RETHINKING REASONING CONTEXT FOR VISION-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning in vision-language navigation (VLN) can serve as an action-generation trace, auxiliary supervision, or policy context. Policy context is particularly suited to sequential navigation, making task-level guidance explicit at execution time and reusable across action decisions. Yet what this context should contain remains underexplored. Scene description, progress assessment, and subgoal planning address complementary navigation needs, suggesting that combining them should provide better guidance. Surprisingly, we find that more comprehensive reasoning context can lead to worse navigation. Through component-wise comparisons on matched navigation trajectories and action annotations, we find that subgoal planning alone outperforms the full reasoning context. However, the best way to complete a subgoal can depend on what follows it. We therefore propose Context-Think, which uses *final-goal replanning* to regenerate a language plan for the remaining route from the agent's current state to the original destination. Each updated plan explicitly represents the remaining subgoals, allowing future route requirements to inform current action selection while adapting to execution progress. Context-Think improves navigation success and path efficiency over both subgoal planning and full reasoning context on the widely used R2R-CE and RxR-CE benchmarks. Our findings challenge completeness as a design principle for policy context: include fewer reasoning components, but plan all the way to the final goal.

1 INTRODUCTION

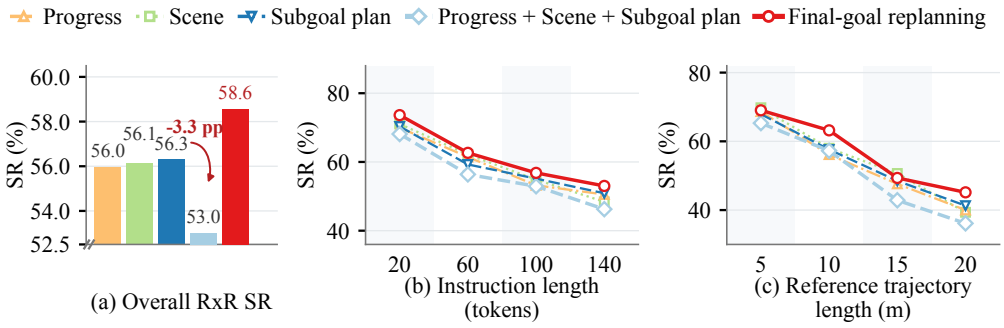

Fig. 1: Comparison of reasoning contexts on RxR-CE. Combining multiple reasoning components does not necessarily improve navigation, whereas final-goal replanning provides the most effective policy context, particularly for long-horizon instructions and trajectories.

Vision-language navigation (VLN) requires an agent to follow natural-language instructions through unseen environments by grounding language in visual observations (Anderson et al., 2018; Ku et al., 2020; Krantz et al., 2020). Success in VLN requires more than recognizing objects or matching instructions to individual observations: an agent must maintain a coherent understanding of its progress, anticipate future navigation requirements, and make decisions whose consequences may unfold over long horizons.

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.

Language reasoning has been used as an *action-generation trace* (Lin et al., 2025), as *auxiliary supervision* (Wang et al., 2026a), or as persistent *policy context* during execution (Wang et al., 2026b; Guo et al., 2026). We focus on the last paradigm, in which generated reasoning is retained across multiple actions alongside the original instruction and observation history. This context may describe the current scene, assess progress through the instruction, or formulate a navigation plan. Although these components provide complementary information, it remains unclear whether combining more reasoning actually provides better guidance.

We systematically study this question by independently constructing matched *scene*, *progress*, and *subgoal-plan* supervision, three recurring forms of reasoning context in prior VLN work, at the same navigation states and comparing them under matched trajectories, action annotations, and training conditions. **Surprisingly, more comprehensive reasoning context is not better.** The full scene–progress–plan context underperforms simpler single-component alternatives, while *subgoal planning* provides the strongest success and path efficiency on both R2R-CE and RxR-CE. We term this phenomenon **Reasoning-Context Degradation**. The result suggests that useful policy context need not reconstruct the entire navigation state: scene and progress largely describe what the policy can already infer from observations and instructions, whereas planning directly constrains what the agent should accomplish next.

Inspired by this finding, we propose **Context-Think**, a policy-context framework that preserves the focused guidance of subgoal planning while extending its planning horizon to the final destination. Conventional subgoal planning represents only the next navigation phase. Yet the best way to complete a local subgoal can depend on what follows: actions that appear equally suitable for the immediate objective may lead to very different states for subsequent turns, landmarks, or the final stopping condition. Context-Think addresses this limitation through *final-goal replanning*. At each reasoning update, *final-goal replanning* regenerates a state-conditioned language plan covering the remaining route from the agent's current state to the original destination. The plan removes completed stages while preserving upcoming subgoals, their order, and the final stopping objective, allowing downstream constraints to inform current action selection.

