabilities. However, its inference time increases significantly with longer video streams, making real-world deployment challenging. Building upon NaVid, Uni-NaVid [75] introduces a video-stream compression mechanism to control the number of visual tokens. Moreover, Uni-NaVid extends the model to multiple categories of embodied navigation tasks, achieving state-of-the-art performance across diverse benchmarks. Similarly, NaVILA [10] and StreamVLN [56] adopt similar architectures; they further incorporate large-scale open-world navigation data and leverage KV cache to jointly improve both generalization and inference speed. JanusVLN [70] enhances 3D understanding by fusing spatial features produced by VGGT [51], thereby exhibiting strong instruction-following performance. Notably, all the aforementioned works represent robot actions as discrete tokens. This simplification leads to inefficient action quality and weak adaptability in dynamic scenarios. To address this limitation, TrackVLA [52] designs an anchor-based diffusion policy that directly outputs the robot's motion trajectory, substantially improving both action quality and efficiency. NavFoM [74] further extends the model by introducing TVI tokens, enabling inputs from cross-embodiment navigation data.

Nevertheless, existing navigation VLA models rely solely on action labels for finetuning and thus fail to exploit the inherent reasoning capabilities of VLMs [2, 13]. In addition, they maintain history only through implicit visual features, without explicit memory, which ultimately prevents fully unlocking the potential of the VLM backbone.

2.2 Embodied Chain-of-Thought

With the chain-of-thought significantly enhancing the performance of LLM and VLM on complex tasks [20, 54, 55], several studies have attempted to extend this paradigm to embodied tasks. By explicitly outputting the reasoning process before a robot executes actions, the inherent reasoning capabilities of the VLM can be better leveraged. This approach aims to enhance the model's competencies in task decomposition, environmental perception, and decision-making, ultimately improving the accuracy and quality of the actions generated by the model, as well as its generalization ability and performance in real-world scenarios. Embodied-CoT [68] first utilizes structured textual instructions enriched with spatial localization information. CoT-VLA [83] and VPP [19, 72] integrate reasoning via future image prediction. $\pi_{0.5}$ [3] performs task decomposition and reasoning through text. ChatVLA-2 [87] enhances the model's performance in complex visual reasoning tasks by introducing additional open-world visual reasoning pre-training data. ThinkAct [20] designs a dual-system framework that bridges high-level reasoning with low-level action. However, the aforementioned methods are limited to tabletop manipulation tasks and have not been extended to navigation in open spaces. OctoNav [14] improves the model's performance in navigation tasks and enhances interpretability by executing CoT at fixed frequency. However, the requirement for manual configuration of the CoT frequency impedes the full exploitation of CoT's potential. Aux-Think [53] constructs a VLN dataset with CoT labeling, and experiments show that using CoT as an auxiliary task during training enhances the model's navigation performance, while excessive reasoning affects the model's efficiency and performance. $NavA^3$ [76] adopts GPT-4o as the reasoning-VLM for task decomposition and 3D spatial localization, but it suffers from long reasoning latency, making it difficult to deploy on real robots.

In contrast to previous work, we propose an adaptive thinking strategy that balances reasoning efficiency with navigation capability.

2.3 Memory in VLA Models

For long-horizon embodied tasks, VLA models must possess robust memory capabilities. RoboFlamingo [27], for instance, compresses vision–language representations into latent tokens and propagates them through

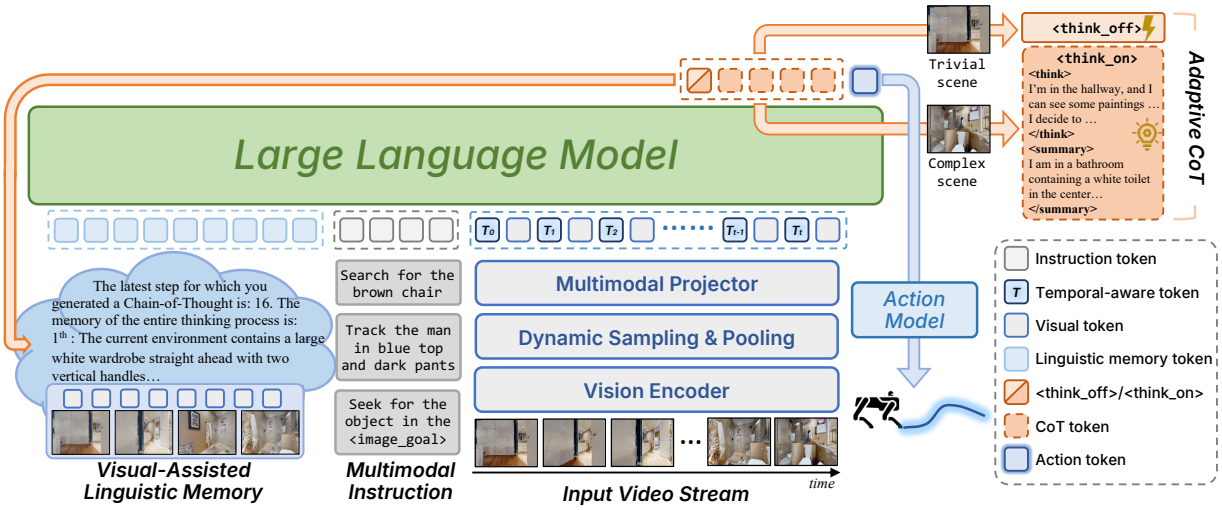

Figure 2 The overall framework of VLingNav. The framework takes video streams and multimodal instruction as input to produce robot action for navigation with tailored linguistic designs. AdaCoT can adaptively generate linguistic thinking according to its observation, while VLingMem summarizes CoT cues with key visual features for globally informed decision-making.

a Long Short-Term Memory (LSTM) network. However, the resulting latent representations are relatively coarse-grained, leading to a significant loss of fine-grained perceptual history. In contrast, MemoryVLA [48] integrates high-level cognitive semantics and fine-grained perceptual details within a unified memory framework, enabling effective temporal modeling for long-horizon manipulation tasks. However, it only employs a single implicit cognitive token to serve as the semantic memory, failing to fully leverage the reasoning capabilities of LLM. On the navigation side, video-based VLA models [10, 30, 52, 56, 73, 75] commonly encode historical image observations as inputs to provide implicit visual memory. However, such implicit memory can hinder learning to focus on key regions, and semantic information is further degraded as visual features are repeatedly compressed. Finally, Mem2Ego [78] and MapNav [77] incorporate global map information into VLA models as memory components. Yet current VLM backbones lack native support for map-format inputs, and the representation design of maps for VLAs remains under-explored.

Compared with latent-, vision-, or map-based memories, language memory is better aligned with the VLA framework, thanks to large-scale language pretraining. Therefore, we design the memory module from a linguistic perspective and use visual features as auxiliary signals.

2.4 Post-training for VLA Models

Reinforcement Learning enhances the exploration capability of large models, unlocks their reasoning potential, and shows promise for mitigating issues such as covariate shift and causal confusion induced by imitation learning. Notably, OctoNav [14], VLN-R1 [37], and Nav-R1 [31] have convergently integrated GRPO [47] into navigation VLA models, enabling the simultaneous optimization of CoT outputs and actions.

Recent advances in large reasoning models (*e.g.*, DeepSeek-R1 [16]) show that RL can drive remarkable progress even when relying solely on outcome-based rewards. Several studies have also attempted to leverage outcome-based rewards for the RL post-training of VLA models. For instance, SimpleVLA-RL [26] pioneers outcome-based rewards for the RL post-training of OpenVLA-OFT [23], achieving a substantial improvement in success rate on the manipulation benchmarks. ActiveVLN [82], caches all historical actions and states into model tokens and leverages GRPO to implement outcome-based RL through this mechanism.

The aforementioned work remains confined to autoregressive action outputs, failing to support more advanced continuous action prediction. Recently, ReinFlow [80] addresses this by formulating flow matching as an MDP, enabling RL training via PPO [45] or GRPO.

Existing VLA-RL frameworks either adopt discrete autoregressive action with limited policy space or continual flow-based action with slow inference speed. We adopt an MLP-based continuous action to overcome the above drawbacks. In addition, we introduce prior expert knowledge into the RL framework to improve online learning efficiency and performance.

3 Methodology

3.1 Navigation Task Definitions

Embodied navigation tasks can be defined as follows: a mobile robot is provided with an instruction $\mathcal{I}$ and a sequence of visual observations $\mathcal{O}_{1:t} \in \mathbb{R}^{W \times H \times 3}$, captured by the egocentric camera mounted on the robot at each time step $\{1, ..., t\}$. Given these observations and the instruction, the policy model π is required to output the next action $a_t \in \mathbb{A} = \{v, \omega\}$ for the robot. The robot accomplishes the navigation task by executing the action predicted by the model, which can be formulated as $a_t = \pi(\mathcal{I}, \mathcal{O}_{1:t})$. VLingNav is capable of performing multiple embodied navigation tasks, including ObjectNav, EVT, and ImageNav. ObjectNav requires the robot to explore unseen environments given a textual description of an object category and to locate an object that matches the specified goal. EVT focuses on identifying the correct target described by textual instructions in dynamic, crowded scenarios and maintaining continuous tracking of the moving target. ImageNav is analogous to ObjectNav, with the key difference that the goal is specified by an image rather than text. Similarly, the robot must explore unseen environments and find the location corresponding to the image goal.

3.2 VLingNav Overview

VLingNav extends a video-based VLM, specifically LLaVA-Video-7B [81], and integrates an action model to enable simultaneous text token generation and trajectory planning. For text token prediction, the model follows a conventional autoregressive paradigm. For trajectory planning, the action model conditions on the VLM backbone's outputs to predict a motion trajectory $\tau = \{a_1, a_2, \dots a_n\}$, where n is the trajectory horizon and each $a \in \mathbb{R}^3 = (x, y, \theta)$ denotes a waypoint that encapsulates both position and orientation.

3.3 VLingNav Architecture

3.3.1 Observation Encoding

For the video-based VLA model, the number of image frames grows over time during online inference. This substantially increases computational burden, making it difficult to ensure inference efficiency when deploying on real robots. Moreover, for low-speed mobile robots, adjacent egocentric frames captured at high FPS contain substantial redundant visual information. Prior studies explore two main strategies to mitigate this issue. One merges visual tokens from historical frames to reduce redundancy among adjacent frames [4, 75]; however, this operation often distorts original semantic features and introduces additional computation. The other uniformly samples the video stream to reduce frame count [10], which inevitably causes delayed and inaccurate decisions due to insufficient short-term observations at low sampling rates.

