StageNav: A Diagnostic Benchmark and State-Aligned RL Framework for Long-Horizon Navigation

ChengJie Fang[1], Yujia Yang[1], Yu Liu[2], Ziyu Sun[1] and Jungang Xu[3*]

***Abstract*— Large vision-language models (LVLMs) show promise in vision-language navigation (VLN), yet their performance degrades in long-horizon tasks due to error accumulation. Early deviations caused by hallucinations, context drift, or imprecise spatial grounding often prevent models from accurately assessing their execution progress and completion status, leading to cascading failures. The core limitation lies in the absence of structured state alignment mechanisms for long-horizon tasks. Existing benchmarks and sparse-reward reinforcement learning (RL) paradigms provide only task-level feedback, lacking verifiable supervision over intermediate stages and thus offering unreliable progress signals. To address this issue, we introduce `StageNav`, a unified benchmark-and-training framework for long-horizon navigation. We first construct `StageNav-Bench`, a diagnostic benchmark that reformulates long-horizon instructions as geometrically constrained directed acyclic graphs (DAGs), equipped with a Geometric Verification Oracle (GVO) for node-level objective evaluation. Building upon the geometrically validated training environments, we propose `StageNav-RFT`, a state-aligned reinforcement fine-tuning (RFT) method in which the agent explicitly predicts its subjective cognitive state alongside physical actions. By jointly verifying cognitive predictions against the GVO's objective geometric states, the method generates dense, logic-aligned process rewards. This dual-track state alignment mitigates long-horizon credit assignment, suppresses overconfident hallucinations, and intrinsically motivates honest failure recognition to enable proactive topological backtracking. Experiments demonstrate that StageNav significantly improves navigation success and recovery rates in continuous long-horizon environments, highlighting the critical role of fine-grained state alignment in embodied planning. The code is available at https://github.com/Son4ta/StageNav.**

I. INTRODUCTION

LVLMs have been widely adopted in vision-language navigation. However, when executing long-horizon instructions, agents remain vulnerable to hallucinations, context drift, and insufficient fine-grained spatial grounding. These issues often introduce early deviations that accumulate over extended decision sequences. When a model cannot accurately determine its current task stage or execution progress, initial errors compound and ultimately lead to failure. The underlying limitation is the absence of an explicit structured task-state alignment mechanism in existing end-to-end VLN systems, preventing stable coupling between cognitive states and physical execution states.

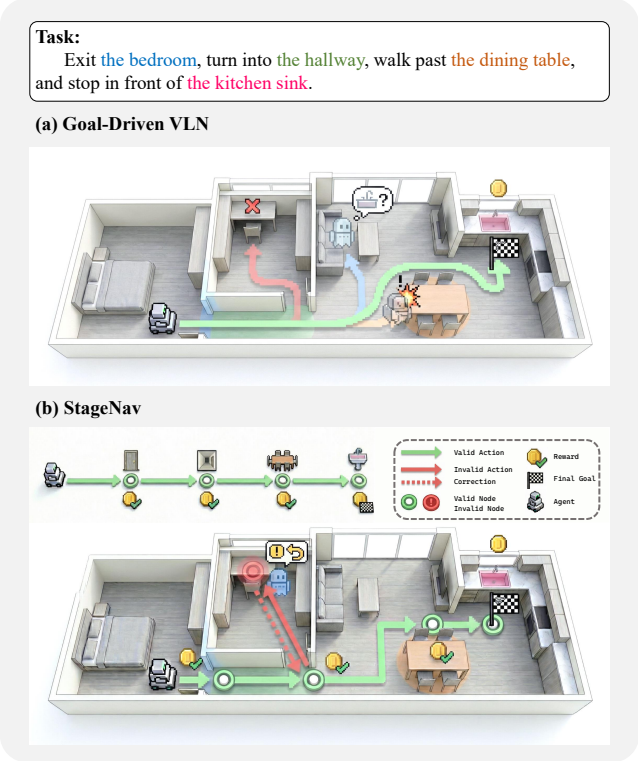


Fig. 1. **Comparison of learning paradigms in long-horizon navigation. (a)** Standard Goal-Driven VLN relies on implicit state representations and sparse terminal rewards, suffering from severe credit assignment issues and error accumulation. This lack of timely correction signals leads to various execution failures. **(b)** The proposed StageNav framework reformulates the instruction into a DAG. By aligning the agent's subjective cognitive state with a Geometric GVO, it provides dense logic-aligned process rewards at every step and enables proactive topological backtracking for error recovery.

This deficiency in state awareness is reinforced by current training paradigms and evaluation benchmarks. Imitation learning (IL) [1] trains agents by fitting expert trajectories, limiting generalization in out-of-distribution (OOD) environments. To encourage exploration, reinforcement learning [2], [3], [4] has been introduced to navigation tasks. However, conventional RL suffers from extreme reward sparsity in long-horizon settings, typically providing a scalar reward only upon reaching the final goal. This design results in severe credit assignment failure: agents receive no structured incentive for correct intermediate decisions and lack timely correction signals when deviating from the target. Furthermore, existing navigation benchmarks [1], [5], [6] primarily rely on task-level outcome metrics, offering limited

[1]ChengJie Fang, Yujia Yang and Ziyu Sun are with Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou, China. {fangchengjie24, yangyujia24, sunziyu24}@mails.ucas.ac.cn

[2]Yu Liu is with Beijing Institute of Graphic Communication, Beijing, China. friedabigc@gmail.com

[3]Jungang Xu is with School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China. xujg@ucas.ac.cn

*Corresponding author.

diagnostic insight into intermediate reasoning and execution processes.

To address the lack of state awareness and coarse diagnostic granularity in long-horizon navigation, we reformulate long-horizon instructions as directed acyclic graphs (DAGs) [7] and construct a diagnostic benchmark, **StageNav-Bench**. While the DAG provides a topological representation of task structure, StageNav-Bench further introduces a GVO operating in continuous 3D physical space, transforming natural language subgoals into mathematically verifiable conditions. This design shifts evaluation granularity from the task level to the node level, enabling objective quantification of physical completion at each DAG node.