To learn such plans, we transform the remaining portions of training trajectories into state-conditioned remaining-route supervision and jointly train the policy to generate these plans and predict actions conditioned on them. At inference, no future trajectory information is available: the policy predicts the remaining plan from the original instruction and observed history, retains it across subsequent actions, and regenerates it as navigation unfolds. Experiments on R2R-CE and RxR-CE (Fig. 1) show that final-goal replanning consistently improves success and path efficiency over both local subgoal planning and full reasoning context, with its advantage becoming more pronounced on longer navigation tasks. Together, our results suggest a simple principle for reasoning context: *reason less comprehensively, but plan farther ahead.*

Our main contributions are:

- We first systematically investigate reasoning-context composition in VLN using independently generated, state-aligned scene, progress, and planning supervision. We uncover **Reasoning-Context Degradation**: combining complementary reasoning components can hurt navigation, while subgoal planning provides the strongest single-component context.
- We introduce **Context-Think**, a policy-context framework based on *final-goal replanning*. By explicitly representing the complete unfinished route, Context-Think allows downstream subgoals and the final destination to guide current action selection.
- Experiments on R2R-CE and RxR-CE show that Context-Think improves navigation success and efficiency over both full reasoning context and local subgoal planning, with larger gains on longer instructions and trajectories.

2 Related Work

Roles of reasoning in VLN. Language-based reasoning in VLN mainly serves three roles: direct action generation, auxiliary supervision, and explicit intermediate reasoning (Zhou et al., 2024b; Long et al., 2024b; Zhu et al., 2020). NavGPT performs sequential decision making through language reasoning (Zhou et al., 2024a), while NavCoT uses a more structured chain that anticipates observations, matches visual evidence, and predicts actions (Lin et al., 2025). Aux-Think instead

uses reasoning as training supervision while predicting actions directly at inference (Wang et al., 2026a); FantasyVLN extends this idea to textual, visual, and multimodal reasoning (Zuo et al., 2026). A third line explicitly predicts intermediate navigation information: Progress-Think estimates semantic instruction progress (Wang et al., 2026b), whereas AwareVLN jointly reasons about scene, progress, and upcoming navigation decisions (Guo et al., 2026). We focus on this explicit intermediate-reasoning paradigm and study which reasoning information is most useful for navigation.

Reasoning context composition. Explicit navigation reasoning commonly contains three components: *scene*, *progress*, and *plan*. Scene reasoning describes navigation-relevant environments, landmarks, and spatial relations (Pan et al., 2024; Zhang et al., 2025c); progress reasoning tracks completed and remaining instruction content (Ma et al., 2019; Wang et al., 2026b); and plan reasoning specifies the next navigation step or subgoal (Zhou et al., 2024a; Guo et al., 2026; Zhang et al., 2024c). AwareVLN combines all three in a unified representation (Guo et al., 2026), while other methods emphasize individual components. Because these components are usually studied under different models and training settings, their individual contributions remain unclear. We therefore compare scene, progress, and plan under the same policy, data, and supervision setup.

Planning horizon. Planning-based VLN methods commonly decompose long instructions into local subgoals. NavGPT updates subgoals from observations and navigation history (Zhou et al., 2024a); AwareVLN plans for the next navigation phase (Guo et al., 2026); and MapGPT, InstructNav, and related hierarchical methods organize navigation around intermediate nodes, subgoals, or subinstructions (Chen et al., 2024; Long et al., 2024a; Zhang et al., 2025b; Huang et al., 2026; Qiao et al., 2024; Chen et al., 2025). Such short-horizon planning simplifies each decision, but can omit downstream route constraints: locally similar choices may have different consequences for later stages. This motivates a natural question: *should planning cover only the next subgoal, or the remaining route to the final goal?* We study *final-goal replanning*, which regenerates the remaining plan from the current state to the original destination, and compare it directly with next-subgoal planning in long-horizon navigation.

3 METHOD

We study how language-based reasoning should be represented and retained as policy context for vision-language navigation, with the aim of answering two questions: (1) *What reasoning content provides the most effective guidance for action selection?* and (2) *How far ahead should the policy reason to support long-horizon navigation?* We first establish a unified policy-context formulation and conduct a controlled comparison using matched supervision for representative reasoning components. Motivated by our findings, we introduce Context-Think, which retains planning as policy context while extending its horizon from the immediate subgoal to the final destination.