To address the limitations of these two approaches, we propose a dynamic FPS sampling strategy. Inspired by the Ebbinghaus forgetting curve [12], historical frames are sampled according to their time intervals relative to the current frame. Specifically, older historical frames, regarded as long-term memory, are sampled at a lower rate to simulate the forgetting process. In contrast, recent historical frames, considered as short-term memory, are sampled at a guaranteed higher rate. The relationship between the sampling rate and the time interval approximately satisfies the following:

$$f_s(i) = f_s^{max} e^{-\frac{\Delta T}{s}} \tag{1}$$

where f_s denotes the sampling rate, f_s^{max} represents the maximum sampling rate, $\Delta T = t - i$ stands for the time interval from latest frame t to frame i, and s signifies the stability of memory. Through this approach, we can control the number of input image tokens while selectively preserving more important images.

After sampling the input visual observations, we need to encode and map the visual observations into the latent space of the VLM backbone. Following LLaVA-Video, we employ a pre-trained vision encoder (SigLIP-400M [71]), to encode the input egocentric video stream $\mathcal{O}_{1:t} = \{\mathbf{o}_1, \cdots, \mathbf{o}_t\}$ of the robot. This encoding process yields visual features $\mathbf{V}_{1:t} \in \mathbb{R}^{N \times C}$, where N stands for the number of image patches ($N = 729$) and C denotes the embedding dimension ($C = 1152$). To efficiently summarize historical visual information, we process past observations using a grid pooling strategy. This approach downsamples the feature maps of historical observations, enabling the model to capture high-level semantic features while effectively controlling computational costs. Similar to dynamic FPS, we also determine the downsampling ratio for grid pooling based on time intervals. The specific operation is defined as follows:

$$g(i) = \lfloor e^{-\frac{\Delta T}{g}} \rfloor \tag{2}$$

$$\mathbf{V}'_{t_i} = \mathcal{G}(\mathbf{V}_{t_i}, g(i)) \tag{3}$$

where $\mathbf{V}'_{t_i}$ is the i-th visual feature after grid pooling operation $\mathcal{G}(\cdot)$ with stride $g(i)$.

Furthermore, to eliminate the temporal inconsistency in the video stream caused by dynamic FPS sampling, we incorporate timestamp information for each frame within the visual observations. Specifically, a temporal-aware indicator token $E^T(\cdot) \in \mathbb{R}^C$ is introduced prior to each frame, which can reflect the time interval between a given historical visual observation and the current observation. By encoding timestamp information using Rotary Position Embedding (RoPE) [49], E^T enables the model to perceive the absolute time interval between different historical frames and the current frame. It can be expressed as follows:

$$E^T(\Delta T) = E^T_{base} + RoPE(\Delta T) \tag{4}$$

For the projection of visual features, we follow the well-established framework of VLMs [29]. Specifically, a cross-modality projector based on a two-layer Multi-Layer Perceptron (MLP) $\mathcal{P}(\cdot)$ is employed to map the visual features $\mathbf{V}$ into the latent space of the VLM, yielding the projection result as $\mathbf{E}^V_t = \mathcal{P}(\mathbf{V}'_t)$, where $\mathbf{E}^V_t$ represents the projected visual token.

3.3.2 Adaptive CoT & Visual-Assisted Linguistic Memory

As illustrated in Fig. 2, we concatenate the visual tokens $\mathbf{E}^V_t$ with the language tokens $\mathbf{E}^I$ and the temporal-aware indicator tokens $\mathbf{E}^T$ to form the input sequence of the VLM. To balance the model's inference performance and efficiency, we train the model using a large scale high-quality adaptive CoT data (detailed in Sec. 4) endowing it with the ability to autonomously decide whether to perform CoT reasoning for a given input. Specifically, for the current input, the VLM first predicts a CoT indicator token (`<think_on>` or `<think_off>`). Upon outputting `<think_on>`, the model generates the specific content of CoT in an autoregressive manner, which consists of two components:

- The reasoning content, enclosed within `<think>` and `</think>` tokens. This content includes perception of the visual observation, task decomposition and analysis, assessment of whether the current location has been visited, and determination of the next action.
- The environmental summary of the current observation, enclosed within `<summary>` and `</summary>` tokens. This summary is incorporated into subsequent inputs as linguistic memory.

3.3.3 Action Model

To transfer the reasoning and decision-making knowledge of the VLM backbone into the robot-specific action space, we integrate an MLP-based action model $\mathcal{A}_\theta(\cdot)$ into VLingNav. Specifically, the hidden state vector $\mathbf{h}^{pred}_t$ corresponding to the final token predicted by the VLM backbone is used as the condition to guide the action model in converting this representation into robot motion trajectory τ, which can be formulated as:

$$\hat{\tau}_t = \mathcal{A}_\theta \left(\mathbf{h}^{pred}_t \right) \tag{5}$$

Table 1 Comparison of existing navigation datasets. Nav-AdaCoT-2.9M is the first dataset to integrate three navigation tasks (ObjNav, Track, ImageNav) and provide adaptive chain-of-thought reasoning.

Dataset	Scenes			Instruction Capability				N_{step}	N_{cot}	Action
	HM3D	MP3D	N_{scene}	ObjNav	Track	ImageNav	Modality			
HM3D ObjNav [41]	✓	×	80	✓	×	×	L	-	-	Des.
MP3D ObjNav [7]	×	✓	56	✓	×	×	L	-	-	Des.
SOON [88]	✓	×	90	✓	×	×	L	30K	-	Des.
HM3D OVON [66]	✓	×	181	✓	×	×	L	53K	-	Des.
EVT-Bench [52]	✓	✓	703	×	✓	×	L	855K	-	Traj.
HM3D ImgNav [25]	✓	×	145	×	×	✓	L	-	-	Des.
OctoNav-Bench [14]	✓	✓	438	✓	×	✓	V, L	45K	10K	Des.
Nav-CoT-110K [31]	✓	✓	342	✓	×	×	V, L	110K	110K	Des.
Nav-AdaCoT-2.9M (Ours)	✓	✓	**718**	✓	✓	✓	V, L	**2.9M**	**472K**	Traj.

where $\hat{\tau}_t$ is the predicted motion trajectory in current timestamp t. The pseudocode presented in Alg. 1 illustrates the complete online inference process of VLingNav in detail.

Algorithm 1 VLingNav Online Inference

Input: Observation video stream $\mathcal{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t\}$, Instruction I
Initialize: Memory $\mathcal{M} \leftarrow \emptyset$, Visual Cache $\mathcal{V} \leftarrow \emptyset$
procedure OnlineInference($I, \mathcal{O}$)
  while true do
    $\mathbf{E}^I \leftarrow \text{Tokenizer}(I)$
    $\mathbf{v}_t \leftarrow \text{VisionEncoder}(\mathbf{o}_t)$ ▷ Encode the current visual frame
    $\mathcal{V} \leftarrow \text{Cache}(\mathcal{V}, \mathbf{v}_t)$ ▷ Update visual cache with the new feature
    $\mathbf{E}^V \leftarrow \text{Sampling\&Pooling}(\mathcal{V})$ ▷ Obtain visual tokens from cache
    $\mathbf{E}^T \leftarrow \text{RoPE}(\Delta t)$ ▷ Create temporal-aware indicator token
    $\mathbf{E}^M \leftarrow \text{Tokenizer}(\mathcal{M})$
    $\mathbf{E}^{\text{CoT}} \leftarrow \text{LLM.forward}(\mathbf{E}^I, \mathbf{E}^T, \mathbf{E}^V, \mathbf{E}^M)$
    if $\mathbf{E}^{\text{CoT}} =$ `<think_on>` **then**
      $c_t \leftarrow \text{LLM.generate}(\mathbf{E}^I, \mathbf{E}^T, \mathbf{E}^V, \mathbf{E}^M, \mathbf{E}^{\text{CoT}})$
      $\mathcal{M} \leftarrow \text{UpdateMemory}(\mathcal{M}, c_t)$
    end if
    $h_t^{\text{pred}} \leftarrow \mathbf{E}^{\text{CoT}}[-1]$ ▷ Use the hidden state of the last token as input for the action model
    $\hat{\tau}_t \leftarrow A_\theta(h_t^{\text{pred}})$ ▷ Generate the next trajectory
    if $\hat{\tau}_t =$ `stop` **then**
      break
    else
      ExecuteAction($\hat{\tau}_t$)
    end if
  end while
end procedure

4 Data Collection

Our framework is trained on Nav-AdaCoT-2.9M, a large-scale dataset we constructed, supplemented by public open-world video datasets. Tab. 1 presents a statistical comparison between Nav-AdaCoT-2.9M and existing public embodied navigation datasets, evaluating metrics such as scene number, task type, total steps, CoT annotation number, and action modalities. Notably, our dataset surpasses others in scene number, task variety, and input modality richness. It also features the largest count of CoT annotations to date. Furthermore, Nav-AdaCoT-2.9M employs trajectory-based annotation, which provides finer-grained supervision compared

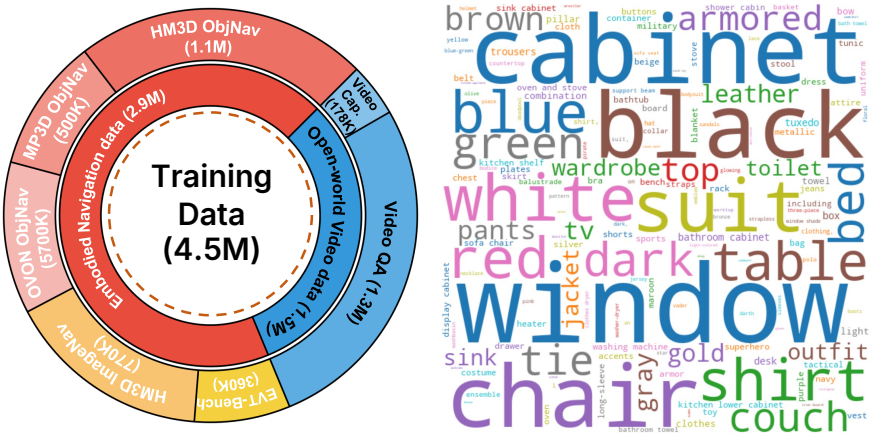

Figure 3 Data distribution and instruction word cloud for the VLingNav training dataset.

to discrete action-based datasets.

4.1 Embodied Navigation Data

4.1.1 Navigation Data Generation

To ensure both diversity and comparability, we construct our training data from several widely used embodied navigation benchmarks.

Object-Goal Navigation. We use data from three benchmarks:

- HM3D ObjNav [41]: For this category-level search task, we utilize a subset of the human demonstration data provided by Habitat-Web [42].
- MP3D ObjNav [7]: We collect shortest-path trajectories to serve as training data.
- HM3D OVON [66]: For this zero-shot, open-vocabulary task, we also collect shortest-path trajectories.

Visual Tracking. We leverage EVT-Bench [52] to curate a multi-person indoor tracking dataset.

Image-Goal Navigation. We use the HM3D Instance ImageNav [24] benchmark. For this, we also generate shortest-path trajectories and derive step-by-step action labels.