Building upon the training split of the constructed data, we propose a state-aligned RFT framework, **StageNav-RFT**. The method transforms an implicit LVLM policy into an explicit cognitive state machine. During execution, the DAG topology guides progress tracking, while the agent is required to predict its current subtask completion status in addition to generating physical actions. During fine-tuning, the agent's subjective cognitive state is cross-validated against the GVO's objective physical state, converting sparse task-level rewards into dense node-level logical alignment rewards. This dual-track state alignment mitigates long-horizon credit assignment issues, suppresses overconfidence and redundant exploration caused by cognitive–physical misalignment, and, crucially, incentivizes honest failure recognition to enable proactive structured backtracking for safe error recovery.

Our contributions are summarized as follows:

- We introduce **StageNav-Bench**, a diagnostic benchmark that reformulates long-horizon VLN as DAGs and incorporates a **GVO** for node-level process diagnostics in continuous 3D environments.
- We propose **StageNav-RFT**, a state-aligned RFT framework that leverages DAG topology and GVO-guided process rewards for structured progress tracking and dual-track state alignment.
- Extensive experiments demonstrate significant improvements in navigation success (SR/SPL) and recovery rates in continuous long-horizon environments. The proposed state-alignment mechanism enables proactive topological backtracking, effectively mitigating error accumulation.

II. RELATED WORK

A. Vision-Language Navigation

Vision-Language Navigation formulates multimodal instruction following as a sequential decision-making process. Early paradigms decoupled this task into modular pipelines evaluated in discrete spaces [1]. The modeling paradigm has recently shifted towards end-to-end continuous generation driven by LVLMs, either leveraging closed-source models for zero-shot navigation [8], [21] or fine-tuning open-source models via SFT/RFT [9], [24]. While this unified modeling simplifies the architecture, it relies on implicit state representations that lack mechanisms to strictly align internal cognitive states with intermediate physical geometries. Consequently, the absence of explicit structural priors limits the system's ability to manage error accumulation and context degradation over extended action sequences.

B. Vision-Language Navigation Benchmarks

VLN evaluation has transitioned from discrete topological graphs such as Room-to-Room (R2R) [1] and Room-Across-Room (RxR) [5] to continuous 3D environments including R2R-CE [14] and RxR-CE, better approximating real-world physical dynamics [13]. Despite this evolution, the predominant paradigm remains outcome-oriented metrics such as Success Rate (SR) and Success weighted by Path Length (SPL), evaluating task completion solely based on the terminal state. This creates an evaluation granularity limitation: existing benchmarks cannot quantify intermediate step-level execution fidelity or the capability to recognize and recover from early deviations, structurally limiting diagnostic analysis of multi-stage spatial reasoning.

C. Reinforcement Fine-Tuning for VLN Agents

RFT formulates policy optimization as continuous environmental interaction. Frameworks like PPO [10] and GRPO [11] are standard mechanisms for LVLM agents. However, standard GRPO relies on sparse outcome rewards, introducing temporal credit assignment challenges in long-horizon tasks. To incorporate step-level supervision, recent work has explored process rewards. GiGPO [12] requires strict identical state grouping across trajectories, suffering from significant sampling inefficiency in continuous 3D spaces and limiting scalability to complex navigation topologies.

Current VLN modeling paradigms, evaluation protocols, and RFT mechanisms operate largely in isolation. The absence of a unified framework aligning intermediate cognitive reasoning with physical geometry and policy optimization creates structural mismatches in long-horizon continuous navigation. Bridging this gap requires joint formulation of diagnostic evaluation and process-level supervision under consistent geometric constraints, motivating the integrated benchmark and logic-aligned optimization framework proposed in this work.

III. METHODOLOGY

A. Problem Formulation

Graph-Constrained MDP. The long-horizon navigation task is formulated as a Graph-Constrained Markov Decision Process (MDP). Given a natural language instruction $\mathscr{I}$, an autoregressive language model parses it into a Directed Acyclic Graph (DAG) denoted as $\mathscr{G} = (\mathscr{V}, \mathscr{E})$.

Nodes and Edges. A node $v_i \in \mathscr{V}$ represents a physical entity in the environment, parameterized by its 3D center coordinate $\mathbf{o}_i \in \mathbb{R}^3$. A directed edge $e_{i,j} = (v_i, v_j) \in \mathscr{E}$ defines a constraint set $\mathscr{C}_{i,j}(\tau)$ containing geometric verification conditions that the agent's continuous trajectory τ must satisfy during the transition from v_i to v_j.

State Space and Observations. At discrete time step t, the agent state incorporates its 3D environmental coordinate