3.1 PROBLEM FORMULATION

Given a route instruction I, the policy interleaves navigation actions with textual reasoning updates. At policy-query step t, it conditions on the observation–action history H_t and the latest reasoning context C_t:

$$Y_t \sim p_\theta(\cdot \mid I, H_t, C_t), \qquad Y_t \in \mathcal{A} \cup \mathcal{V}^*,$$
$$C_{t+1} = \begin{cases} Y_t, & t \in \mathcal{B}, \\ C_t, & t \notin \mathcal{B}, \end{cases} \tag{1}$$

where $\mathcal{A}$ is the action space, $\mathcal{V}^*$ is the space of textual reasoning sequences, and $\mathcal{B}$ denotes the reasoning positions, determined by the policy outputs during deployment. Each reasoning update replaces the previous context and persists until the next update, while the original instruction and interaction history remain available throughout navigation.

All variants use the same learning procedure. At reasoning positions, the policy is supervised to generate the target context; at action positions, it is supervised to predict expert actions conditioned on the most recently provided context. We jointly optimize the two objectives as

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{action}} + \lambda_{\text{reason}} \mathcal{L}_{\text{reason}}. \tag{2}$$

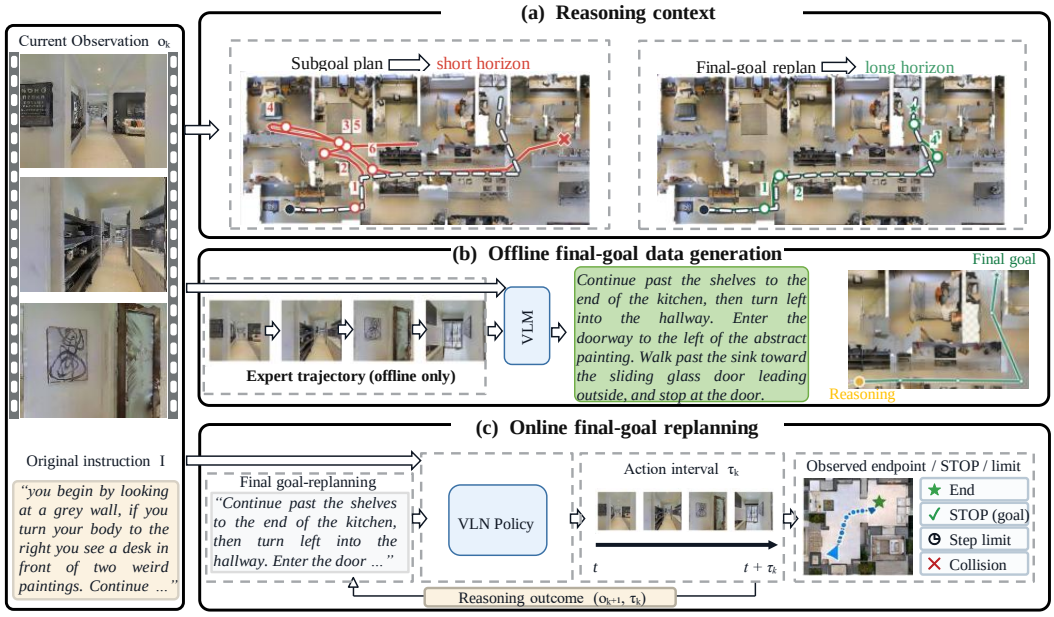

Fig. 2: Overview of Context-Think. (a) Reasoning context comparison between subgoal planning and final-goal replanning. While subgoal planning focuses on the next navigation phase, final-goal replanning maintains a plan from the current state to the final destination. (b) Offline final-goal data generation. Expert trajectories are transformed into state-conditioned remaining-route plans using a vision-language teacher. (c) Online final-goal replanning. The VLN policy updates the remaining-route plan from the original instruction and observed history, and uses the generated plan as persistent reasoning context for action prediction.

This common setup allows us to change only the content of reasoning context in the following study, and later its planning horizon.

3.2 Reasoning-Context Supervision Construction

Reasoning used as VLN policy context commonly captures three aspects of navigation: *scene description*, *progress assessment*, and *subgoal planning*. They respectively describe what the agent currently observes, what part of the instructed task has been completed, and what should be achieved next.

