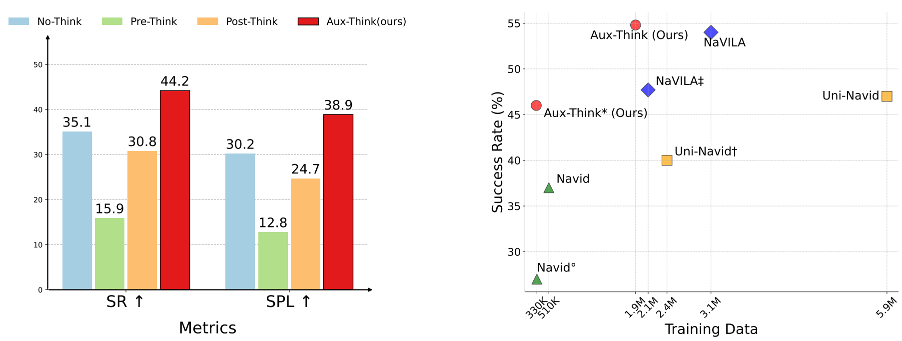
Aux-Think: Exploring Reasoning Strategies for Data-Efficient Vision-Language Navigation

Shuo Wang[1,3]*, **Yongcai Wang**[1]†, **Wanting Li**[1], **Xudong Cai**[1],
Yucheng Wang[3]‡, **Maiyue Chen**[3], **Kaihui Wang**[3], **Zhizhong Su**[3], **Deying Li**[1], **Zhaoxin Fan**[2]†

[1]Renmin University of China, [2]Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, [3]Horizon Robotics

https://horizonrobotics.github.io/robot_lab/aux-think



(a) Comparison of navigation performance of different reasoning strategies, which are only trained on R2R-CoT-320k without receding-horizon action planning. The proposed Aux-Think (Ours) method consistently outperforms other reasoning strategies.

(b) Aux-Think (Ours) is Pareto-optimal in data efficiency and success rate. Variants: Aux-Think*: only trained on R2R-CoT-320k. Navid° : no DAgger [1] and instruction data. Uni-Navid†: no VQA data. NaVILA‡: no Internet data.

Figure 1: Aux-Think outperforms alternative reasoning approaches in navigation tasks (a) and achieves a favorable trade-off between data usage and success rate (b).

Abstract

Vision-Language Navigation (VLN) is a critical task for developing embodied agents that can follow natural language instructions to navigate in complex real-world environments. Recent advances driven by large pretrained models have significantly improved generalization and instruction grounding compared to traditional approaches. However, reasoning strategies in this task remain underexplored. Navigation is action-centric and long-horizon, while Chain-of-Thought (CoT) reasoning has mainly shown success in static tasks such as visual question answering. To address this gap, we conduct the first systematic evaluation of reasoning strategies, including No-Think (direct action prediction), Pre-Think (reasoning before action), and Post-Think (reasoning after action). Surprisingly, our findings reveal a Test-time Reasoning Collapse issue, where reasoning during testing degrades navigation accuracy, highlighting the challenges of integrating reasoning into embodied navigation. Based on this insight, we propose Aux-Think, a framework

*This work was done while Shuo Wang was a Research Intern with Horizon Robotics.
†Corresponding authors.
‡Project leader.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

that trains models to internalize structured reasoning patterns via CoT supervision, while predicting actions directly without explicit reasoning at test time. To support this framework, we release R2R-CoT-320k, the first Chain-of-Thought annotated dataset for VLN. Extensive experiments show that Aux-Think substantially reduces training effort and achieves state-of-the-art performance on success rate.

1 Introduction

Vision-and-Language Navigation (VLN) [2, 3, 4] represents a groundbreaking step towards enabling robots to understand natural language instructions and navigate complex, unfamiliar environments. As a foundational capability for embodied AI systems, VLN bridges the gap between perception and action, empowering robots to seamlessly interact with the real world. In particular, Vision-Language Navigation in continuous environments (VLN-CE) [5, 6, 7] has emerged as a critical research focus, pushing the boundaries of autonomy and adaptability in dynamic, real-world scenarios.

Traditional Vision-and-Language Navigation methods often rely on waypoint predictors [8, 5, 9] or topology maps [10, 11, 12, 13] but struggle with generalization and the sim-to-real gap. With the advancements in Large Language Models (LLMs) [14, 15] and Vision-Language Models (VLMs) [16, 17, 18], recent studies [19, 20, 21] shift toward action prediction via supervised fine-tuning on paired videos and instructions. Despite these advancements, most efforts emphasize training strategies [22, 23], data organization [24], or model architecture [25] for VLN. Chain-of-Thought (CoT), which explicitly generates intermediate reasoning before producing final answers [26], has shown success in enhancing reasoning capabilities across various LLM- and VLM-driven tasks, like video understanding [27] and tool usage [28]. However, its application to VLN remains unexplored.

Motivated by this gap, we present the first systematic study of reasoning strategies in VLN, comparing three paradigms: (1) No-Think, which predicts actions without explicit reasoning; (2) Pre-Think [29], which performs CoT before action selection; and (3) Post-Think [30], which reasons after action prediction. **Our key findings reveal a phenomenon we term "Test-time Reasoning Collapse" (TRC)**: Introducing CoT via Pre-Think or Post-Think consistently harms navigation performance (Fig. 1a). CoT errors or hallucinations during testing result in incorrect actions, as shown in Fig. 3. We attribute this to a training–testing mismatch: CoT is only trained on optimal (oracle) trajectories, but for testing, agents often enter unfamiliar, off-distribution states where reasoning fails, leading to error accumulation along the trajectory and cascading navigation failures. Even with DAgger, coverage of off-distribution states is limited and CoT supervision remains biased toward the optimal region. This phenomenon highlights a fundamental limitation of multi-turn explicit reasoning in dynamic, partially observable environments, unlike the single-turn static tasks such as VQA or image captioning [31, 32].

To address the TRC issue, we propose **Aux-Think**, inspired by the dual-process theory of human learning [33]: during training, humans often rely on explicit reasoning to understand principles, but during execution, they focus on actions without consciously recalling those principles, much like a driver who no longer recites traffic rules while driving. Similarly, Aux-Think uses CoT as an auxiliary signal during training to guide the model in internalizing reasoning patterns. At testing time, the model no longer generates explicit reasoning, but instead directly predicts actions based on the internalized reasoning learned during training. This separation between learning and execution improves decision focus, reduces testing overhead and hallucinations, and leads to more accurate and stable navigation.

