VLN-MEM: SCENE-MEMORY-GUIDED VISION-AND-LANGUAGE NAVIGATION FOR RELIABLE ARRIVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-and-Language Navigation (VLN) asks an agent to follow instructions to a goal from egocentric observations. Recent VLN systems have improved grounding, yet they still fail in two distinct modes: they fail to *reach* the goal, and when they do reach it they fail to *stop* there. Both failures share one cause: the goal is specified as a *destination*, a pose and a view, which says *where* the goal is but not *how to reach it* nor *whether it has been reached*. In this paper, we propose VLN-MEM, a scene-memory-guided VLN for reliable arrival, which supplies both conditions from a persistent scene memory: a frozen per-scene graph of places, built offline from reference trajectories, from which retrieval returns next-hop geometry for the reachability and a stored goal view for the recognition. VLN-MEM adopts zero-initialized residual fusion to feed the reachability into the planner's visual stream, and training-free stop verification to confirm arrival against the goal's appearance and proximity. VLN-MEM reaches 65.0% and 59.8% SR on R2R-CE and RxR-CE val-unseen in a pre-mapped, goal-specified setting, gaining 14.5 SR and 13.6 SPL over a training-budget-matched backbone; against stronger full-data references it matches or exceeds their SR while navigating goal-directly. A component ablation confirms that each mechanism contributes.

Keywords: Vision-and-Language Navigation, VLN, Scene Memory, Reachability, Embodied AI

1 INTRODUCTION

Vision-and-Language Navigation (VLN) requires an agent to interpret natural language instructions and navigate to a goal using egocentric observations. It is a central capability for embodied assistants operating in complex indoor environments. Recent vision-language models (VLMs) have advanced instruction grounding and navigation generalization (Zhang et al., 2024b; 2025a; Cheng et al., 2025; Wei et al., 2026), planning and acting directly from the current observation and a short history of frames.

Despite this progress, VLN systems still fail in two distinct modes: they fail to *reach* the goal, and when they do reach it they fail to *stop* there. Reaching means entering the goal region at any point; stopping means terminating inside it. A strong vision-language baseline reproduces both on R2R-CE: it never enters the goal region in 35.4% of episodes, and in a further 14.1% it enters the region but terminates outside it (Figure 4a). These are not failures to interpret the instruction; they are failures to complete the navigation.

The agent receives a natural-language route instruction and egocentric observations, but no explicit goal pose or goal view. The instruction describes the route linguistically, leaving the agent to ground it in the unseen scene and determine from its observations when the destination has been reached. We separate these requirements into *reachability*, scene-level connectivity aggregated from multiple prior episodes and used to identify a feasible next hop toward the goal, and *recognition*, place-level evidence that confirms arrival. A short observation history provides neither a persistent account of how the current location connects to the destination nor a stable reference for recognizing it. Our goal-specified setting therefore provides the goal pose and view as additional inputs and binds them to a prebuilt scene memory; they are not part of the standard R2R task.

Existing methods do not supply both forms of evidence from prior scene knowledge. Typical VLN agents reason from the current view and a short history (Zhang et al., 2024b; 2025a; Cheng et al.,

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.

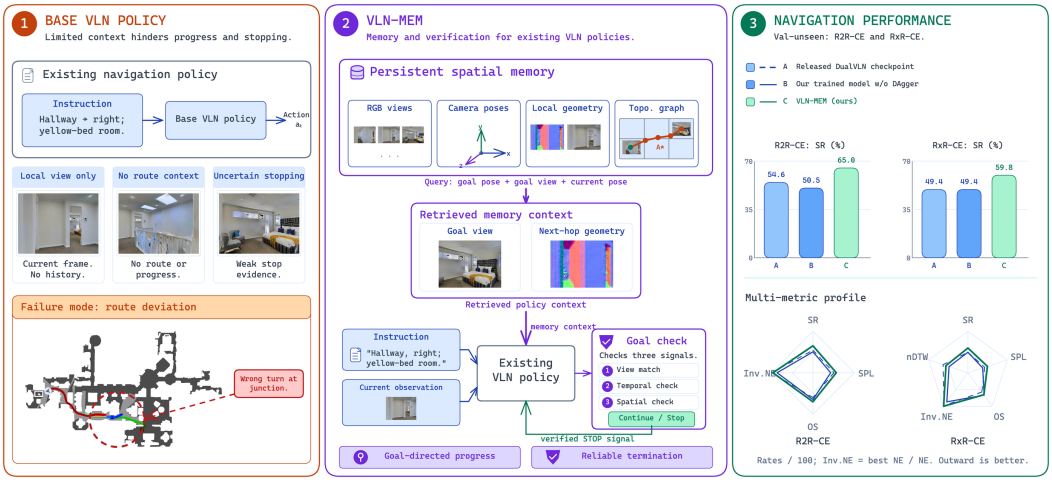


Figure 1: Motivation, design, and outcome of VLN-MEM. (1) A base VLN policy acting from local observations has neither route context nor strong evidence of arrival; the example trajectory deviates from the instructed route. **(2)** VLN-MEM binds the specified goal pose and view in persistent spatial memory, anchors the agent by its current pose, conditions the existing policy on the retrieved goal view and next-hop geometry, and verifies its stop decision. **(3)** Results on R2R-CE val-unseen and RxR-CE val-unseen English compare VLN-MEM with *Baseline A*, the released DualVLN checkpoint whose use of DAgger is not established by the paper, and *Baseline B*, our independently trained w/o-DAgger model. The radar plots summarize the corresponding multi-metric profiles (outward is better).

2025; Wei et al., 2026), a momentary trace of that direction. Some map-assisted VLN agents record the places they visit into a map (Wang et al., 2023a; An et al., 2025; Zhang et al., 2025b), a fuller record of the same trajectory. Either way, the goal side of the scene is learned only after the agent arrives, when the two conditions are no longer needed. What is missing is scene knowledge acquired outside the agent's own trajectory, from which the agent derives reachability and recognition during navigation.

To address these problems, we propose VLN-MEM, a scene-memory-guided VLN framework for reliable arrival. Three key mechanisms realize it. First, a scene memory, a per-scene graph of places built offline from reference trajectories, supplies the next-hop geometry for reachability and a stored goal view for recognition. Second, zero-initialized residual fusion feeds this reachability into the planner's visual stream, so the planner steers from local observations while remaining globally oriented. Third, training-free stop verification confirms arrival against the goal's appearance and proximity, supplying the recognition. The three mechanisms act as evidence, not control: they condition the planner and the stop decision without adding a separate map encoder, and applying them to another planner requires only access to its visual tokens.

Figure 1 summarizes the motivation, design, and outcome of VLN-MEM. A current-view-only policy lacks route context and reliable stopping evidence; VLN-MEM retrieves both from persistent spatial memory, conditions the existing policy, and verifies its stop decision. On both benchmarks, the resulting system improves success rate over both the released DualVLN checkpoint and our own w/o-DAgger training.

We evaluate VLN-MEM on R2R-CE and RxR-CE in a pre-mapped, goal-specified setting, where scenes are mapped offline and the goal pose and view are provided. VLN-MEM reaches 65.0% and 59.8% SR on R2R-CE and RxR-CE val-unseen. Relative to our trained w/o-DAgger baseline, this is a gain of 14.5 SR and 13.6 SPL on R2R-CE, and 10.3 SR and 12.1 SPL on RxR-CE. Results against stronger full-data references remain mixed. A component ablation adds each mechanism in turn, confirming that the scene memory, the residual fusion, and the stop verification each contribute to the final performance.

Our contributions are threefold:

- We propose VLN-MEM, a scene-memory-guided VLN framework for reliable arrival. Its core idea is to supply the two conditions a destination omits, reachability and recognition, from prior knowledge of the scene rather than the agent's own trajectory.

- We develop three mechanisms that realize the core idea: a scene memory that retrieves next-hop geometry and a stored goal view from prior knowledge of the scene, zero-initialized residual fusion that feeds the geometry into the planner as reachability, and training-free stop verification that uses the goal view to confirm recognition.
- We evaluate VLN-MEM on R2R-CE and RxR-CE in a pre-mapped, goal-specified setting, where it reliably reaches and stops at the goal and outperforms the training-budget-matched backbone. An ablation attributes the gain to the two conditions the method supplies: reachability improves arrival, and recognition improves stopping.

2 Related Work

Typical VLN Agents. VLN began on discrete nav-graphs (Anderson et al., 2018; Ku et al., 2020) and moved to continuous control with VLN-CE (Krantz et al., 2020), helped by large-scale data generation (Wang et al., 2023b) and physically realistic platforms (Wang et al., 2025a). Current foundation models treat navigation as VLM-conditioned next-step prediction from the current view and a short history: NaVid (Zhang et al., 2024b) and Uni-NaVid (Zhang et al., 2025a) predict actions from a single-view video stream, NaVILA (Cheng et al., 2025) pairs a VLA planner with a legged controller, and StreamVLN (Wei et al., 2026) adds slow-fast streaming context. AdaNav (Ding et al., 2025) instead triggers explicit reasoning adaptively from action uncertainty to avoid unnecessary computation. DualVLN (Wei et al., 2025) decouples a VLM planner from a diffusion controller through learnable latent queries. These methods improve temporal reasoning or control, but a short history is a momentary trace of the agent's own trajectory and cannot supply the two conditions a destination omits, reachability and recognition.

Memory-enhanced VLN Agents. GridMM (Wang et al., 2023a), ETPNav (An et al., 2025), and MapNav (Zhang et al., 2025b) give the policy explicit spatial structure through grid, topological, or annotated semantic-map representations. However, they record the places the agent visits into a map online within a single episode, a fuller record of the same trajectory, so the goal side of the scene is learned only after the agent arrives; they also couple a dedicated map encoder into the policy and do not separate the goal into reachability and recognition. Our scene memory is instead acquired outside the agent's own trajectory: a per-scene graph of places built offline from reference trajectories and kept frozen, whose retrieval supplies the next-hop geometry for reachability and a stored goal view for recognition. These conditions are consumed as evidence through zero-initialized residual fusion, without a separate map encoder. Poses and per-pixel geometry come from feed-forward reconstruction (CUT3R (Wang et al., 2025b), after DUSt3R (Wang et al., 2024)), so no depth sensor or panorama is needed; known episode start poses align reconstructions into the shared Habitat scene frame.

Retrieval augmentation for embodied agents. What a scene memory encodes is separate from how the agent retrieves from it; for the latter we follow retrieval-augmented generation (Lewis et al., 2020), which indexes an external store and retrieves from it to condition generation. Applied to the scene memory above, the query is the specified goal pose and view; retrieval gates candidates by proximity and reranks them by CLIP (Radford et al., 2021) appearance similarity, and the retrieved node supplies the next-hop geometry for reachability and a stored goal view for recognition.

3 Method

VLN-MEM adds a persistent scene memory to a vision-language navigation agent and draws from it the two conditions a goal destination does not supply: reachability and recognition. Figure 2 shows how the three mechanisms divide across the navigation pipeline. Given the goal, scene memory supplies next-hop geometry for reachability and a stored goal view for recognition. Zero-initialized residual fusion feeds the retrieved context into the planner's visual tokens. Stop verification retains the planner's stop signal and adds two ways to end the trajectory: a temporally consistent appearance match, or a joint appearance-and-proximity match. These mechanisms require access only to the visual stream and stop signal, so they apply to an end-to-end or modular policy alike. The additional scene and goal inputs are specified in Section 4.

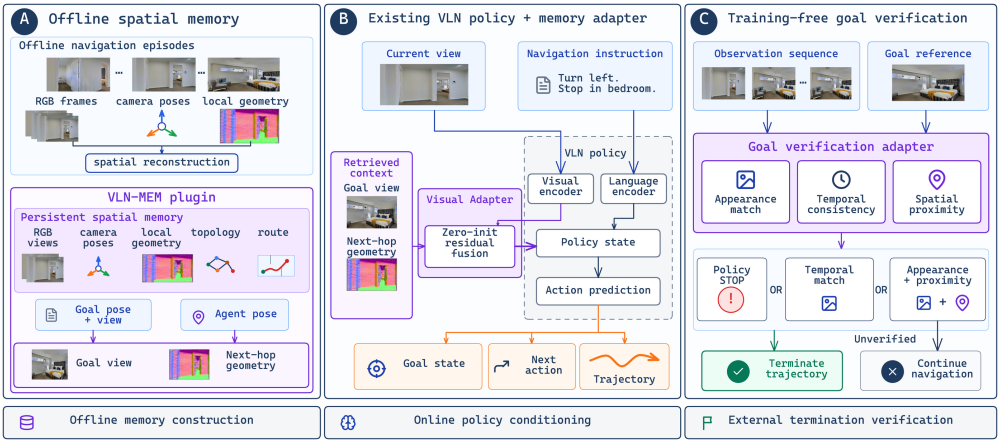


Figure 2: VLN-MEM architecture. (A) Offline reconstruction organizes reference episodes into RGB views, camera poses, local geometry, and topology. At inference, the specified goal pose and view bind the goal node, while the current pose anchors the agent for retrieval of a goal view and next-hop geometry. **(B)** A zero-initialized residual adapter injects the retrieved context into the policy's visual stream while preserving its action interface. **(C)** The verifier terminates on the policy's own stop signal, a temporally confirmed appearance match, or joint appearance-and-proximity evidence; otherwise navigation continues.

3.1 Scene Memory

Offline construction. The scene memory is the source of prior knowledge of the scene, and it must be acquired outside the agent's own trajectory. Reference trajectories describe each scene episode-by-episode: the same place reappears across many episodes, but each episode records it in its own coordinate frame. We collapse these per-episode observations into one per-scene graph of *places*, built offline. A place is a node storing a 6-DoF pose, an RGB view, a CLIP (Radford et al., 2021) appearance embedding, and a dense point map. Feed-forward reconstruction (Wang et al., 2025b) recovers per-frame geometry from each pre-rendered monocular sequence, and the known episode start pose lifts the camera-relative estimates into the shared scene frame (Savva et al., 2019). Two observations are merged into one place only when they agree in both pose and appearance, which collapses duplicate views of the same place across overlapping episodes while keeping visually similar but distant rooms distinct (merge thresholds in Appendix A.1, "Offline scene memory construction"). Edges encode observed local traversability: temporal edges connect consecutive observations within a trajectory, and loop closures, spatial bridges, and a connectivity pass connect compatible places across episodes. The scene is assumed static once built, so nodes are reusable places rather than isolated timesteps.

Retrieval. The scene memory is useful only if the agent can pull the right place out of it for the current episode. Retrieval answers two queries: the specified goal pose and view bind a goal place, and the agent's current pose anchors the agent's place. No single signal identifies the right node: appearance is ambiguous across visually similar rooms, while proximity cannot separate nearby viewpoints. We therefore retrieve in two stages. A *position gate* first keeps only candidates near the query; an *appearance rerank* then selects, among those, the candidate whose view best matches the query view in CLIP (Radford et al., 2021) similarity, returning one place $n^\star$. The two stages are complementary: proximity resolves which room, appearance resolves which viewpoint within it. If the best match is not confident enough, both outputs are blanked and the episode proceeds without them, so an uncertain retrieval degrades to the base planner rather than misdirecting it (thresholds in Appendix A.1, "Scene memory and retrieval"). The specified goal view is used only for this retrieval and is never passed directly to the planner. Goal binding emphasizes view identity, whereas the start anchor emphasizes current location.

Next-hop geometry. Binding the goal place locates the goal but does not say how to reach it. We derive reachability by searching the scene memory for a path from the agent's anchor to the goal place and taking the first hop: A^* (Hart et al., 1968) over the graph of places yields a next-hop place whose point map is fused to supply reachability (Section 3.2). Reachability is thus a function of the goal place and the agent's current place in the graph, derived by graph search rather than by replaying the instruction's demonstrated route: the agent is steered goal-directly and completes

without retracing the demonstration. Completion is thereby decoupled from path fidelity, which we measure in Section 4.2. If no path exists, the goal view is retained and the next-hop slot is blanked, so recognition survives even when reachability cannot be supplied. No online mutation or frontier expansion occurs; the scene memory is frozen. During training, the planner is exposed to a degraded reachability signal on a fraction of steps so that it does not learn to depend unconditionally on a perfect retrieval (schedule in Appendix A.1, "Oracle/noisy/empty training schedule"). The two retrieved signals operate at different horizons: the goal view is a fixed recognition reference once bound, whereas the next-hop geometry is recomputed as the agent moves. The planner thus receives reachability and recognition through separate residuals.

3.2 Zero-initialized residual fusion

The planner must integrate reachability and recognition with its current observation without disrupting the pretrained visual stream. The retrieved goal view and the rendered next-hop point map are encoded by the *same* visual transformer that encodes the current frame, then added onto the current frame's merged visual tokens through two zero-initialized branches:

$$\hat{h}_{\text{cur}}^{(k)} \;=\; h_{\text{cur}}^{(k)} \;+\; w_{\text{geo}}\,\mathrm{Geo}_{\phi}\big(h_{P_{n_{\text{next}}}}^{(k)}\big) \;+\; w_{\text{goal}}\,\big[\mathrm{Goal}_{\psi}\big(h_{n^{\star}}\big)\big]^{(k)}, \tag{1}$$

where Geo_{ϕ} is a zero-initialized linear projection of the point map's aligned token and Goal_{ψ} a zero-initialized cross-attention resampler compressing the goal view's tokens onto the current grid (architecture in Appendix A.1, "Fusion architecture"). At initialization $\hat{h}_{\text{cur}} = h_{\text{cur}}$ exactly, so the pretrained visual stream is preserved before adaptation; this initialization does not imply that later gains arise from fusion alone. Sharing the ViT places the current frame, the goal view, and the rendered geometry in one feature space without a separate map encoder. Empty or rejected retrieval produces zero residuals and recovers the current-frame stream of the adapted planner. The interface requires only an accessible token-level visual stream; adapting a planner without one would require a different fusion point.

3.3 Training-free stop verification

Reaching the goal region does not guarantee successful termination: a baseline without scene memory enters the success region but terminates outside it in 14.1% of episodes (Figure 4a). Stop verification addresses this by supplementing the planner's stop signal s_t with recognition evidence from the goal place $n^{\star}$: a DINOv2 (Oquab et al., 2024) appearance check v_t between the current frame and the goal view, and a joint appearance and distance check j_t. The appearance branch requires a strict visual match; the joint branch accepts a weaker match only when the agent is near the goal place, trading recognition strictness against proximity. Each branch is temporally gated to require a sustained match, which reduces sensitivity to transient visual matches:

$$\text{STOP} \;\Leftarrow\; s_t \;\vee\; \widetilde{v}_t \;\vee\; \widetilde{j}_t, \tag{2}$$

where tildes denote temporal gating (thresholds in Appendix A.1, "Inference-time stop verification"). The rule requires no additional training labels and permits adjusting thresholds without retraining the planner. The OR rule preserves the planner's stop decisions along the executed prefix and can only terminate a rollout earlier, which supports the paired analysis in Section 4.2. It targets success within the benchmark's goal region, not exact goal-pose localization.

4 Experiments

Data. R2R-CE (Anderson et al., 2018; Krantz et al., 2020) and RxR-CE (Ku et al., 2020) trajectories are re-rendered with four camera triplets, $(125\,\mathrm{cm}, 0^{\circ}, 30^{\circ})$, $(125\,\mathrm{cm}, 0^{\circ}, 45^{\circ})$, $(60\,\mathrm{cm}, 15^{\circ}, 30^{\circ})$, and $(60\,\mathrm{cm}, 15^{\circ}, 45^{\circ})$, where the entries are camera height, level-view pitch, and downward-view pitch. Rendering uses Matterport3D (Chang et al., 2017) in Habitat (Savva et al., 2019). We evaluate on R2R-CE val-unseen (n=1839) and RxR-CE val-unseen English (n=3669).

Metrics. We report NE, OS, SR, and SPL, plus nDTW on RxR. Reliable arrival decomposes into the two conditions our mechanisms supply: OS records whether the trajectory ever enters the 3 m success region, an empirical proxy for reachability; SR additionally requires termination there, an

Table 1: R2R-CE val-unseen (n=1839). *Goal Spec.* marks whether the goal pose and rendered goal view are additionally provided; DualVLN rows are our evaluations of the released checkpoint, other external values come from the cited papers, and the training-budget-matched *Base* is in Table 3. A dash in the DAgger column indicates that its use is not established by the cited paper.

Method	DAgger	Goal Spec.	NE↓	OS↑	SR↑	SPL↑
Seq2Seq (Krantz et al., 2020)	Yes	Lang.	7.77	37.0	25.0	22.0
CMA (Krantz et al., 2020)	Yes	Lang.	7.37	40.0	32.0	30.0
HPN+DN (Krantz et al., 2021)	–	Lang.	6.31	40.0	36.0	34.0
R2R-CMTP (Chen et al., 2021)	Yes	Lang.	7.90	38.0	26.4	22.7
LAW (Raychaudhuri et al., 2021)	Yes	Lang.	6.83	44.0	35.0	31.0
CM2 (Georgakis et al., 2022)	No	Lang.	7.02	41.5	34.3	27.6
WS-MGMap (Chen et al., 2022)	Yes	Lang.	6.28	47.6	38.9	34.3
GridMM (Wang et al., 2023a)	Yes	Lang.	5.11	61.0	49.0	41.0
ETPNav (An et al., 2025)	Yes	Lang.	4.71	65.0	57.0	49.0
ScaleVLN (Wang et al., 2023b)	Yes	Lang.	4.80	–	55.0	51.0
InstructNav (Long et al., 2024)	No	Lang.	6.89	–	31.0	24.0
NaVid (Zhang et al., 2024b)	Yes	Lang.	5.47	49.1	37.4	35.9
MapNav (Zhang et al., 2025b)	Yes	Lang.	4.93	53.0	39.7	37.2
NaVILA (Cheng et al., 2025)	No	Lang.	5.22	62.5	54.0	49.0
Uni-NaVid (Zhang et al., 2025a)	Yes	Lang.	5.58	53.3	47.0	42.7
StreamVLN (Wei et al., 2026)	Yes	Lang.	4.98	64.2	56.9	51.9
DualVLN (released checkpoint) (Wei et al., 2025)	–	Lang.	5.07	59.9	54.6	50.7
DualVLN (full-data, our eval) (Wei et al., 2025)	–	Lang.	**4.13**	**70.2**	63.9	**57.8**
VLN-MEM (ours)	No	Lang.+Pose/Img	5.24	**70.2**	**65.0**	56.5

empirical proxy for recognition. NE measures final goal distance, and SPL penalizes inefficient successful paths. Hence OS minus SR isolates "arrived but did not stop" failures, while RxR nDTW separately measures adherence to the demonstrated path.

Baselines. We compare against a mini-data reproduction of the planner without scene memory (*Base*, Table 3), which shares our exact data budget; against our evaluations of released DualVLN weights with an unspecified DAgger status, treated as external references under a different input protocol; and against prior VLN-CE methods (Krantz et al., 2021; 2020; Wang et al., 2023a; An et al., 2025; Wang et al., 2023b; Chen et al., 2021; Raychaudhuri et al., 2021; Georgakis et al., 2022; Chen et al., 2022; Long et al., 2024; Zhang et al., 2024b; 2025b; Cheng et al., 2025; Zhang et al., 2025a; Wei et al., 2026).

Model and training. Our backbone is DualVLN (Wei et al., 2025), a dual system that decouples a VLM planner from a diffusion controller: the planner is Qwen2.5-VL-7B-Instruct (Bai et al., 2025), and the controller is initialized from a DualVLN-pretrained checkpoint. The scene memory is built offline from reference trajectories (Section 3.1); the planner, the residual branches, and the depth head are then fine-tuned together with a language-modelling cross-entropy over action, pixel-goal, and STOP tokens; finally, the planner is frozen and the four `<TRAJ>` queries and the controller diffusion policy are trained with the original flow-matching objective (Wei et al., 2025; Lipman et al., 2023) over 32-waypoint trajectories. VLN-MEM and *Base* train on the same *mini-data* budget: planner fine-tuning uses only the two *125 cm* configurations of R2R and RxR (four of the eight full-data configurations), and controller adaptation draws from the same configurations at a 30% per-config sampling rate. The released DualVLN references use the full budget (the exact configuration list is in Appendix A.1). Monocular geometry uses CUT3R (Wang et al., 2025b), appearance rerank uses CLIP ViT-B/32 (Radford et al., 2021), and the planner is fine-tuned with DeepSpeed ZeRO-2 (Rajbhandari et al., 2020); full hyperparameters are in Appendix A.1. Evaluation inference and per-episode seeds are fixed for bit-for-bit reproducibility, and Table 3 steps are compared with exact paired McNemar tests (Section 4.3).

Goal specification. At test time the specified goal pose and view bind a node in the scene memory, and A^* supplies the next-hop geometry; memory construction and anchoring additionally use evaluation-split reference trajectories and known start poses. Because these inputs exceed the language-only setting, Tables 1 and 2 identify goal specification explicitly.

Table 2: RxR-CE val-unseen English (n=3669); columns and setup follow Table 1.

Method	DAgger	Goal Spec.	NE↓	OS↑	SR↑	SPL↑	nDTW↑
Seq2Seq (Krantz et al., 2020)	Yes	Lang.	12.10	–	13.9	11.9	30.8
CMA (Krantz et al., 2020)	Yes	Lang.	8.76	–	26.5	22.1	47.0
LAW (Raychaudhuri et al., 2021)	Yes	Lang.	10.90	–	8.0	8.0	38.0
ETPNav (An et al., 2025)	Yes	Lang.	5.64	–	54.7	44.8	61.9
NaVILA (Cheng et al., 2025)	No	Lang.	6.77	–	49.3	44.0	58.8
Uni-NaVid (Zhang et al., 2025a)	Yes	Lang.	6.24	–	48.7	40.9	–
StreamVLN (Wei et al., 2026)	Yes	Lang.	6.22	–	52.9	46.0	61.9
DualVLN (released checkpoint) (Wei et al., 2025)	–	Lang.	6.20	57.4	49.4	42.7	63.2
DualVLN (full-data, our eval) (Wei et al., 2025)	–	Lang.	**4.70**	**68.9**	59.7	**50.7**	**69.8**
VLN-MEM (ours)	No	Lang.+Pose/Img	6.09	65.7	**59.8**	49.7	55.8

4.1 Main results

R2R-CE results. As reported in Table 1, VLN-MEM achieves 65.0 SR and 56.5 SPL without DAgger. It exceeds the reported language-only StreamVLN result by 8.1 SR and 4.6 SPL, and reaches the same 70.2 OS as the full-data DualVLN reference while improving SR by 1.1 points. More importantly, under the controlled mini-data comparison in Table 3, the complete method improves the matched *Base* from 50.5 to 65.0 SR and from 42.9 to 56.5 SPL. The larger SR gain (+14.5) than OS gain (+5.5) shows that the method improves not only goal arrival but also termination after arrival. The corresponding +13.6 SPL demonstrates better success-weighted path efficiency, although SPL alone does not establish shorter trajectories.

RxR-CE val-unseen English results. Table 2 shows that the improvement transfers to longer and more diverse instructions: VLN-MEM obtains 59.8 SR and 49.7 SPL, improving over the reported language-only StreamVLN result by 6.9 SR and 3.7 SPL. Its SR is comparable to the full-data DualVLN reference (59.7), while its OS, SPL, and nDTW are lower (65.7 vs. 68.9, 49.7 vs. 50.7, and 55.8 vs. 69.8), despite using the smaller mini-data budget and no DAgger. Because the systems also differ in goal inputs, this comparison provides context rather than a same-input estimate of the scene memory's effect. Together, Tables 1 and 2 show consistent gains in successful arrival and stopping across both benchmarks; Table 3 next identifies which components produce these gains.

Qualitative interpretation. Figure 3 compares three episode-matched checkpoint replays. Memory-free trajectories continue into distant corridors and terminate 24.5–35.7 m from the goal, whereas VLN-MEM terminates within 0.2–2.7 m. Examples show the intended combination of goal-directed progress and termination inside the success region. They are selected single-run cases, not evidence of a population-level path-length reduction or of the contribution of any individual module.

4.2 Arrival and stopping analysis

Figure 4a decomposes every episode into three outcomes. The memory-free *Base* model reaches within 3 m of the goal in 64.7% of episodes but succeeds in only 50.5%: 260 episodes enter the success region and subsequently terminate outside it. With memory but without verification, the proportion that never reaches the goal decreases from 35.4% to 28.8%, while the arrival–stopping gap remains 14.5%. Verification reduces this gap to 5.2%. Thus, the results distinguish a goal-arrival bottleneck from a terminal-decision bottleneck.

Matching episodes one-to-one between the verified and unverified agents isolates the change in terminal outcomes: of 1839 episodes, verification converts 166 failures to successes and 14 successes to failures, a net +8.3 SR. Among the recovered episodes, 55 terminate through the DINOv2 appearance check and 111 through the bank-distance gate. Both branches contribute, although these counts do not isolate their independent effects or interactions.

Figure 4b identifies the corresponding precision trade-off. Because verification can truncate a trajectory before the policy STOP, VLN-MEM stops *earlier* and therefore less precisely than the unverified agent at tight radii: within 1 m it succeeds on 14.6% of episodes against 39.9%. Linear interpolation between sampled radii places the crossing near 2.2 m. The observed benefit concerns success at the benchmark's 3 m criterion, not sub-metre localization. At 2 m the verified model is still slightly behind (47.6% versus 50.5%), but by 2.5 m it is ahead (58.7% versus 53.8%). The full distance curve therefore reveals an operating-point change hidden by a single 3 m score.

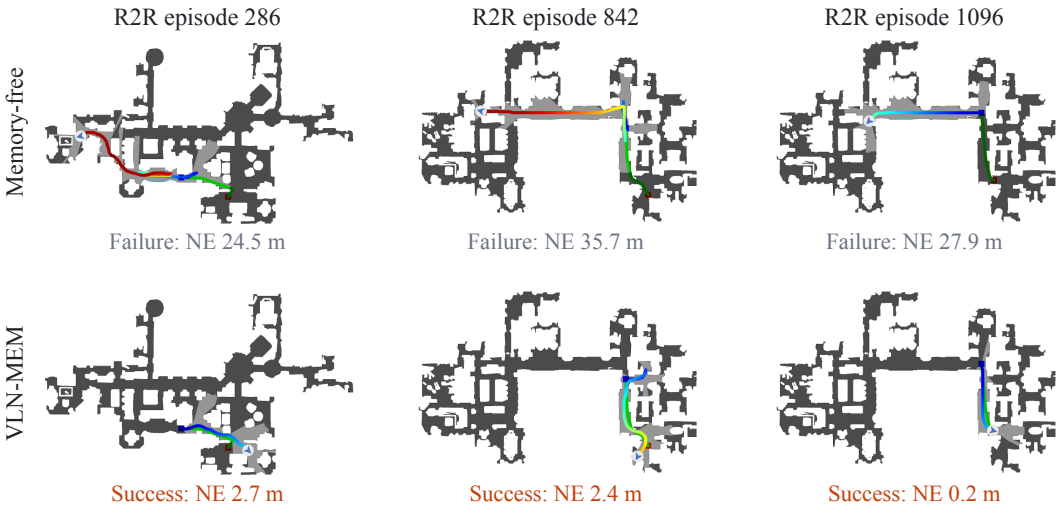


Figure 3: Qualitative trajectory comparison on three R2R-CE val-unseen episodes from single-run, episode-matched checkpoint replays. **Top:** the memory-free checkpoint terminates 24.5–35.7 m from the goal. **Bottom:** VLN-MEM retrieves the goal viewpoint and stops within 0.2–2.7 m in these examples. Blue/red squares indicate the start/goal; triangular markers indicate the final agent pose.

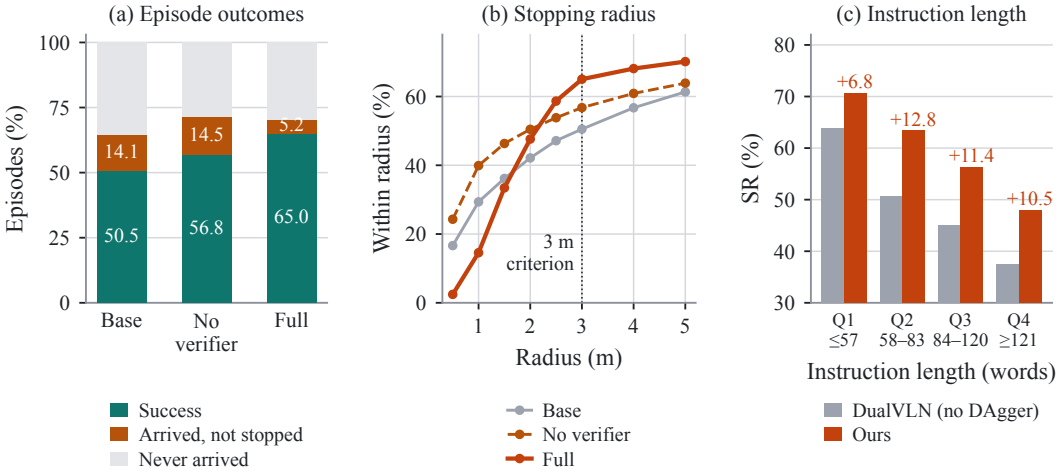


Figure 4: Arrival and stopping analysis (R2R-CE val-unseen, n=1839, except (c)). **(a)** The complete pre-verifier system reduces failures to reach the goal; stop tests reduce failures to stop. **(b)** Verification improves success at 3 m but not below ≈2.2 m. **(c)** RxR-CE val-unseen English gains by instruction-word-count quartile: Q1–Q4 span 10–57, 58–83, 84–120, and 121–505 words, respectively.

Finally, Figure 4c stratifies the episode-matched RxR-CE val-unseen English sample by instruction word count: Q1–Q4 contain 10–57, 58–83, 84–120, and 121–505 words, respectively. VLN-MEM differs from the language-only released DualVLN reference by +6.8, +12.8, +11.4, and +10.5 SR from the shortest to longest quartile. The difference is positive across all four ranges and larger in Q2–Q4 than Q1, but not monotonic. Since the systems also differ in goal inputs, this is descriptive rather than an isolated estimate of memory's effect.

4.3 Ablations

Table 3 explains the gains in the main results by adding each component to the same navigation backbone. Retrieval first improves goal arrival, raising OS from 64.7 to 69.8 and SR from 50.5 to 55.4. Confidence-gated goal binding adds a further 0.6 SR, while the dense appearance verifier and bank-distance gate contribute 4.5 SR each, ultimately reaching 65.0 SR and 56.5 SPL. The first two components therefore primarily improve goal-directed progress, whereas the two verification branches convert arrival into correct termination. This division is consistent with the OS–SR decomposition

in Figure 4a and accounts for the main R2R-CE improvement. Because both verifiers require the retrieved node $n^\star$, a verifiers-only *Base* is undefined, and the cumulative rows should not be interpreted as interaction-free marginal effects.

Table 3: Component ablation of VLN-MEM on R2R-CE val-unseen (n=1839). Each row adds one component cumulatively; *Base* has no memory bank, retrieval, or stop verifiers. The pre-verifier row is this incremental run and is distinct from the paired no-verifier checkpoint used in Figure 4a.

Configuration	NE↓	OS↑	SR↑	SPL↑
Base (parallel visual fusion, no retrieval)	5.85	64.7	50.5	42.9
+ Retrieval-augmented memory (goal + geometry)	5.47	69.8	55.4	44.3
+ Confidence-gated goal binding	5.36	69.6	56.0	45.5
+ Dense DINOv2 stop verifier	5.25	69.0	60.5	51.8
+ Bank-distance joint gate (full VLN-MEM)	5.24	**70.2**	**65.0**	**56.5**

Appendix Figure 5 plots the ablation alongside the VLM fine-tuning budget sweep used to select the 8k-step checkpoint.

Determinism and paired tests. Fixed inference and episode seeds make each checkpoint deterministic. The verifiers change 166 failures to successes and 14 successes to failures (exact two-sided McNemar $p < 10^{-33}$). This paired interpretation applies only to verifier variants, which truncate a shared rollout prefix. The reported checkpoint comparisons do not include repeated training seeds and therefore do not estimate variation across independent training runs.

5 Discussion and Limitations

Scene preparation and goal specification. VLN-MEM targets scenes whose memory is built offline from reference trajectories, not first-visit VLN-CE. Evaluation-split reference trajectories and known start poses construct the frozen scene memory; goal binding uses a specified pose and view. Language-only goal resolution, online scene memory construction, and drift handling remain necessary for first-visit deployment. Reference coverage is also consequential: missing connectors can disconnect the graph, and uncovered goals require nearest-node fallback. Coverage-controlled ablations are needed to distinguish these effects from the benefit of persistent storage.

Dynamic scenes and first visit. VLN-MEM assumes the scene is static between construction and evaluation, and it does not navigate scenes it has not seen: first-visit and open-world deployment fall back to language-only navigation, which is outside our contribution. When a scene changes, the frozen memory degrades rather than collapses. A goal whose stored appearance no longer matches blanks both retrieved signals at the 0.90 binding threshold, returning control to the planner; the next-hop geometry is a soft residual cue, and the agent still reacts to local obstacles from the current observation; and the oracle/noisy/empty schedule has already exposed the planner to blank or degraded retrieval. Active recovery, online memory update, and change detection are untested and left to future work.

Robustness and stopping precision. The 80/10/10 oracle/noisy/empty schedule trains with degraded retrieval. The single-run sweep of the VLM fine-tuning budget suggests saturation around 8k steps but not multi-seed convergence (Figure 5b). Verification improves 3 m success but reduces precision inside ≈2.2 m (Figure 4b). A fallback path with blanked scene memory inputs covers low retrieval confidence. Fixed verification thresholds may likewise require recalibration for new cameras and success radii.

6 Conclusion

This paper proposes VLN-MEM, which supplies the two missing conditions, reachability and recognition, from one persistent scene memory: retrieval returns next-hop geometry for the reachability and a stored goal view for the recognition, zero-initialized residual fusion feeds the reachability into the planner, and training-free stop verification uses the goal view to confirm arrival. VLN-MEM reaches 65.0% and 59.8% SR on R2R-CE val-unseen and RxR-CE val-unseen English, respectively, outperforms a training-budget-matched backbone. An ablation separates the two effects: reachability closes the reaching gap and recognition closes the stopping gap.

Reproducibility statement

Sections 3.1 and 4 specify the data, models, hyperparameters, memory construction, retrieval, and evaluation; Appendix A.1 gives implementation details and stop thresholds. Section 4 discloses scene-preparation and goal-input assumptions. We report deterministic paired verifier analysis and will release code.

AI use statement

Generative AI assisted method implementation, debugging, prose and figure editing, literature verification, manuscript consistency review, and checks of result interpretation, but not research ideas, experimental design, or data generation. The authors tested AI-assisted code and manually checked reported numbers and citations against experiment exports and primary sources. We take responsibility for the final content.

References

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*, 47(7):5130–5145, 2025. doi: 10.1109/TPAMI.2024.3386695. URL https://arxiv.org/abs/2304.03047.

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 (CVPR)*, pp. 3674–3683, 2018. doi: 10.1109/CVPR.2018.00387. URL https://arxiv.org/abs/1711.07280.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. URL https://arxiv.org/abs/2502.13923.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *International Conference on 3D Vision (3DV)*, pp. 667–676, 2017. doi: 10.1109/3DV.2017.00081. URL https://arxiv.org/abs/1709.06158.

Kevin Chen, Junshen K. Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11276–11286, 2021. doi: 10.1109/CVPR46437.2021.01112. URL https://arxiv.org/abs/2012.05292.

Peihao Chen, Dongyu Ji, Kunyang Lin, Runhao Zeng, Thomas H. Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 38149–38161, 2022. URL https://arxiv.org/abs/2210.07506.

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 (RSS)*, 2025. URL https://arxiv.org/abs/2412.04453.

Xin Ding, Jianyu Wei, Yifan Yang, Shiqi Jiang, Qianxi Zhang, Hao Wu, Fucheng Jia, Liang Mi, Yuxuan Yan, Weijun Wang, Yunxin Liu, Zhibo Chen, and Ting Cao. Adanav: Adaptive reasoning with uncertainty for vision-language navigation. *arXiv preprint arXiv:2509.24387*, 2025. URL https://arxiv.org/abs/2509.24387.

Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15460–15470, 2022. doi: 10.1109/CVPR52688.2022.01502. URL https://arxiv.org/abs/2203.05137.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: 10.1109/TSSC.1968.300136. URL https://ieeexplore.ieee.org/document/4082128.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, pp. 104–120, 2020. doi: 10.1007/978-3-030-58604-1_7. URL https://arxiv.org/abs/2004.02857.

Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15162–15171, 2021. doi: 10.1109/ICCV48922.2021.01488. URL https://arxiv.org/abs/2110.02207.

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)*, pp. 4392–4412, 2020. doi: 10.18653/v1/2020.emnlp-main.356. URL https://aclanthology.org/2020.emnlp-main.356/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 9459–9474, 2020. URL https://arxiv.org/abs/2005.11401.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.02747.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. In *Conference on Robot Learning (CoRL)*, 2024. URL https://arxiv.org/abs/2406.04882.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://arxiv.org/abs/2304.07193.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1–16, 2020. doi: 10.1109/SC41405.2020.00024. URL https://arxiv.org/abs/1910.02054.

Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X. Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4018–4028, 2021. doi: 10.18653/v1/2021.emnlp-main.328. URL https://aclanthology.org/2021.emnlp-main.328/.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410. URL https://arxiv.org/abs/1908.10084.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9339–9347, 2019. doi: 10.1109/ICCV.2019.00943. URL https://arxiv.org/abs/1904.01201.

Liuyi Wang, Xinyuan Xia, Hui Zhao, Hanqing Wang, Tai Wang, Yilun Chen, Chengju Liu, Qijun Chen, and Jiangmiao Pang. Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. *arXiv preprint arXiv:2507.13019*, 2025a. URL https://arxiv.org/abs/2507.13019.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b. URL https://arxiv.org/abs/2501.12387. Oral.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, 2024. doi: 10.1109/CVPR52733.2024.01956. URL https://arxiv.org/abs/2312.14132.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 5776–5788, 2020. URL https://arxiv.org/abs/2002.10957.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15625–15636, 2023a. doi: 10.1109/ICCV51070.2023.01432. URL https://arxiv.org/abs/2307.12907.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12009–12020, 2023b. doi: 10.1109/ICCV51070.2023.01105. URL https://arxiv.org/abs/2307.15644.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. *arXiv preprint arXiv:2512.08186*, 2025. URL https://arxiv.org/abs/2512.08186.

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 *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://arxiv.org/abs/2507.05240.

Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. *arXiv preprint arXiv:2403.15378*, 2024a. URL https://arxiv.org/abs/2403.15378.

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 (RSS)*, 2024b. URL https://arxiv.org/abs/2402.15852.

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. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025a. URL https://arxiv.org/abs/2412.06224.

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. *arXiv preprint arXiv:2502.13451*, 2025b. URL https://arxiv.org/abs/2502.13451.

A Appendix

A.1 Full implementation details

Mini-data budget. Planner fine-tuning (SFT) uses only the two *125 cm* camera configurations of R2R and RxR (4 dataset configs: R2R/RxR $\times$ $\{(125\,\mathrm{cm}, 0°, 30°), (125\,\mathrm{cm}, 0°, 45°)\}$), *omitting* the four 60 cm configurations used by the full-data recipe (which additionally uses $(60\,\mathrm{cm}, 15°, 30°)$ and $(60\,\mathrm{cm}, 15°, 45°)$ for each dataset and spans all 8). Each triplet denotes camera height, level-view pitch, and downward-view pitch. Controller adaptation draws from the 125 cm R2R and RxR configs at a 30% per-config sampling rate. In aggregate this restricts training to the eye-level (125 cm) camera regime that matches the evaluation viewpoint and uses roughly half of the camera-configuration coverage of the full DualVLN recipe; the released DualVLN references are trained on the full budget.

World-frame anchoring. Each predicted CUT3R pose/point-map T_t^{cut3r} (in CUT3R's own arbitrary local frame) is lifted into a single, fixed Habitat-world (Savva et al., 2019) scene frame via

$$T_t^{\text{world}} \;=\; M\, T_t^{\text{cut3r}}\, C_4, \qquad M \;=\; T_{\text{start}}^{\text{world}}\, C_4 \left(T_0^{\text{cut3r}}\right)^{-1}, \qquad C_4 = \mathrm{diag}(1, -1, -1, 1),$$

where C_4 is the OpenCV$\rightarrow$OpenGL/Habitat axis flip, T_0^{cut3r} is CUT3R's arbitrary-frame estimate at the first step, and $T_{\text{start}}^{\text{world}}$ is the episode's known starting pose, injected once per reference episode.

Offline scene memory construction (Section 3.1). We group all reference episodes in a split by scene, reconstruct their pre-rendered observations, and merge them into one scene memory per scene. R2R val-unseen scene memories use the corresponding pre-rendered reference episodes; RxR val-unseen scene memories use 1,223 deduplicated reference trajectories covering all 11 scenes. Nodes merge at a 0.5 m pose threshold and 0.85 CLIP cosine threshold. Temporal edges are added within episodes, followed by cross-episode loop closures, spatial bridges, and a connectivity pass. Evaluation loads the resulting scene `.pkl`; there is no online write-back.

Oracle/noisy/empty training schedule (Section 3.1). An 80/10/10 sampling schedule mixes the deterministic oracle next-hop geometry with (i) a synthetically perturbed next-hop point map and (ii) an empty slot, so the planner does not learn to depend unconditionally on a perfect retrieval.

Fusion architecture (Section 3.2). Geo_ϕ is a single zero-initialized linear projection per token, applied because the point map's rendered-image tokens are already aligned 1:1 with the current frame's 14×14 token grid. Goal_ψ is a zero-initialized cross-attention query resampler that compresses the goal view's ViT tokens to that same grid shape. The scalar gains $w_{\text{geo}}, w_{\text{goal}}$ are fixed constants (both 1.0); a learnable per-branch scalar was no more effective and less stable early in planner fine-tuning. An optional third residual (frozen DINOv2 (Oquab et al., 2024) features, similarly resampled) is available as an ablation but disabled by default. Next-hop RGB is not fused as a fourth residual branch. A dense log-depth auxiliary head supervises the fused representation during planner fine-tuning only. The DINOv2 stop verification (Section 3.3) separately compares the goal node's goal view with the current frame.

Scene memory and retrieval. Monocular geometry uses CUT3R (Wang et al., 2025b) (`cut3r_512_dpt_4_64`, 512×512 input, bf16 autocast). Scene memory nodes store OpenAI CLIP ViT-B/32 (Radford et al., 2021) image embeddings (512-d), which also drive the appearance rerank. The position gate uses a per-scene k-d tree with radius r=2.0 m for goal binding and 5.0 m for the A^* start anchor. An empty goal-radius gate falls back to the nearest node. Goal binding then applies a 0.90 CLIP similarity confidence threshold; below it, the goal view and next-hop geometry are both blanked. If binding succeeds but A^* finds no graph path, the goal view is retained and only the next-hop geometry slot is blanked.

Instruction-side retrieval (unused in reported results). An alternative retrieval configuration, explored during development but not used in any table or ablation in this paper (Section 3.1), additionally mixes instruction-text similarity into the rerank score using a Long-CLIP text encoder (Zhang et al., 2024a) and an all-MiniLM-L6-v2 sentence encoder (Wang et al., 2020; Reimers & Gurevych, 2019): $\text{score}(v) = \alpha \cdot \text{sim}_{\text{img}}(f_v, f_g^{\text{query}}) + (1-\alpha) \cdot \text{sim}_{\text{text}}(t_v, t_{\text{instr}})$, with $\alpha = 0.7$ (image-weighted) in our early trials, where t_v is a text description of node v (when available) and t_{instr} is the episode instruction. We mention it here only for completeness and reproducibility of code paths present in the released implementation; the main results use appearance-only rerank (Section 3.1).

Inference-time stop verification (Section 3.3). Both branches use frozen DINOv2 ViT-B/14 (Oquab et al., 2024) CLS-token cosine similarity between the current frame and the goal view. On R2R, the main branch uses similarity 0.75, one-frame confirmation, and no minimum step. Its joint appearance and distance branch uses similarity 0.40, distance 1.75 m, four-frame confirmation, and minimum step 40. On RxR, the joint branch uses similarity 0.30, distance 1.75 m, six-frame confirmation, and no minimum step. The final decision is the strict OR of the planner's stop signal, the temporally gated main branch, and the temporally gated joint branch.

Metric aggregation. SR and OS are means of binary episode indicators. SPL treats non-finite per-episode values as zero before averaging; NE is averaged over finite values only. This convention yields the reported RxR aggregate of 49.73 SPL rather than the biased mean obtained by silently dropping undefined SPL entries.

Optimization. The planner (Qwen2.5-VL-7B) is fine-tuned with DeepSpeed ZeRO-2 (Rajbhandari et al., 2020) on 8 GPUs at a global batch of 128 (LR 2×10^{-5}, vision-tower LR 5×10^{-6}, cosine decay to zero, one epoch). The controller is the DualVLN diffusion policy (hidden 384, 12 layers, 6 heads, latent projected $3584\to768$), trained with flow matching at AdamW LR 1×10^{-4}, cosine decay to a floor of 1×10^{-5}, global batch 128, for 3 epochs (or an equivalent fixed step budget on the mini-data profile). Write-back into the ViT's token tensor (Section 3.2) uses an out-of-place `index_copy` rather than in-place indexing, which is required for correct autograd behavior under gradient checkpointing and ZeRO.

A.2 VLM FINE-TUNING STEP SWEEP

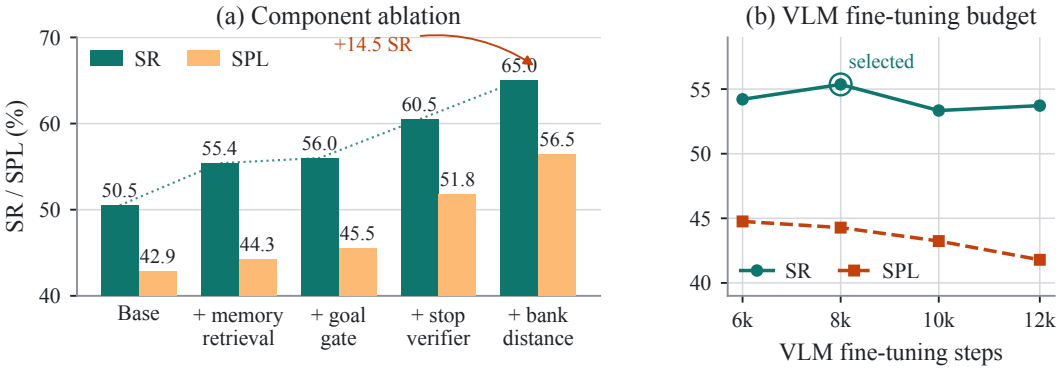


Figure 5: Ablation and VLM fine-tuning budget on R2R-CE val-unseen. **(a)** Cumulative additions from Table 3. **(b)** Pre-verification VLM fine-tuning sweep with the controller frozen. SR peaks at 8k steps, which determines the selected checkpoint.

Figure 5(b) sweeps the VLM fine-tuning budget while holding the controller checkpoint fixed. To isolate grounding, the sweep evaluates the VLM planner with scene memory *before* applying the stop checks, using the "+ scene memory" configuration from Table 3. Thus, the plotted SR peaks at 55.4 after 8k steps, rather than at the complete system's 65.0. Performance decreases slightly from 8k to 12k, but the single-run spread is insufficient to establish overfitting. We therefore select the 8k VLM checkpoint before applying the training-free stop checks.