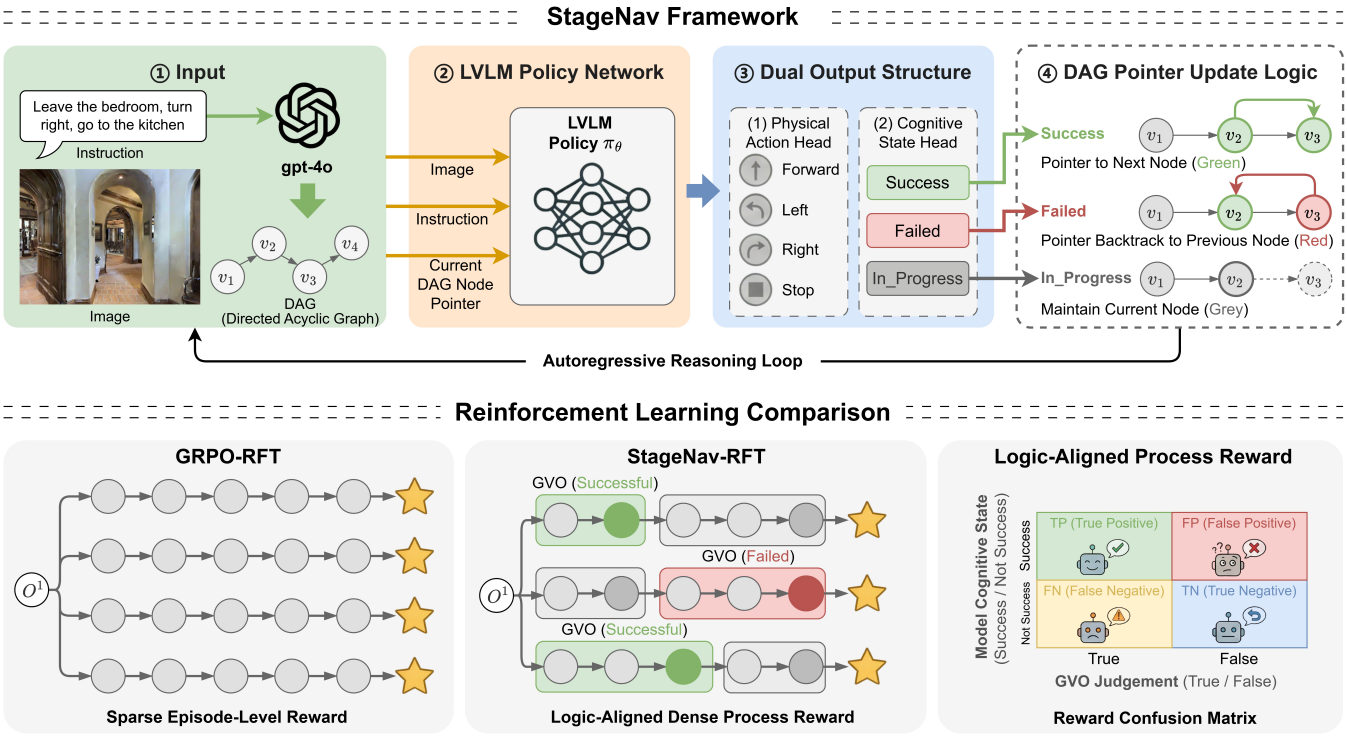


Fig. 2. **Overview of the StageNav framework. (Top)** The autoregressive reasoning loop. The LVLM policy network takes the current DAG pointer, instruction, and visual observation to jointly output a physical action and a subjective cognitive state. The DAG pointer is dynamically updated based on this cognitive state to guide the next step. **(Bottom)** Comparison of RL paradigms. Standard GRPO-RFT samples N trajectories from the same initial state and relies on sparse episode-level rewards. In contrast, StageNav-RFT leverages a GVO to evaluate the cognitive states, generating logic-aligned dense process rewards defined by a reward confusion matrix.

$\mathbf{p}_t \in \mathbb{R}^3$ and a real-time DAG pointer $g_t \in \mathcal{V}$ tracking the current subtask. The joint observation space consists of the egocentric visual observation $\mathbf{O}_t^{FV}$, the global instruction $\mathcal{I}$, and the topological constraints of the current target node g_t.

Action Space and State Transition. The policy network π_θ takes the joint observation to autoregressively output a joint action space comprising a physical action a_t and an explicit cognitive state flag s_t^{cog}:

$$(a_t, s_t^{cog}) \sim \pi_\theta(\cdot|\mathbf{O}_t^{FV}, \mathcal{I}, g_t) \tag{1}$$

where $a_t \in \{\text{FORWARD}, \text{TURN_LEFT}, \text{TURN_RIGHT}, \text{STOP}\}$ and $s_t^{cog} \in \{\text{IN_PROGRESS}, \text{SUCCESS}, \text{FAILED}\}$. During inference, the transition of the DAG pointer g_t is entirely driven by the agent's subjective cognitive prediction s_t^{cog}. If $s_t^{cog} = \text{SUCCESS}$, the agent assumes the current subtask is physically completed, and the pointer advances to the next node ($g_{t+1} \leftarrow v_{j+1}$). Conversely, if the agent recognizes an execution deviation and outputs $s_t^{cog} = \text{FAILED}$, a topological backtracking mechanism is triggered, reverting the pointer to the most recently verified node ($g_{t+1} \leftarrow v_{prev}$). If $s_t^{cog} = \text{IN_PROGRESS}$, the pointer remains at the current node. The sequential discrete displacements yield a continuous trajectory $\tau = \{\mathbf{p}_0, \mathbf{p}_1, ..., \mathbf{p}_T\}$. The ultimate objective of the agent is to complete all node subtasks of the DAG constructed from the natural language instruction $\mathcal{I}$, strictly satisfy all geometric constraints defined by $\mathcal{G}$, and maintain accurate cognitive judgments of the task status throughout the process.

B. Geometric Verification Oracle (GVO)

Constraint-Driven Evaluation. To provide step-level objective supervision, we construct the GVO. Assuming the current graph pointer is at v_j with predecessor v_i, the trajectory segment $\tau_{i \to j}$ is verified using three fundamental algebraic triggers. The specific constraint types are jointly determined by the DAG's edge semantics and node attributes. These triggers can be combinatorially applied (e.g., combining "turn left" and "pass the table" requires simultaneous activation of Type B and Type C checks), thereby covering diverse navigation semantics with $\mathcal{O}(1)$ time complexity.

Type-A: Metric Inclusion. Governing scenarios such as reaching a destination (Stop at/Go to) or walking past an entity, this trigger verifies whether the agent reaches the effective region of a target entity. Given the center coordinate $\mathbf{o}_j$ of v_j, an effective closed set $\mathcal{S}_j$ with radius d_{thresh} is defined on the ground plane. The current position $\mathbf{p}_t$ must strictly satisfy:

$$||\mathbf{p}_t^{2D} - \mathbf{o}_j^{2D}||_2 \leq d_{thresh} \tag{2}$$

Upon successful verification, $\mathbf{p}_t$ is anchored as a new reference point $\mathbf{p}_{ref} \leftarrow \mathbf{p}_t$ for subsequent manifold directional computations.

Type-B: Topological Transversal. Governing actions like passing through doors or moving between entities, this trigger evaluates traversal through physical gaps. The gap is abstracted as a 2D truncated hyperplane $\mathcal{H}_{ab} = \overline{\mathbf{ab}}$. The intersection between the trajectory segment $\mathcal{T} = \overline{\mathbf{p}_{t-1}\mathbf{p}_t}$

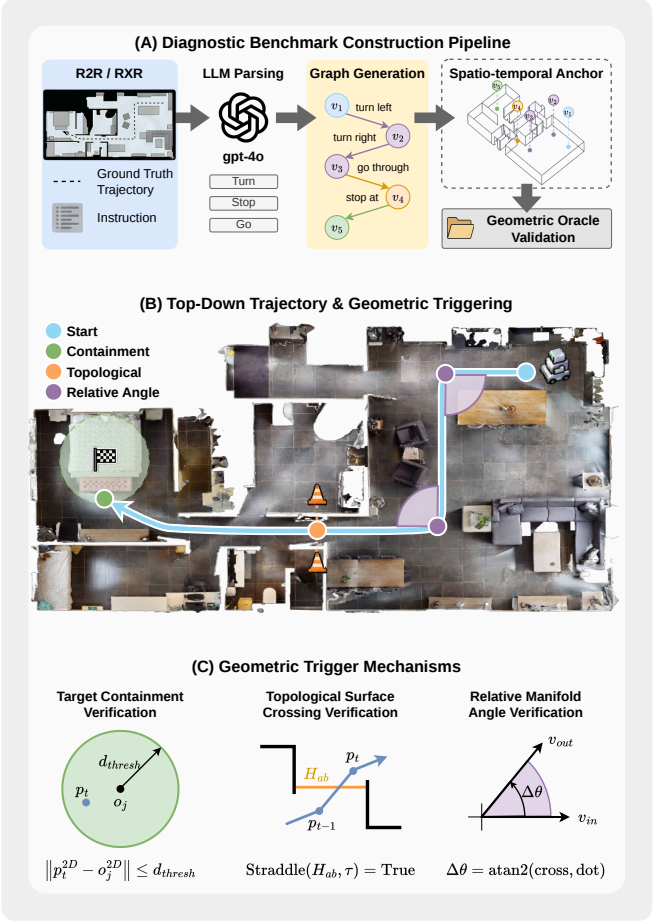


Fig. 3. **Construction pipeline of the StageNav-Bench and the GVO. (a)** Diagnostic benchmark construction pipeline. Long-horizon instructions are parsed by an LLM to generate a DAG, which is then spatially anchored and strictly filtered via GVO validation. **(b)** Top-down trajectory and geometric triggering. The continuous navigation trajectory activates different geometric constraints in a 3D environment. **(c)** (a) Diagnostic benchmark construction pipeline. The three fundamental algebraic triggers used by the GVO.

and the hyperplane is strictly verified using standard 2D line-segment intersection criteria based on cross-products. Let $f(\mathbf{u},\mathbf{v},\mathbf{w}) = (v_x - u_x)(w_y - u_y) - (v_y - u_y)(w_x - u_x)$, the intersection is valid if and only if:

$$\begin{aligned} f(\mathbf{a},\mathbf{b},\mathbf{p}_{t-1}) \cdot f(\mathbf{a},\mathbf{b},\mathbf{p}_t) < 0, \\ f(\mathbf{p}_{t-1},\mathbf{p}_t,\mathbf{a}) \cdot f(\mathbf{p}_{t-1},\mathbf{p}_t,\mathbf{b}) < 0. \end{aligned} \tag{3}$$

Type-C: Directional Alignment. Applied to explicit directional commands (Turn left/right/around, Go straight), this trigger maps discrete turning actions into continuous angular deviations. Let $\mathbf{p}_{ref}$ denote the anchor coordinate associated with v_i. The macroscopic outgoing displacement vector is defined as $\vec{\mathbf{v}}_{out} = \mathbf{p}_t - \mathbf{p}_{ref}$. Letting the reference incoming vector be $\vec{\mathbf{v}}_{in} = (x_1, y_1)$ and $\vec{\mathbf{v}}_{out} = (x_2, y_2)$, the yaw angle $\Delta\theta$ is computed via:

$$\Delta\theta = \text{atan2}(x_1 y_2 - x_2 y_1, x_1 x_2 + y_1 y_2) \tag{4}$$

The angle $\Delta\theta$ must satisfy semantic bounds defined by $e_{i,j}$: $[\theta_{min}, \theta_{max}]$ for left turns, $[-\theta_{max}, -\theta_{min}]$ for right turns, and $[-\theta_{thresh}, \theta_{thresh}]$ for straight movements.

C. The StageNav-Bench Construction

Based on the proposed GVO, we build **StageNav-Bench** by converting the English portions of R2R and RxR into graph-structured task representations. Each instruction is decomposed into semantically grounded nodes and edges, instantiated within the continuous 3D environments of Matterport3D, thereby enabling explicit node-level verification and diagnostic evaluation beyond trajectory-level metrics.

Instruction Parsing and Spatial Grounding. Initial DAGs are generated by utilizing an LLM to extract critical waypoint entities and directional edges from complex instructions $\mathscr{I}$. Extracted entity nodes are then geometrically grounded using 3D instance segmentation labels provided by the continuous simulation environment. This mapping assigns an objective physical coordinate $\mathbf{o}_i$ and specific geometric boundaries to each node v_i, activating corresponding GVO triggers.

GT Verification and Quality Assurance. To ensure high-fidelity spatial grounding and physical achievability, the human-annotated ground truth (GT) trajectories τ_{GT} are evaluated using the GVO. Trajectories that are initially misaligned with the parsed DAG undergo a human-in-the-loop refinement process to correct LLM parsing hallucinations and bounding box mismatches. This systematic curation process maximizes data utility, yielding a high-quality, geometrically aligned benchmark.

Dataset Statistics and Splits. As summarized in Table I, StageNav-Bench contains 27,576 geometrically validated long-horizon instructions across 86 Matterport3D scenes, maintaining a robust retention rate of approximately 70.1%. Crucially, to strictly prevent data leakage during reinforcement fine-tuning, StageNav-Bench inherits the environment splits from the original datasets. Models are trained exclusively on the isolated training split, guaranteeing that diagnostic evaluations on the *val-unseen* split reflect true zero-shot spatial generalization rather than topological overfitting.

TABLE I
STATISTICS OF STAGENAV-BENCH DATASET SPLITS.

Dataset	Split	#Instr. (Orig.)	#Validated	Avg. Nodes	Scenes
R2R-CE	Train	10,819	7,594	5.6	58
	Val-Seen	778	548	5.5	49
	Val-Unseen	1,839	1,289	5.8	10
RxR-CE	Train	19,996	13,997	9.4	56
	Val-Seen	2,255	1,580	9.1	52
	Val-Unseen	3,669	2,568	9.3	10
StageNav-Bench (Total)		-	27,576	8.3	86

D. State-Aligned Reinforcement Fine-Tuning

Leveraging the isolated training split of our constructed dataset as the interactive environment, we introduce StageNav-RFT. To mitigate accumulated errors in long-horizon navigation, the framework converts sparse task-level outcome rewards into structured node-level process rewards through oracle-aligned logical verification. This design enables step-wise supervision grounded in geometric and semantic consistency, facilitating stable policy improvement under complex multi-stage instructions.

Logic-Aligned Process Reward. At time step t, the base process reward r_t is formulated as:

$$r_t = r_{t,\text{logic}} + r_{t,\text{step}} - \lambda_{\text{invalid}} \cdot r_{t,\text{invalid}}. \tag{5}$$

Here, $r_{t,\text{invalid}}$ penalizes illegal actions (e.g., collisions), and $r_{t,\text{step}} = -\eta$ penalizes sequence length to encourage efficient convergence, where $\eta > 0$ is a fixed per-step penalty coefficient. The logic-aligned reward $r_{t,\text{logic}}$ evaluates the alignment between the agent's subjective cognitive state s_t^{cog} and the objective output of the GVO, denoted as $V_{\text{GVO}}^{(t)} \in \{\text{True}, \text{False}\}$. To minimize computational overhead, the GVO is only triggered upon a cognitive state transition ($s_t^{\text{cog}} \in \{\text{SUCCESS}, \text{FAILED}\}$):

$$r_{t,logic} = \begin{cases} +r_{TP}, & \text{if } s_t^{cog} = \text{SUCCESS} \wedge V_{GVO}^{(t)} = \text{True} \\ -r_{FP}, & \text{if } s_t^{cog} = \text{SUCCESS} \wedge V_{GVO}^{(t)} = \text{False} \\ -r_{FN}, & \text{if } s_t^{cog} = \text{FAILED} \wedge V_{GVO}^{(t)} = \text{True} \\ +r_{TN}, & \text{if } s_t^{cog} = \text{FAILED} \wedge V_{GVO}^{(t)} = \text{False} \end{cases} \tag{6}$$

This confusion matrix design is specifically tailored for state alignment and error recovery. Here, $+r_{TP}$ rewards accurate node completion, $-r_{FP}$ heavily penalizes overconfident hallucinations (claiming success when failing), and $-r_{FN}$ penalizes the agent for prematurely abandoning a successfully completed subtask. Crucially, the $+r_{TN}$ (True Negative) reward intrinsically motivates the agent to proactively recognize execution failures (e.g., being lost or physically blocked). This honest self-assessment serves as the prerequisite for triggering proactive topological backtracking. Finally, when $s_t^{cog} = \text{IN_PROGRESS}$, $r_{t,logic} = 0$, leaving the agent subject only to the intrinsic time-step penalty to prevent stalling.

E. Grouped Trajectory Sampling

To mitigate cross-task reward scale variance and obtain stable relative policy improvement signals, we sample N independent trajectories $\{\tau_1, \ldots, \tau_N\}$ under identical initial observation $\mathbf{O}_0$ and instruction constraints using the current policy $\pi_{\theta_{old}}$.

The cumulative dense return of trajectory τ_i is computed as:

$$R(\tau_i) = \sum_{t=1}^{T_i} \gamma^{t-1} r_t^{(i)}. \tag{7}$$

where $\gamma \in (0, 1]$ is the discount factor.

Since the GVO provides semantically grounded logical rewards at meaningful cognitive transitions, the resulting returns remain informative even without reaching the terminal goal.

We estimate the normalized intra-group advantage as:

$$A_i = \frac{R(\tau_i) - \mu_R}{\sigma_R}, \tag{8}$$

where μ_R and σ_R denote the mean and standard deviation of returns within the group. This grouped normalization removes task-level heterogeneity and ensures that advantage estimation reflects relative behavioral quality under identical task constraints.

F. Oracle-Guided Policy Update

The normalized trajectory-level advantage A_i is distributed across all decision steps within trajectory τ_i. Given that logical rewards are triggered only at semantically meaningful cognitive transitions, this trajectory-level ranking preserves temporal consistency while avoiding high-variance step-wise advantage estimation.

We optimize the policy using a clipped surrogate objective:

$$\begin{aligned} \mathscr{J}_{StageNav}(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta_{old}}} \Bigg[\frac{1}{N} \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} \min \Big(\rho_{i,t}(\theta) A_i, \\ \text{clip}(\rho_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon) A_i \Big) - \beta \mathbb{D}_{KL}(\pi_\theta \,\|\, \pi_{ref}) \Bigg], \end{aligned} \tag{9}$$

where

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(a_t^{(i)}, s_t^{cog(i)} \mid \mathbf{O}_t^{(i)})}{\pi_{\theta_{old}}(a_t^{(i)}, s_t^{cog(i)} \mid \mathbf{O}_t^{(i)})} \tag{10}$$

denotes the importance sampling ratio over the joint action–cognitive space.

This joint formulation ensures that policy improvement consistently aligns both physical execution and cognitive commitment decisions. The KL divergence regularization term constrains deviation from the reference policy π_{ref}, preventing catastrophic forgetting of foundational vision–language alignment during navigation fine-tuning.

IV. EXPERIMENTS

A. Experimental Setup

Environments and Benchmarks. We evaluate our framework in continuous 3D environments via the Habitat simulator [13]. Macro-level navigation performance is assessed on the standard VLN-CE datasets, including R2R-CE and the long-horizon RxR-CE. For micro-level cognitive and diagnostic evaluation, we utilize our proposed StageNav-Bench. Crucially, to strictly prevent data leakage, all diagnostic metrics are evaluated exclusively on the isolated *val-unseen* split, ensuring that the results reflect genuine zero-shot spatial generalization.

Baselines. To ensure a comprehensive comparison, we evaluate against representative methods from three paradigms: (1) classic and mapping-based VLN systems, (2) zero-shot or training-free LLM agents, and (3) recent LVLM-based navigation models. The specific implementations included in each category are detailed in Table II.

Evaluation Metrics. For traditional macro-level evaluation, we report Navigation Error (NE), Oracle Success Rate (OS), SR, and SPL. For the micro-level diagnostic evaluation on the StageNav-Bench, we formally define four novel metrics based on the confusion matrix of cognitive-physical state alignment. Let N_{total} and $N_{\text{completed}}$ denote the total and physically reached sub-goal nodes, respectively. Furthermore, we define True Positives (TP) as verified `SUCCESS` declarations, False Positives (FP) as unverified `SUCCESS` (hallucinations), True Negatives (TN) as accurate `FAILED` declarations, and False Negatives (FN) as missed completions or timeouts:

TABLE II

COMPARISON ON R2R AND RXR TASKS(ENGLISH ONLY). METRICS WITH ARROWS INDICATE IMPROVEMENT DIRECTION (↓: LOWER IS BETTER, ↑: HIGHER IS BETTER). THE BEST RESULTS ARE IN **BOLD** AND THE SECOND-BEST ARE UNDERLINED.

Method	Pano.	S. RGB	Depth	Zero-shot	R2R				RXR			
					NE↓	OS↑	SR↑	SPL↑	NE↓	OS↑	SR↑	SPL↑
Seq2Seq [ECCV20] [14]		✓	✓		7.9	36.0	23.0	21.0	12.2	25.5	13.5	11.5
VLN-BERT [CVPR22] [15]	✓		✓		5.68	54.0	45.0	39.5	8.85	41.5	28.0	23.1
CMA [CVPR22] [16]	✓		✓		6.3	51.0	40.0	35.0	8.8	37.5	26.0	21.8
Sim2Sim [ECCV22] [17]	✓		✓		6.01	53.0	44.0	37.0	–	–	–	–
Ego^2-Map [ICCV23] [18]	✓		✓		5.61	55.0	46.0	40.0	–	–	–	–
GridMM [ICCV23] [19]	✓		✓		5.15	60.0	48.0	40.0	–	–	–	–
Reborn [CVPRW22] [20]	✓		✓		5.45	56.0	49.0	45.0	6.05	–	47.5	41.0
ZSON [NeurIPS22] [21]		✓	✓	✓	–	–	20.1	9.8	–	–	14.8	5.1
NavGPT-CE [AAAI24] [22]	✓		✓	✓	8.5	26.5	15.8	9.9	–	–	–	–
CA-Nav [TPAMI25] [23]		✓	✓	✓	7.4	49.0	26.1	11.2	10.2	–	19.5	6.3
InstructNav [CoRL24] [24]	✓		✓	✓	7	–	30.5	23.5	–	–	–	–
GC-VLN [CoRL25] [7]		✓		✓	7.1	42.5	34.2	17.0	8.6	45.0	34.5	14.2
NaVid [RSS24] [25]		✓			5.55	48.5	36.8	35.2	8.5	34.0	23.2	20.8
Uni-NaVid [RSS25] [26]		✓			5.5	54.0	47.8	43.5	6.1	56.1	49.5	41.6
NaVILA [RSS24] [27]		✓			5.3	61.5	53.0	48.0	6.85	–	48.5	42.7
ETPNav [TPAMI24] [28]	✓		✓		4.8	64.0	56.0	48.1	5.75	60.5	53.5	**44.0**
StageNav (Ours)		✓			**4.68**	**65.5**	**59.1**	**49.9**	**5.53**	**61.9**	**56.8**	43.1

- **Node Success Rate (NSR):** It measures the proportion of physical sub-goal nodes successfully reached within the DAG topology, evaluating the pure execution capability in long-horizon tasks.

$$\text{NSR} = \frac{N_{\text{completed}}}{N_{\text{total}}}$$

- **Cognitive Accuracy (CA):** It measures the global alignment consistency between the agent's cognitive judgment and the physical reality. Note that cases where the agent exhausts the maximum step limit without declaring a status (Timeout) are classified as False Negatives (FN), as the agent fails to recognize the objective inability to proceed.

$$\text{CA} = \frac{TP+TN}{TP+FP+FN+TN}$$

- **Success Credibility (SC):** It denotes the precision of the positive class. It evaluates the reliability of the agent's state declarations by answering the proportion of true physical successes among all `SUCCESS` outputs.

$$\text{SC} = \frac{TP}{TP+FP}$$

- **Active Recall Rate (ARR):** It measures the active failure awareness capability. It represents the proportion of proactive `FAILED` declarations (triggering backtracking) among all sub-tasks that ultimately fail to reach the physical goal.

$$\text{ARR} = \frac{TN}{FP+TN}$$

Implementation Details. Our policy network is initialized with Qwen2.5-VL-7B-Instruct [29] and optimized via LoRA [30] on four NVIDIA A100 (40GB) GPUs. The training pipeline comprises a Supervised Fine-Tuning (SFT) phase using standard teacher-forcing on the original training sets, followed by a Reinforcement Fine-Tuning (RFT) phase. During RFT, the agent interacts exclusively with the isolated *training split* of StageNav-Bench. For the oracle-guided GRPO, the trajectory sampling group size is set to $N = 16$. Logic-aligned reward parameters are empirically defined as $r_{TP} = 1.0$, $r_{FP} = 1.5$, $r_{FN} = 0.5$, and $r_{TN} = 0.5$, which severely penalizes cognitive hallucinations while encouraging proactive failure recognition. The maximum interaction steps per sub-task node are capped at 50.

B. Main Results on Continuous Environments

Table II presents the quantitative evaluation on the continuous R2R-CE and RxR-CE environments. On the R2R unseen validation set, StageNav achieves a leading SR of 59.1% and an SPL of 49.9%. This performance directly validates the necessity of explicit cognitive state machines compared to alternative paradigms. Specifically, it surpasses the mapping-based state-of-the-art ETPNav (56.0% SR) and the implicit LVLM baseline NaVILA (53.0% SR), revealing that explicit spatial maps or implicit policy representations alone are insufficient without fine-grained cognitive-physical state alignment. Furthermore, zero-shot and training-free methods, such as GC-VLN and InstructNav, suffer severe structural failures, yielding SRs of only 34.2% and 30.5% respectively. This stark contrast demonstrates that prompting LLMs is inherently inadequate for resolving complex spatio-temporal alignment without geometrically grounded process supervision.

The architectural advantages of StageNav are further magnified in the long-horizon RxR environment, which serves as a rigorous stress test for error accumulation. Under these extended decision sequences, our framework achieves an SR of 56.8%, establishing an absolute improvement of 3.3% over ETPNav and 8.3% over NaVILA. This amplified performance margin is not coincidental; it confirms the hypothesis that explicit DAG-guided process verification structurally suppresses the compounding errors that typically paralyze

TABLE III

PERFORMANCE COMPARISON OF VARIOUS VISION-LANGUAGE MODELS (VLMS) AND ABLATION STUDY ON STAGENAV-BENCH.

Model	NE↓	OS↑	SR↑	SPL↑	NSR↑	CA↑	SC↑	ARR↑
Close-Source VLMs								
Claude-3.7-Sonnet	7.41	47.1	25.2	11.8	38.5	41.2	39.5	18.5
Claude-3.5-Sonnet	7.35	41.5	20.3	9.5	31.5	35.8	32.1	14.2
GPT-4o	7.50	43.2	21.8	10.5	33.2	37.5	34.5	16.0
GPT-4o-mini	8.12	24.1	12.5	8.2	20.5	26.4	22.8	8.5
GPT-4.1	7.55	46.5	24.8	11.5	37.8	40.5	38.2	17.8
Gemini-1.5-Pro	8.45	18.5	9.5	6.8	16.2	23.5	18.5	6.2
Gemini-2.0-flash	7.62	45.8	24.2	11.2	36.5	39.8	37.0	15.5
Open-Source VLMs								
LLama-3.2-90B-Vision-Ins	7.65	39.8	19.5	10.2	29.5	33.5	30.2	11.8
LLama-3.2-11B-Vision-Ins	8.25	21.5	10.2	7.5	17.5	25.1	19.5	7.0
Qwen2.5-VL-72B-Ins	7.38	40.5	20.8	10.0	28.5	32.8	29.5	11.2
Qwen2.5-VL-7B-Ins	7.75	29.5	16.5	9.2	25.5	29.8	27.5	10.5
Qwen2.5-VL-3B-Ins	8.85	13.5	6.8	4.8	12.5	19.5	14.5	4.5
gemma-3-27b-it	6.85	42.5	21.2	10.8	32.0	35.5	32.5	12.8
Ablation								
SFT Only	5.75	50.5	39.5	31.8	55.5	34.5	35.8	16.5
SFT + Standard GRPO	5.35	55.5	45.8	36.5	60.2	41.2	42.5	24.5
StageNav-RFT (w/o r_{TN})	4.85	62.5	55.2	46.5	67.5	63.5	62.2	46.5
StageNav-RFT (Full)	**4.68**	**65.5**	**59.1**	**49.9**	**70.5**	**66.5**	**65.8**	**59.8**

long-sequence Markov Decision Processes.

On RxR, StageNav attains a higher success rate but a slightly lower SPL (43.1% vs. 44.0%). This reflects a deliberate design trade-off: the diagnostic mechanism triggers backtracking upon detecting deviations, increasing path length but preventing irreversible failures. The resulting exploratory recovery mildly reduces SPL while improving global task completion.

C. Diagnostic Evaluation on StageNav-Bench

Table III presents the diagnostic metrics evaluated on the proposed StageNav-Bench. This evaluation specifically assesses the models across three structural dimensions of cognitive reliability: execution extent (NSR), declaration reliability (SC), and diagnostic awareness (ARR), thereby establishing a causal link between micro-level cognitive behaviors and macro-level task success (SR).

Under zero-shot settings, proprietary LVLMs (e.g., GPT-4.1) achieve a baseline physical execution with an NSR of 37.8%. However, they exhibit a severe *over-declaration bias*. Specifically, GPT-4.1 yields an SC of only 38.2%, revealing that the majority of its `SUCCESS` declarations are textual hallucinations structurally decoupled from physical reality. Furthermore, an ARR of 17.8% indicates a systemic inability to intrinsically recognize execution failures. This structural misalignment between cognitive assertions and geometric states inevitably cascades into terminal task failures, capping its final SR at 24.8%.

A parallel structural flaw is observed in open-source baselines. The un-finetuned Qwen2.5-VL-7B-Ins demonstrates an acute deficit in spatial-cognitive alignment, recording an SC of 27.5% and an ARR of 10.5%.

In contrast, the proposed StageNav-RFT framework explicitly enforces state alignment via the Physical Oracle (GVO). By architecturally grounding the cognitive state machine, the framework significantly elevates SC to 65.8% and ARR to 59.8%. This structural intervention causally suppresses overconfident hallucinations and intrinsically activates autonomous error recovery, driving the global SR to a leading 59.1%.

D. Ablation Study

To validate the causal contribution of each structural component within the StageNav-RFT framework, we conduct an ablation study as detailed in Table III.

The *SFT Only* variant achieves an NSR of 55.5% but exhibits weak cognitive grounding (CA: 34.5%, SC: 35.8%, ARR: 16.5%). This confirms that while behavioral cloning successfully fits expert physical trajectories, it structurally fails to induce out-of-distribution diagnostic behaviors; the agent executes actions without genuinely monitoring its execution state.

Incorporating standard GRPO (*SFT + Standard GRPO*) yields limited cognitive improvements (CA: 41.2%, SC: 42.5%). This exposes the fundamental limitation of sparse, episode-level rewards in Graph-Constrained MDPs: they fail to resolve step-level credit assignment, leaving the agent's internal state transitions misaligned with the environment's topological constraints.

The integration of the Logic-Aligned Process Reward fundamentally shifts this paradigm. Specifically, removing the True Negative incentive (*StageNav-RFT w/o* r_{TN}) restricts the ARR to 46.5%. While the model learns to advance, it lacks sufficient intrinsic motivation to proactively flag deviations. The full *StageNav-RFT* configuration establishes a complete theoretical loop by explicitly rewarding honest failure recognition. This necessary incentive structure maximizes diagnostic awareness (ARR: 59.8%) and declaration reliability (SC: 65.8%), proving that the r_{TN} mechanism causally triggers the topological backtracking essential for robust long-horizon navigation (SR: 59.1%).

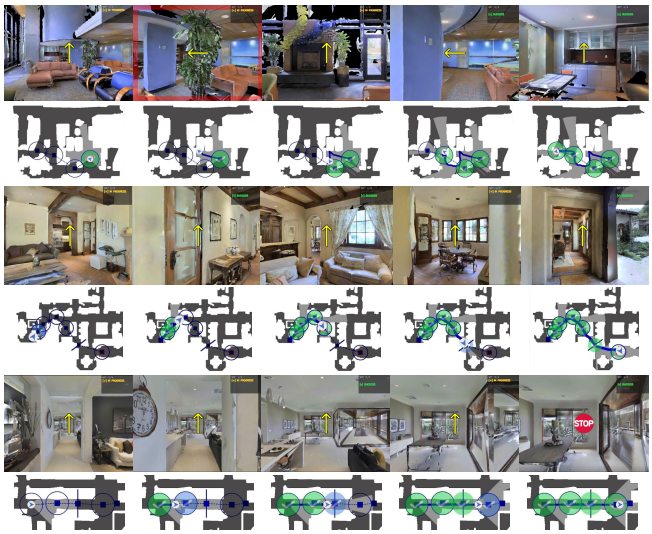


Fig. 4. **Qualitative navigation examples of the StageNav.** Each example displays the agent's first-person egocentric views **(top)** alongside the corresponding top-down Bird's-Eye View (BEV) trajectories **(bottom)**. The BEV maps explicitly visualize the spatial layout of the three GVO triggers evaluated during execution: circles denote Type-A triggers (Target Containment), lines represent Type-B triggers (Topological Surface Crossing), and arrows indicate Type-C triggers (Relative Manifold Angle).

E. Qualitative Analysis

To intuitively illustrate the structural alignment mechanism, Figure 4 provides qualitative navigation examples of the StageNav agent. The top-down BEV trajectories explicitly visualize how the agent's spatial progress is continuously grounded by the predefined GVO triggers. This visual evidence further corroborates the quantitative improvements in declaration reliability and diagnostic awareness, confirming that StageNav effectively bridges the gap between language generation and embodied physical execution.

V. CONCLUSIONS

In this work, we propose StageNav to address structural mismatches in long-horizon Vision-and-Language Navigation. We introduce a constraint-driven benchmark that shifts evaluation from coarse outcomes to fine-grained geometric diagnostics. The StageNav-RFT method reformulates policy learning as a cognitive state machine, using logic-aligned process rewards and homologous trajectory sampling to reduce temporal credit assignment and environmental bias while improving autonomous error recovery. Future work will extend this paradigm to multi-floor and open-world settings and incorporate memory with topological supervision for better cross-scene generalization. We hope StageNav provides a unified foundation for reliable long-horizon planning in embodied AI.

REFERENCES

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Hengel. "Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3674–3683.

[2] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y. Wang, W.Wang, and L. Zhang. "Reinforced cross-Modal matching and self-supervised imitation learning for vision-language navigation," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6629-6638.

[3] J. Li and M. Bansal. "Improving vision-and-language navigation by generating future-view image semantics," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 10803-10812.

[4] P. Xu, X. Gong, and Y. Mu. "NavQ: Learning a Q-Model for foresighted vision-and-language navigation," in IEEE/CVF International Conference on Computer Vision (ICCV), 2025, PP. 6327-6341.

[5] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge. "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 4392-4412.

[6] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Wang, C. Shen, and A. Hengel. "Reverie: Remote embodied visual referring expression in real indoor environments," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9982-9991.

[7] H. Yin, H. Wei, X. Xu, W. Guo, J. Zhou, and J. Lu. "GC-VLN: Instruction as graph constraints for training-free vision-and-language navigation,". in Conference on Robot Learning (CoRL), 2025.

[8] J. Chen, B. Lin, R. Xu, Z. Chai X. Liang, and K. Wong. "MapGPT: Map-Guided prompting with adaptive path planning for vision-and-language Navigation," in Meeting of the Association for Computational Linguistics (ACL), 2024, pp. 9796-9810.

[9] Z. Qi, Zhang Z, Yu Y, J. Wang, and H. Zhao. "VLN-R1: Vision-language navigation via reinforcement fine-tuning," arXiv print arXiv:2506.17221, 2025.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[11] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, and D. Guo. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," arXiv preprint arXiv:2402.03300, 2024.

[12] L. Feng, Z. Xue, T. Liu, and B. An. "Group-in-Group policy optimization for LLM agent training," in Neural Information Processing Systems (NeurIPS), 2025.

[13] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D.v Batra. "Habitat: A platform for embodied ai research," in IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9339-9347.

[14] J. Krantz, E. Wijmans, A. Majundar, D. Batra, and S. Lee. "Beyond the nav-graph: Vision and language navigation in continuous environments," in European Conference on Computer Vision (ECCV), 2020, pp. 104–120.

[15] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould, "VLN BERT: A recurrent vision-and-language BERT for navigation," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 1643–1653.

[16] Y. Hong, Z. Wang, Q. Wu, and S. Gould. "Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 15439-15449.

[17] J. Krantz and S. Lee. "Sim-2-sim transfer for vision-and-language navigation in continuous environments," in European Conference on Computer Vision (ECCV), 2022. pp. 588–603.

[18] Y. Hong, Y. Zhou, R. Zhang, F. Dernoncourt, T. Bui, S. Gould, and H. Tan. "Learning navigational visual representations with semantic map supervision," in IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 3055-3067.

[19] Z. Wang, X. Li, J. Yang, Y. Liu, and S.Jiang. "Gridmm: Grid memory map for vision and-language navigation," in IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 15625-15636.

[20] D. An, Z. Wang, Y. Li, Y. Wang, Y.Hong, Y. Huang, L. Wang, and J.Shao. "1st place solutions for rxr-habitat vision-and-language navigation competition," in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022.

[21] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra. "Zson: Zero-shot object goal navigation using multimodal goal embeddings," in Neural Information Processing Systems (NeurIPS), 2022, pp. 32340-32352.

[22] G. Zhou, Y. Hong, and Q. Wu. "Navgpt: Explicit reasoning in vision-and-language navigation with large language models," in AAAI Conference on Artificial Intelligence (AAAI), 2024, vol. 38, no. 7, pp. 7641–7649.

[23] K. Chen, D. An, Y. Huang, R. Xu, Y. Su, Y. Ling, I. Reid, and L. Wang. "Constraint aware zero-shot vision-language navigation in continuous environments," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 47, no. 11, pp. 10441-10456, 2025.

[24] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong. "Instructnav: Zero-shot system for generic instruction navigation in unexplored environment," in Conference on Robot Learning (CoRL), 2024.

[25] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and W. He. "Navid: Video-based vlm plans the next step for vision-and-language navigation," in Robotics: Science and Systems (RSS), 2024.

[26] J. Zhang, K. Wang, S. Wang, M. Li, H. Liu, S. Wei, Z. Wang, Z. Zhang, and H. Wang. "Uninavid: A video-based vision-language-action model for unifying embodied navigation tasks," in Robotics: Science and Systems (RSS), 2025.

[27] A. Cheng, Y. Ji, Z. Yang, X. Zou, J. Kautz, E. Biyik, H. Yin, S. Liu, and X. Wang "Navila: Legged robot vision-language-action model for navigation," in Robotics: Science and Systems (RSS), 2025.

[28] D. An, H. Wang, W. Wang, Z. Wang, Y. Huang, K. He, and L. Wang. "Etpnav: Evolving topological planning for vision-language navigation in continuous environments," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 47, no. 7, pp. 5130-5145, 2025.

[29] 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.

[30] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. "Lora: Low-rank adaptation of large language models," in International Conference on Learning Representations (ICLR), 2022.