To compare these roles without inheriting dependencies from an existing multi-component reasoning trace, we independently regenerate all three components at the same trajectory-aligned reasoning positions. For each expert trajectory and update position, a fixed vision-language teacher receives the original instruction, the marked current state, and trajectory information available for offline annotation. Only the requested reasoning role differs.

The *scene* prompt asks the teacher to describe the current navigation-relevant environment, including useful landmarks, room type, and spatial relations, while excluding task progress and future actions. The *progress* prompt asks which part of the original instruction has been completed and the agent's current task stage, while excluding future planning. The *subgoal-plan* prompt asks for one concise high-level objective for the next navigation phase, while excluding completed progress and the complete remaining route. Exact prompts and generation details are provided in Appendix B.1.

We denote the k-th reasoning-update position by $b_k \in \mathcal{B}$. Let the independently generated targets be $S^*_{b_k}$, $P^*_{b_k}$, and $L^*_{b_k}$. The single-component conditions use them individually, while the full-context condition concatenates the same matched targets:

$$C^{\text{full}}_{b_k} = [S^*_{b_k}; P^*_{b_k}; L^*_{b_k}]. \tag{3}$$

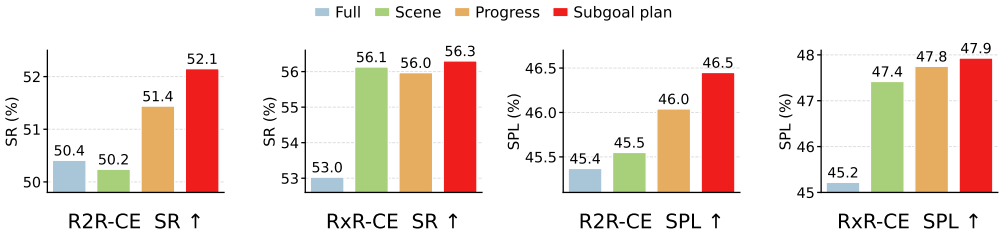

Fig. 3: **Reasoning-context composition.** Combining all reasoning components does not improve navigation; among single-component contexts, subgoal planning achieves the strongest SR and SPL.

Navigation trajectories, action labels, reasoning-update positions, and optimization settings are kept identical across all conditions. Thus, the controlled comparison changes the composition of reasoning context rather than the underlying navigation supervision.

3.3 What should reasoning context contain?

Scene, progress, and planning appear complementary: scene provides perceptual grounding, progress identifies the agent's position within the instructed task, and planning specifies future intent. A natural expectation is therefore that combining all three should provide the strongest reasoning context. Surprisingly, our controlled comparison shows otherwise.

Finding 1: more complete context is not necessarily better. Fig. 3 compares the four reasoning contexts under matched navigation trajectories, action supervision, reasoning-update positions, and training settings, with only the retained reasoning content changed. Surprisingly, the full scene–progress–plan context does not achieve the strongest navigation performance. On R2R-CE, progress and planning both improve SR and SPL over full context, while scene remains comparable in SR and slightly improves SPL. On RxR-CE, all three single-component contexts outperform full context in both SR and SPL.

We refer to this phenomenon as *reasoning-context degradation*: adding complementary reasoning information does not necessarily make persistent policy context more useful.

A plausible explanation is that descriptive completeness differs from decision usefulness. Scene and progress largely summarize information already recoverable from the visual history and original instruction. Moreover, both characterize the state when reasoning is generated and can become stale as execution proceeds, introducing redundant or temporally mismatched information when retained across subsequent decisions.

Finding 2: planning is the strongest single component. Among the individual components, subgoal planning achieves the highest SR and SPL on both benchmarks. The distinction lies in how directly each component constrains future behavior: scene describes *what is around the agent*, progress describes *what has been completed*, whereas planning specifies *what should be achieved next*. Planning therefore provides the most action-oriented representation for subsequent decisions.

These findings suggest that persistent reasoning context need not be comprehensive; it should emphasize information that organizes future behavior. They also expose a remaining question. The strongest component above plans only for the next navigation phase. *If planning is the useful context, how far should it plan?*

3.4 Context-Think: Final-Goal Replanning

From local subgoals to the final goal. Subgoal planning is effective because it compresses a long instruction into a simple and actionable near-term objective. However, this compression also removes downstream route constraints. As illustrated in Fig. 2(a), several locally plausible actions may accomplish the same immediate subgoal while leading to different states for later turns, land-

mark transitions, or the final stopping condition. These future constraints can already influence how the current subgoal should be completed. The limitation is therefore not that subgoal planning is incorrect, but that its planning horizon can be insufficient.

Final-goal replanning. At each reasoning update, