LongNav-R1: Horizon-Adaptive Multi-Turn RL for Long-Horizon VLA Navigation

Yue Hu[1], Avery Xi[1], Qixin Xiao[1], Seth Isaacson[1], Henry X. Liu[1], Ram Vasudevan[1], Maani Ghaffari[1]
[1] University of Michigan, Ann Arbor
{huyu,axi,qxiaocs,sethgi,henryliu,ramv,maanigj}@umich.edu

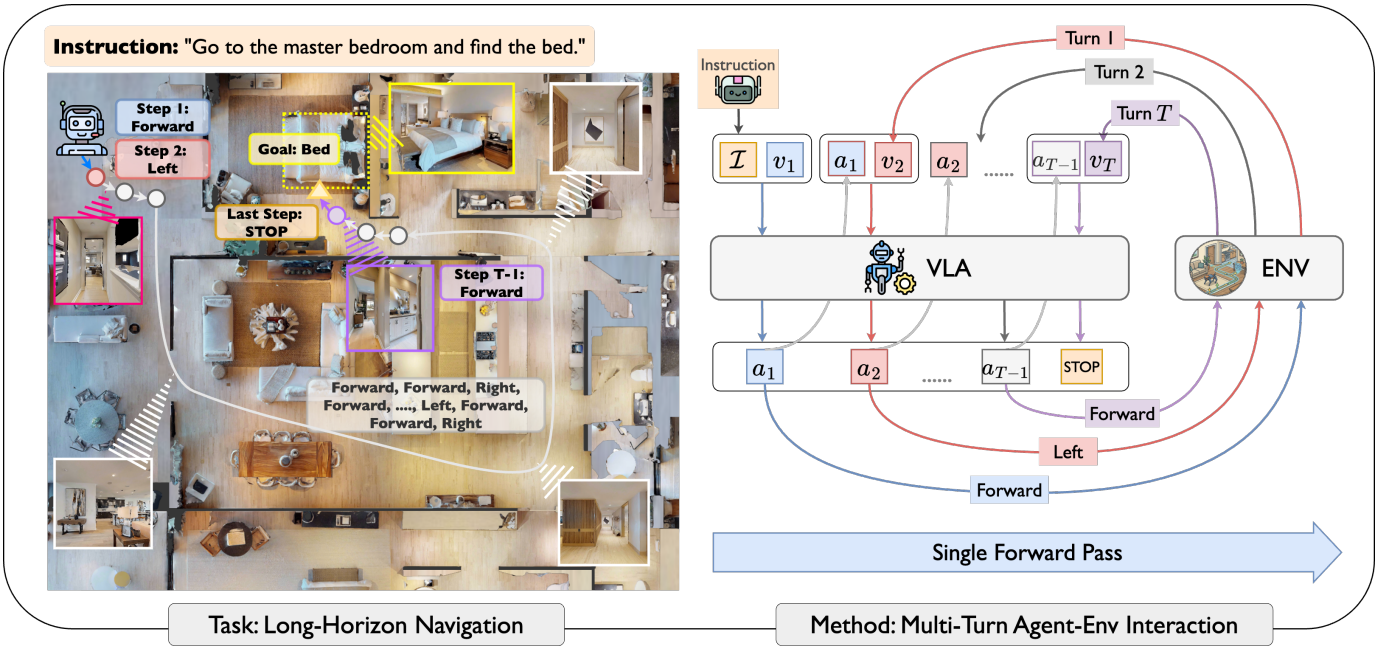


Fig. 1: LongNav-R1 formulates the navigation process as a multi-turn conversation between the VLA policy and the embodied environment. This end-to-end multi-turn RL framework enables the VLA policy to optimize multi-step decision-making based on cumulative, sequential outcomes.

Abstract*—This paper develops LongNav-R1, an end-to-end multi-turn reinforcement learning (RL) framework designed to optimize Visual-Language-Action (VLA) models for long-horizon navigation. Unlike existing single-turn paradigm, LongNav-R1 reformulates the navigation decision process as a continuous multi-turn conversation between the VLA policy and the embodied environment. This multi-turn RL framework offers two distinct advantages: i) it enables the agent to reason about the causal effects of historical interactions and sequential future outcomes; and ii) it allows the model to learn directly from online interactions, fostering diverse trajectory generation and avoiding the behavioral rigidity often imposed by human demonstrations. Furthermore, we introduce Horizon-Adaptive Policy Optimization. This mechanism explicitly accounts for varying horizon lengths during advantage estimation, facilitating accurate temporal credit assignment over extended sequences. Consequently, the agent develops diverse navigation behaviors and resists collapse during long-horizon tasks. Experiments on object navigation benchmarks validate the framework's efficacy: With 4,000 rollout trajectories, LongNav-R1 boosts the Qwen3-VL-2B success rate from 64.3% to 73.0%. These results demonstrate superior sample efficiency and significantly outperform state-of-the-art methods. The model's generalizability and robustness are further validated by its zero-shot performance in long-horizon real-world navigation settings. All source code is open-sourced at https://github.com/UMich-CURLY/LongNav-R1.

I. Introduction

Navigation is a fundamental capability for intelligent embodied agents, serving as the cornerstone for robots to assist humans in physical environments. Historically, navigation systems relied on modular pipelines [52, 2, 7, 72, 35] involving separate perception [28], mapping [74, 62], and planning [19, 60, 61] components. However, recent progress has shifted toward end-to-end Vision-Language-Action (VLA) models [69, 68, 48]. These models leverage large-scale pre-training to enable general semantic navigation, allowing agents to interpret complex visual cues and linguistic commands directly into actions.

Despite these advancements, current navigation approaches remain far from achieving human-level performance, particularly in long-horizon tasks. This is because existing state-of-the-art methods [69, 68, 27, 48] adopt a single-turn imitation learning paradigm, which reduces navigation to a sequence of isolated action predictions based on immediate local context.

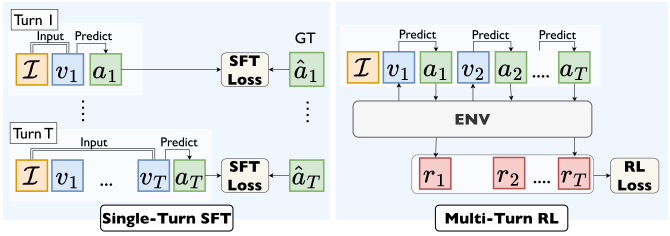


Fig. 2: Comparison of single-turn SFT and multi-turn RL.

This formulation introduces two critical deficiencies: first, it lacks causal reasoning by treating steps independently, thereby overlooking the sequential dependencies where early-stage exploration serves as a prerequisite for late-stage, goal-directed efficiency. Second, it leads to behavioral rigidity by strictly imitating expert trajectories instead of optimizing for goal success. Consequently, the agent becomes myopic and brittle, incapable of recovering from errors or adapting to distribution shifts.

To bridge this gap, this paper proposes LongNav-R1, an end-to-end framework that reformulates navigation as a multi-turn Reinforcement Learning (RL) process. Unlike the single-turn paradigm, LongNav-R1 treats the navigation task as a continuous conversation between the VLA policy and the physical environment. This multi-turn formulation offers two key advantages. First, it provides the model with comprehensive state and objective awareness, allowing it to learn the causal relationship between current actions and distant rewards. Second, by learning directly from online interactions, the agent is encouraged to explore diverse trajectories, thereby overcoming the rigidity of human demonstrations and improving robustness against environmental stochasticity.

While multi-turn RL offers a promising framework for long-horizon VLA navigation, its deployment is bottlenecked by the challenge of temporal credit assignment, as the relative contribution of each turn to the final objective varies significantly over time. While actor-critic methods like PPO [40] manage this via learned value functions, they incur prohibitive computational overhead during long-horizon training. Conversely, efficient critic-free methods from the LLM domain, such as GRPO [41] and REINFORCE++ [18], are optimized for single-turn tasks with outcome-level rewards and fail to capture the evolving temporal statistics inherent in multi-step robotic decision-making.

To address this bottleneck, we introduce Horizon-Adaptive Policy Optimization (HAPO), a framework for critic-free advantage estimation. HAPO eliminates the need for a separate value network by regressing a baseline directly from the local rollout buffer via kernel regression. Drawing inspiration from variance reduction techniques in verifiable RL [66], we introduce a general formulation that unifies the advantage estimation of critic-free methods with the value function approximation of actor-critic architectures. Under this formulation, existing optimization heuristics such as GRPO and REINFORCE++ emerge as special cases determined by specific kernel choices. By explicitly designing a temporal kernel function, HAPO enables the derived baseline to capture the reward's temporal dynamics of long-horizon navigation, offering the temporal precision of a critic without the associated computational overhead. HAPO enables optimizing VLA models for the varying sequence lengths inherent in long-horizon robotic tasks.

To validate the effectiveness of LongNav-R1 and HAPO, we conducted three key evaluations. First, we validated LongNav-R1 across real-world settings and four widely-used simulation benchmarks. Despite being trained on only six object categories in HM3D V1 [36], LongNav-R1 successfully localizes unseen objects in zero-shot real-world scenarios and outperforms previous SOTA methods on HM3D V1 [36], V2 [57], MP3D [22] and OVON [63], demonstrating robust generalizability. Second, we conducted ablation studies on core components. Results indicate that multi-turn RL contributes a substantial improvement of 8.7% on success rate compared to SFT, while HAPO is critical for unlocking long-horizon navigation, boosting success rates from 0% to 15.4%. Third, we demonstrated that LongNav-R1 yields highly sample-efficient agents. The success rate of a Qwen3-VL-2B backbone increased from 0.51% to 73.0% using only 34k trajectories, outperforming SFT baselines trained on millions of data.

To summarize, the primary contributions of this paper are:

(i) We propose LongNav-R1, a end-to-end multi-turn reinforcement learning framework that optimizes the VLA policy for multi-step navigation.

(ii) We introduce HAPO, a critic-free advantage estimation framework that facilitates precise temporal credit assignment, allowing large VLA models to improve multi-step decision-making without the significant computational burden of auxiliary critic networks.

(iii) We provide a comprehensive experimental validation of LongNav-R1 in real-world and diverse navigation benchmarks, demonstrating that LongNav-R1 significantly outperforms existing methods.

II. Related Works

Semantic navigation. Semantic navigation requires agents to navigate to the specific target in unseen environments based on human instructions. There are two representative tasks that involve both visual information and language instructions: Object Goal Navigation [10, 36, 57, 8] and Vision-and-Language Navigation [4, 23, 24]. Early methods [50, 32, 59, 30, 31] largely focused on acquiring task-specific skills via imitation learning [37, 38] or RL [9, 16, 13, 67]. While these methods can achieve strong performance in trained environments, they often suffer from poor generalization due to domain gaps. Recently, the field has shifted toward leveraging the generalization capabilities of Large-Language Models (LLMs) and Vision-Language Models (VLMs) to improve multi-task navigation. These approaches [21, 33, 43, 5, 44, 6, 14, 26, 64, 73, 29, 20, 60, 69, 68, 27] offer greater flexibility and adaptability in novel environments, but often lack optimized task execution and navigation efficiency, resulting in inferior performance. In contrast, our method trains a VLA model end-to-end with navigation objective, offering both task-aware efficiency and generalization capability.

Large language model for embodied navigation. Large language models have been introduced into robotic navigation due to their generalization capabilities in understanding and planning. Current research follows two primary trajectories. The first utilizes off-the-shelf LLMs and VLMs in a zero-shot fashion. These methods employ LLMs and VLMs as high-level policies to identify landmarks [65, 64] and select frontiers [29, 29, 20, 60]. The second trajectory [68, 69, 48] involves fine-tuning VLMs as end-to-end VLA models that directly generate actions via Supervised Fine-Tuning (SFT). However, SFT-based methods are often bottlenecked by the need for exhaustive expert demonstrations and are susceptible to the domain drift inherent in behavior cloning. To overcome these limitations, we train our VLA model using end-to-end RL, which facilitates the discovery of diverse navigation skills and enhances final navigation performance.

Reinforcement learning for large language model. Reinforcement learning has become a crucial post-training technique for large language models, facilitating alignment with human preferences [34] and significantly bolstering reasoning capabilities [6]. While these advancements have primarily been applied to web search [49, 53] and tool-use [46, 54, 47], adapting these benefits to embodied navigation remains an open challenge. A pioneering effort, Nav-R1 [27], adopts the DeepSeek-R1 paradigm [17] by generating navigation action sequences in a single turn and training the VLA model via GRPO [41]. However, embodied navigation is inherently a multi-turn decision-making process involving hundreds of steps; a single-turn formulation fails to capture its sequential and long-horizon nature. To address this, we propose an end-to-end, multi-turn RL framework for VLA models. Furthermore, rather than directly adopting standard advantage estimations from GRPO [41], REINFORCE++ [18], or RLOO [1], we introduce horizon-adaptive advantage estimation. This technique stabilizes interaction-adaptive optimization and improves long-horizon decision-making performance in embodied navigation.

III. Problem Formulation

This section establishes the mathematical formulation for robotic navigation using a VLA policy. We first introduce the navigation objective and frame the multi-step navigation decision process as a multi-turn conversation. Then we define a multi-turn RL formulation tailored for VLAs. Building on this foundation, we introduce critic-free horizon-adaptive policy optimization to optimize VLA for long-horizon objective.

A. Navigation task objective

We address the task of general robotic navigation in unknown environments. In this setting, an agent is provided with a open instruction $\mathcal{I}$, describing either a target object or a sequence of steps, and iteratively perceives and interacts with the environment to locate the goal. At each time step t, the agent receives an observation v_t and constructs a state s_t. Action selection is governed by a parameterized policy π_θ conditioned on the current state s_t:

$$a_t \sim \pi_\theta(\cdot \mid s_t). \tag{1}$$

An episode terminates when the agent invokes a 'stop' action. Success is defined as reaching the target's vicinity within a fixed step budget. Our objective is to optimize π_θ to maximize both navigation efficiency and success rate.

B. Multi-turn RL formulation of VLA policy

To solve the general navigation tasks, we utilize recent advances in large foundation models to optimize a VLA policy under the navigation objective. Given that navigation is a long-horizon process often spanning hundreds of steps, we frame the task as a multi-turn conversation between the VLA policy and its embodied environment. We model actions auto-regressively, conditioned on the global instruction $\mathcal{I}$, history $\{v_{1:t-1}, a_{1:t-1}\}$, and current observation v_t:

$$a_t \sim \pi_\theta(\cdot \mid \mathcal{I}, \{v_{1:t-1}, a_{1:t-1}\}, v_t). \tag{2}$$

This multi-turn RL framework offers two advantages. First, it enables KV cache reuse, significantly enhancing computational efficiency during training and inference. The history $\{v_{1:t-1}, a_{1:t-1}\}$ is saved in cache and only need to compute new observation. This optimization is critical for long-horizon navigation, which involves hundreds of steps and generates hundreds of thousands of visual tokens. Second, the sequential nature of our formulation allows the model to explicitly account for the causal dependencies of prior interactions. By attributing the final trajectory outcome to individual actions, we improve both learning efficiency and performance on long-horizon tasks.

Comparison with existing VLA policies. In contrast to UniNavid [69] and Nav-R1 [27], which adopt a single-turn paradigm, our multi-turn paradigm explicitly models the sequence effects. These prior works predict actions and optimize at the step-level, failing to capture the sequential nature of navigation. Furthermore, while StreamVLN [48] utilizes a multi-turn structure, it relies on SFT. Consequently, it suffers from the severe distribution shift issues inherent to behavior cloning and lacks a mechanism to directly optimize for the final, long-term success of the navigation task.

C. Horizon-adaptive policy optimization

To optimize VLA policies for long-horizon navigation, we propose Horizon-Adaptive Policy Optimization (HAPO). While recent advancements such as REINFORCE++ [18] and GRPO [41] have demonstrated efficacy in LLM training, their applicability is constrained by a reliance on single-turn paradigms and sparse, outcome-level rewards. By treating all time steps uniformly, these frameworks fail to capture the temporal dynamics of robotic navigation. In reality, moments like critical turns are far more determinative of success than routine straight-line movements. HAPO addresses this temporal credit assignment challenge by employing a kernel-based advantage estimator that uses step-level dense rewards, allowing the model to explicitly weigh contributions across varying horizons. HAPO has two phases: i) formulating advantage estimation as a generalized non-parametric value estimation framework via kernel regression, and ii) introducing a temporal kernel design to enable horizon-adaptive advantage estimation.

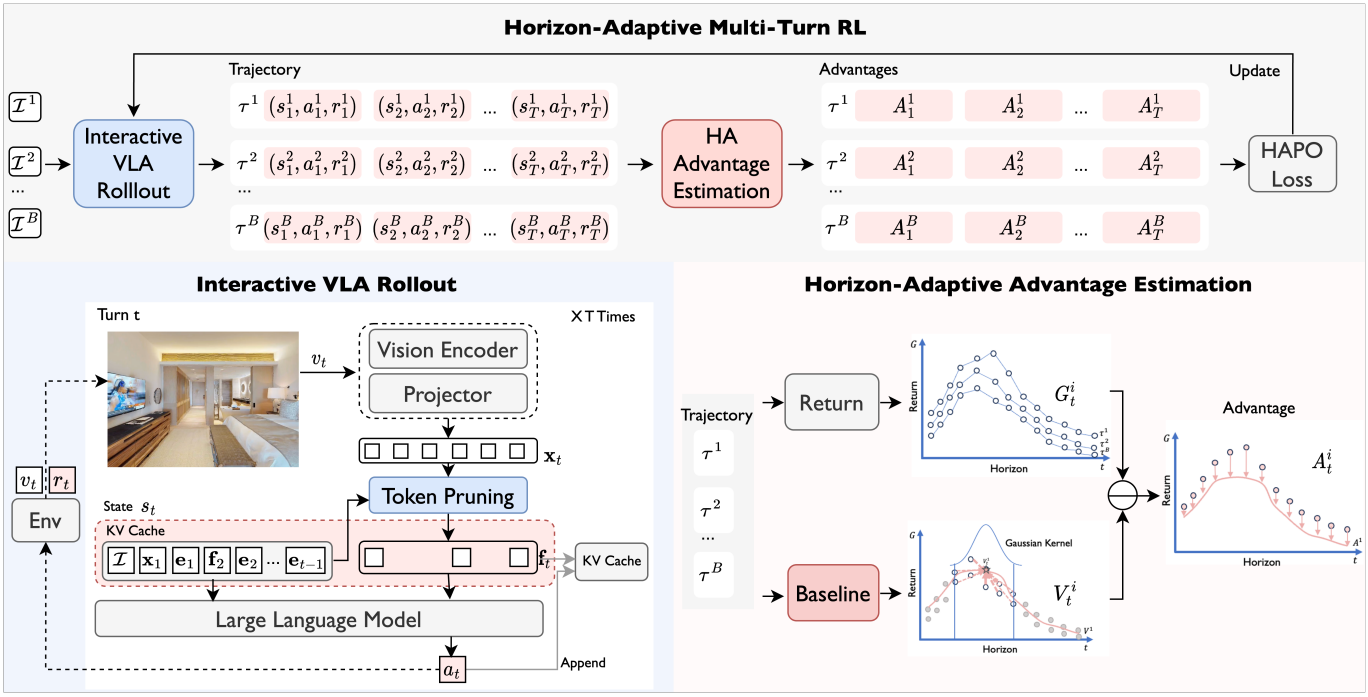


Fig. 3: The overall pipeline of LongNav-R1. The framework optimizes the VLA policy through a three-stage iterative process: i) collecting long-horizon trajectories via multi-turn interactive rollout; ii) computing action-level advantages using the proposed horizon-adaptive estimator; and iii) updating the VLA model via the aggregated optimization objective.

General kernel-based advantage estimation. Following the theoretical framework established in [66], we model critic-free baselines as non-parametric value function approximations. Unlike parametric critics that approximate values over the global state space, non-parametric critic-free methods perform local regression based strictly on the current rollout buffer. The key idea is to formalize value estimation as a kernel regression problem over the rollout buffer, which induces a generalized framework for advantage estimation, capable of adapting to diverse task objectives through specialized kernel designs.

Specifically, let $\mathcal{B} = \{\tau^i\}_{i=1}^{B}$ denote a buffer of complete rollout trajectories $\tau^i = \{(s_t^i, a_t^i, r_t^i)\}_{t=1}^{T}$. For each action a_t^i at timestep t in trajectory τ^i, the advantage A_t^i is defined as

$$A_t^i = G_t^i - V_{\mathcal{B}}^K(s_t^i), \quad G_t^i = \sum_{t'=t}^{|\tau^i|} \gamma^{t'-t} r_t \tag{3}$$

where G_t^i is the return with discount factor γ, and $V_{\mathcal{B}}^K(\cdot)$ denotes a kernel-based baseline. Specifically, the baseline is computed via leave-one-out kernel regression:

$$V_{\mathcal{B}}^K(s_t^i) = \frac{\sum_{j\neq i}^{B}\sum_{t'=1}^{T} K\big(f(s_t^i), f(s_{t'}^j)\big) G_{t'}^j}{\sum_{j\neq i}^{B}\sum_{t'=1}^{T} K\big(f(s_t^i), f(s_{t'}^j)\big)}. \tag{4}$$

where $f(\cdot)$ is a feature mapping and $K(\cdot,\cdot)$ is a similarity kernel defined in the feature space. The leave-one-out construction $j \neq i$ excludes samples from the current trajectory to avoid self-bias and ensure unbiased estimation. This formulation subsumes a broad class of critic-free advantage estimators through appropriate choices of the kernel K and feature representation f, enabling task-dependent value estimation without learning a parametric critic. For instance, in scenarios governed by temporal dynamics, a time-aware kernel can be employed to capture sequential dependencies; conversely, in environments where geometric configuration is paramount, a spatial-aware kernel is more appropriate for modeling structural regularities. Explicitly integrating these state features yields significantly more accurate value estimates.

Temporal-aware kernel design. To explicitly account for the sequential structure of navigation, HAPO introduces a time-aware kernel. We account for the temporal dynamics by explicitly defining the discrete time step as the feature, $f(s_i^t) = t$, and employ a Gaussian kernel $K(x, x') = \exp\left(-\frac{|x-x'|^2}{2\sigma^2}\right)$ with bandwidth σ, to provide temporal smoothing for the value estimates. By applying this formulation, HAPO regresses a baseline that is strictly specific to the current horizon of the episode. This enables the advantage estimator to adapt to the shifting reward distributions inherent in navigation, effectively distinguishing between the high-variance exploration phase (early horizon) and the high-precision target acquisition phase (late horizon).

Comparison with REINFORCE++. HAPO provides a general advantage estimation framework while existing methods such as REINFORCE++ emerges as special cases. Specifically, REINFORCE++ corresponds to the use of a constant kernel $K(\cdot,\cdot) = 1$ with features defined only at the final outcome state. Under this setting, the value estimate reduces to the mean final reward of the rollout buffer: $V_{\mathcal{B}}^K(s_T^i) = \frac{1}{B}\sum_{j=1}^{B} G_T^j$. This implies that the entire batch shares a uniform baseline,

effectively assigning the same value to all actions across all time steps. It introduces significant bias by disregarding the varying step-specific contributions to the final return, a critical oversight in long-horizon navigation tasks.

IV. LongNav-R1: Multi-Turn Navigation Framework

This section provides a comprehensive overview of our proposed pipeline, illustrated in Fig. 3. We begin by detailing the interactive VLA rollout within embodied environments in section IV-A, including state encoding, token pruning, and action prediction. Subsequently, we describe our horizon-adaptive advantage estimation in section IV-B, which processes rollout trajectories to provide dense advantages for long-horizon navigation tasks. Finally, we describe our training strategy in section IV-C, which enables scalable and efficient VLA optimization through SFT warm-up phase followed by multi-turn RL training.

A. Interactive VLA rollout

State encoding. State encoding transforms the multi-modal trajectory history into a sequence of tokens to provide global context for action prediction. Specifically, at each time step t, we process the observation-action pair in two stages: i) visual and action encoding. The current visual observation v_t is processed by a vision encoder $\mathbf{E}$ to extract latent features (visual tokens) $\mathbf{x}_t$. Simultaneously, the previous action a_{t-1} is embedded into a textual token $\mathbf{e}_t$ via the tokenizer $\mathbf{D}(\cdot)$:

$$\mathbf{x}_t = \mathbf{E}(v_t), \quad \mathbf{e}_{t-1} = \mathbf{D}(a_{t-1}). \tag{5}$$

Online token pruning. We filter redundant visual tokens before updating the state. Since navigation tasks are inherently long-horizon, often involving hundreds of steps, a single trajectory can generate hundreds of thousands of visual tokens. This massive volume creates significant computational overhead. To enhance representation efficiency, we introduce online token pruning at each rollout step. The key idea is to retain only informative visual tokens, defined as those exhibiting low feature similarity to the historical context.

Let $\{\mathbf{k}_{1:t-1}, \mathbf{v}_{1:t-1}\}$ be the cached keys and values from the historical trajectory. We compute a binary selection mask $\mathbf{m}_t \in \{0,1\}^M$, where an element is set to 1 only if the token's maximum similarity to the history falls below a predefined threshold δ (indicating novel information). The retained sparse visual tokens $\mathbf{f}_t$ are computed as:

$$\mathbf{f}_t = \mathbf{m}_t \odot \mathbf{x}_t, \quad \text{where } \mathbf{m}_t = \mathbb{I}(\max(\mathbf{x}_t \cdot \mathbf{k}_{1:t-1}^\top) < \delta). \tag{6}$$

Here, M denotes the number of visual tokens in the current observation, and $\mathbb{I}(\cdot)$ is the indicator function. Only the non-zero (informative) tokens are preserved. Consequently, the sparse state s_t at time step t is updated to include the history cache, the new sparse visual tokens, and the action token:

$$s_t = \{\mathbf{k}_{1:t-1}, \mathbf{v}_{1:t-1}, \mathbf{e}_{t-1}, \mathbf{f}_t\}. \tag{7}$$

Note that at the initial step, with the KV cache and action history empty, the model's state is derived from the task instruction $\mathcal{I}$ and the first observation. Subsequently, the state is maintained via the KV cache, which encodes the full history of actions and observations.

Action prediction. Given the encoded state representation, the agent predicts the optimal next action to advance toward the goal. By encapsulating the full history of observations and actions, the state allows the agent to reason about the causal effects of past decisions and maintain global context, crucial for preventing redundant exploration and enabling targeted search. In the VLA framework, action generation is cast as next-token prediction. At each timestep t, the policy outputs a categorical distribution over the vocabulary, from which the action token is sampled:

$$a_t \sim \pi_\theta(\cdot|s_t), \quad \text{where } \pi_\theta(\cdot|s_t) \in \Delta^{|D|-1}. \tag{8}$$

Here, π_θ denotes the probability distribution derived from the VLA's softmax-normalized logits, and $|D|$ represents the vocabulary size. While this work focuses on a discrete action space where each command corresponds to a single token, the framework is readily extensible to continuous control via action quantization or tokenization strategies.

Upon predicting action a_t, the agent executes it within the embodied environment, prompting a state transition, a reward r_t and a new observation v_{t+1}. This sequential rollout iterates until a termination condition, either a stop action or the maximum horizon T is met, yielding the complete trajectory $\tau = \{(s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_T, a_T, r_T)\}$. This loop allows the agent to continuously update its state and refine its decision-making to satisfy the navigation instruction.

B. Horizon-adaptive advantage estimation

To optimize VLA policies within long-horizon, multi-turn settings, we implement HAPO solution proposed in sec. III-C for navigation task. The key idea of HAPO is the regression of a temporal-aware baseline via kernel regression over the rollout buffer. This enables accurate advantage estimation that accounts for temporal dynamics. Given full trajectories in the rollout buffer, we compute advantages for every timestep to provide dense process feedback for the VLA policy learning with three main steps.

Return estimation. Given a full trajectory $\tau^i \in \mathcal{B}$, we compute the empirical returns at t-th step $G_t^i = \sum_{t'=t}^{|\tau^i|} \gamma^{t'-t} r_t$ using a predefined discount factor γ. For the navigation task, we define the step-level reward r_t based on the geodesic distance to the goal, providing a dense proxy for incremental progress. We calibrate the discount factor γ to modulate the effective temporal horizon based on environmental semantic density. In indoor settings, high semantic density induces complex, high-frequency temporal dynamics; we therefore restrict the horizon to mitigate variance and maintain a high signal-to-noise ratio. Conversely, sparse outdoor environments exhibit low-frequency temporal dynamics, allowing a larger γ to capture long-range dependencies without signal degradation.

Kernel-based baseline. We regress a non-parametric baseline by applying a temporal kernel function across the rollout buffer. To mitigate the high variance inherent in long-horizon tasks,

we utilize a critic-free baseline based on Gaussian smoothing. By grouping returns in proximal timestamps, HAPO constructs a stable baseline at each horizon scale. The baseline V_t^i for state s_t^i is calculated as

$$V_t^i = \frac{\sum_{j\neq i}^{B}\sum_{t'=1}^{|\tau_j|}\exp\left(-\frac{|t-t'|^2}{2\sigma^2}\right)G_{t'}^j}{\sum_{j\neq i}^{B}\sum_{t'=1}^{|\tau_j|}\exp\left(-\frac{|t-t'|^2}{2\sigma^2}\right)} \tag{9}$$

where $\mathcal{B}$ represents the rollout buffer and σ is the bandwidth parameter controlling the temporal smoothing window. The kernel bandwidth is calibrated to the environment's temporal dynamics. Note that, while standard long-horizon tasks suffer from high variance due to noisy trajectories, our non-parametric kernel regression effectively reduces this variance with temporal smoothing window.

Dense advantage estimation. The advantage at each timestep is computed as the residual between the return and the temporal baseline: $A_t^i = G_t^i - V_t^i$. By calculating dense advantage signals at each timestamp, HAPO provides the VLA policy with informative process signals. This allows the model to learn complex navigation dynamics and maintain a precise correspondence between visual-linguistic inputs and sequential actions. This approach significantly enhances sampling efficiency and training stability compared to standard outcome-reward critic-free methods that lack temporal awareness.

C. Training strategy

We utilize an off-the-shelf VLM [58] as our backbone. Since generic VLMs lack embodied navigation priors, we first employ a warm-start phase using imitation learning (IL) on human demonstrations [37]. Subsequently, we transition the agent to an Online RL phase, optimizing the policy via our HAPO.

Balanced IL Warm-up. In this phase, we utilize teacher forcing to bootstrap the VLA policy with fundamental navigation behaviors. We process the full trajectory sequence in a single forward pass, computing gradients exclusively on the generated action tokens. The objective is to minimize the standard negative log-likelihood:

$$\mathcal{L}_{\text{IL}}(\theta) = -\sum_{i=1}^{B}\sum_{t=1}^{T}\log\pi_\theta(a_t^i \mid \mathcal{I}^i), \tag{10}$$

where $\mathcal{I}^i$ represents the i-th instruction in the each batch with size B. This supervised pre-training instills essential search strategies, such as panoramic scanning and collision avoidance. However, naive IL induces a strong length bias, particularly affecting the 'stop' action; agents tend to terminate episodes near the average length of training trajectories regardless of goal proximity.

Online multi-turn RL training. Following the warm-up, the agent enters the multi-turn RL stage. Here, the objective is to maximize the expected cumulative reward over the full horizon:

$$\mathcal{J}_{\text{HAPO}}(\theta) = \mathbb{E}_{\{\tau^i\}_{i=1}^{B}}\left[\frac{1}{B}\sum_{i=1}^{B}\frac{1}{|\tau^i|}\sum_{t=1}^{|\tau^i|}\left(\min\left(\rho_t^i A_t^{\text{norm},i}, \text{clip}(\rho_t^i, 1-\epsilon_l, 1+\epsilon_h)A_t^{\text{norm},i}\right) - \beta D_{\text{KL}}\left(\pi_\theta(a_t^i|\mathcal{I}^i) \parallel \pi_{\text{ref}}(a_t^i|\mathcal{I}^i)\right)\right)\right]. \tag{11}$$

where $A_t^{\text{norm},i}$ denotes the normalized advantage, computed via global batch normalization to enhance training stability. Following REINFORCE++ [18], the transformation is defined as: $A_t^{\text{norm},i} = \frac{A_t^i - \text{mean}(A|A\in\mathcal{B})}{\text{std}(A|A\in\mathcal{B})} + \epsilon$, where ϵ is a small constant for numerical stability. And $\rho_t^i = \pi_\theta(a_t^i|\mathcal{I}^i)/\pi_{\text{ref}}(a_t^i|\mathcal{I}^i)$ denotes the importance sampling ratio. The hyperparameters ϵ_l, ϵ_h and β serve to constrain the gradient range and policy update magnitude, respectively, ensuring stable optimization by preventing large architectural shifts during learning. The reference model π_{ref} and KL loss D_{KL} are used for regularization to make the training more stable. We also use the entropy masking trick in [12]. Note that, comparing to REINFORCE++, our advantage estimator moves beyond simple batch averages to a time-aware baseline, enabling more precise credit assignment. This allows the policy to better manage stochastic dynamics and rectify errors accumulated during the imitation phase, ensuring robustness over long execution horizons.

V. Experiments

A. Experimental setup

We conduct experiments to evaluate LongNav-R1 on three specific aspects: i) How does LongNav-R1 perform compared to existing state-of-the-arts (SOTA)? ii) Does LongNav-R1 improve long-horizon decision-making capacity for VLA? iii) Is the key design of our method effective?

Benchmarks. To evaluate our general-purpose navigation method, we conduct extensive experiments on object-goal and open-vocabulary tasks across four benchmarks: HM3D V1 [36], V2 [57], MP3D [8], and HM3D-OVON [63]. Unlike previous baselines that often prioritize either efficiency or open-vocabulary capabilities on limited subsets, we provide a comprehensive comparison across diverse baselines.

Evaluation metrics. We report two metrics including *success rate (SR)* and *success rate weighted by path length (SPL)* [3]. SR is the core metric of object-goal navigation task, representing the success rate of navigation episodes. SPL measures the ability of the agent to find the optimal path. If success, SPL $= \frac{\text{optimal path length}}{\text{path length}}$, otherwise SPL $= 0$. Higher is better for both metrics.

Implementation details. For real-world, an Orbbec Femto Bolt sensor is mounted approximately 1m off the ground. To mitigate the sim-to-real gap, the real-world RGB camera is calibrated to approximate the intrinsics of the simulated camera used during training. For simulation, we follow the standard settings [36]. The camera of the agent is $0.88m$ above the ground. The camera outputs 640×480 RGB images. The success distance threshold is set as 1m, and the step budget is 500. The discrete action set includes MOVE FORWARD (0.25m), TURN LEFT/RIGHT (30°), and STOP. Ablations are conducted on randomly sampled 200 episodes. For the VLA policy training, we start with a Qwen-3-VL-2B [58] as base.

Computation resource. Both SFT and RL stages were trained on 4 A100 GPUs (72 GPUh per stage). For consistency, efficiency metrics were benchmarked on a single A100. In real-world experiments, an A100 server handles inference, with ROS managing observation and action communication.

TABLE I: **Comparison on object goal navigation.** LongNav-R1 outperforms previous SOTAs across different metrics on object goal navigation benchmark HM3D V1 [36], V2 [57] and MP3D [8].

Method	Observations			HM3D V1		HM3D V2		MP3D	
	Odom.	Depth	RGB	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
CoWs [15]	✓	✓	✓	-	-	-	-		
DD-PPO [50]	✓	✓	✓	27.9	14.2	-	-	-	-
ESC [73]	✓	✓	✓	39.2	22.3	-	-	28.7	14.2
Habitat-Web [37]	✓	✓	✓	41.5	16.0	-	-	31.6	8.5
VoroNav [52]	✓	✓	✓	42.0	26.0	-	-	-	-
L3MVN [64]	✓	✓	✓	50.4	23.1	-	-	34.9	14.5
OpenFMNav [25]	✓	✓	✓	52.5	24.1	-	-	37.2	15.7
VLFM [62]	✓	✓	✓	52.5	30.4	-	-	36.4	17.5
GAMap [20]	✓	✓	✓	53.1	26.0	-	-	-	-
SG-Nav [60]	✓	✓	✓	54.0	24.9	-	-	40.2	16.0
UniGoal [61]	✓	✓	✓	54.5	25.1	-	-	41.0	16.4
InstructNav [29]	✓	✓	✓	58.0	20.9	-	-	-	-
SGM [70]	✓	✓	✓	-	-	60.2	30.8	37.7	14.7
BeliefMapNav [74]	✓	✓	✓	61.4	30.6	-	-	37.3	17.6
SGImagineNav [19]	✓	✓	✓	65.4	30.0	-	-	-	-
ImagineNav [71]	✓		✓	53.0	23.8	-	-	-	-
ProcTHOR [13]	✓		✓	54.4	31.8	-	-	-	-
OVRL [56]	✓		✓	62.0	26.8	-	-	28.6	7.4
OVRL-v2 [55]	✓		✓	64.7	28.1	-	-	-	-
PIRLNav-IL [38]	✓		✓	64.1	27.1	52.0	20.6	-	-
PIRLNav-RL [38]	✓		✓	70.4	34.1	61.9	27.9	-	-
ZSON [30]			✓	25.5	12.6	-	-	15.3	4.8
PixNav [7]			✓	37.9	20.5	-	-	-	-
PSL [42]			✓	42.4	19.2	-	-	18.9	6.4
UniNavid [69]			✓	73.7	37.1	-	-	-	-
LongNav-R1			✓	**76.0**	**44.3**	**83.7**	**43.4**	**63.0**	**30.2**

TABLE II: **Comparison on open-vocabulary object goal navigation.** Zero-shot LongNav-R1 outperforms previous SOTAs across different metrics on OVON [63]. ZS denotes zero-shot. **LongNav-R1** is trained only on HM3D without fine-tuning on the OVON. Val-Seen-Syn is the Val-Seen-Synonyms split.

Method	ZS	Val-Seen		Val-Seen-Syn		Val-Unseen	
		SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
BC	✗	11.1	4.5	9.9	3.8	5.4	1.9
DAgger [39]	✗	11.1	4.5	9.9	3.8	5.4	1.9
RL [40]	✗	18.1	9.4	15.0	7.4	10.2	4.7
BCRL [45]	✗	39.2	18.7	27.8	11.7	18.6	7.5
DAgRL [11]	✗	41.3	21.2	29.4	14.4	18.3	7.9
VLFM [62]	✓	35.2	18.6	32.4	17.3	35.2	19.6
DAgRL+OD [63]	✗	38.5	21.1	39.0	21.4	37.1	19.8
Uni-NaVid [69]	✗	41.3	21.1	43.9	21.8	39.5	19.8
MTU3D [75]	✗	55.0	23.6	45.0	14.7	40.8	12.1
Nav-R1 [27]	✗	58.4	26.3	48.1	23.1	42.2	20.1
LongNav-R1	✓	**59.2**	**32.1**	**58.7**	**28.3**	**53.8**	**22.8**

B. Comparison with SOTAs

LongNav-R1 outperforms previous SOTAs on object-goal navigation benchmarks: HM3D V1, V2, and MP3D. Tab. I compares LongNav-R1 against state-of-the-art object navigation methods using various observation modalities, including odometry, depth, and RGB. LongNav-R1 consistently outperforms existing baselines across all datasets, improving success rates to **76.0**%, **83.7**%, and **63.0**% on HM3D V1, V2, and MP3D, respectively. We also observe that: i) LongNav-R1 (using RGB-only) surpasses methods that rely on full odometry and RGBD sensor setups. This demonstrates the effectiveness of our approach and suggests strong generalizability to real-world scenarios, where odometry and depth sensors are often noisy or range-limited. We attribute this robustness to the informative navigation cues provided by large model priors and the spatial understanding enriched by multi-turn causal reasoning; and ii) LongNav-R1 outperforms the prior end-to-end VLA model UniNavid [69] on path efficiency by **7.2**%. This superior efficiency stems from our RL training, which enables the agent to learn diverse behaviors to solve the navigation objective. In contrast, UniNavid uses SFT imitation learning, which is limited by the strict mimicry of human demonstrations and struggles to adapt efficiently to unseen testing environments.

TABLE III: **Compare to existing large-scale learning strategies for VLAs.** Results are reported on OVON Val-Unseen.

Method	Turn	Learning	GPUh	SR↑	SPL↑
UniNavid [69]	Single-turn	Imitation learning	1400	39.5	19.8
Nav-R1 [27]	Single-turn	Critic-free RL	-	42.2	20.1
StreamVLN [48]	Multi-turn	Imitation learning	1500	11.4	9.4
SFT [34]	Multi-turn	Imitation learning	72	45.5	19.9
LongNav-R1	Multi-turn	Critic-free RL	144	53.8	22.8

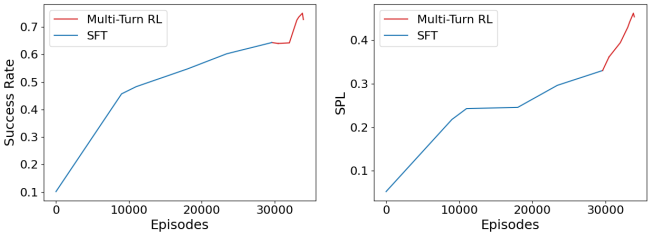


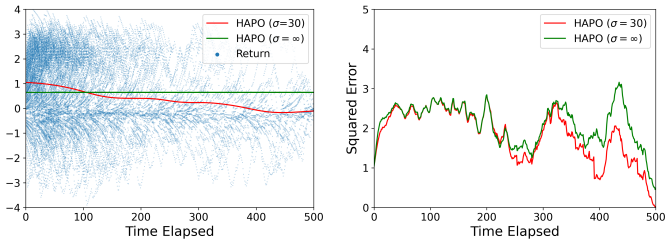
Fig. 4: RL demonstrates high efficiency, improving performance by 30% with 4k iterations. Evolution of success rate and path efficiency during training. The training pipeline consists of two phases: an initial SFT phase followed by a RL phase.

LongNav-R1 outperforms previous SOTAs on the open-vocabulary object-goal navigation benchmark: HM3D-OVON. Tab. II compares LongNav-R1 against several state-of-the-art open-vocabulary object navigation methods. To validate the generalization ability of LongNav-R1, we trained exclusively on HM3D and evaluated on all other benchmarks via direct inference. LongNav-R1 demonstrates strong generalization: without any fine-tuning, it achieves SOTA. We also observe that: i) LongNav-R1 significantly surpasses traditional transformer-based RL methods by about 30%, such as RL [40], BCRL [45], and DAgRL [11]. This highlights the potential of using pretrained large models as policies for robotic tasks; ii) LongNav-R1 outperforms the single-turn RL baseline, Nav-R1 [27]. We attribute this to our multi-turn RL approach, which better captures sequential dependencies and long-term navigation objectives.

Evaluation against other large-scale learning strategies for VLAs. We benchmarked against SOTA VLAs (UniNavid, Nav-R1) and RL baselines (REINFORCE++). LongNav-R1 excels across turn types and learning setups (critic-free RL and imitation learning). i) It surpasses multi-turn (StreamVLN, SFT) by overcoming behavior cloning's distribution shift and fostering better generalization, while StreamVLN fails to

TABLE IV: **Effectiveness of multi-turn reinforcement learning.** Performance across different horizon length (steps). The physical distances are 12.5m (50 steps) and 50m (200 steps)

Training Strategy	Overall		< 50		50 - 200		> 200	
	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
No Training [58]	0.51	0.50	1.9	1.8	0.0	0.0	0.0	0.0
SFT [34]	64.3	33.0	74.1	38.9	64.7	32.8	0.0	0.0
REINFORCE++ [18]	47.4	26.0	68.0	37.0	42.5	23.5	0.0	0.0
HAPO ($\sigma = \infty$)	71.6	40.4	82.7	47.3	72.2	40.5	0.0	0.0
HAPO ($\sigma = 30$)	73.0	44.3	86.2	53.2	69.9	41.7	15.4	9.4



(a) Baseline: Kernel t (b) Val Err: Kernel t

Fig. 5: We compare two HAPO variants, using different kernel sizes σ. Subplot (a) displays the returns in a batch buffer alongside the regressed baselines, while (b) illustrates the value estimation errors.

Fig. 6: Ablation of σ and γ.

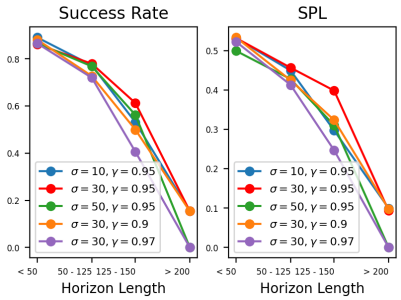


TABLE V: Feature.

Feature	SR↑	SPL↑
Timestep	76.4	45.7
Per-Traj	71.0	40.5
Distance	73.4	45.3

TABLE VI: Horizon.

σ	<200		>200	
	VE ↓	SR↑	VE ↓	SR↑
30	230	75.1	455	15.4
∞	230	75.0	530	0.0

generalize. ii) Lacking existing critic-based RL for navigation VLAs, we reimplement PPO, however it collapses from value-head optimization complexities. LongNav-R1's critic-free estimator removes this need, achieving superior performance and computational efficiency.

Training statistics analysis. Fig. 4 shows the evolution of success rate and path efficiency throughout the training process, which consists of an initial SFT phase followed by RL. We see that: i) during the warm-up stage, SFT facilitates rapid early learning, elevating performance from 0 to 64.3%; however, it encounters a bottleneck at approximately 30k iterations, where further data scaling yields diminishing returns; and ii) in contrast, the RL phase successfully breaks this plateau, demonstrating superior efficiency by improving performance by 8.7% within only 4k iterations. This is attributed to the multi-turn design, which enables the model to learn from hundreds of decision-making points per sample, and the use of on-policy rollouts, which allow for effective error correction and behavioral refinement.

C. Ablation studies

Effectiveness of multi-turn reinforcement learning. Tab. IV compares two HAPO variants against various training strategies, including zero-shot (no training), SFT [34], and REINFORCE++ [18]. Specifically, REINFORCE++ uses sparse binary outcome rewards (1/0 for success/failure), while HAPO incorporates step-wise distance-to-goal rewards. We see that: i) both HAPO variants outperform all other training strategies; and ii) RL with dense rewards outperforms SFT, whereas RL with sparse rewards does not. This demonstrates the importance of dense reward shaping in multi-horizon robotics tasks.

Effectiveness of horizon-adaptive advantage estimation. Fig. 5 and Tab. IV compares HAPO variants, using different bandwidth sizes σ. We visualize the estimated baseline and the value estimation error of a randomly sampled batch buffer in subplots (a) and (b). The value estimation error is the average difference between the real return and the regressed baseline at each step. HAPO with restricted temporal window outperforms infinite-window (uniform) baseline. We see that: i) HAPO with $\sigma = 30$ can achieve smaller estimation error compared to $\sigma = \infty$; ii) this reduced estimation error directly correlates with improved performance, as evidenced in Tab. IV. This highlights the necessity of temporal-adaptive advantage estimation, as simple uniform dense estimation fails to account for the complex temporal dynamics inherent in robotic navigation.

Ablation on kernel bandwidth σ and discount factor γ, see Fig. 6. We see that an intermediate temporal window ($\sigma = 30, \gamma = 0.95$) yields optimal performance. This balance is critical because increasing temporal window via higher γ prevents short-sighted policies by encouraging long-term predictions, yet setting it too high makes return values unpredictable. Increasing temporal window via higher σ prevents the baseline from overfitting to local returns, but an excessively large σ eliminates horizon awareness. We optimize this configuration via value-error analysis on an offline rollout dataset (Fig. 4 in main paper). This kernel regression is computationally efficient and strongly correlates with final model performance. Moreover, this technique can be extended to automated bandwidth tuning using an online rollout buffer.

Stronger justification for timestep feature design, see Tab. V. HAPO is a flexible framework capable of accommodating different factors via feature switching. Specifically, HAPO's timestep feature outperforms per-trajectory and distance metrics. The reasons are: i) HAPO reduces per-trajectory bias (similar to REINFORCE++ v.s GRPO), ensuring generalized navigation over instance-specific overfitting; ii) distance fails to credit critical in-place actions (turning to unlock regions). This is reflected in the data: elapsed timesteps maintain a higher correlation with returns (0.24) than distance-to-goal (0.09).

Clearer explanation for value-error analysis, see Tab. VI. i) Navigation success requires minimizing accumulated value error across all elapsed timesteps. ii) Thus, longer horizons (>200 steps) yield larger accumulated errors and lower performance than shorter ones (<200 steps). iii) For >200 steps, HAPO (30) sustains lower error and outperforms the ∞-baseline, while

TABLE VII: Computation analysis.

Turns	Inference Time (s)				Memory Cost (GB)			
	50	100	200	400	50	100	200	400
LongNav-R1	0.12	0.15	0.19	0.23	6.25	7.34	9.33	11.62

TABLE VIII: Ablation of pruning.

Pruning (δ)	Speed (s)	Mem (GB)	SR↑
0% (1.0)	0.203	15.6	53.1
20% (0.97)	0.172	13.7	56.5
35% (0.95)	0.168	11.8	54.4
45% (0.93)	0.141	10.8	52.4
50% (0.92)	0.132	10.4	45.6
63% (0.90)	0.118	8.9	25.8
80% (0.85)	0.089	7.3	0.0

both perform identically for <200 steps. iv) The peak variations (<200) in Tab. III (main paper) arise from fine-grained binning across unbalanced intervals (<50, 50-200).

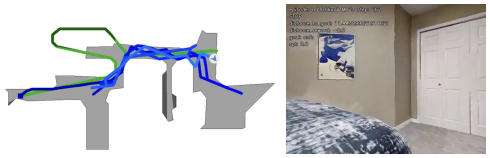
Computation analysis. Tab. VII compares the computational overhead of LongNav-R1 in terms of inference latency and memory consumption. We see that while inference time scales with the horizon, LongNav-R1 remains highly performant, requiring only 0.23 s to process a 400-step sequence. Achieving about 5 FPS, the model can be deployed in practical real-world environments. This is attributed to two architectural advantages: the multi-turn setup, which enables KV cache reuse to avoid redundant processing of past observations, and online token pruning, which restricts attention mechanisms to only the most informative new tokens.

Effectiveness of online pruning. Tab. VIII shows that increasing the pruning ratio consistently improves speed and reduces memory usage, navigation accuracy follows an inverted-U trend. Performance suffers at both extremes: unpruned models quickly exceed memory limits in long tasks (peak memory), while over-pruning discards essential data. ii) We demonstrate superior efficiency compared to baselines (StreamVLN requires 24GB memory and 0.37s). Moreover, performance drops after 150 steps (37.5m) due to the training focus on nearby targets and context-heavy degradation at 60k tokens, see Fig.6.

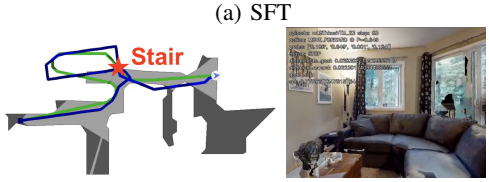
D. Case study

Qualitative analysis of SFT and LongNav-R1. Fig. 7 visualizes the navigation process of SFT and LongNav-R1 as they search for a sofa. We see that: i) despite the challenge of a distant, cross-floor goal, LongNav-R1 successfully and efficiently reaches the target while SFT fails; and ii) LongNav-R1 natively handles cross-floor transitions, a task where modular designs often struggle, enhancing its adaptability in real-world environments.

Visualization of LongNav-R1 in real-world deployment. Fig. 8 visualizes the navigation process of LongNav-R1 as it searches for an elevator, an unseen object category, in a real-world office environment. For the hardware mobile base, we adopt the mobile base design from TidyBot++ [51], omitting the manipulator. Equipped with LongNav-R1, the robot successfully locates the target, demonstrating zero-shot real-world effectiveness of the navigation policy.



(a) SFT



(b) LongNav-R1

Fig. 7: Visualization of the navigation process in habitat. Blue and green lines represent the optimal and inferred trajectories.

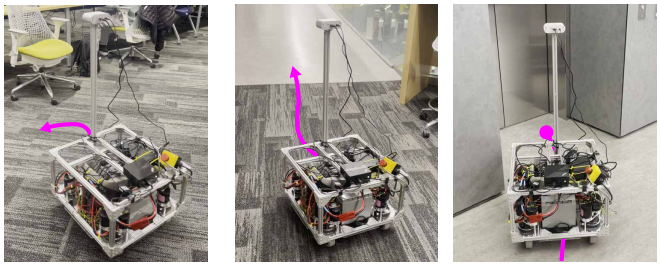


Fig. 8: Visualization of the navigation process in **real-world**. The robot successfully navigates to an elevator, an unseen category. Displayed trajectories are approximate for visualization.

VI. Discussion

Generalize to tasks lacking geometric proxies. i) We want to clarify that HAPO is not constrained to dense geometric proxies. As a versatile, critic-free framework for long-horizon tasks, it accommodates diverse factors via feature switching and natively supports sparse reward (producing the REINFORCE++ in Tab. IV); ii) HAPO's strong sim-to-real generalization, evidenced by its superior zero-shot performance in real-world demo, makes it highly feasible to get these priors in simulation before deploying in real world.

VII. Conclusion

Conclusion. In this paper, we provide a recipe for training end-to-end VLA policies for long-horizon navigation using multi-turn reinforcement learning and horizon-adaptive policy optimization. By utilizing a critic-free advantage estimation, HAPO effectively manages dense rewards and sequential decision-making without the prohibitive computational overhead of an auxiliary critic model, a key advantage when scaling large models. Our model, LongNav-R1, achieves state-of-the-art results across four simulation benchmarks and demonstrates robustness in real-world deployment.

Limitation and future works. A current limitation is that our online token pruning retains the full KV cache to maintain causal consistency. To mitigate memory bottlenecks in extreme-horizon tasks, future work will explore selective KV cache eviction and spatially aware global memory mechanisms. Beyond navigation, we aim to extend this framework to diverse applications, such as mobile manipulation, and incorporate world models for real-world multi-turn reinforcement learning.

References

[1] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, 2024.

[2] 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.

[3] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. doi: 10.48550/arXiv.1807.06757.

[4] 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*, 2018.

[5] Anthropic. Claude ai. https://claude.ai, 2023. Accessed: 2025-05-05.

[6] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[7] Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5228–5234. IEEE, 2024.

[8] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *2017 International Conference on 3D Vision (3DV)*, pages 667–676, 2017.

[9] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. *Advances in Neural Information Processing Systems*, 33:4283–4294, 2020.

[10] 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.

[11] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019.

[12] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

[13] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

[14] A. Dubey et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[15] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23171–23181, 2023.

[16] Nandiraju Gireesh, DA Sasi Kiran, Snehasis Banerjee, Mohan Sridharan, Brojeshwar Bhowmick, and Madhava Krishna. Object goal navigation using data regularized q-learning. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 1092–1097. IEEE, 2022.

[17] 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.

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

[19] Yue Hu, Junzhe Wu, Ruihan Xu, Hang Liu, Avery Xi, Henry X Liu, Ram Vasudevan, and Maani Ghaffari. Imaginative world modeling with scene graphs for embodied agent navigation. *arXiv preprint arXiv:2508.06990*, 2025.

[20] Hao Huang, Yu Hao, Congcong Wen, Anthony Tzes, Yi Fang, et al. Gamap: Zero-shot object goal navigation with multi-scale geometric-affordance guidance. *Advances in Neural Information Processing Systems*, 37:39386–39408, 2024.

[21] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14809–14818, 2021.

[22] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Schacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X. Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16384–16393, 2023.

[23] Jacob Krantz, Erik Wijmans, Arjun Majundar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision and language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, 2020.

[24] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[25] Yuxuan Kuang, Hai Lin, and Meng Jiang. OpenFMNav: Towards open-set zero-shot object navigation via vision-language foundation models. In *Findings of the Association for Computational Linguistics (NAACL)*, pages 338–351, 2024.

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

[27] Qingxiang Liu, Ting Huang, Zeyu Zhang, and Hao Tang. Nav-r1:

Reasoning and navigation in embodied scenes. *arXiv preprint arXiv:2509.10884*, 2025.

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

[29] Yuxing Long, Wenzhe Cai, Hongchen Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. In *Conference on Robot Learning*, 2024.

[30] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352, 2022.

[31] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. Thda: Treasure hunt data augmentation for semantic navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15374–15383, 2021.

[32] Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson. Visual representations for semantic target driven navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8846–8852. IEEE, 2019.

[33] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[35] Yanyuan Qiao, Wenqi Lyu, Hui Wang, Zixu Wang, Zerui Li, Yuanyuan Zhang, Mingkui Tan, and Qi Wu. Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms. *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6710–6717, 2024.

[36] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

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

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

[39] 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.

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

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

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

[43] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[44] Gemini Robotics Team. Gemini robotics: Bringing ai into the physical world. *ArXiv*, abs/2503.20020, 2025.

[45] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6629–6638, 2019.

[46] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

[47] Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

[48] Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. StreamVLN: Streaming vision-and-language navigation via SlowFast context modeling. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026.

[49] Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. WebAgent-R1: Training web agents via end-to-end multi-turn reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7909–7928, 2025.

[50] Erik Wijmans, Abhishek Kadian, Ari S. Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2019.

[51] Jimmy Wu, William Chong, Robert Holmberg, Aaditya Prasad, Yihuai Gao, Oussama Khatib, Shuran Song, Szymon Rusinkiewicz, and Jeannette Bohg. Tidybot++: An open-source holonomic mobile manipulator for robot learning. In *Conference on Robot Learning*, 2024.

[52] Pengying Wu, Yao Mu, Bin Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. Voronav: Voronoi-based zero-shot object navigation with large language model. In *International Conference on Machine Learning*, 2024.

[53] Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. AgentGym-RL: Training LLM agents for long-horizon decision making through multi-turn reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

[54] Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. SimpleTIR: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

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

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

[57] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4927–4936, 2023.

[58] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[59] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[60] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *Advances in Neural Information Processing Systems*, 37:5285–5307, 2024.

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

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

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

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

[65] Bangguo Yu, Yuzhen Liu, Lei Han, Hamidreza Kasaei, Tingguang Li, and Ming Cao. Vln-game: Vision-language equilibrium search for zero-shot semantic navigation. *IEEE Transactions on Robotics*, 42:1824–1839, 2024.

[66] Guanning Zeng, Zhaoyi Zhou, Daman Arora, and Andrea Zanette. Shrinking the variance: Shrinkage baselines for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2511.03710*, 2025.

[67] Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross Girshick, Aniruddha Kembhavi, and Luca Weihs. Poliformer: Scaling on-policy rl with transformers results in masterful navigators. In *Conference on Robot Learning*, 2024.

[68] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *Robotics: Science and Systems*, 2024.

[69] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *Robotics: Science and Systems*, 2025.

[70] Sixian Zhang, Xinyao Yu, Xinhang Song, Xiaohan Wang, and Shuqiang Jiang. Imagine before go: Self-supervised generative map for object goal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16414–16425, 2024.

[71] Xinxin Zhao, Wenzhe Cai, Likun Tang, and Teng Wang. ImagineNav: Prompting vision-language models as embodied navigator through scene imagination. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[72] 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.

[73] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning*, pages 42829–42842. PMLR, 2023.

[74] Zibo Zhou, Yue Hu, Lingkai Zhang, Zonglin Li, and Siheng Chen. BeliefMapNav: 3D voxel-based belief map for zero-shot object navigation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

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