Specifically, in Aux-Think, the generation of CoT reasoning and navigation actions is decoupled into two distinct tasks during training: (1) generating navigation actions based on stepwise observations and language instructions as the primary task, and (2) generating the reasoning process for each step as an auxiliary task. By leveraging prompts to switch between these tasks, we independently supervise navigation actions and reasoning processes, effectively avoiding the negative interference caused by jointly training both tasks. During testing, Aux-Think directly predicts actions without intermediate reasoning, thereby eliminating the risk of errors introduced by CoT hallucinations.

To validate the effectiveness of Aux-Think, we introduce R2R-CoT-320k, the first CoT dataset for VLN, which is large-scale and specifically tailored for the R2R-CE benchmark [34]. As existing datasets lack aligned CoT-style reasoning, we construct this dataset by generating reasoning traces that can faithfully lead to the correct next action. R2R-CoT-320k consists of over 320,000 diverse reasoning traces grounded in natural instructions and photo-realistic navigation trajectories. It covers a wide range of scenarios and CoT content, making it a rich and challenging resource for training and evaluation. We show that Aux-Think, when trained with R2R-CoT-320k, matches the performance of state-of-the-art VLN methods, while using only a fraction of their training data (Fig. 1b).

Our contributions are as follows:

- **New Finding:** We conduct a systematic comparison of reasoning strategies in VLN and reveal that test-time reasoning, including Pre-Think and Post-Think, consistently underperforms direct action prediction (No-Think), termed the TRC issue. To our knowledge, this is the first exploration of CoT strategies on the VLN-CE task.
- **New Method:** We propose Aux-Think, a novel training paradigm that uses CoT as auxiliary supervision while maintaining No-Think testing, achieving superior performance over other reasoning methods. Aux-Think pioneers a new perspective on CoT utilization and achieves the best performance on the navigation success rate.
- **New Dataset:** We introduce R2R-CoT-320k, a large-scale, diverse, and challenging Chain-of-Thought dataset tailored for the R2R-CE benchmark, which enables more effective training of reasoning-aware VLN agents.

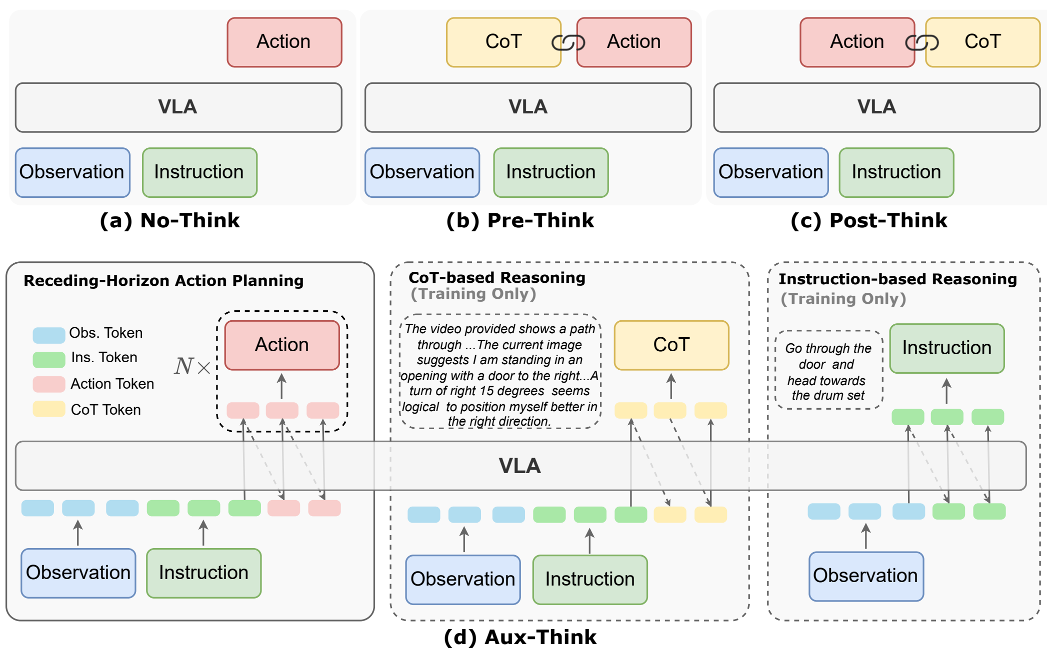


Figure 2: Illustration of Aux-Think and other reasoning strategies. Unlike No-Think, Pre-Think, and Post-Think, our Aux-Think introduces auxiliary CoT- and instruction-based reasoning during training while maintaining efficient action planning at testing.

2 Related Work

2.1 Traditional VLN Methods

Before the emergence of large-scale pretrained models, VLN methods are primarily built on modular pipelines trained via imitation [35, 36, 37] or reinforcement learning [38, 39], often with handcrafted visual features and auxiliary objectives such as progress monitoring or instruction reweighting. These models typically operate with panoramic observations and discretized actions, as in benchmarks like Room-to-Room (R2R) [3] and Room-Across-Room (RxR) [40]. The traditional auxiliary reasoning tasks have also been explored, which rely on specific network outputs to predict structured, low-level reasoning results like task progress or trajectory alignment. More recent work has explored navigation in continuous action spaces using egocentric visual inputs [34, 41], which is the setting adopted in our

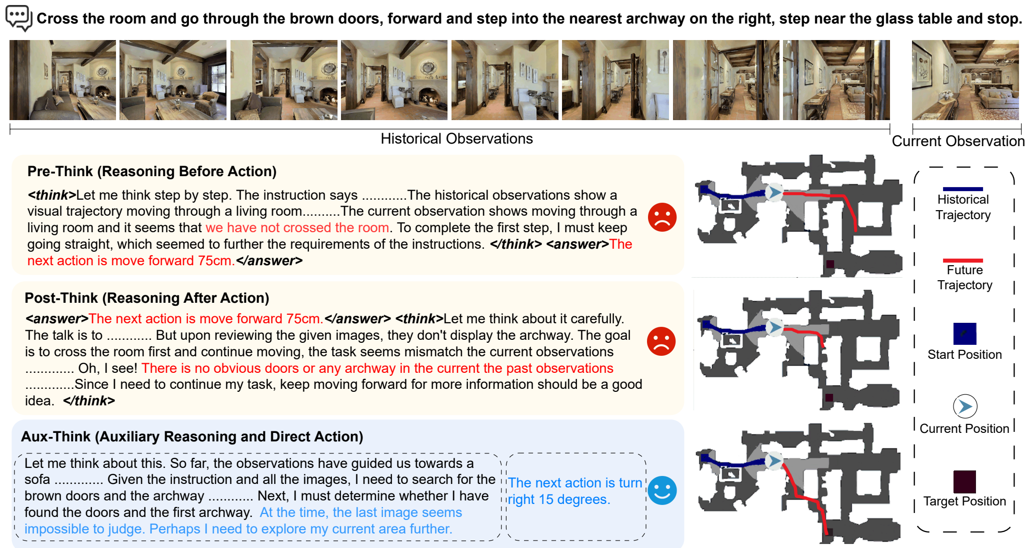


Figure 3: Illustration of CoT and Action Prediction Results Using Different Reasoning Strategies. Pre-Think generates incorrect actions (e.g., "move forward 75cm") due to flawed CoT reasoning, such as "we have not crossed the room," leading to significant trajectory deviation. Post-Think, which builds on Pre-Think's output, inherits similar reasoning errors (e.g., "no obvious door or archway") and makes the same wrong prediction. In contrast, Aux-Think correctly predicts "turn right 15 degrees" and follows a trajectory aligned with the ground truth. While Aux-Think does not rely on CoT during testing, it can optionally produce CoT via prompt switching—yet its action prediction remains accurate even when the generated CoT is of moderate quality. This highlights Aux-Think's robustness to imperfect reasoning and its superior reliability in action prediction.

study. Our method replaces handcrafted pipelines with vision-language models that directly predict agent actions, aiming to improve instruction following in realistic environments.

2.2 VLN with Large Pretrained Models

Recent advancements have seen the integration of large pre-trained models [17, 16, 14, 42], into VLN tasks. Early explorations of LLM in the VLN field usually use off-the-shelf large language models to select landmarks or waypoints in a zero-shot manner [43, 44, 45, 41]. Recent works have focused on fine-tuning the VLM to obtain the navigational Vision-Language-Action model. Notably, Poliformer [23] and NaVid [19] introduce a video-based monocular VLN, demonstrating navigation capabilities using monocular RGB videos without maps or depth input. Uni-NaVid [20] unifies various navigation tasks, including VLN, ObjectNav [46], Embodied Question Answering [47], and Human-following [48, 49], into a single model trained on a diverse dataset. NaVILA [21] further extends this approach by integrating VLN with legged robot locomotion skills in complex environments.

While these models have improved the alignment among visual understanding, language instructions, and navigation actions, they predominantly employ No-Think testing strategies, lacking reasoning mechanisms. Moreover, their performance gains often stem from leveraging extensive datasets, whereas our approach focuses on exploring reasoning strategies.

2.3 Reasoning Models

Recent advances like Chain-of-Thought (CoT) [26], ReAct [50], and Toolformer [51] highlight the potential of LLMs to perform explicit reasoning in static and multimodal tasks, including VQA [52], visual grounding [53], and video understanding [27], where Pre-Think strategies have shown success. Similar ideas have been explored in embodied tasks like manipulation [54] and control [55]. However, a recent study [30] argues that small models may benefit more from No-Think or Post-Think strategies due to limited CoT quality.

In our work, we conduct the first systematic comparison of Pre-Think, Post-Think, and No-Think reasoning strategies for VLN. Based on our findings, we propose Aux-Think, a novel framework that leverages CoT reasoning as auxiliary supervision during training while maintaining No-Think testing, thereby enhancing data efficiency and performance in VLN.

3 Method

3.1 Problem Setup

We study monocular Vision-and-Language Navigation in continuous environments (VLN-CE), where an embodied agent navigates photo-realistic indoor environments by following natural language instructions. VLN-CE emphasizes generalization to unseen environments and supports both forward and reverse navigation, offering a comprehensive test of spatial reasoning and language grounding.

At each time step, the agent receives: (1) a natural language instruction, typically a short paragraph specifying the navigation goal; (2) a RGB observation from the agent's current viewpoint; and (3) historical observations, including 8 frames uniformly sampled from all historical frames (always including the first frame). The agent selects an action (e.g., move forward, turn left/right by a specific degree, or stop). The objective is to generate an action sequence that follows the instruction as accurately and efficiently as possible until the agent reaches the target position.

Within the Supervised Fine-Tuning (SFT) framework, our VLN model based on NVILA 8B [17], is learned by imitating expert demonstrations from the dataset, where each trajectory provides sequences of <navigation context, expert action> pairs, with navigation context denoting the combination of historical observations, the current observation, and the instruction.

3.2 R2R-CoT-320k Dataset Construction

We present R2R-CoT-320k, the first VLN dataset annotated with CoT reasoning, tailored for the R2R-CE benchmark. We reconstruct step-wise navigation trajectories in the Habitat simulator [56]. Each sample in the dataset comprises the current view, the historical visual context, the corresponding instruction, and the ground-truth action. We employ Qwen-2.5-VL-72B [16], one of the strongest publicly available VLMs, to generate detailed CoT for each navigation sample (Fig. 4). For Pre-Think and Post-Think strategies, we format reasoning traces with <think></think> and <answer></answer> tags, following recent reasoning models [57]. See Appendix A.2 for more details on our dataset.

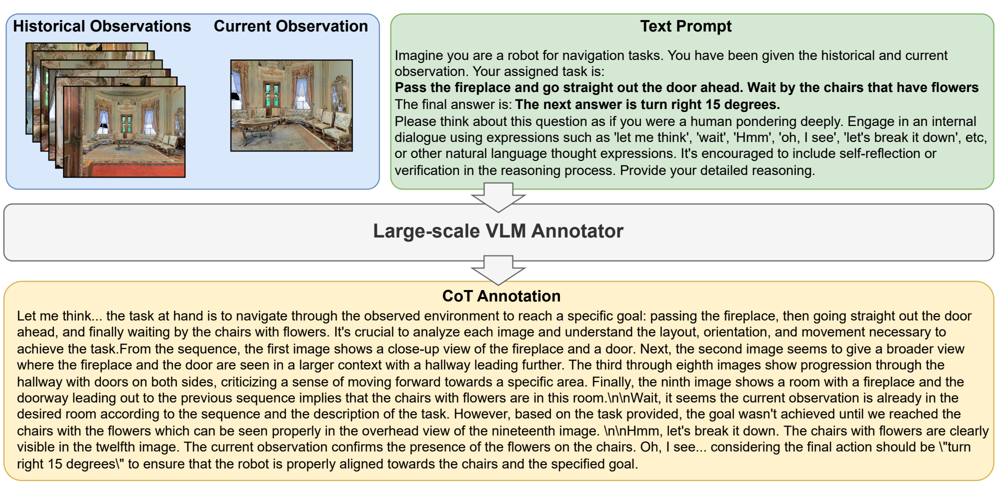


Figure 4: The annotation pipeline of our R2R-CoT-320k dataset.

3.3 Systematic Investigation on Reasoning Strategies for VLN

To investigate the impact of reasoning on VLN, we study and evaluate three distinct strategies for integrating Chain-of-Thought (CoT) reasoning during training and testing.

No-Think: The agent directly predicts the next action based on the current observation and instruction, without any intermediate reasoning.

Pre-Think: The agent first generates an explicit reasoning trace based on the instruction and current observation. The following predicted actions are conditioned on the CoT output.

Post-Think: The agent first predicts an action and then retrospectively generates a reasoning trace that explains the decision.

The training loss for the VLN model π_θ with above three strategies is:

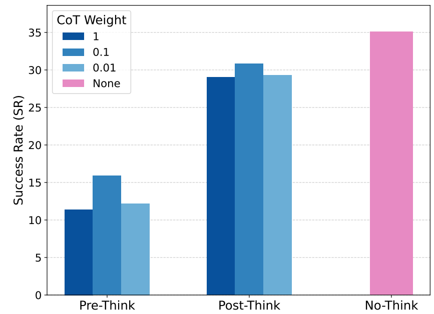


Figure 5: Comparision of success rate on Pre-Think, Post-Think, and No-Think.

$$L(\theta) = -\sum_{\tau \in D} \sum_{t=0}^{T} \begin{cases} \log \pi_\theta(a_t^* | \mathcal{O}_t, o_t, I_\tau) & \text{for No-Think} \\ \log \pi_\theta(< c_t^*, a_t^* > | \mathcal{O}_t, o_t, I_\tau) & \text{for Pre-Think} \\ \log \pi_\theta(< a_t^*, c_t^* > | \mathcal{O}_t, o_t, I_\tau) & \text{for Post-Think} \end{cases} \tag{1}$$

where D is the training dataset; T is the total number of timesteps in a trajectory τ, and t is the current timestep. a_t^* is the ground truth action and c_t^* is the ground truth reasoning trace. $\mathcal{O}_t$ represents the history of observations up to timestep t, o_t is the current visual observation at timestep t, and I_τ is the natural language instruction provided to the agent.

We isolate the impact of reasoning on overall navigation performance. In addition, we adjust the loss weight of the CoT part (c_t^* in Equation 1) to make the model more focused on learning the action (Fig. 5). Our key findings include:

Finding 1: Pre-Think and Post-Think perform significantly worse than No-Think. Despite involving explicit reasoning, both strategies lead to lower navigation success and efficiency, highlighting the unreliability of test-time CoT in dynamic environments.

Finding 2: Careful Balancing of Explicit CoT Loss Weight Improves Performance. By carefully tuning the explicit CoT weight for Pre-Think and Post-Think, we find that balanced supervision yields slight gains, indicating that training emphasis on reasoning is a key, strategy-dependent factor despite test-time unreliability.

We further analyze the severe performance degradation observed with Pre-Think and Post-Think strategies. During training, the model is only exposed to optimal sequences of states and actions in oracle trajectories. However, VLN environments are inherently complex, dynamic, and partially observable, increasing the likelihood of agents deviating from optimal, oracle-guided training trajectories. Consequently, when encountering non-oracle, out-of-distribution states during testing, the CoT reasoning generated by these strategies is susceptible to drift, potentially yielding inaccurate or hallucinated interpretations of the environment and instructions.

In Pre-Think, the model's actions rely heavily on the preceding reasoning chain, making decisions fragile. Any hallucination or misstep in reasoning directly leads to wrong actions. In Post-Think, although actions are generated first, the need to produce follow-up explanations still alters the model's hidden states. This subtle interference, like reserving capacity or shifting attention, can compromise the quality of the initial action decision. A detailed example is provided in Fig. 3.

Moreover, agents perform multi-step action prediction, where an error at any intermediate step further pushes the agent away from the correct state distribution. This compounding effect leads to cascading errors in both CoT reasoning and subsequent actions, ultimately resulting in trajectory-level failures.

3.4 Aux-Think: Reasoning-Aware Co-Training Strategy

To address the challenges from the explicit CoT training to VLN, we propose Aux-Think, a novel reasoning strategy designed to enhance navigation performance without incurring test-time problems. Aux-Think leverages reasoning exclusively during training through auxiliary tasks. We design two reasoning-based auxiliary tasks and one action-based primary task during training.

CoT-based Reasoning. The model is trained to generate CoT traces conditioned on the given instruction I, current observation o_t, and historical observations $\mathcal{O}_t$. This encourages the acquisition of structured reasoning patterns and strengthens the connection between language, vision, and actions. The loss of CoT-based reasoning for each trajectory τ is:

$$L_\tau^{CoT}(\theta) = -\sum_{t=0}^{T} \log \pi_\theta(c_t^* | \mathcal{O}_t, o_t, I_\tau) \tag{2}$$

where c_t^* is the ground-truth reasoning trace at step t.

Instruction-based Reasoning. Given a sequence of visual observations $\sum_{t=0}^{T} o_t$, the model is trained to reconstruct the corresponding instruction I. This reverse reasoning task provides complementary supervision beyond CoT-based signals, further enriching the model's semantic grounding. The training loss is:

$$L_\tau^{Ins}(\theta) = -\log \pi_\theta(I_\tau | \sum_{t=0}^{T} o_t) \tag{3}$$

Receding-Horizon Action Planning. We introduce Receding-Horizon Action Planning as our primary task. During training, the model predicts a sequence of the next n actions $(a_t, a_{t+1}, ..., a_{t+n-1})$ based on the instruction I, current observation o_t, and navigation history $\mathcal{O}_t$ for the sample at time step t . This setup encourages short-term forecasting while retaining reactivity to new observations. The training objective for each trajectory τ is defined as:

$$L_\tau^{Act}(\theta) = -\sum_{t=0}^{T} \sum_{k=0}^{n} \log \pi_\theta(a_{t+k}^* | \mathcal{O}_t, o_t, I_\tau) \tag{4}$$

where a_{t+k}^* denotes the ground-truth action at future step $t+k$.

During training, we co-train the three tasks and switch between different tasks by changing the prompt (Appendix A.3). The final loss function is:

$$L = \sum_{\tau \in D} L_\tau^{Act}(\theta) + L_\tau^{CoT}(\theta) + L_\tau^{Ins}(\theta) \tag{5}$$

where D is the set of training trajectories.

For testing, we only activate the prediction of actions, and the model predicts the next n actions and executes only the first one. This ensures fast, reactive navigation without reasoning overhead. We demonstrate its stabilizing effect in long-horizon trajectories through ablation studies (Table 5).

4 Experimental Results

4.1 Experiment Setup

Simulated environments. We evaluate our method on the VLN-CE benchmarks R2R-CE [34] and RxR-CE [40] following the standard VLN-CE settings. All the methods are evaluated on the R2R val-unseen split and RxR val-unseen split.

Metrics. We follow the standard VLN evaluation protocol [34, 40] to evaluate the navigation performance for all the methods, including success rate (SR), oracle success rate (OSR), success weighted by path length (SPL), and navigation error from goal (NE).

4.2 Implementation Details

Model training. We use NVILA-lite 8B [17] as the base pretrained model, which consists of a vision encoder (SigLIP [58]), a projector, and an LLM (Qwen 2 [16]). We use supervised finetuning

(SFT) to train our VLN model from stage 2 of NVILA-lite, as it has finished visual language corpus pre-training. Our model is trained with 8 NVIDIA H20 GPUs for one epoch (around 60 hours), with a learning rate of 1e-5.

Action design. The action space is designed into four categories: move forward, turn left, turn right, and stop. The forward action includes step sizes of 25 cm, 50 cm, and 75 cm, while the turn actions are parameterized by rotation angles of 15° , 30° , and 45° . This fine-grained design allows for more precise and flexible control, which is critical in complex environments.

4.3 Comparison on VLN-CE Benchmarks

We evaluate our method on the VLN-CE benchmarks, which provide continuous environments for navigational actions in reconstructed photorealistic indoor scenes. We first focus on the val-unseen split in **R2R-CE** dataset in Table 1. To be fair, we distinguish between methods by marking those based on waypoint predictors (∗) and those that are not based on large models (†).

In large model-based methods, we additionally mark the amount of data used by the method in addition to the R2R-CE training split (**Extra Data**). We further scale up by the RxR training split (600K), DAgger data (500K) and web data (500K), and our performance achieves the SOTA Success Rate (SR) to those using a much larger amount of training data.

Table 1: Comparison of different methods on the R2R Val-Unseen split. Observations used include Monocular (Mono.) and Panoramic view (Pano.). ∗ indicates methods based on the waypoint predictor [5]. † indicates methods without using LLMs. ° indicates the models using the training data only from R2R-CE training split, and we compare with the results reported in their paper for a fair evaluation. The training data structures of traditional non-LLM-based methods are quite different, so we do not compare them with their extra data.

Method	Venue	Observation		R2R Val-Unseen				Training
		Mono.	Pano.	NE ↓	OSR ↑	SR ↑	SPL ↑	Extra Data
BEVBert∗†[7]	ICCV2023		✓	4.57	67.0	59.0	50.0	-
ETPNav∗†[59]	TPAMI2024		✓	4.71	65.0	57.0	49.0	-
ENP–ETPNav∗†[60]	Neurips2024		✓	4.69	65	58	50	-
Seq2Seq†[34]	ECCV2020	✓		7.77	37.0	25.0	22.0	-
CMA†[34]	ECCV2020	✓		7.37	40.0	32.0	30.0	-
LAW†[61]	EMNLP2021	✓		6.83	44.0	35.0	31.0	-
CM2†[62]	CVPR2022	✓		7.02	41.0	34.0	27.0	-
WS-MGMap†[12]	Neurips2022	✓		6.28	47.0	38.0	34.0	-
sim2real†[63]	CoRL2024	✓		5.95	55.8	44.9	30.4	-
NaVid°[19]	RSS2024	✓		<u>6.33</u>	<u>30.8</u>	<u>24.7</u>	<u>23.6</u>	0K
Aux-Think (ours)°	-	✓		**6.01**	**52.2**	**46.0**	**40.5**	0K
Uni-NaVid[20]	RSS2025	✓		<u>5.58</u>	53.3	47.0	42.7	5570K
NaVILA[21]	RSS2025	✓		**5.22**	**62.5**	<u>54.0</u>	**49.0**	2770K
Aux-Think (ours)	-	✓		6.08	<u>60.0</u>	**54.8**	<u>46.9</u>	**1600K**

As in Table 1, our proposed **Aux-Think** achieves strong performance with and without extra data. We attribute these results to multilevel reasoning supervision. Our joint training on CoT-based reasoning, instruction reconstruction, and receding-horizon action prediction enriches the model's semantic grounding and decision-making ability, allowing it to better generalize from limited data. Our method benefits from reasoning-induced supervision signals that align more closely with the high-level semantic structure of the instructions, making each training example more informative.

To further assess the ability of Aux-Think, we evaluate it on the RxR-CE [40] Val-Unseen split (Table 2). Compared to R2R-CE, RxR-CE includes more natural instructions and longer trajectories, making it a more realistic and challenging benchmark. Aux-Think achieves strong overall performance, particularly on the Success Rate (SR) metric, where it surpasses Uni-NaVid and NaVILA while using much fewer training data (1920K vs. 5900K and 3100K). This demonstrates the effectiveness of reasoning supervision under limited data. It is noted that performance on NE and SPL is relatively modest. This is likely due to the model's internalized reasoning behavior, learned during CoT training, which encourages broader exploration. Although this can lead to longer paths and reduced efficiency, it improves SR and OSR by increasing the likelihood of reaching the goal, especially in unfamiliar environments.

Table 2: Comparison of different methods on the RxR Val-Unseen split. † indicates methods without using LLMs. The training data structures of traditional non-LLM-based methods are quite different, so we do not include their training data in this table.

Method	Venue	Observation		RxR Val-Unseen				Training
		Mono.	Pano.	NE ↓	OSR ↑	SR ↑	SPL ↑	Data
ETPNav†[59]	TPAMI2024		✓	5.64	-	54.7	44.8	-
ENP–ETPNav†[60]	Neurips2024		✓	5.51	-	55.27	45.11	-
Seq2Seq†[34]	ECCV2020	✓		11.8	-	13.9	11.9	-
LAW†[61]	EMNLP2021	✓		10.87	21.0	8.0	8.0	-
CM2†[62]	CVPR2022	✓		12.29	25.3	14.4	9.2	-
sim2real†[63]	CoRL2024	✓		8.79	36.7	25.5	18.1	-
Uni-NaVid[20]	RSS2025	✓		**6.24**	<u>55.5</u>	48.7	<u>40.9</u>	5900K
NaVILA[21]	RSS2025	✓		<u>6.77</u>	-	<u>49.3</u>	**44.0**	3100K
Aux-Think (ours)	-	✓		**6.24**	**61.9**	**52.2**	40.2	**1920K**

4.4 Comparison Between Different Reasoning Strategies

As shown in Table 3, we compare different reasoning strategies on R2R-CE, with only the R2R-CoT-320K dataset as training data for fairness. We find that the SR performance of Pre-Think and Post-Think is significantly lower than No-Think. In Pre-Think, the action prediction is conditioned on the generated CoT; thus, low-quality or poorly learned CoT directly impairs action accuracy. While Post-Think partially mitigates this issue by generating CoT after the action, suboptimal CoT representations can still degrade overall performance. In contrast, the proposed Aux-Think decouples CoT and action learning by implicitly internalizing CoT knowledge into feature representations.

Table 3: Comparison of different reasoning strategies on R2R-CE Val-Unseen split.

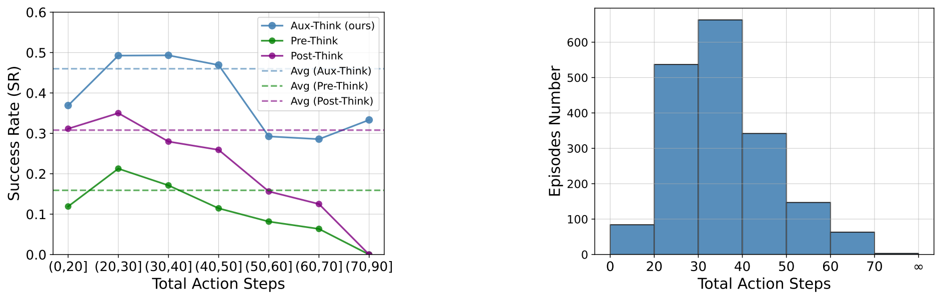
Reason Strategies	NE↓	OSR↑	SR↑	SPL↑	Avg. time↓
No-Think	7.78	43.7	35.1	30.2	**1.25s**
Pre-Think	9.23	19.3	11.4	8.6	30.62s
Post-Think	8.59	35.1	29.0	23.8	28.97s
Aux-Think (ours)	**7.09**	**47.6**	**41.3**	**35.8**	**1.25s**

In Fig. 6, we evaluate the Success Rate (SR) per test step for Aux-Think (ours), Pre-Think, and Post-Think, with results grouped by the number of steps required for task completion. Across all step ranges, Aux-Think consistently outperforms both baselines. A key observation is that the performance of Pre-Think and Post-Think degrades sharply as the required steps increase, with SR approaching zero for tasks exceeding 70 steps. In contrast, Aux-Think maintains strong performance even on longer-horizon tasks, exhibiting markedly higher robustness and generalization to complex, multi-step navigation scenarios. These results highlight the superior scalability of Aux-Think in handling extended reasoning and decision-making under increased task complexity.

4.5 Ablation Studies

4.5.1 Impact of Different Auxiliary Tasks and Receding-Horizon Action Planning

Table 4 presents the ablation of three components: CoT-based Reasoning (A), Instruction-based Reasoning (B), and Receding-Horizon Action Planning (C). Introducing CoT Reasoning (A) leads to a noticeable improvement across all metrics, indicating its effectiveness in guiding action decisions. Adding Non-CoT Reasoning (A+B) further enhances performance, suggesting that the two forms of reasoning are complementary. The full model (A+B+C), which incorporates receding-horizon planning, achieves the best results, particularly in terms of SPL and SR, demonstrating that long-term planning grounded in implicit reasoning yields the most robust behavior. These results validate the necessity of integrating both reasoning and planning for optimal performance.



(a) (b)

Figure 6: (a) Success Rate of reasoning strategies on different steps during testing. (b) The epsodes number on different steps during testing. The steps indicate the actions required by the instruction.

Table 4: Ablation study on different components. A: CoT Reasoning, B: Non-CoT Reasoning, C: Receding-Horizon Action Planning.

Configuration A	B	C	Metrics NE↓	OSR↑	SR↑	SPL↑
			7.78	43.7	35.1	30.2
✓			7.08	47.6	41.3	35.8
	✓		7.12	46.3	40.6	35.7
		✓	7.14	47.3	37.1	32.2
✓	✓		6.92	49.1	44.2	38.9
✓	✓	✓	**6.01**	**52.2**	**46.0**	**40.5**

Table 5: Ablation studies on the predicted steps in our Receding-Horizon Action Planning. The best one across all metrics is when the number of steps is 3.

#Steps	NE↓	OSR↑	SR↑	SPL↑
1	7.78	43.7	35.1	30.2
2	7.88	44.2	35.8	30.9
3	**7.14**	**47.3**	**41.4**	**36.1**
4	7.50	43.6	36.4	31.8
5	7.54	43.7	37.1	32.2

4.5.2 Impact of Steps in Receding-Horizon Action Planning

Based on Table 5, the model achieves the best performance when the number of predicted steps is set to 3. We disable the CoT auxiliary task to analyze the impact of actions more clearly. The results highlight our Receding-Horizon Action Planning, which encourages the model to anticipate future actions and enhances its planning capabilities. However, increasing the number of predicted steps beyond this point leads to performance degradation. We attribute this to the limited perceptual field of monocular observations without additional global knowledge, which makes long-horizon prediction more challenging and can cause the model to generate suboptimal or collapsed navigation behaviors.

5 Limitation and Future Work

This work evaluates Aux-Think's data efficiency under a controlled, widely adopted setup: SFT on the R2R dataset with monocular RGB input. This enables fair comparison and isolates the effect of reasoning-aware supervision. While constrained, this setting opens future directions, scaling to larger navigation datasets and incorporating richer supervision (e.g., depth, panorama, localization) [42].

6 Conclusion

We conduct the first systematic investigation of reasoning strategies in Vision-and-Language Navigation, revealing a key limitation, *Test-time Reasoning Collapse*, where errors in generated reasoning can compound and degrade navigation performance. Motivated by this finding, we propose **Aux-Think**, a reasoning-aware co-training framework that leverages Chain-of-Thought as auxiliary supervision during training, while relying on efficient No-Think testing. Extensive experiments demonstrate that Aux-Think achieves performance on par with state-of-the-art methods while using significantly less training data, highlighting its robustness and data efficiency. We also release **R2R-CoT-320k**, the first CoT dataset for VLN, to facilitate future research on reasoning models.

7 Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant No. 62441617. It was supported by the Postdoctoral Fellowship Program and China Postdoctoral Science Foundation under Grant No. 2024M764093 and Grant No. BX20250485, the Beijing Natural Science Foundation under Grant No. 4254100, the Fundamental Research Funds for the Central Universities under Grant No. KG16336301, and by Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing. Deying Li was supported in part by the National Natural Science Foundation of China Grant No.12071478. Yongcai Wang was supported in part by the National Natural Science Foundation of China Grant No. 61972404, Public Computing Cloud, Renmin University of China, and the Blockchain Lab, School of Information, Renmin University of China. Shuo Wang was supported in part by the Outstanding Innovative Talents Cultivation Funded Programs 2024 of Renmin University of China.

References

[1] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[2] Wansen Wu, Tao Chang, Xinmeng Li, Quanjun Yin, and Yue Hu. Vision-language navigation: a survey and taxonomy. *Neural Computing and Applications*, 36(7):3291–3316, 2024.

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[4] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Eric Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. *arXiv preprint arXiv:2203.12667*, 2022.

[5] Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15439–15449, 2022.

[6] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 15625–15636, 2023.

[7] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022.

[8] Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15162–15171, 2021.

[9] Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *European conference on computer vision*, pages 588–603. Springer, 2022.

[10] Muhammad Zubair Irshad, Niluthpol Chowdhury Mithun, Zachary Seymour, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Sasra: Semantically-aware spatio-temporal reasoning agent for vision-and-language navigation in continuous environments. *arXiv e-prints*, pages arXiv–2108, 2021.

[11] Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11276–11286, 2021.

[12] Peihao Chen, Dongyu Ji, Kunyang Lin, Runhao Zeng, Thomas Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 35:38149–38161, 2022.

[13] Shuo Wang, Wanting Li, Yongcai Wang, Zhaoxin Fan, Zhe Huang, Xudong Cai, Jian Zhao, and Deying Li. Mambavo: Deep visual odometry based on sequential matching refinement and training smoothing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1252–1262, 2025.

[14] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[15] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[16] 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.

[17] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. *arXiv preprint arXiv:2412.04468*, 2024.

[18] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[19] 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. *arXiv preprint arXiv:2402.15852*, 2024.

[20] 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. *arXiv preprint arXiv:2412.06224*, 2024.

[21] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024.

[22] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10012–10022, 2020.

[23] 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. *arXiv preprint arXiv:2406.20083*, 2024.

[24] Mingfei Han, Liang Ma, Kamila Zhumakhanova, Ekaterina Radionova, Jingyi Zhang, Xiaojun Chang, Xiaodan Liang, and Ivan Laptev. Roomtour3d: Geometry-aware video-instruction tuning for embodied navigation. *arXiv preprint arXiv:2412.08591*, 2024.

[25] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13624–13634, 2024.

[26] 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.

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

[28] Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing action prediction of gui agents by reinforcement learning. *arXiv preprint arXiv:2503.21620*, 2025.

[29] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 37:333–356, 2024.

[30] Ming Li, Jike Zhong, Shitian Zhao, Yuxiang Lai, and Kaipeng Zhang. Think or not think: A study of explicit thinking in rule-based visual reinforcement fine-tuning. *arXiv e-prints*, pages arXiv–2503, 2025.

[31] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[32] Yan Wang, Yawen Zeng, Jingsheng Zheng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. Videocot: A video chain-of-thought dataset with active annotation tool. *arXiv preprint arXiv:2407.05355*, 2024.

[33] Jonathan St BT Evans. In two minds: dual-process accounts of reasoning. *Trends in cognitive sciences*, 7(10):454–459, 2003.

[34] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 104–120. Springer, 2020.

[35] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12527–12537, 2019.

[36] Hanqing Wang, Wei Liang, Luc V Gool, and Wenguan Wang. Towards versatile embodied navigation. *Advances in neural information processing systems*, 35:36858–36874, 2022.

[37] Qiaoyun Wu, Xiaoxi Gong, Kai Xu, Dinesh Manocha, Jingxuan Dong, and Jun Wang. Towards target-driven visual navigation in indoor scenes via generative imitation learning. *IEEE Robotics and Automation Letters*, 6(1):175–182, 2020.

[38] Zifan Xu, Bo Liu, Xuesu Xiao, Anirudh Nair, and Peter Stone. Benchmarking reinforcement learning techniques for autonomous navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9224–9230. IEEE, 2023.

[39] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Vision-language navigation policy learning and adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4205–4216, 2020.

[40] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020.

[41] 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.

[42] Shuo Wang, Yongcai Wang, Wanting Li, Yucheng Wang, Maiyue Chen, Kaihui Wang, Zhizhong Su, Xudong Cai, Yeying Jin, Deying Li, et al. Monodream: Monocular vision-language navigation with panoramic dreaming. *arXiv preprint arXiv:2508.02549*, 2025.

[43] Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. Discuss before moving: Visual language navigation via multi-expert discussions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17380–17387. IEEE, 2024.

[44] 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.

[45] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR, 2023.

[46] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.

[47] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2018.

[48] Md Jahidul Islam, Jungseok Hong, and Junaed Sattar. Person-following by autonomous robots: A categorical overview. *The International Journal of Robotics Research*, 38(14):1581–1618, 2019.

[49] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

[50] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[51] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

[52] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025.

[53] Zhenyi Liao, Qingsong Xie, Yanhao Zhang, Zijian Kong, Haonan Lu, Zhenyu Yang, and Zhijie Deng. Improved visual-spatial reasoning via r1-zero-like training. *arXiv preprint arXiv:2504.00883*, 2025.

[54] Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression. *arXiv preprint arXiv:2412.03293*, 2024.

[55] 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.

[56] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[57] 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.

[58] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[59] Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[60] Rui Liu, Wenguan Wang, and Yi Yang. Vision-language navigation with energy-based policy. *arXiv preprint arXiv:2410.14250*, 2024.

[61] Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel Chang. Language-aligned waypoint (LAW) supervision for vision-and-language navigation in continuous environments. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4018–4028, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[62] Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15460–15470, 2022.

[63] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Sim-to-real transfer via 3d feature fields for vision-and-language navigation. *arXiv preprint arXiv:2406.09798*, 2024.

[64] Peihao Chen, Xinyu Sun, Hongyan Zhi, Runhao Zeng, Thomas H Li, Gaowen Liu, Mingkui Tan, and Chuang Gan. A2 nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models. *arXiv preprint arXiv:2308.07997*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Refer to Abstract and Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Refer to Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Refer to Section 3.3 and 3.4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Refer to Section 3.2 and 4.1

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Refer to Section 3.2 and Appendix A.2

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Refer to Section 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Refer to Secton 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Refer to 4.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Refer to Appendix.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: VLN research typically focuses on specific technical aspects or objectives that may not directly address broader societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: VLN poses no such risks

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the corresponding original papers.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Refer to Section 3.2 and 4.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Qwen 2.5 VL 72B was used to assist with CoT data annotation. However, the model was used strictly as a labeling aid under human supervision, and its outputs did not constitute a core, original, or non-standard component of the method itself. The annotated data was manually reviewed and curated, and the LLM's role was limited to speeding up the annotation process.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Technical Appendices and Supplementary Material

A.1 More Experimental Results

We evaluate the cross-data performance on **RxR-CE** Val-Unseen split, as shown in Table 6. Even without using RxR-CE training data, our Aux-Think model achieves new state-of-the-art performance on the RxR-CE Val-Unseen split. This confirms the strong generalization of our reasoning-augmented co-training, enabling the model to transfer across datasets with different instructions and scenes.

Table 6: Cross-dataset performance on the RxR-CE Val-Unseen split. All results are obtained without training on the RxR-CE training set.

Method	Venue	Observation		RxR Val-Unseen			
		Mono.	Pano.	NE ↓	OSR ↑	SR ↑	SPL ↑
Seq2Seq[34]	ECCV2020	✓		11.8	5.02	3.51	3.43
CMA[34]	ECCV2020	✓		11.7	10.7	4.41	2.47
LAW[61]	EMNLP2021	✓		10.87	21.0	8.0	8.0
CM2[62]	CVPR2022	✓		8.98	25.3	14.4	9.2
WS-MGMap[12]	Neurips2022	✓		9.83	29.8	15.0	12.1
A^{2}NAV[64]	Arxiv2023	✓		-	-	16.8	6.3
NaVid[19]	RSS2024	✓		**8.41**	<u>34.5</u>	<u>23.8</u>	<u>21.2</u>
Aux-Think (ours)	-	✓		<u>8.98</u>	**39.6**	**29.5**	**23.6**

A.2 R2R-CoT-320k

The action labels in R2R-CoT-320k are derived from the original annotations in R2R-CE. To generate the Chain-of-Thought (CoT) annotations, we employ Qwen-VL 2.5 (72B). Specifically, for each navigation step, we provide the model with the agent's historical observations, the current visual input, and the next action. The model is then prompted to produce intermediate reasoning steps that reflect human-like decision-making processes. The annoation prompt is:

> Imagine you are a robot programmed for navigation tasks. You have been given a video of historical observations: <image>,...,<image> and and current observation: <image>. Your assigned task is: [Instruction]. Analyze this series of images to decide your next move, which could involve turning left or right by a specific degree, moving forward a certain distance, or stop if the task is completed. The final answer is [Action]. Please think about this question as if you were a human pondering deeply. Engage in an internal dialogue using expressions such as 'let me think', 'wait', 'Hmm', 'oh, I see', 'let's break it down', etc, or other natural language thought expressions. It's encouraged to include self-reflection or verification in the reasoning process.

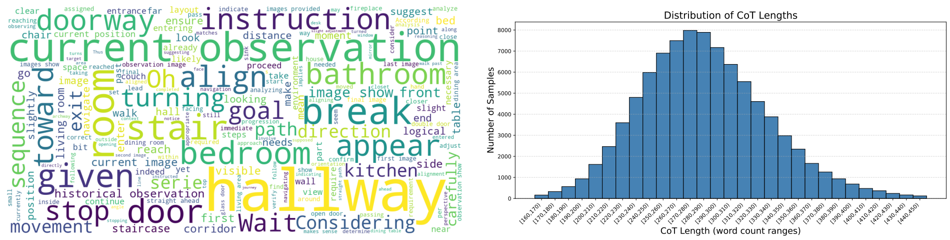
To provide a deeper quantitative understanding of our proposed R2R-CoT-320k dataset, we present statistics on CoT content and complexity. As shown in Fig. 7a, the word cloud reveals frequent reasoning patterns grounded in spatial semantics, such as "doorway," "current observation," "hallway," "turning," and "goal." These tokens suggest that the dataset captures rich, step-by-step reasoning tightly aligned with embodied navigation semantics.

Fig. 7b shows the distribution of CoT lengths, where most reasoning chains fall within the 200–300 word range, but with a long tail reaching beyond 450 words. This indicates that the dataset covers both concise and highly detailed reasoning processes, posing a greater challenge than typical short-form CoT datasets used in static tasks.

Overall, R2R-CoT-320k represents the first large-scale reasoning-augmented dataset for VLN with diverse, high-coverage CoT annotations. It offers a valuable benchmark to study the role of language-based reasoning in long-horizon, partially observable navigation tasks.

A.3 Navigation Prompts

We use the following prompt to drive the model to predict navigation actions:



(a) (b)

Figure 7: (a) Word cloud of Chain-of-Thought in the R2R-CoT-320k dataset, highlighting frequent visual and spatial reasoning patterns. (b) Distribution of CoT lengths (in word count), showing a wide and diverse range of reasoning complexity.

> Imagine you are a robot programmed for navigation tasks. You have been given a video of historical observations: <image>,...,<image> and and current observation: <image>. Your assigned task is: [Instruction]. Analyze this series of images to decide your next move, which could involve turning left or right by a specific degree, moving forward a certain distance, or stop if the task is completed.

Among them, [Instruction] is the language instruction given for the current task. For the auxiliary task of CoT-based reasoning, we add "Please provide your step-by-step reasoning process" after the above prompt.

For the Non-CoT Instruction Reasoning, we set the prompt as:

> Assume you are a robot designed for navigation. You are provided with captured image sequences: <image>,...,<image>. Based on this image sequence, please describe the navigation trajectory of the robot.

A.4 More Details About Pre-Think and Post-Think

To further investigate Test-time Reasoning Collapse (TRC) phenomenon, we introduce special tokens to delineate the reasoning and action prediction components. During training, we assign different loss weights to the reasoning component, as shown in Table 7. We observe that moderately reducing the CoT (Chain-of-Thought) loss weight improves the performance of both Pre-Think and Post-Think. However, their performance still lags behind that of No-Think and Aux-Think.

Our results reveal a consistent trend: reducing the CoT loss weight moderately improves performance for both Pre-Think and Post-Think. For example, in the Post-Think setting, decreasing the CoT weight from 1 to 0.1 leads to a +1.8% SR improvement (from 29.0 to 30.8). This suggests that over-reliance on CoT can introduce noise, potentially due to its misalignment with the suboptimal, off-distribution states encountered during testing.

However, even with carefully tuned weights, both strategies still fall short of No-Think. For instance, No-Think achieves 35.1% SR, significantly outperforming the best Post-Think variant (30.8%) and Pre-Think (15.9%). This persistent gap underscores a deeper issue: while CoT can serve as useful supervision during training, explicitly generating and relying on CoT during testing is inherently brittle in VLN due to compounding errors and distribution shift. This reinforces our finding that VLN agents suffer from reasoning collapse when required to generate structured thoughts in real time within dynamic, partially observable environments.

In summary, despite our extensive efforts to optimize Pre-Think and Post-Think through CoT loss reweighting and architectural adjustments, these strategies fail to match the robustness and effectiveness of CoT-free testing (No-Think). These findings motivate our proposal of Aux-Think, which leverages the strengths of CoT through auxiliary supervision while circumventing the vulnerabilities of test-time reasoning.

Table 7: Experiments on the impact of CoT Loss weight on Pre-Think and Post-Think strategies.

	Weight of CoT Loss	NE ↓	OSR↑	SR↑	SPL↑
Pre-Think	1	9.23	19.3	11.4	8.6
	0.1	9.42	28.3	15.9	12.8
	0.01	8.84	20.6	12.2	9.3
Post-Think	1	8.59	35.1	29.0	23.8
	0.1	8.50	37.7	30.8	24.7
	0.01	8.25	36.6	29.3	24.9
No-Think	-	**7.78**	**43.7**	**35.1**	**30.2**