Leveraging these existing resources, we propose Nav-AdaCoT-2.9M, a large-scale dataset encompassing 2.9M step-by-step adaptive Chain-of-Thought trajectories. Distinct from prior datasets that predominantly furnish only instructions and expert action labels, Nav-AdaCoT-2.9M explicitly integrates structured reasoning that is aligned with observations and instructions. This design effectively bridges the domains of perception, language, and action. As the cornerstone for the supervised fine-tuning phase of VLingNav, this dataset facilitates the acquisition of structured reasoning capabilities in VLingNav prior to the reinforcement learning-based post-training.

4.1.2 Autonomous Adaptive CoT labeling pipeline

We propose an autonomous adaptive Chain-of-Thought data labeling pipeline, specifically designed to construct high-quality CoT labels for embodied navigation and reasoning tasks. This pipeline leverages the reasoning capabilities of Vision-Language Models to generate coherent, step-by-step CoT rationales that justify navigation decisions in complex environments. As illustrated in Fig. 4, we applied adaptive CoT labeling to the entire embodied navigation dataset described in the previous section.

To generate high-quality Chain-of-Thought labels, we designed a composite prompt for Qwen2.5-VL-72B [2] that incorporates five essential components: 1) navigation instructions, 2) egocentric visual stream input

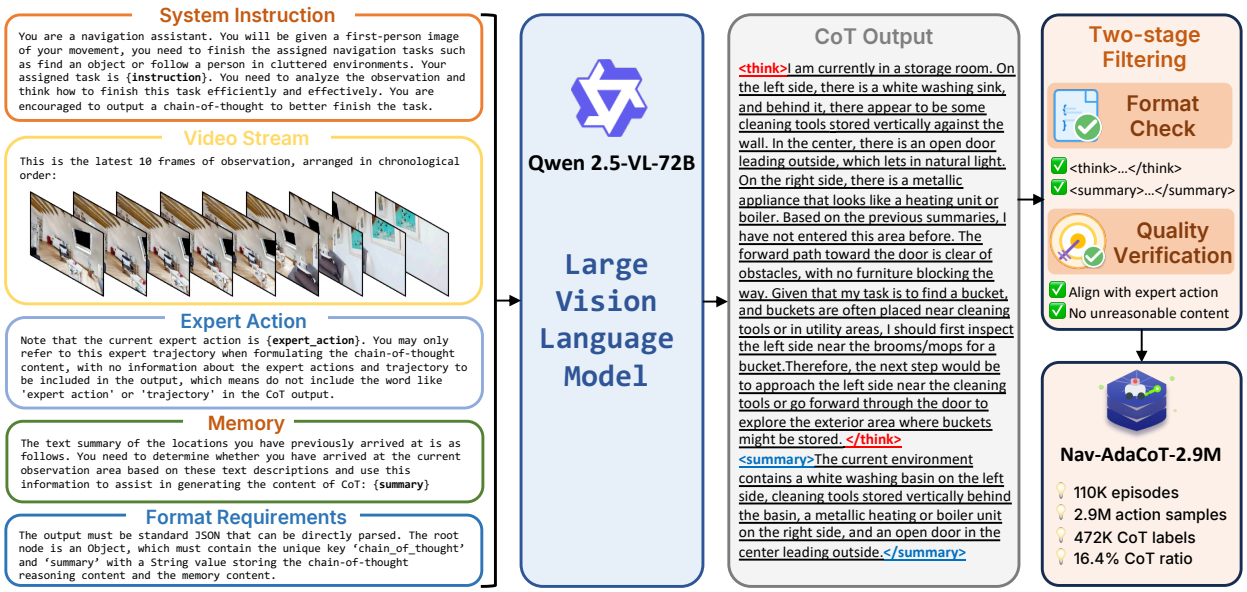

Figure 4 The autonomous adaptive CoT labeling pipeline of VLingNav.

(the most recent 10 frames, included to reduce the computational load of the VLM), 3) prior memory content, 4) expert trajectories at each step, and 5) explicit formatting requirements. This prompt guides the VLM to reason about spatial relationships, environmental constraints, and the semantic meaning of instructions, thereby generating structured, step-by-step CoT sequences. The outputs follow a standardized format: reasoning processes are enclosed within `<think>···</think>` tags, while summaries are contained in `<summary>···</summary>` tags. This formatting ensures transparent alignment among observations, reasoning, and memory. When this pipeline was executed across our diverse set of environments, approximately 472K CoT responses were generated from 2.9M samples. Each response includes a detailed CoT analysis and decision-making process for the navigation scenario, as well as linguistic memory describing the current environmental context. These raw outputs were further refined through a two-stage filtering procedure: 1) Rule-based checks: Responses that were incomplete or logically inconsistent were discarded. 2) Quality verification: Decisions were cross-validated against expert navigation trajectories to ensure accuracy. Following refinement, we constructed the Nav-AdaCoT-2.9M dataset. Serving as the Supervised Fine-Tuning data for VLingNav, this dataset provides rich reasoning trajectories that tightly integrate perception, instruction following, and navigation decision-making.

4.2 Open-World Video Data

Furthermore, co-training with open-world video data has been shown in multiple studies [3, 56, 75] to enhance model generalization and reduce the sim-to-real transfer gap. Consistent with these findings, we incorporate a variety of publicly available open-world video datasets [1, 13, 81] into our training data. Beyond prior efforts, our approach not only improves general visual understanding but also further strengthens adaptive reasoning through additional adaptive CoT annotations. Specifically, we utilize three datasets, LLaVA-Video-178K [81], Video-R1 [13], and ScanQA [1], comprising a total of 1.6M samples, and construct an adaptive CoT-based video dataset by categorizing samples according to difficulty. In particular, the Video-R1 dataset, which contains relatively challenging video QA pairs, is organized as a CoT-annotated subset, whereas the other two datasets are formatted as non-CoT subsets. This design enables the model to further develop the ability to autonomously decide whether reasoning is required for a given input.

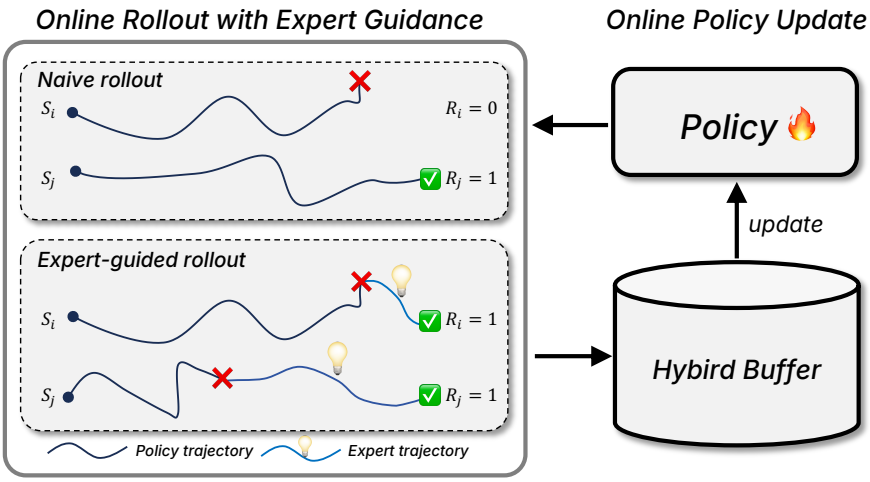

Figure 5 Online post-training with a hybrid rollout procedure.

4.3 Dataset Statistics

Ultimately, the training dataset for VLingNav comprises the two aforementioned types of datasets, totaling 4.5M training samples. Specifically, it includes 2.9M samples of embodied navigation data and 1.6M samples of open-world video data, with the detailed data distribution illustrated in Fig. 3.

5 Training Recipe

5.1 Model Pre-train

The VLM backbone used in VLingNav does not natively support adaptive reasoning. To address this, we first conduct a pre-training stage on our custom open-world adaptive CoT video dataset (detailed in Sec. 4). Following standard VLM training paradigms, we fine-tune the model for a single epoch. This process equips the model with the foundational ability to perform adaptive visual reasoning. The training is supervised using a standard cross-entropy (CE) loss, applied at the token level.

5.2 Supervised Fine-Tuning

Following the pre-training phase, we perform supervised fine-tuning (SFT) to establish robust navigation and video reasoning capabilities. Specifically, we train the model using standard imitation learning on a combined dataset that integrates our embodied navigation data with the open-world video data. This co-training strategy ensures the model retains general-purpose visual reasoning while acquiring task-specific navigation skills. The training objective can be formalized as:

$$\min_{\theta} \quad \mathcal{L}_{\text{SFT}}(\theta) = \alpha \mathcal{L}_{\text{MSE}}(\hat{\tau}_t, \tau_t^{gt}) + (1-\alpha)\mathcal{L}_{\text{CE}}(E_t^{pred}, E_t^{gt}) \tag{6}$$

where $\mathcal{L}_{\text{MSE}}$ is the Mean Squared Error loss that supervises the predicted action trajectory $\hat{\tau}_t$ against the ground-truth trajectory τ_t^{gt}, $\mathcal{L}_{\text{CE}}$ is the Cross-Entropy loss that supervises the generation of all textual outputs, including both the CoT reasoning and the VQA responses. α is a hyperparameter that balances the contribution of the two loss components.

5.3 Online Expert-guided Post-training

To address the limitations of offline imitation learning, such as covariate shift, and to better align the VLM's high-level representations with the closed-loop robot continuous action, we introduce an online post-training stage. Starting from the SFT checkpoint, the agent actively interacts with the simulation environment to collect fresh, on-policy trajectories. The policy is then updated using a hybrid objective function. This

objective combines outcome-driven optimization with expert-guided supervision. This dual approach allows the model to explore more effective strategies while preventing catastrophic forgetting of the expert policy.

5.3.1 Probabilistic Continuous Action Model

Existing VLA architectures often employ discrete tokenization for actions, which sacrifices precision; others use generative models like diffusion or flow matching, which incur high computational costs due to iterative denoising. To address this trade-off between high-precision continuous control and efficient inference, we propose a lightweight probabilistic projection head.

Let $\mathbf{h}_t$ denote the visual-linguistic features extracted from the VLM backbone at timestep t. We parameterize the policy $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ as a multivariate Gaussian distribution. Specifically, the action head projects $\mathbf{h}_t$ to predict the mean $\boldsymbol{\mu}_\theta(\mathbf{h}_t)$ and the logarithm of the standard deviation $\log \boldsymbol{\sigma}_\theta(\mathbf{h}_t)$:

$$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t) = \mathcal{N}\left(\boldsymbol{\mu}_\theta(\mathbf{h}_t), \operatorname{diag}\left(\boldsymbol{\sigma}_\theta(\mathbf{h}_t)^2\right)\right) \tag{7}$$

During online post-training rollout, stochastic exploration is implemented by sampling actions $\mathbf{a}_t$ from the policy distribution $\mathbf{a}_t \sim \pi_\theta(\cdot \mid \mathbf{s}_t)$. In contrast, for the validation phase, deterministic execution is adopted, where actions $\mathbf{a}_t$ are set to the mean value $\mathbf{a}_t = \boldsymbol{\mu}_\theta(\mathbf{h}_t)$ of the policy's action distribution conditioned on the hidden state $\mathbf{h}_t$.

