ANCHORNAV: LONG-HORIZON 3D GAME NAVIGATION WITH PERSISTENT MEMORY AND WAYPOINT-CONDITIONED CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in multimodal agents have steadily improved autonomous gameplaying, yet reliable long-range navigation in photorealistic 3D games remains challenging due to open terrain, extended horizons, and the absence of privileged localization. Here we present ANCHORNAV, a *Vision-Language-Navigation* framework built around two complementary anchors: a local visual anchor for grounding route-level decisions into short-horizon actions, and a persistent global anchor for maintaining spatial consistency over time. The local anchor projects the next navigation target into the current view as a 2D waypoint, which guides a waypoint-conditioned visuomotor controller together with the language instruction. The global anchor is provided by a persistent topological memory, where visual place recognition periodically re-anchors visual odometry to previously established locations. In photorealistic 3D games, ANCHORNAV effectively completes long-horizon navigation and remains competitive on RxR under the VLN-CE protocol.

1 INTRODUCTION

General-purpose multimodal agents can increasingly follow natural-language instructions, interpret rendered observations, and act through native control interfaces in complex 3D games (Tan et al., 2025; SIMA Team, 2025). Reliable navigation is a fundamental capability underlying many such tasks.To explore unfamiliar areas, interact with distant objects, or accomplish long-horizon objectives, an agent must often traverse environments far beyond its immediate field of view. In 3D games, this long-range navigation is especially challenging without privileged maps, forcing the agent to rely solely on visual observations. This challenge is compounded by the limited spatial reasoning and memory of current VLMs over long navigation horizons (Qiao et al., 2025; Xu et al., 2025; Zhu et al., 2026). As a result, agents may lose track of previously visited locations, repeatedly backtrack, and fail to maintain consistent progress toward distant goals (Zhang et al., 2026), as illustrated in Fig. 1(a).

These failure modes can be traced to two complementary challenges in long-horizon navigation. *Globally*, the agent must maintain a persistent spatial reference that preserves where it has been and where it is despite limited visual history and localization drift. *Locally*, it must ground the next route-level target into an unambiguous direction of motion in the current view. Existing game agents improve long-horizon behavior through planning, skill libraries, and multimodal memory (Wang et al., 2024; 2023; Tan et al., 2025; SIMA Team, 2025), while video-based VLN methods extend temporal context for navigation (Zhang et al., 2024b;a; Wei et al., 2025). However, temporal context alone does not provide a persistent global spatial reference, and route-level instructions may remain geometrically ambiguous in the current view. Reliable long-horizon navigation therefore requires both global spatial anchoring and local visual grounding.

To address these challenges, we introduce ANCHORNAV, a vision-language-navigation system for long-horizon navigation in 3D games based on *dual-scale anchoring*. As shown in Fig. 1(b), ANCHORNAV establishes complementary spatial references at global and local scales. At the global scale, it builds a persistent topological memory from streaming or offline gameplay videos and periodically re-anchors online motion estimates through visual place recognition and visual odometry,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

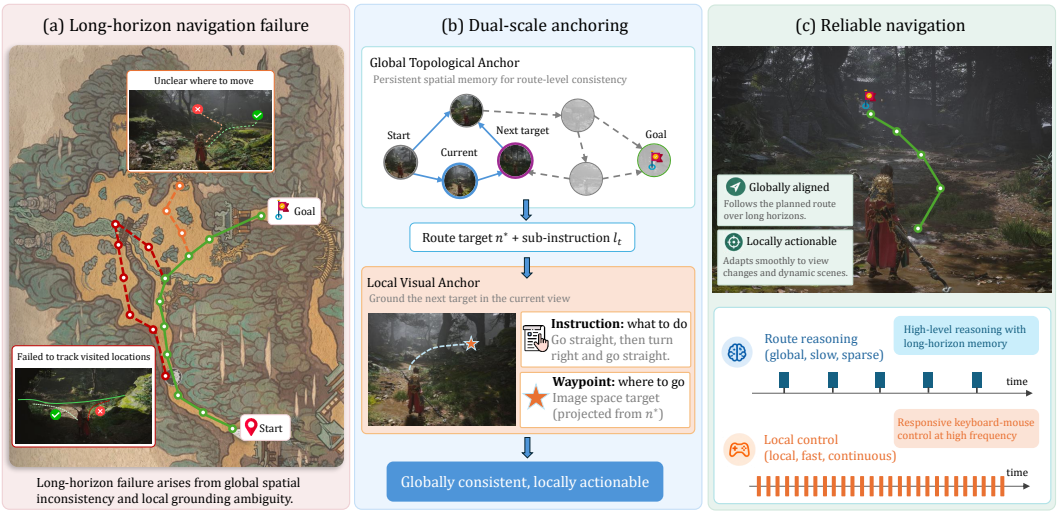


Figure 1: **Dual-scale anchoring for long-horizon game navigation.** (a) Long-horizon failure arises from global spatial inconsistency and local grounding ambiguity; the map is shown only for illustration and is not available to the agent. (b) ANCHORNAV combines a persistent global topological anchor with a local image-space waypoint. (c) Together, they enable globally aligned navigation and responsive local control across two timescales.

preserving route consistency over extended trajectories. At the local scale, it projects the next topological target into the current view as a 2D waypoint paired with a natural-language sub-instruction, providing explicit guidance for visuomotor control. The system further operates at two timescales, with slower route-level reasoning updating global navigation decisions and fast local control continuously reacting to the current observation. As illustrated in Fig. 1(c), this design couples long-term spatial consistency with responsive local control, enabling reliable real-time navigation through native keyboard–mouse actions.