5.3.2 Hybrid Rollout

To balance exploration with successful task completion, we employ a hybrid data collection strategy. As illustrated in Fig. 5, we alternate between two rollout modes:

- Naive rollout: The current policy π_θ interacts with the environment independently. We store the complete interaction trajectories $\tau = \{(\mathbf{s}_t, \mathbf{a}_t, r_t)\}$, and only successful trajectories are filtered out and incorporated into the hybrid buffer. As on-policy data, this dataset accurately reflects the current policy's capabilities and provides high-quality positive examples for reinforcing successful action sequences.
- Expert-guided rollout: To address inefficient exploration and mitigate erroneous behaviors, the system incorporates an expert policy π^* (implemented via a Shortest Path planner in simulator). When the agent triggers an irrational condition (*e.g.*, oscillating or stuck for k steps, here $k = 15$) or eventually fails. By taking control and demonstrating a recovery path, the expert provides high-quality, corrective trajectories. These demonstrations are then added to the hybrid buffer, enriching it with valuable examples of how to escape difficult states and improving overall agent robustness.

5.3.3 Online Fine-tuning with Augmented Loss

Pure reinforcement learning can be unstable and sample-inefficient under sparse rewards and long horizons, while pure imitation learning may overfit to the expert state distribution and suffer from covariate shift. We therefore adopt a demonstration-augmented online post-training scheme [40], where interaction data provides an outcome-driven learning signal and expert-guided trajectories provide a stabilizing supervised signal. Specially, we optimize the following composite objective:

$$\begin{aligned} \min_\theta \quad & \mathcal{L}_{\text{post}}(\theta) = \lambda \mathcal{L}_{\text{RL}}(\theta) + (1-\lambda)\mathcal{L}_{\text{SFT}}(\theta) \\ \text{where} \quad & \mathcal{L}_{\text{RL}}(\theta) = -\mathbb{E}_t\left[\min\left(r_t(\theta)A_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\, A_t\right)\right], \end{aligned} \tag{8}$$

where $\mathcal{L}_{\text{RL}}$ is a PPO-style policy-gradient objective,and we use REINFORCE++ [18] to calculate the advantage A_t. $\mathcal{L}_{\text{SFT}}$ is the imitation loss defined in Eq. 6.

5.4 Implementation Details

5.4.1 Training Details

VLingNav is trained on a cluster with 128 NVIDIA A100 GPUs using a three-stage training pipeline. In the first stage, we leverage open-world video data for pre-training to endow the model with adaptive general visual reasoning capabilities. Consistent with standard VLM practices [29], this pre-training runs for a single epoch. In the second stage, all embodied navigation data and open-world video data are mixed and randomly shuffled for co-training 20K steps with a total batch size of 512. In the online post-training phase, the policy undergoes 10 rollout iterations of updates using the training datasets of HM3D OVON, HM3D Instance ImageNav and EVT-Bench DT benchmarks. For each iteration, we use the current policy to collect 128 episodes of on-policy data, which are then added to the hybrid buffer before the model is updated. For open-world video data, all videos are sampled at 1 FPS to reduce redundancy between consecutive frames. Throughout all stage training, only the visual encoder's parameters are frozen; all other components are updated. The hyperparameters are set to $\alpha = 0.5$ and $\lambda = 0.01$, which is determined by the scale of different losses.

5.4.2 Inference Details

During inference, we maintain a compact and consistent model architecture by not using task-specific tokens for task partitioning. Instead, at each step, the model autoregressively predicts a CoT indicator token. Based on this indicator, it may then generate CoT content. Finally, the hidden state corresponding to the last generated token is fed into the action module, which predicts the robot's future motion trajectory.

6 Experiments

To comprehensively evaluate the performance of VLingNav, we conducted a series of extensive experiments in both simulation and the real world. We first quantitatively compare VLingNav against state-of-the-art methods on several standard embodied navigation benchmarks. We then conduct detailed ablation studies to validate the effectiveness of each of our proposed key components. Furthermore, we have validated that the proposed model framework and training recipe in VLingNav demonstrate emergent generalization capabilities across diverse domains and tasks. Finally, we demonstrate VLingNav's ability to transfer to a real-world robot and complete practical navigation tasks in a zero-shot manner, verifying its real-world generalization and utility.

6.1 Experiment Setups

6.1.1 Benchmarks.

Our method is evaluated on multiple public benchmarks, including those for Object Goal Navigation (HM3Dv1 ObjNav, HM3Dv2 ObjNav, MP3D ObjNav, and HM3D OVON), Embodied Visual Tracking (EVT-Bench), and Image Goal Navigation (HM3D Instance ImageNav). Notably, a shared model checkpoint is used across all tasks, with no additional fine-tuning performed for any individual task.

6.1.2 Baselines.

We conduct a comprehensive comparison of VLingNav against current state-of-the-art models, categorized into three groups: (1) modular methods, often separate the model into perception, mapping and planning, *i.e.* [8, 25, 62, 63, 66, 67, 74, 79], (2) end-to-end small-scale models often leverage a pre-trained network for visual feature extraction, which are then integrated with a policy network to output robot actions, *i.e.* [42, 43, 64, 66, 69, 84], and (3) VLA models [30, 31, 52, 74, 75, 89].

6.1.3 Metrics.

To evaluate navigation performance, we use standard metrics from public benchmarks, including Success Rate (SR), Success-weighted Path Length (SPL), Tracking Rate (TR), and Collision Rate (CR).

Table 2 Performance on object goal navigation. Comparison on HM3D ObjNav [41] and MP3D ObjNav [7] benchmarks. The **best** and the second best results are denoted by **bold** and underline.

Method	HM3Dv1		HM3Dv2		MP3D	
	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
VLFM [65]	52.5	30.4	63.6	32.5	36.4	17.5
SG-Nav [62]	54.0	24.9	49.6	25.5	40.2	16.0
L3MVN [67]	54.2	25.5	36.6	15.7	-	-
UniGoal [63]	54.5	25.1	-	-	41.0	16.4
Habitat-Web [66]	57.6	23.8	31.6	8.5		
InstructNav [33]	-	-	58.0	20.9	-	-
ApexNav [79]	59.6	33.0	76.2	38.0	39.2	17.8
OVRL [60]	62.0	26.8	-	-	28.6	7.4
OVRL-v2 [59]	62.8	28.1	-	-	-	-
LFG [46]	68.9	36.0	-	-	-	-
PirlNav [43]	70.4	34.1	-	-	-	-
FiLM-Nav [64]	61.7	37.3	77.0	**41.3**	-	-
CogNav [6]	72.5	26.2	-	-	46.6	16.1
Uni-NaVid [75]	73.7	37.1	-	-	-	-
VLingNav (SFT)	70.6	38.2	76.4	32.6	47.4	25.8
VLingNav	**79.1**	**42.9**	**83.0**	40.5	**58.9**	**26.5**

Table 3 Performance on object goal navigation. Comparison on HM3D-OVON [66] benchmark. The **best** and the second best results are denoted by **bold** and underline.

Method	Val Seen		Val Seen Synonyms		Val Unseen	
	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
BC	11.1	4.5	9.9	3.8	5.4	1.9
DAgger	11.1	4.5	9.9	3.8	5.4	1.9
RL	18.1	9.4	15.0	7.4	10.2	4.7
BCRL	39.2	18.7	27.8	11.7	18.6	7.5
DAgRL	41.3	21.2	29.4	14.4	18.3	7.9
VLFM [65]	35.2	18.6	32.4	17.3	35.2	19.6
DAgRL+OD [66]	38.5	21.1	39.0	21.4	37.1	19.8
Uni-NaVid [75]	41.3	21.1	43.9	21.8	39.5	19.8
TANGO [90]	-	-	-	-	35.5	19.5
FiLM-Nav [64]	44.9	24.5	40.1	23.1	40.8	24.4
MTU3D [89]	55.0	23.6	45.0	14.7	40.8	12.1
NavFoM [74]	37.7	25.5	43.3	29.9	43.6	**31.3**
Nav-R1 [31]	58.4	26.3	48.1	23.1	42.2	20.1
VLingNav (SFT)	45.9	26.5	44.8	27.1	41.5	22.4
VLingNav	**59.3**	**29.7**	**56.8**	**30.1**	**50.1**	24.6

6.2 Simulation Experiments

6.2.1 Object Goal Navigation

First, we compared the performance metrics of VLingNav with those of state-of-the-art methods on the Object Goal Navigation task. Specifically, our evaluations were conducted across multiple publicly available benchmarks including HM3Dv1, HM3Dv2, MP3D and HM3D OVON.

As presented in Tab. 2, VLingNav achieved SOTA performance on three closed-vocabulary benchmarks, significantly outperforming prior methods on both SR and SPL metrics. On HM3Dv1, VLingNav reaches 79.1

SR and 42.9 SPL, improving over previous SOTA video-based VLA model Uni-NaVid (73.7/37.1) by +5.4 SR (+7.3%) and +3.9 SPL (+15.6%). A comparable performance improvement is also observed in HM3Dv2, where VLingNav achieves 83.0 SR and 40.5 SPL, surpassing FiLM-Nav (77.0/41.3) by +6.0 SR (+7.8%). It is noted that the SPL result achieved by our method is slightly lower than that of FiLM-Nav. This discrepancy primarily arises because the FiLM-Nav model only selects the next frontier position and then relies on a shortest path planner to reach it—an approach that confers greater advantages in the simulator compared to our method, which directly outputs trajectory-based actions. On the MP3D benchmark—where long-range exploration scenarios predominate—VLingNav achieves an SR of 58.9 and an SPL of 26.5. These results significantly outperform those of the prior SOTA methods CogNav (46.6/16.1) and ApexNav (39.2/17.8). Specifically, our method yields an +26.4% improvement in SR and a substantial +32.8% enhancement in SPL. This impressive result demonstrates that VLingNav possesses robust exploration and memory capabilities, validating its effectiveness in complex long-range navigation tasks. Collectively, these results show that VLingNav not only exhibits enhanced object-exploration capabilities in diverse and challenging unseen environments, but also produces substantially shorter and more efficient trajectories across benchmark tests, highlighting the benefits of adaptive reasoning and long-horizon linguistic memory in the Object Goal Navigation task.

To further validate the generalization capability of VLingNav, we evaluate its performance on HM3D OVON—a more challenging open-vocabulary object navigation benchmark. This benchmark comprises three distinct test splits: (1) *val seen*, which includes object categories present in the training set; (2) *val seen synonym*, which consists of goal categories synonymous with those encountered during training; and (3) *val unseen*, which contains object categories not present in the training dataset. As illustrated in Tab. 3, VLingNav achieves the best performance across all three test splits, with SRs improved by 0.9 (+1.5%), 8.7 (+18.1%), and 6.6 (+15.1%) respectively compared to the previous SOTA methods. This result demonstrates the strong cross-domain generalization capability of VLingNav.

6.2.2 Embodied Visual Tracking

Table 4 Performance on embodied visual tracking. Comparison on EVT-Bench [52]. †: Use GroundingDINO [32] as the open-vocabulary detector. ‡: Use SoM [61] with GPT-4o [36] as the visual foundation model.

Method	Single Target Tracking			Distracted Tracking		
	SR↑	TR↑	CR↓	SR↑	TR↑	CR↓
IBVS† [17]	42.9	56.2	3.75	10.6	28.4	6.14
PoliFormer† [69]	4.67	15.5	40.1	2.62	13.2	44.5
EVT [84]	24.4	39.1	42.5	3.23	11.2	47.9
EVT‡ [84]	32.5	49.9	40.5	15.7	35.7	53.3
Uni-NaVid [75]	53.3	67.2	12.6	31.9	50.1	21.3
TrackVLA [52]	85.1	78.6	<u>1.65</u>	57.6	63.2	5.80
NavFoM [74]	86.0	80.5	-	61.4	68.2	-
NavFoM* [74]	88.4	80.7	-	62.0	67.9	-
TrackVLA++ [30]	86.0	<u>81.0</u>	2.10	<u>66.5</u>	68.8	**4.71**
VLingNav (SFT)	<u>87.2</u>	78.9	**1.23**	66.1	<u>69.7</u>	<u>4.78</u>
VLingNav	**88.4**	**81.2**	2.07	**67.6**	**73.5**	5.51

To evaluate the efficacy of the proposed method for the Embodied Visual Tracking (EVT) task, we conduct a comprehensive comparative analysis on the EVT-Bench. Specifically, we evaluate two representative and challenging splits: (1) *Single-Target Tracking*: The agent must continuously track a single designated target in complex unseen environments. (2) *Distracted Tracking*: A more complex scenario in which the agent must sustain stable tracking of the correct target under instructions while resisting interference from multiple distractors. As illustrated in Tab. 4, VLingNav demonstrates SOTA performance across both splits. In the *Single Target Tracking* task, VLingNav achieves an SR of 88.4 and a TR of 81.2, matching or slightly surpassing the previous best methods like NavFoM and TrackVLA++. In the more challenging *Distracted Tracking* scenario, VLingNav establishes a clear advantage, achieving an SR of 67.6 and a TR of 73.5. This represents a significant improvement of 1.1 (+1.7%) in SR and 4.7 (+6.8%) in TR compared to the previous

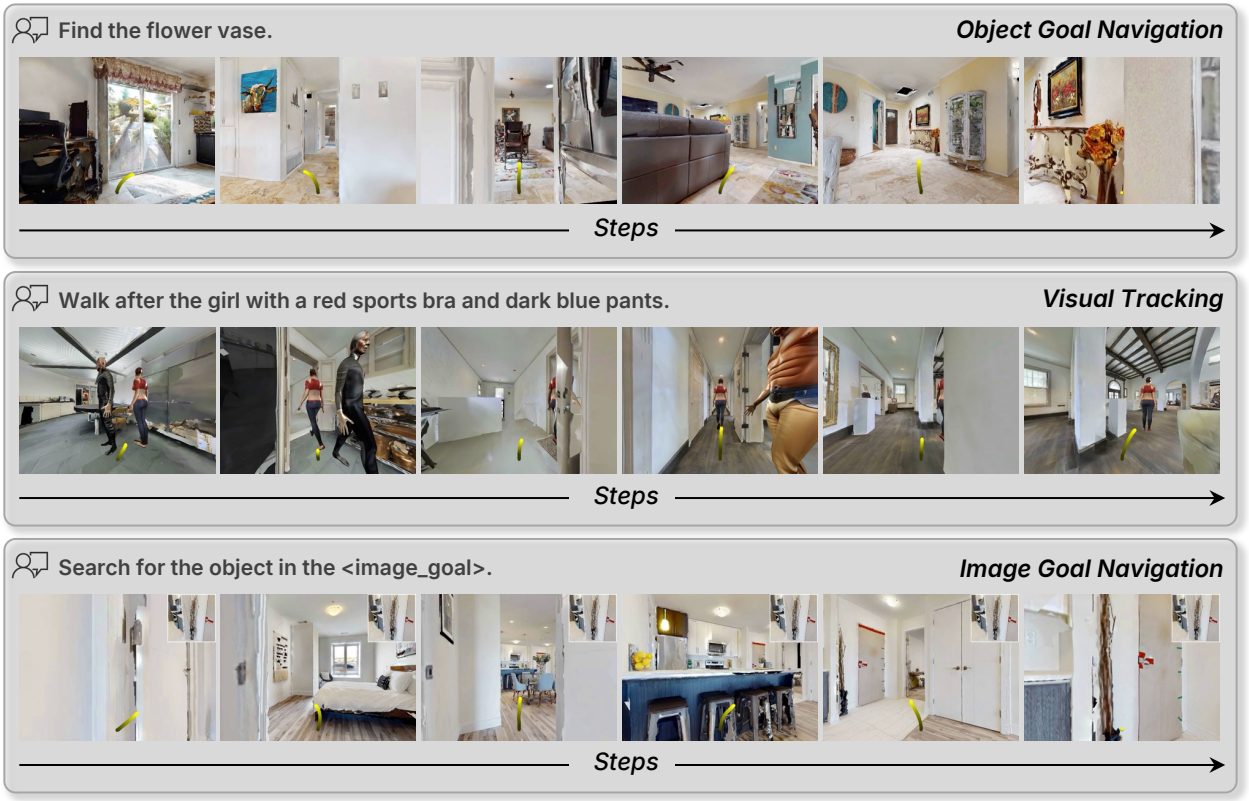

Figure 6 Performance visualization of VLingNav across various navigation benchmarks.

Table 5 Performance on image goal navigation. Comparison on HM3D Instance ImageNav benchmark [41]. The **best** and the second best results are denoted by **bold** and underline.

Method	HM3D Instance ImageNav	
	SR↑	SPL↑
Krantz et al. [24]	8.3	3.5
OVRL-v2-IIN [59]	24.8	11.8
PSL [50]	23.0	11.4
GOAT [8]	37.4	16.1
Mod-IIN [25]	56.1	23.3
UniGoal [63]	60.2	23.7
VLingNav (SFT)	51.1	32.6
VLingNav	**60.8**	**37.4**

SOTA method TrackVLA++. Notably, VLingNav outperforms NavFoM with multi-view setting while using only a monocular camera, demonstrating robust tracking and precise recognition. These results strongly validate the superior tracking capability and robustness of our approach, especially in complex environments with distractors.

6.2.3 Image Goal Navigation

To further evaluate the Image Goal Navigation capabilities of our model, we evaluate VLingNav on the HM3D Instance ImageNav benchmark. This task requires the agent to navigate to a specific object instance depicted in a goal image within complex and unseen environments. As presented in Table 5, VLingNav achieves

Figure 7 Real-world robot platform setup.

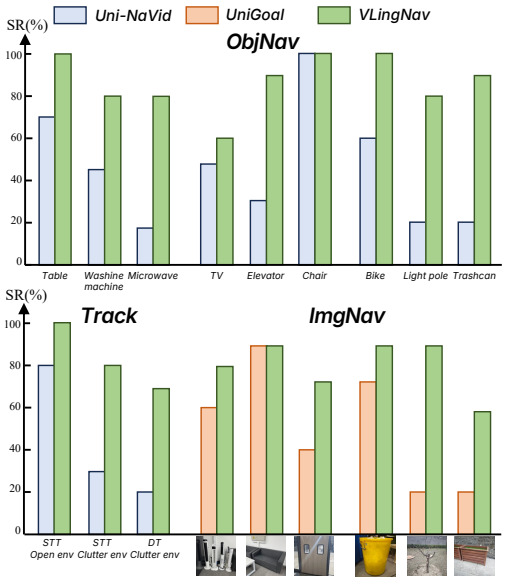

Figure 8 Real-world experiment results.

state-of-the-art results on this benchmark. It achieves an SR of 60.8, which is slightly higher than the previous SOTA method UniGoal (60.2/23.7). Note that, UniGoal leverages the LightGlue [28] keypoint-matching algorithm as an additional criterion, whereas VLingNav relies solely on the model's implicit reasoning. More impressively, VLingNav demonstrates a substantial improvement in navigation efficiency, achieving an SPL of 37.4. This represents a remarkable 13.7 (+57.8%) improvement over UniGoal. This significant gain in SPL underscores our model's ability to not only successfully find the target instance but also to do so via much more direct and efficient paths, highlighting the advanced reasoning and planning abilities of VLingNav.

6.2.4 Visualization Results

Fig. 6 illustrates several visualizations of VLingNav on the simulation benchmarks, including the robot's egocentric visual observations, a top-down scene map, input instructions, and the outputs of adaptive CoT and the predicted trajectory. These examples show that our model efficiently accomplishes multiple embodied navigation tasks while adaptively generating CoT reasoning. This not only enhances interpretability but also improves the overall quality of the navigation process.

6.3 Real-World Experiments

6.3.1 Robot Platform and Deployments

We provide a visualization of our robot platform in Fig. 7. The platform is based on the Unitree Go2 quadruped robot, equipped with an Intel RealSense D457 camera on its head. In our work, we only utilize the RGB frames with a resolution of 1280×800 from the camera, under a horizontal field of view (HFOV) of 90°. Additionally, a portable Wi-Fi is mounted on the back of the robot to enable communication with the remote server through the Internet.

VLingNav is deployed on a remote server equipped with an NVIDIA RTX 4090 GPU. During real-world deployment, the server receives the instructions and images captured by the camera via the Internet. To ensure efficient communication, the images are compressed before transmission. After receiving the incoming data, the model performs inference and predicts the future trajectory, which is then transmitted to the quadruped robot for execution. Given that real-world navigation is an online process, we cache visual tokens from historically observed images. As a result, at each step the model only encodes the latest frame, which significantly improves inference efficiency. Furthermore, by leveraging VLingNav's visual memory compression strategy, our model maintains an inference latency of under 300 ms across 500 video frames. Including

communication overhead (approximately 100 ms), VLingNav achieves an effective inference speed of around 2.5 FPS during long-horizon, real-world robot experiments.

Upon receiving the predicted trajectory, the robot employs a nonlinear model predictive control (NMPC) module for trajectory tracking [15]. Formulating the task as an optimization problem based on a kinematic unicycle model, the controller computes optimal linear and angular velocities over a receding horizon.

6.3.2 Object Goal Navigation

We evaluated the Object Goal Navigation performance of VLingNav against the SOTA method Uni-NaVid, across three representative scenarios: home, office, and outdoor environment. For each scenario, we selected three distinct target objects: (i) the table, washing machine, and microwave for the home environment; (ii) the TV, elevator, and trashbin for the office environment; (iii) the bike, light pole, and tree for the outdoor environment. To mitigate the effects of randomness, we conducted 10 repeated trials for each target object. As shown in Fig. 8, VLingNav achieves a significantly higher success rate than Uni-NaVid across all tested scenarios. These results validate the robust object recognition, exploration, and cross-scenario generalization capabilities of our model.

6.3.3 Embodied Visual Tracking

We evaluated the embodied visual tracking performance of our method against Uni-NaVid across three representative scenarios: (i) single-target tracking in open spaces, (ii) single-target tracking in cluttered indoor environments, and (iii) distracted tracking in crowded scenes with frequent occlusions and nearby distractors. To mitigate randomness, we conducted 10 repeated trials per scenario. As shown in Fig. 8, our method consistently outperforms Uni-NaVid in tracking success rate, with the largest margins appearing in the distracted setting where transient occlusions and target switches are common. These results validate the effectiveness of our adaptive reasoning for re-identification after occlusion and the benefit of precise trajectory control, highlighting strong generalization to dynamic, cluttered environments.

6.3.4 Image Goal Navigation

We further evaluated Image Goal Navigation by comparing our method with UniGoal across three representative scene categories—home, office, and outdoor environments. For each category, we selected two image-specified goals and conducted 10 repeated trials per goal. As shown in Fig. 8, our approach achieves a substantially higher success rate than UniGoal in all categories. These results suggest that multi-task training induces robust cross-modal grounding from text to images, while the combination of Adaptive CoT and linguistic memory supports reliable localization and efficient long-horizon navigation to visually specified targets under variations in camera intrinsics, viewpoints, and lighting.

6.3.5 Visualization Results

Real-world experimental results are shown in Fig. 9, where we evaluate the navigation capabilities of VLingNav under challenging scenarios. Specifically, we test the model across three representative scenario categories: office environments, household settings, and outdoor scenes. Within each category, we assess three core capabilities: object goal navigation, embodied visual tracking, and image goal navigation. Notably, the model weights deployed on the real-world robot are the same as those used in the simulation experiments described in the previous section; no additional fine-tuning on real-world data is performed. The results demonstrate that VLingNav exhibits strong sim-to-real transfer in both recognition and planning while sustaining high-frequency inference in real-world scenarios, enabling zero-shot deployment in complex environments.

6.4 Emergence of Cross-Task and Cross-Domain Capabilities

Joint training on multi-task navigation datasets leads VLingNav to exhibit emergent behaviors that generalize beyond any single task, yielding both cross-task and cross-domain capabilities.

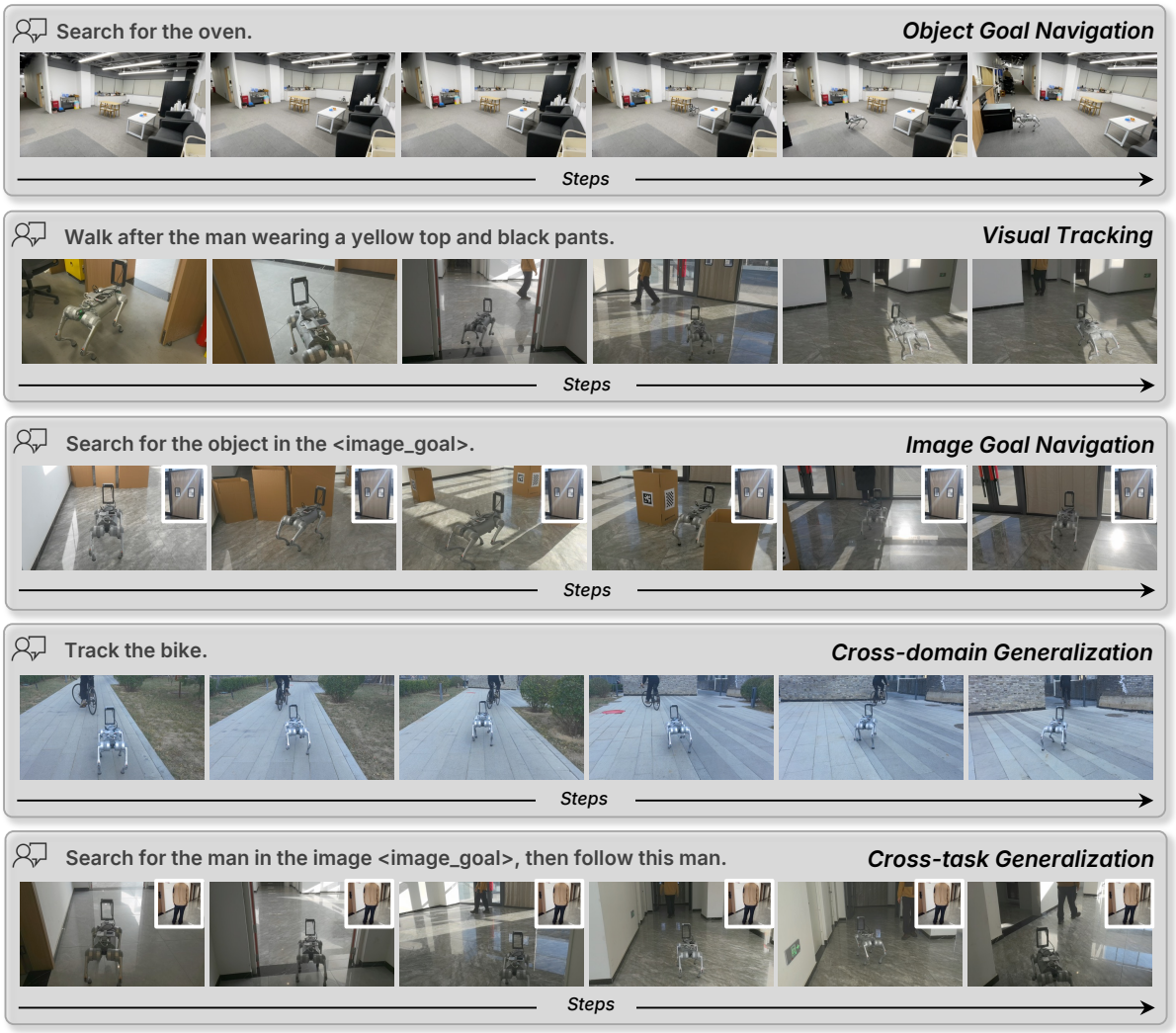

Figure 9 Qualitative performance of VLingNav in real-world deployments.

6.4.1 Cross-task Performance

We observe clear cross-task transfer in real-world experiments (Fig. 9). For instance, although the visual tracking task contains only language-format instructions, VLingNav can directly track targets specified by image goals in a zero-shot manner. Moreover, it composes behaviors across tasks: (1) Search for a language-described target, then switch to tracking that target. (2) Search for the target in the image goal and subsequently track it after locating it. This compositionality arises from the VLA model's shared, unified architecture and co-training on multi-task navigation datasets. Together, these factors enable the model to learn common navigation priors and transfer them successfully across diverse navigation tasks.

6.4.2 Cross-domain Performance

Second, we observe robust cross-domain generalization. Although trained only to track humans, VLingNav reliably tracks dynamic non-human targets. Moreover, VLingNav successfully localizes and navigates to out-of-distribution objectives specified only by fine-grained textual instructions, including category-ambiguous objects disambiguated by color, spatial constraints, or detailed descriptions. These behaviors indicate that multi-task learning, when co-trained with general visual understanding data, can substantially enhance VLingNav's generalization across domains.

6.5 Ablation Studies

To evaluate the contribution of each component and training strategy in VLingNav, we conducted comprehensive ablation studies. These studies were performed on the ObjectNav task using the HM3D OVON *val unseen* benchmark, the EVT task using the EVT-Bench *Distracted Tracking* benchmark, and the ImageNav task using the HM3D Instance ImageNav *val* benchmark. For consistency, we adhered to the same training procedures and evaluation settings as those used for the full model. Below, we summarize the empirical results and analyze the key findings.

6.5.1 Adaptive CoT

Table 6 Ablation study on Chain-of-Thought strategies. r_{CoT} indicates the average percentage of steps where CoT reasoning is activated.

CoT Strategy	ObjNav		Track			ImageNav		r_{CoT} (%)
	SR↑	SPL↑	SR↑	TR↑	CR↓	SR↑	SPL↑	
w/o CoT	36.2	16.5	62.7	68.5	6.28	56.3	27.3	0.0
Dense CoT (Per-step)	25.3	13.0	59.8	70.1	26.3	19.6	13.2	100.0
Fixed Interval ($k = 5$)	42.5	23.5	**68.5**	**74.2**	9.18	48.2	28.7	20.0
Fixed Interval ($k = 20$)	39.7	19.4	66.2	70.8	11.9	51.3	31.2	5.0
Adaptive CoT (Ours)	**50.1**	**24.6**	67.6	73.5	**5.51**	**60.8**	**37.4**	**2.1**

To assess the impact of different reasoning strategies, we conducted an ablation study detailed in Tab. 6. The results show that both a complete lack of reasoning ("w/o CoT") and exhaustive reasoning at every step ("Dense CoT") lead to suboptimal performance. While fixed-interval reasoning provides a moderate improvement, it remains inflexible. Our proposed Adaptive CoT strategy demonstrates clear superiority. It achieves the highest performance across all benchmarks. Remarkably, it accomplishes this while maintaining an exceptionally low reasoning frequency ($r_{CoT} = 2.1\%$), far more efficient than even the sparse fixed-interval method. This highlights that dynamically and intelligently activating reasoning only when needed is crucial for creating high-performing, efficient embodied agents.

6.5.2 Visual-assisted Linguistic Memory

Table 7 Ablation study on memory modalities.

Memory Mode	ObjNav		Track			ImageNav	
	SR↑	SPL↑	SR↑	TR↑	CR↓	SR↑	SPL↑
w/o Memory	15.4	3.5	37.5	59.1	**1.90**	21.0	3.7
Visual-only	45.2	20.3	66.8	70.6	7.85	57.9	33.7
Language-only	18.8	4.4	40.2	55.2	3.25	23.3	7.5
VLingMem (Ours)	**50.1**	**24.6**	**67.6**	**73.5**	5.51	**60.8**	**37.4**

We conducted an ablation study to evaluate the VLingMem module and assess how long-horizon context affects navigation performance. As summarized in Tab. 7, removing the memory module entirely ("w/o Memory") leads to a substantial performance drop. This is particularly pronounced in large or multi-room layouts, where agents frequently get stuck in loops or revisit dead ends. Using a naive replay buffer that stores only visual features ("Visual-only") or only linguistic memory ("Linguistic-only") partially recovers performance but remains inferior to our full approach. In contrast, our proposed VLingMem achieves the best results with minimal latency overhead. Qualitatively, VLingMem enables the agent to remember the environment layout and avoid revisiting explored regions, yielding higher success rates with more efficient paths.

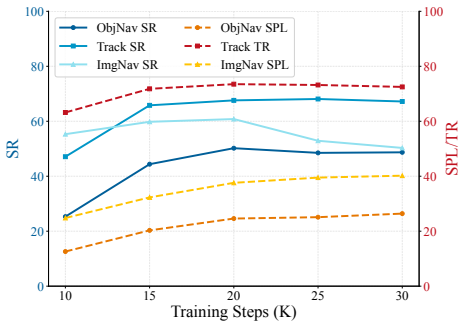

Figure 10 Ablation study on training steps.

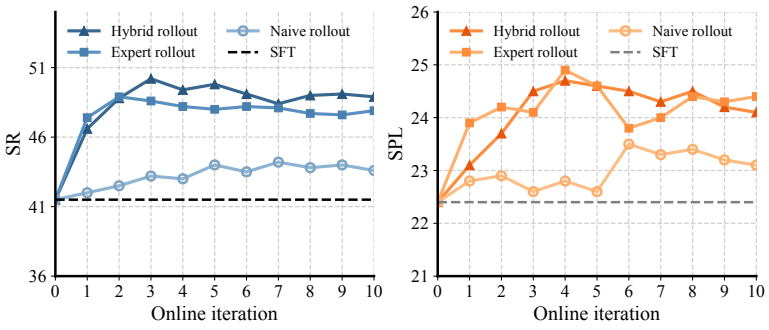

Figure 11 Ablation study on online post-training iteration steps.

6.5.3 Co-train with Open-world Video Data

Table 8 Ablation study on open-world video co-training.

Training Data	ObjNav		Track			ImageNav	
	SR↑	SPL↑	SR↑	TR↑	CR↓	SR↑	SPL↑
w/o Co-training	43.1	20.6	66.5	70.2	7.62	50.2	32.7
w/ Co-training	**50.1**	**24.6**	**67.6**	**73.5**	**5.51**	**60.8**	**37.4**

We further evaluated the impact of co-training with open-world video data. As presented in Tab. 8, the results demonstrate a significant performance improvement compared to the model trained solely on embodied navigation data. This co-training strategy effectively enriches the model's semantic priors, thereby enhancing cross-modal grounding and generalization capabilities while notably reducing the sim-to-real gap.

6.5.4 SFT Training Steps

We investigated the relationship between model performance and the number of training steps. As shown in Fig. 10, model performance scales positively with the number of training steps (where 1 epoch $\approx$ 10K training steps). The success rate rises steadily as the model is exposed to more data. Notably, we found that excessive training leads to diminishing returns and eventual performance degradation, likely due to overfitting on the simulation data. This highlights the need for a balanced training strategy that maximizes performance without incurring unnecessary computational cost or risking overfitting.

6.5.5 Online Post-training

We evaluated the effect of our online post-training phase, which follows the SFT stage. Across all benchmarks (Tab. 2,3,4,5), the post-trained VLingNav model significantly outperforms the SFT checkpoint. This phase is critical for teaching the agent to find shortcuts, recover from errors, and handle the distribution shift that occurs beyond the static demonstration data. As shown in Fig. 11, the ablation studies on the rollout strategy demonstrate that the proposed Hybrid Rollout exhibits the highest effectiveness, yielding the optimal performance. Here, we evaluate on the HM3D OVON *val unseen* split. While the Expert Rollout (DAgger-like) also delivers strong performance, it still exhibits a performance gap compared to the Hybrid Rollout. However, the Naive Rollout fails to improve performance, likely due to the sparse reward signals and the long-horizon nature of the task making value estimation too difficult. This confirms that our expert-guided framework successfully optimizes the policy. It uses expert data to correct faulty behaviors while simultaneously using on-policy data to explore and discover better strategies, thereby outperforming pure imitation learning and finding a more robust policy.

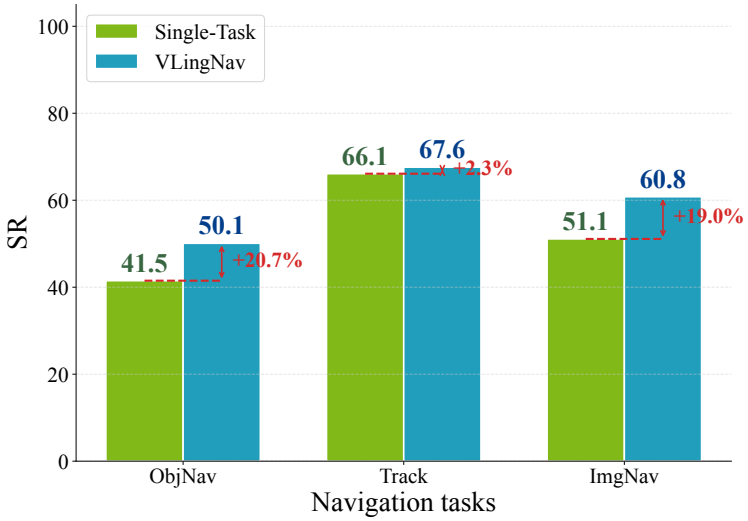

Figure 12 Ablation study on multi-task learning. We present the multi-task synergy of VLingNav and illustrate the performance comparison between models trained on a single task and those trained on multiple tasks.

6.5.6 Multi-task Synergy

Finally, we investigated how jointly training on ObjectNav, EVT, and ImageNav affects generalization. As shown in Fig. 12, models trained on a single task consistently underperform the multi-task model, even on their respective specialized benchmarks. More importantly, this multi-task training strategy fosters emergent cross-domain and cross-task capabilities, leading to a notable performance improvement on out-of-distribution tasks. These findings demonstrate that multi-task learning facilitates the transfer of skills across different domains. This synergy enhances the model's ability to reason and plan across diverse modalities, tasks, and target categories.

7 Discussion

We further discuss the core contributions of our approach, their broader implications, and the key insights revealed by our experimental results.

1) Effectiveness of Adaptive Thinking: Inspired by dual-process theory, our adaptive Chain-of-Thought (AdaCoT) mechanism autonomously allocates "cognitive resources", balancing efficiency and deliberation. When faced with simple, unambiguous navigation scenarios, such as traversing a straight corridor, the model opts for "fast thinking" (`<think_off>`) and directly outputs actions, ensuring fluid and real-time navigation. Conversely, at critical decision positions, in complex environments, or when encountering ambiguity—like choosing a direction at an intersection or searching for an occluded object—the model triggers "slow thinking" (`<think_on>`), generating a detailed reasoning output. This adaptability not only significantly enhances decision quality but also proves that deliberate thought at a small fraction of key steps (shown to be average 2.1% in our experiments) is sufficient to substantially boost overall task success. This finding is crucial for deploying efficient, intelligent navigation on resource-constrained robot platforms.

2) Synergy of Visual-Assisted Linguistic Memory: Navigation is a long-horizon decision-making process by its nature. Our proposed Visual-Assisted Linguistic Memory (VLingMem) module effectively addresses the memory deficiencies in traditional VLA models. Unlike methods that rely solely on implicit visual features, VLingMem distills key visual observations into concise linguistic summaries (`<summary>···</summary>`) and integrates them into the model's context. This design offers two primary advantages. First, linguistic memory is more robust against information decay than compressed visual features, enabling the model to clearly recall critical semantic information such as "I have already checked this room" or "there is a locked door on the left," thereby effectively preventing redundant exploration and inefficient paths. Second, this linguistic memory

forms a powerful synergy with the AdaCoT mechanism. When the model chooses not to engage in detailed CoT, the persistent linguistic memory still provides the necessary historical context, ensuring a coherent decision-making process. This synergy serves as a pivotal factor in enhancing the robustness of VLingNav in long-horizon and complex environments, while significantly improving the efficiency and quality of VLingNav's exploration.

3) Beyond Imitation Learning: The Value of Online Expert-guided RL: Our research confirms that VLA models trained exclusively via imitation learning (SFT) are constrained by both the quality and coverage of expert data. Such models are additionally prone to critical issues, including causal confusion and covariate shift. To address these limitations, we introduce a post-training phase using expert-guided reinforcement learning. VLingNav enables autonomous exploration and policy refinement through real-time online interaction with the environment, while directly deriving rewards from prior expert policy. Compared to rule-based RL, the introduction of expert knowledge allows the model to discover superior or more robust navigation strategies with higher efficiency. The significant performance gains observed in our experiments underscore that RL post-training is a critical step to unlock the full potential of VLA models, transforming them from mere "imitators" into genuine "problem solvers."

4) Generality and Real-world Generalization: A notable achievement of VLingNav is its generality. By training on the large-scale, multi-task Nav-AdaCoT-2.9M dataset, VLingNav achieves state-of-the-art or competitive performance across all these tasks using a single, unified set of model weights, obviating the need for task-specific fine-tuning. This demonstrates that our approach successfully captures the underlying, universal cognitive structures of embodied navigation. Even more encouraging is VLingNav's ability to transfer to real-world robot platforms in a zero-shot manner and complete practical navigation tasks. This indicates that, through high-quality simulation training and a powerful cognitive architecture, the model learns generalizable representations of space, language, and action, rather than just patterns specific to the simulator, successfully bridging the sim-to-real gap. In summary, VLingNav, with its unique cognitive architecture, provides a powerful paradigm for developing more intelligent, efficient, and interpretable embodied agents. It demonstrates the immense potential of combining principles from human cognition, such as adaptive thinking and episodic memory, with advanced machine learning paradigms like VLAs and RL.

8 Conclusion and Limitation

In this work, we introduce VLingNav, a Vision-Language-Action model grounded in linguistic-driven cognition to address critical challenges in embodied navigation. By synergistically integrating adaptive reasoning, multi-modal memory, and online expert-guided RL post-training, VLingNav achieves state-of-the-art performance across a range of embodied navigation benchmarks and can directly transfer to real-world robot platforms in a zero-shot manner.

While VLingNav has achieved progress in embodied navigation, it has several limitations that point to promising directions for future research. First, the current model primarily relies on monocular egocentric observations as input. Due to the limited field of view (FOV) inherent in monocular vision, such input constrains the model's perceptual capabilities. Following recent work [74], we will explore integrating multi-view observations to improve navigation efficiency. Second, the current model adopts a single-system architecture, which restricts its prediction frequency. This limitation impedes rapid decision-making and obstacle handling in highly dynamic environments. To address this, we plan to upgrade VLingNav to a dual-system structure that supports high-frequency action outputs, thereby enhancing fundamental navigation performance, such as obstacle avoidance. Finally, the current approach uses only an MPC-based waypoint controller and lacks a more flexible locomotion model [35]. Incorporating such a locomotion controller could increase movement speed and expand the robot's reachable areas. We therefore plan to integrate locomotion capabilities into VLingNav in future work.

9 Acknowledgements

We sincerely thank Yunke Cai, Haiquan Chen, Shuai Chu, Taifeng Gao, Bo Jiang, Yunfei Li, Yunfei Liu, Tao Wang, Xibin Wu, and Tingshuai Yan for their strong support and fruitful discussions.

References

[1] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 19129–19139, 2022.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923, 2025.

[3] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In 9th Annual Conference on Robot Learning, 2025.

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461, 2022.

[5] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE transactions on robotics, 37(6): 1874–1890, 2021.

[6] Yihan Cao, Jiazhao Zhang, Zhinan Yu, Shuzhen Liu, Zheng Qin, Qin Zou, Bo Du, and Kai Xu. Cognav: Cognitive process modeling for object goal navigation with llms. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9550–9560, 2025.

[7] 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), pages 667–676. IEEE, 2017.

[8] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. Goat: Go to any thing. arXiv preprint arXiv:2311.06430, 2023.

[9] Sheng Chen, Peiyu He, Jiaxin Hu, Ziyang Liu, Yansheng Wang, Tao Xu, Chi Zhang, Chongchong Zhang, Chao An, Shiyu Cai, et al. Astra: Toward general-purpose mobile robots via hierarchical multimodal learning. arXiv preprint arXiv:2506.06205, 2025.

[10] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. In RSS, 2025.

[11] Guilherme N DeSouza and Avinash C Kak. Vision for mobile robot navigation: A survey. IEEE transactions on pattern analysis and machine intelligence, 24(2):237–267, 2002.

[12] Hermann Ebbinghaus. [image] memory: A contribution to experimental psychology. Annals of neurosciences, 20 (4):155, 2013.

[13] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. arXiv preprint arXiv:2503.21776, 2025.

[14] Chen Gao, Liankai Jin, Xingyu Peng, Jiazhao Zhang, Yue Deng, Annan Li, He Wang, and Si Liu. Octonav: Towards generalist embodied navigation. arXiv preprint arXiv:2506.09839, 2025.

[15] Ruben Grandia, Fabian Jenelten, Shao Yang, Farbod Farshidian, and Marco Hutter. Perceptive locomotion through nonlinear model predictive control. IEEE Transactions on Robotics, 39(5):3402–3421, 2023.

[16] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

[17] Meenakshi Gupta, Swagat Kumar, Laxmidhar Behera, and Venkatesh K Subramanian. A novel vision-based tracking algorithm for a human-following mobile robot. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(7):1415–1427, 2016.

[18] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. arXiv preprint arXiv:2501.03262, 2025.

[19] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In Forty-second International Conference on Machine Learning.

[20] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. arXiv preprint arXiv:2507.16815, 2025.

[21] Sertac Karaman, Matthew R Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime motion planning using the rrt. In 2011 IEEE International Conference on Robotics and Automation, pages 1478–1483. ieee, 2011.

[22] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation, 12(4):566–580, 2002.

[23] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645, 2025.

[24] Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances. arXiv preprint arXiv:2211.15876, 2022.

[25] Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10916–10925, 2023.

[26] Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. arXiv preprint arXiv:2509.09674, 2025.

[27] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. arXiv preprint arXiv:2311.01378, 2023.

[28] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In Proceedings of the IEEE/CVF international conference on computer vision, pages 17627–17638, 2023.

[29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In NeurIPS, 2023.

[30] Jiahang Liu, Yunpeng Qi, Jiazhao Zhang, Minghan Li, Shaoan Wang, Kui Wu, Hanjing Ye, Hong Zhang, Zhibo Chen, Fangwei Zhong, et al. Trackvla++: Unleashing reasoning and memory capabilities in vla models for embodied visual tracking. arXiv preprint arXiv:2510.07134, 2025.

[31] Qingxiang Liu, Ting Huang, Zeyu Zhang, and Hao Tang. Nav-r1: Reasoning and navigation in embodied scenes. arXiv preprint arXiv:2509.10884, 2025.

[32] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In European conference on computer vision, pages 38–55. Springer, 2024.

[33] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. arXiv preprint arXiv:2406.04882, 2024.

[34] Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking via reinforcement learning. In International conference on machine learning, pages 3286–3295. PMLR, 2018.

[35] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. Science robotics, 7(62):eabk2822, 2022.

[36] OpenAI. Introducing 4o image generation. https://openai.com/index/introducing-4o-image, 2024. Accessed: 2025-04-29.

[37] Zhangyang Qi, Zhixiong Zhang, Yizhou Yu, Jiaqi Wang, and Hengshuang Zhao. Vln-r1: Vision-language navigation via reinforcement fine-tuning. arXiv preprint arXiv:2506.17221, 2025.

[38] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics, 34(4):1004–1020, 2018.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In International Conference on Machine Learning, pages 8748–8763. PMLR, 2021.

[40] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. arXiv preprint arXiv:1709.10087, 2017.

[41] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John 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. arXiv preprint arXiv:2109.08238, 2021.

[42] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5173–5183, 2022.

[43] Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17896–17906, 2023.

[44] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714, 2024.

[45] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[46] Dhruv Shah, Michael Robert Equi, Błażej Osiński, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. In Conference on Robot Learning, pages 2683–2699. PMLR, 2023.

[47] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

[48] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. arXiv preprint arXiv:2508.19236, 2025.

[49] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 568:127063, 2024.

[50] Xinyu Sun, Lizhao Liu, Hongyan Zhi, Ronghe Qiu, and Junwei Liang. Prioritized semantic learning for zero-shot instance navigation. In European Conference on Computer Vision, pages 161–178. Springer, 2024.

[51] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 5294–5306, 2025.

[52] Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. Trackvla: Embodied visual tracking in the wild. arXiv preprint arXiv:2505.23189, 2025.

[53] Shuo Wang, Yongcai Wang, Wanting Li, Xudong Cai, Yucheng Wang, Maiyue Chen, Kaihui Wang, Zhizhong Su, Deying Li, and Zhaoxin Fan. Aux-think: Exploring reasoning strategies for data-efficient vision-language navigation. arXiv preprint arXiv:2505.11886, 2025.

[54] Xuezhi Wang and Denny Zhou. Chain-of-thought reasoning without prompting. Advances in Neural Information Processing Systems, 37:66383–66409, 2024.

[55] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35:24824–24837, 2022.

[56] 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, 2025.

[57] Yuchen Wu, Pengcheng Zhang, Meiying Gu, Jin Zheng, and Xiao Bai. Embodied navigation with multi-modal information: A survey from tasks to methodology. Information Fusion, 113:102532, 2024.

[58] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. IEEE Transactions on Robotics, 38(4):2053–2073, 2022.

[59] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav. arXiv preprint arXiv:2303.07798, 2023.

[60] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation. In Workshop on Reincarnating Reinforcement Learning at ICLR 2023, 2023.

[61] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. arXiv preprint arXiv:2310.11441, 2023.

[62] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. Advances in neural information processing systems, 37:5285–5307, 2024.

[63] Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Unigoal: Towards universal zero-shot goal-oriented navigation. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 19057–19066, 2025.

[64] Naoki Yokoyama and Sehoon Ha. Film-nav: Efficient and generalizable navigation via vlm fine-tuning. arXiv preprint arXiv:2509.16445, 2025.

[65] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 42–48. IEEE, 2024.

[66] Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. Hm3d-ovon: A dataset and benchmark for open-vocabulary object goal navigation. In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5543–5550. IEEE, 2024.

[67] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3554–3560. IEEE, 2023.

[68] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. arXiv preprint arXiv:2407.08693, 2024.

[69] Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross Girshick, Aniruddha Kembhavi, and Luca Weihs. Poliformer: Scaling on-policy rl with transformers results in masterful navigators. In 8th Annual Conference on Robot Learning.

[70] Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, and Xing Wei. Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. arXiv preprint arXiv:2509.22548, 2025.

[71] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In Proceedings of the IEEE/CVF international conference on computer vision, pages 11975–11986, 2023.

[72] Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. In Forty-second International Conference on Machine Learning.

[73] Jiazhao 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. Robotics: Science and Systems, 2024.

[74] Jiazhao 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, 2025.

[75] 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, 2025.

[76] Lingfeng Zhang, Xiaoshuai Hao, Yingbo Tang, Haoxiang Fu, Xinyu Zheng, Pengwei Wang, Zhongyuan Wang, Wenbo Ding, and Shanghang Zhang. Nava^3: Understanding any instruction, navigating anywhere, finding anything. arXiv preprint arXiv:2508.04598, 2025.

[77] Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A novel memory representation via annotated semantic maps for VLM-based vision-and-language navigation. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, pages 13032–13056, Vienna, Austria, July 2025.

[78] Lingfeng Zhang, Yuecheng Liu, Zhanguang Zhang, Matin Aghaei, Yaochen Hu, Hongjian Gu, Mohammad Ali Alomrani, David Gamaliel Arcos Bravo, Raika Karimi, Atia Hamidizadeh, et al. Mem2ego: Empowering vision-language models with global-to-ego memory for long-horizon embodied navigation. arXiv preprint arXiv:2502.14254, 2025.

[79] Mingjie Zhang, Yuheng Du, Chengkai Wu, Jinni Zhou, Zhenchao Qi, Jun Ma, and Boyu Zhou. Apexnav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion. arXiv preprint arXiv:2504.14478, 2025.

[80] Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. Reinflow: Fine-tuning flow matching policy with online reinforcement learning. arXiv preprint arXiv:2505.22094, 2025.

[81] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. arXiv preprint arXiv:2410.02713, 2024.

[82] Zekai Zhang, Weiye Zhu, Hewei Pan, Xiangchen Wang, Rongtao Xu, Xing Sun, and Feng Zheng. Activevln: Towards active exploration via multi-turn rl in vision-and-language navigation. arXiv preprint arXiv:2509.12618, 2025.

[83] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 1702–1713, 2025.

[84] Fangwei Zhong, Kui Wu, Hai Ci, Churan Wang, and Hao Chen. Empowering embodied visual tracking with visual foundation models and offline rl. In European Conference on Computer Vision, pages 139–155. Springer, 2024.

[85] 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, pages 7641–7649, 2024.

[86] 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, pages 260–278, 2025.

[87] Zhongyi Zhou, Yichen Zhu, Xiaoyu Liu, Zhibin Tang, Junjie Wen, Yaxin Peng, Chaomin Shen, and Yi Xu. Chatvla-2: Vision-language-action model with open-world reasoning. In The Thirty-ninth Annual Conference on Neural Information Processing Systems.

[88] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12689–12699, 2021.

[89] Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, et al. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8120–8132, 2025.

[90] Filippo Ziliotto, Tommaso Campari, Luciano Serafini, and Lamberto Ballan. Tango: training-free embodied ai agents for open-world tasks. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 24603–24613, 2025.