- We introduce a global anchoring mechanism based on persistent topological memory. An offline video-derived graph provides cross-episode spatial knowledge, and visual place recognition periodically re-anchors visual odometry, maintaining long-range consistency without privileged localization.
- We develop a local anchoring mechanism via a language-waypoint interface. The next navigation target is projected as a 2D waypoint onto the current frame, and together with a language sub-instruction, it guides a waypoint-conditioned fast system that converts both channels into native game controls.
- We provide a multi-track empirical study in a photorealistic open-world game, covering cross-area instruction following, long-horizon traversal and coverage, a reference comparison on RxR under the VLN-CE setting, and targeted ablations of the memory bundle and each interface channel.

2 RELATED WORK

Generalist game agents and long-horizon navigation. Recent game agents combine multimodal perception, language reasoning, memory, planning, and action generation to operate in complex interactive environments. MineDojo (Fan et al., 2022), Voyager (Wang et al., 2024), and JARVIS-1 (Wang et al., 2023) study open-ended instruction following, skill acquisition, and memory-augmented planning in Minecraft, while SIMA (SIMA Team, 2024) and Cradle (Tan et al., 2025) extend such capabilities to diverse 3D worlds and commercial games through rendered observations and native control interfaces. More recent systems such as SIMA 2 (SIMA Team, 2025) and Learning to Play (Yue et al., 2025) further improve multimodal reasoning and real-time gameplay. Despite this progress, reliable navigation over extended trajectories remains difficult when no privileged pose or map is available. Our work focuses specifically on the spatial consistency and grounding problems that arise in long-horizon game navigation.

Persistent spatial context and action grounding. Existing navigation methods address different parts of this problem. NaVid (Zhang et al., 2024b), Uni-NaVid (Zhang et al., 2024a), and StreamVLN (Wei et al., 2025) use egocentric histories to preserve temporal context, while MapNav (Zhang et al., 2025) maintains more explicit spatial structure. ORB-SLAM (Mur-Artal et al., 2015) and DROID-SLAM (Teed & Deng, 2021) provide continuous motion estimates, whereas AnyLoc (Keetha et al., 2023) offers sparse but stable visual place recognition; these two forms of localization are naturally complementary over long trajectories. At the control level, Continuous VLN (Krantz et al., 2020), NaVILA (Cheng et al., 2025), SmartWay (Shi et al., 2025), and TAMPNav (Feng et al., 2026) use intermediate language commands or waypoints to bridge high-level planning and low-level execution. ANCHORNAV combines both through dual-scale anchoring, using a persistent topological memory for global consistency and an image-space waypoint for local grounding.

3 METHOD

3.1 PROBLEM FORMULATION AND DUAL-ANCHOR OVERVIEW

Given a natural-language instruction $\mathcal{I}$, the agent observes only rendered RGB frames $o_t \in \mathbb{R}^{H\times W\times 3}$ and acts through the game's native keyboard–mouse interface. Each action $a_t \in \mathcal{A}$ consists of relative mouse motion, mouse-button states, keyboard states, and an optional termination decision.

As illustrated in Fig. 2, ANCHORNAV formulates long-horizon navigation around two complementary spatial anchors.The *global anchor* $\mathcal{A}_{\mathrm{G}}$ maintains route-level consistency using persistent topological memory for continuous motion tracking. It estimates the current node n_t, selects the next route target n^*, and produces a semantic sub-instruction ℓ_t. The *local anchor* $\mathcal{A}_{\mathrm{L}}$ then grounds this target in the current view as a 2D waypoint w_t and converts (ℓ_t, w_t) into native game actions:

$$(n_t, n^*, \ell_t) = \mathcal{A}_{\mathrm{G}}(o_t, \mathcal{I}, \mathcal{G})\,, \qquad a_t = \mathcal{A}_{\mathrm{L}}(\mathrm{render}(o_t, n^*), o_{t-K:t}, \ell_t)\,. \tag{1}$$

Here $\mathcal{G}$ denotes the persistent topological memory and $o_{t-K:t}$ is a rolling window of K frames. The global anchor updates route and localization state when needed, while the local anchor continuously executes toward the active target. This separation decouples slow route-level reasoning from fast view-grounded control, enabling responsive execution without sacrificing long-horizon consistency.

3.2 PERSISTENT GLOBAL ANCHOR

The global anchor builds and updates a reusable spatial reference from streaming or offline gameplay videos, while maintaining the agent's route state relative to this memory during execution. It combines place recognition for stable long-range re-localization with visual odometry for continuous local motion tracking. Together, they preserve route-level spatial consistency throughout long navigation horizons.

Geometry-guided topological memory. The memory externalizes spatial knowledge that a finite frame history cannot preserve, providing a global prior for route planning and visual references for re-localization. For each environment, we construct a directed topological graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from offline walkthrough videos. Each node $v \in \mathcal{V}$ represents a landmark and stores semantic descriptions, nearby reference views, visual descriptors, and estimated camera geometry. Each edge $(u, v) \in \mathcal{E}$ represents an observed traversable transition together with its associated navigation information.

We begin by using a VLM (Gemini Team, 2023) to identify landmarks in each video and annotate them with timestamps, names, notes, and semantic tags. Around each landmark timestamp, we collect a temporal window of nearby frames and retain those whose AnyLoc descriptors (Keetha et al., 2023) exceed a similarity threshold, thereby obtaining multiple visually consistent views of the same place. We then augment these views with camera geometry. Because correspondence-based odometry is easily disrupted by dynamic objects and screen-fixed overlays in game footage, we instead apply VGGT-Omega (Wang et al., 2025), a feed-forward geometry model derived from VGGT and trained to handle distractors such as dynamic characters, on-screen UI, and subtitles. Operating over sliding windows, VGGT-Omega estimates robust camera poses and aligns them within a unified memory coordinate system.

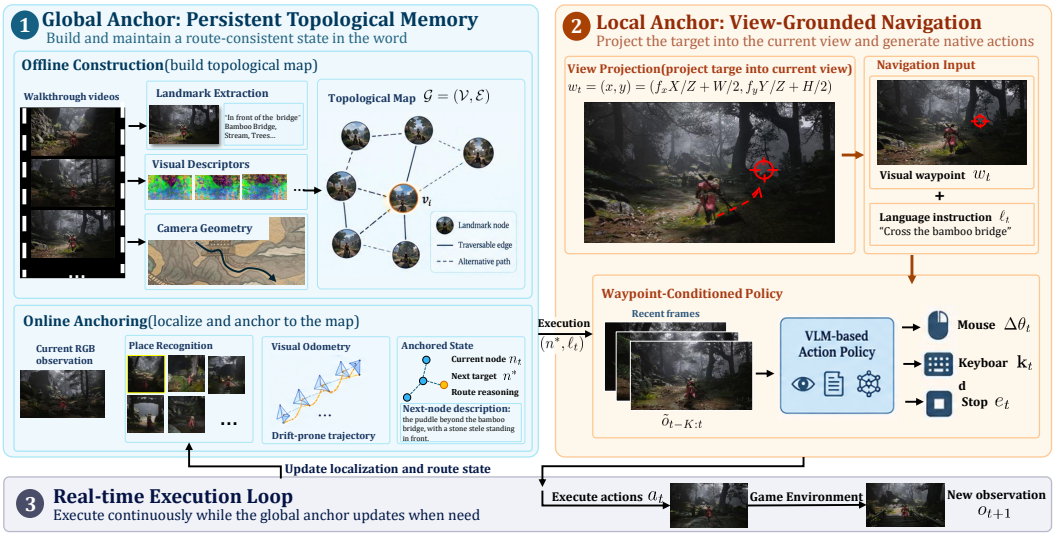


Figure 2: **Overview of ANCHORNAV.** The global anchor builds and maintains a persistent topological memory, localizes the agent, and selects the next route target. The local anchor projects that target into the current view as an image-space waypoint, renders it onto the input frames, and pairs it with a language sub-instruction for native keyboard–mouse control. The two anchors operate asynchronously: the local controller acts continuously, while localization and route state are updated when needed.

Because the node may be constructed from multiple videos with overlapping spatial coverage, we merge observations of the same location into canonical nodes. Each observation o and candidate node v is represented by two descriptors: a visual descriptor ϕ_{vis}, computed as a 49152-dimensional, L2-normalized VLAD over DINOv2 features, and a semantic descriptor ϕ_{sem}, computed as a 1024-dimensional multilingual embedding (Xiao et al., 2023) of the associated tags and note. The merge score is

$$s(o, v) = \lambda_{\text{vis}} \underbrace{\text{mean}_{\text{top-}K} \cos\big(\phi_{\text{vis}}(o),\, \phi_{\text{vis}}(v)\big)}_{\text{VLAD appearance}} + \lambda_{\text{sem}} \underbrace{\cos\big(\phi_{\text{sem}}(o),\, \phi_{\text{sem}}(v)\big)}_{\text{text embedding}}, \tag{2}$$

where $\lambda_{\text{vis}} = 0.7$, $\lambda_{\text{sem}} = 0.3$, and $K = 3$. If $s(o, v)$ exceeds a threshold, o is absorbed into v; otherwise it seeds a new node. The two cues together reduce false merges between visually similar but functionally distinct places, such as repeated rock faces or forest paths.

By combining visual appearance with semantic cues, the node-merging process reduces confusion between visually similar but semantically distinct locations. We then establish graph edges from observed transitions between landmarks, supplemented by connections recovered from intermediate video segments. The resulting topology provides a persistent spatial reference for subsequent localization and route planning.

Online localization and re-anchoring. At runtime, the global anchor combines place recognition with continuous visual odometry. Given the current observation, AnyLoc retrieves visually similar reference views and identifies the corresponding topological node n_t when the match is sufficiently confident. Between such recognized landmarks, VGGT-Omega estimates camera motion over overlapping temporal windows. For frame i in window $\mathcal{W}_k$, it predicts a local camera center $\mathbf{c}_i^{(k)}$ and orientation $\mathbf{R}_i^{(k)}$. The local estimate is aligned to the persistent memory through a similarity transform $(s_k, \mathbf{Q}_k, \mathbf{t}_k)$:

$$\hat{\mathbf{p}}_i^{\mathcal{G}} = s_k \mathbf{Q}_k \mathbf{c}_i^{(k)} + \mathbf{t}_k, \qquad \hat{\mathbf{R}}_i^{\mathcal{G}} = \mathbf{Q}_k \mathbf{R}_i^{(k)}, \quad i \in \mathcal{W}_k. \tag{3}$$

Here s_k, $\mathbf{Q}_k$, and $\mathbf{t}_k$ denote the scale, rotation, and translation that register the local trajectory to the memory coordinate system. Alignment is propagated across overlapping windows following the submap-alignment principle of VGGT-SLAM 2.0 (Maggio & Carlone, 2026). When a new landmark is confidently recognized, the current trajectory is re-aligned to its stored geometry, refreshing the global reference. In this way, visual odometry provides smooth local motion, while periodic place-based re-anchoring prevents drift from accumulating over long trajectories.

Route target selection. Given the current localized state, the global anchor selects the next route node n^* by shortest-path search over the topological graph. A VLM-based route planner (Qwen Team, 2026) then reasons over the current observation, recent visual context, the top-down odometry map M_t, the task instruction, and the current route state:

$$(d_t, \ell_t) = \pi_{\text{route}}(o_t, o_{<t}, M_t, n_t, n^*, \mathcal{I}, s_t)\,. \tag{4}$$

Here d_t determines whether the current sub-instruction should be updated, while ℓ_t specifies the semantic intent associated with the next route target. The compact session summary s_t records completed subgoals and visited regions to reduce redundant backtracking. The global anchor ultimately provides (n^*, ℓ_t) as the navigation interface to the local anchor.

3.3 Local visual anchor

The local anchor grounds the selected route target by rendering an image-space waypoint onto the input frames rather than encoding its coordinates in the language instruction. A language instruction conveys semantic intent but may remain geometrically ambiguous, while one image-plane coordinate supplies a direction without naming the target. We therefore use ℓ_t to specify *what* to accomplish and, when the projection is valid, a coordinate w_t to indicate *where* to move.

View-grounded waypoint. In training, (X, Y, Z) is the target anchor. It is projected by

$$w_t = (x, y) = (f_x X/Z + W/2, f_y Y/Z + H/2)\,, \qquad Z > 0, \tag{5}$$

where (f_x, f_y) are the camera focal lengths and $W \times H$ is the image resolution. If the projected point lies outside the image, it is clamped to the frame boundary, preserving a directional cue toward the off-screen target. This image-space grounding converts the global route target into a view-relative reference that can be directly consumed by the local controller.

Waypoint-conditioned action policy. We instantiate the local anchor with a multimodal action policy that conditions on recent observations, the active waypoint, and the language sub-instruction. Rather than encoding the waypoint as text, we render it directly onto the input frames, preserving its spatial correspondence with the current scene. During training, the waypoint is obtained from future trajectory points; at runtime, it is projected from the globally anchored target n^*.

Let $\tilde{o}_i = \text{render}(o_i, w_t)$ denote an observation marked with the active waypoint. The policy predicts native game controls as

$$(\Delta\theta_t, \mathbf{kb}_t, e_t) = \pi_{\text{act}}(\tilde{o}_{t-K:t}, \ell_t)\,, \tag{6}$$

where $\Delta\theta_t$ denotes relative mouse motion, $\mathbf{kb}_t$ the keyboard state, and e_t the termination decision. By combining semantic intent with explicit visual grounding, the local anchor turns each route-level target into a directly executable, view-conditioned control objective.

4 Experiments

Implementation details. The global anchor uses off-the-shelf VLM endpoints (Bai et al., 2025; Gemini Team, 2023) without gradient updates, AnyLoc at acceptance threshold 0.55, and VGGT-Omega in windows of 8 frames with overlap 3 at resolution 512. The local anchor is Qwen3.5-4B (Qwen Team, 2026), trained for 30,000 steps. Each frame is resized to 448×800. The window holds 20 frames taken every 10 frames from 20 Hz video. Training uses a cross-entropy loss over the action-token vocabulary: mouse motion is expressed as a relative-angle token, key presses as their corresponding key-letter tokens (e.g., `W`, `Space`), and termination as a dedicated `<stop>` token. Decoded actions are issued to the native keyboard–mouse interface at 20 Hz, i.e., each action is held for 50 ms. At inference the policy estimates an action span of variable length rather than a fixed horizon—the span length is decided per decision, not preset—and the span is replayed on the interface action by action at this 50 ms rate. The two anchors run asynchronously.

Dataset. We collect 350 hours of Chapter 1 gameplay from *Black Myth: Wukong* with frame-synchronized keyboard and mouse actions. Annotators segment it into goal-directed clips, each paired with one objective; Gemini (Gemini Team, 2023) then verifies each against the footage and

Table 1: Long-horizon traversal on Chapter 1 of *Black Myth: Wukong*. PCR is route coverage and PCR@15 is that coverage within 15 minutes; PCR-F is the coverage averaged over several runs. RVR is the fraction that advances and then turns back, counting loops and backtracking. TTS is averaged over successful runs only, and a dash means the boss was never reached.

Case	PCR@15	PCR-F	RVR	Succ.	TTS (min)
GPT-5.5	9.3	47.6	45.5	–	–
Claude-ops5	4.3	44.0	42.9	–	–
Gemini3.8flash	14.2	31.2	38.5	–	–
kimi-k3	4.4	39.3	40.0	–	–
ANCHORNAV	**30.3**	**96.9**	**37.0**	**62.5**	**62.29**



Figure 3: Illustrative trajectories on the long-horizon route.

discards mismatches and low-quality segments such as CG cutscenes and wall collisions. We evaluate on four tracks: long-horizon traversal and timed coverage, instruction following on same-area and cross-area splits, a reference comparison on RxR, and ablations of the memory bundle and of the two interface channels.

4.1 LONG-HORIZON TRAVERSAL AND TIMED COVERAGE

Starting from the Chapter 1 spawn point in *Black Myth: Wukong*, the agent must reach the boss arena with no human intervention, privileged map or pose signal, every step issued through the native keyboard–mouse interface from rendered frames; the route crosses several connected areas that a skilled human completes in about 20 minutes. At test time, each method is allotted at most two hours of wall-clock runtime per run and is evaluated over eight independent runs.

We compare against four zero-shot general-purpose VLMs (GPT-5.5, Claude-ops5, Gemini3.8flash and kimi-k3). This is the track in which ANCHORNAV relies on its persistent memory, and both sides do carry an environment prior here, but not the same one: the baselines receive a navigation map in their prompt and may consult the in-game character map, whereas ANCHORNAV navigates from the Geometry-guided topological memory of Section 3.2, which no baseline is given. The two priors differ in kind, in coverage and in construction cost, and we do not quantify that difference; the table is therefore a systems comparison that isolates neither the dual-anchor design. No baseline reaches the boss in any run, so its success rate and time are undefined, whereas ANCHORNAV succeeds on most runs; the four do not stall in the same way, and the difference is visible only when backtracking is read against coverage—the baseline that turns back least (Gemini3.8flash) is also the one that covers the least route by the end, whereas the one that advances furthest (GPT-5.5) is also the one that retraces most, the signature of committing to a heading and then losing it, and Claude-ops5 and kimi-k3, which commit least over the first 15 minutes, finish between the two. Normalising by progress removes this confound (R/C_{full}: 0.38 for ANCHORNAV against 0.96, 0.98, 1.02 and 1.23 for GPT-5.5, Claude-ops5, kimi-k3 and Gemini3.8flash), so a stalled agent no longer looks efficient merely because it backtracks little, and ANCHORNAV both backtracks least in absolute terms and covers almost the whole route.

4.2 SHORT-HORIZON INSTRUCTION FOLLOWING

This track isolates the local anchor: the global anchor fixes the route target and sub-instruction, so errors here are grounding and execution errors rather than planning errors. The benchmark has three levels of 50 tasks with five trials each: *easy* tasks use atomic instructions ("run straight to the wooden gate"), *medium* tasks are compositional, requiring a landmark to be recognised and acted on at the right moment, and *hard* tasks come from chapters outside the Chapter 1 collection above.

General-purpose VLMs (Kimi3, Gemini-3.1-Pro, GPT-5.5) emit actions directly from rendered frames without training on our data, while the VLN methods (DualVLN, ActiveVLN) are retrained on our dataset with the same demonstrations, resolution and action space, so the comparison reflects the method rather than the data and matches the training budget of the ANCHORNAV controller. All baselines share observation stream, control rate, model-call budget and termination prompt. As in

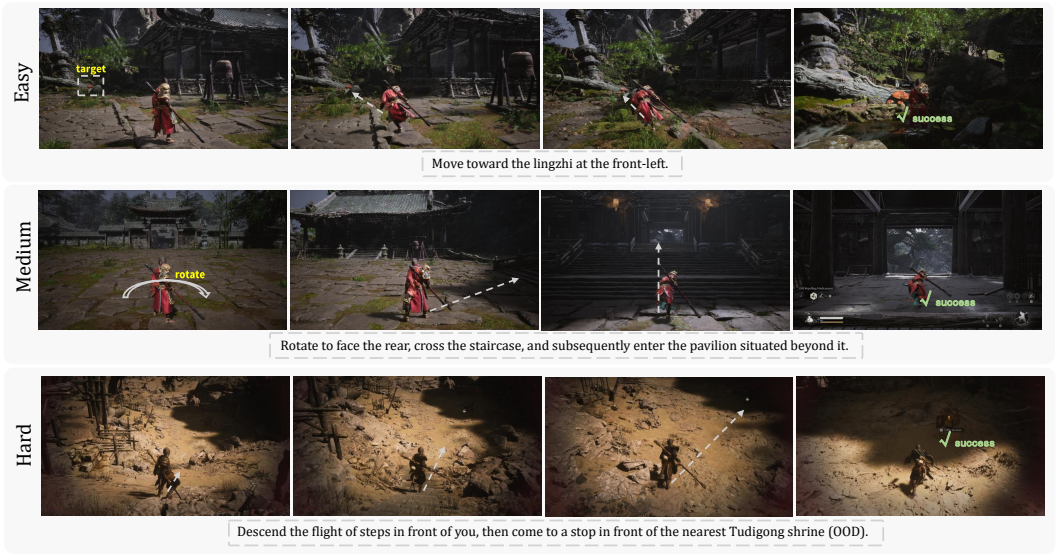


Figure 4: Qualitative rollouts for one easy, one medium, and one hard instruction.

Table 2: Short-horizon instruction following by difficulty level. SR and OSR are percentages; higher is better for all metrics. Best result per column in **bold**.

	Easy				Medium				Hard			
Method	SR	OSR	SPL	nDTW	SR	OSR	SPL	nDTW	SR	OSR	SPL	nDTW
General-purpose VLMs												
Kimi3	42.2	61.4	0.391	0.440	24.4	33.6	0.236	0.402	24.0	33.2	0.230	0.387
Gemini-3.1-Pro	50.6	**83.6**	0.467	0.458	35.6	49.2	0.346	0.465	29.6	48.4	0.283	0.452
GPT-5.5	39.8	76.8	0.365	0.386	23.6	40.4	0.195	0.324	30.0	**57.6**	0.264	0.393
Open-source approach												
DualVLN	15.4	36.2	0.152	0.192	12.0	19.6	0.120	0.050	10.8	38.4	0.098	0.305
ActiveVLN	48.2	62.4	0.482	0.487	22.8	36.0	0.214	0.442	15.2	28.4	0.144	0.401
AnchorNav	**53.8**	77.2	**0.502**	**0.508**	**47.6**	**50.0**	**0.439**	**0.547**	**38.0**	49.2	**0.351**	**0.533**

Section 4.1, the generalist VLMs are zero-shot single models, so we match neither parameter count, training compute nor number of components.

Table 2 averages over the 50 tasks of each level and their five trials; AnchorNav is best on SR, SPL and nDTW at every level, and the margin over the strongest baseline widens with difficulty. The generalist VLMs reproduce the failure mode isolated in the ablation: their OSR is high relative to their SR, i.e. they reach the vicinity without recognising arrival—the stopping criterion the sub-instruction supplies and the waypoint alone cannot express—whereas AnchorNav attains a comparable OSR with a much smaller OSR–SR gap, so the gain is in converting arrival into success rather than in wandering further. The trained VLN baselines fail differently: ActiveVLN is competitive on easy but degrades most sharply, the signature of fitting the training area rather than acquiring transferable grounding, whereas DualVLN is uniformly weak, with nDTW not even monotone across levels. AnchorNav retains 70.6% of its easy SR on hard and remains best in absolute terms.

4.3 Reference comparison on RxR

The two game tracks test behaviour that no public benchmark reproduces, so we add a third track moving the agent to RxR under the continuous-environment (VLN-CE) setting (Ku et al., 2020; Krantz et al., 2020), English val-unseen, where an episode succeeds when the agent stops within 3 m of the goal; we report NE (m, lower better), SR, SPL and nDTW (higher better). The agent receives only the first-person RGB stream—no panorama, depth, odometry or pose—and reaches the VLN-CE action set through a discrete-action adapter. In this new environment it is trained



Figure 5: Qualitative on RxR dataset.

Table 3: Comparison on RxR (VLN-CE), English val-unseen.

Method	NE ↓	SR ↑	SPL ↑	nDTW ↑
Seq2Seq	12.1	13.9	11.9	30.8
NaVILA	6.77	49.3	44.0	58.8
UniNaVid	6.24	48.7	40.9	–
StreamVLN	6.22	52.9	**46.0**	**61.9**
ActiveVLN	5.84	50.7	41.2	58.1
ANCHORNAV	**5.51**	**61.65**	42.12	52.56

Table 4: Ablation of the global memory and of the two interface channels. PCR@15 is the route-coverage quantity of 15 minutes.

Config.	PCR@15	SR	OSR	SPL	nDTW
(A) w/o memory	7.50	46.9	53.8	0.444	0.551
(B) ℓ_t only	16.9	53.6	69.2	0.513	0.624
(C) w_t only	21.4	36.9	71.5	0.253	0.582
(D) $w_t + \ell_t$	**32.7**	**65.7**	**85.4**	**0.610**	**0.629**

on RxR with only a small amount of data, the same low-data regime as ActiveVLN. Its graph is built offline from trajectories of the RxR *training* split only: no val-unseen episode, and hence no scan held out for val-unseen, is replayed or observed at construction time—observing val-unseen footage there would void the generalization this split is meant to test. The memory therefore holds no scan-specific node, view or edge for the environments it is evaluated on in this track: retrieval can only return training-scan nodes, and within a val-unseen episode the global anchor is carried by visual odometry together with the within-episode session summary rather than by a stored map of that scene. This track consequently does not measure memory-supported localization in unseen scenes, and we attribute none of its numbers to the persistent memory; what it reports is whether grounding and arrival survive when the global anchor has no scene-specific prior to offer, while the memory's contribution is measured in Section 4.1 and Section 4.4, where a walkthrough of the evaluated environment precedes the episode. Baseline numbers are quoted and were not rerun with this adapter, so memory construction, control frequency and stopping remain unmatched, and the training-data budget is matched to ActiveVLN's regime rather than to the remaining baselines.

Table 3 shows two regimes rather than a smooth ranking: Seq2Seq (Krantz et al., 2020), with neither a vision–language prior nor a spatial record, is far behind, while the other four baselines (Cheng et al., 2025; Zhang et al., 2024a; Wei et al., 2025) cluster tightly (NE 5.84–6.77 m, SR 48.7–52.9), so among methods mapping history directly onto actions the backbone matters less than having an explicit intermediate goal. ANCHORNAV leaves that cluster on both arrival metrics—NE 5.51 m against 5.84 for the closest baseline, and SR 61.65 against 52.9—but not on path quality: StreamVLN keeps the best SPL (46.0 vs. 42.12) and nDTW (61.9 vs. 52.56), and ANCHORNAV also trails NaVILA and ActiveVLN on nDTW.

What transfers to RxR is thus the ability to arrive, not the shape of the path that gets there (Figure 5). Two facts of the protocol bound this reading: RxR episodes are two orders of magnitude shorter than our game tracks, so the track exercises grounding and arrival but barely the long-horizon consistency that the persistent global anchor is built for; and the agent reaches the VLN-CE action set only through a discrete-action adapter, with baseline numbers quoted rather than rerun, so control frequency and stopping are unmatched and the path-quality gap is not cleanly attributable. RxR is therefore a reference point for arrival, not evidence of a path-quality gain.



Figure 6: Qualitative rollouts for instruction-only, waypoint-only, and the full interface.

4.4 Ablation studies

We ablate two factors with the local-policy checkpoint fixed, namely the persistent global memory (graph, place recognition with periodic re-anchoring, session summary) and the two channels of the mid-level interface, ℓ_t and w_t, giving four configurations: **(A)** without memory, **(B)** instruction only, **(C)** waypoint only and **(D)** the full system. All four are evaluated on coverage under a fixed 15-minute budget (PCR@15, 5 runs each) and on instruction following (14 instructions, 10 runs each, macro-averaged); all metrics are higher-is-better. They are therefore the controlled conditions of this study: they share the same offline video, the same graph construction, the same controller checkpoint and hence the same training budget, and the same evaluation budget, and differ only in the memory bundle and in which interface channel is enabled. (A) is the egocentric-history control with the local controller unchanged, while (B) and (C) are graph retrieval restricted to a single channel. Re-anchoring is still not isolated: (A) removes it together with the rest of the memory and no condition keeps the graph while switching it off, so the gains are attributed to the memory bundle as a whole rather than to re-anchoring in particular.

Table 4 shows a clean division of labour: the memory governs whether the agent keeps making progress at all, whereas the interface channels govern whether a reached target is also recognised as reached. Removing the memory (A) collapses coverage and costs most of the SR: with only egocentric frames the planner keeps no record of covered regions and no correction for odometry drift, so it loses its heading and circles back toward the start. The two channels fail in complementary ways: waypoint-only (C) has the highest OSR among the single-channel variants yet the lowest SR, because a 2D point gives an unambiguous direction but no stopping criterion, so the agent overshoots and terminates at the wrong place or not at all; instruction-only (B) names the target and its completion condition, so it stops correctly but stays geometrically ambiguous when similar landmarks compete or the memory frame has drifted.

5 Conclusion

Long-horizon navigation in photorealistic 3D games fails in two separable ways that extending the frame history does not fix: the agent loses its global spatial reference, and it cannot ground a route-level target unambiguously in the current view. AnchorNav addresses both through dual-scale anchoring. A global anchor keeps a persistent topological memory built from gameplay video and periodically re-anchors visual odometry by place recognition; a local anchor writes one image-plane coordinate of the next target into the instruction, so that a single controller input carries both where to move and what arrival means. Because the two anchors meet only in this narrow interface and run at different timescales, slow route-level reasoning and fast view-grounded control are decoupled rather than traded against each other. Across the four tracks the two anchors pay off where they were built to: AnchorNav is the only system that completes the long-horizon route, it leads instruction following at every difficulty level, and the ablation attributes sustained progress to the memory bundle and successful arrival to the language–waypoint interface.

AI USE STATEMENT

We used generative AI tools as *components of the proposed system*: a commercial VLM proposes landmark observations and annotates edges during offline graph construction, verifies human-annotated demonstration clips against the source footage during data curation (Section 4), and serves as the runtime planner. We additionally used generative AI assistance for code completion and for editing the prose of this manuscript, and all AI-assisted text and code was reviewed and verified by the authors; generative AI was not used to produce research ideas, experimental results or citations without verification, and we take responsibility for the final content of this work.

REFERENCES

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.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.018.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Hongyan Feng, Sunlai Chen, Xuanyu Liu, Miao Pan, Yangfan Xie, Yuxiang Cui, Zhongxiang Zhou, Rong Xiong, Wenqi Zhang, Jianwei Yin, Yueting Zhuang, and Xuhong Zhang. Embodied-navigator: Point, think, memorize, and align for efficient navigation. *arXiv preprint arXiv:2608.17512*, 2026.

Gemini Team. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 9(2):1286–1293, 2023.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Proceedings of the European Conference on Computer Vision*, pp. 104–120, 2020. doi: 10.1007/978-3-030-58604-1_7.

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 European Conference on Computer Vision*, pp. 439–454, 2020. doi: 10.1007/978-3-030-58558-7_26.

Dominic Rosario Maggio and Luca Carlone. VGGT-SLAM 2.0: Real-time dense feed-forward scene reconstruction. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.051.

Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

Yanyuan Qiao, Haodong Hong, Wenqi Lyu, Dong An, Siqi Zhang, Yutong Xie, Xinyu Wang, and Qi Wu. Navbench: Probing multimodal large language models for embodied navigation. In *Advances in Neural Information Processing Systems*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, 2026.

Xiangyu Shi, Zerui Li, Wenqi Lyu, Jiatong Xia, Feras Dayoub, Yanyuan Qiao, and Qi Wu. Smartway: Enhanced waypoint prediction and backtracking for zero-shot vision-and-language navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 16923–16930, 2025. doi: 10.1109/IROS60139.2025.11246147.

SIMA Team. Scaling instructable agents across many simulated worlds. *arXiv preprint arXiv:2404.10179*, 2024.

SIMA Team. SIMA 2: A generalist embodied agent for virtual worlds. *arXiv preprint arXiv:2512.04797*, 2025.

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, Ruyi An, Molei Qin, Chuqiao Zong, Longtao Zheng, Yujie Wu, Xiaoqiang Chai, Yifei Bi, Tianbao Xie, Pengjie Gu, Xiyun Li, Ceyao Zhang, Long Tian, Chaojie Wang, Xinrun Wang, Börje F. Karlsson, Bo An, Shuicheng Yan, and Zongqing Lu. Cradle: Empowering foundation agents towards general computer control. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 58658–58725. PMLR, 2025.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16558–16569, 2021.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025.

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. JARVIS-1: Open-world multi-task agents with memory-augmented multimodal language models. *arXiv preprint arXiv:2311.05997*, 2023.

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. *arXiv preprint arXiv:2507.05240*, 2025.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. *arXiv preprint arXiv:2309.07597*, 2023.

Peiran Xu, Sudong Wang, Yao Zhu, Jianing Li, and Yunjian Zhang. Spatialbench: Benchmarking multimodal large language models for spatial cognition, 2025.

Yuguang Yue, Irakli Salia, Samuel Hunt, Christopher Green, Wenzhe Shi, and Jonathan J. Hunt. Learning to play: A multimodal agent for 3d game-play. *arXiv preprint arXiv:2510.16774*, 2025.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024a.

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. In *Proceedings of Robotics: Science and Systems*, 2024b. doi: 10.15607/RSS.2024.XX.079.

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A novel memory representation via annotated semantic maps for VLM-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13032–13056. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.638.

Shaoheng Zhang, Zhichen Li, and Jie Mei. Dart-vln: Test-time memory decay and anti-loop regularization for discrete vision-language navigation, 2026.

Qihui Zhu, Shouwei Ruan, Xiao Yang, Hao Jiang, Yao Huang, Shiji Zhao, Hanwei Fan, Hang Su, and Xingxing Wei. Mind over space: Can multimodal large language models mentally navigate? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 11488–11497, 2026.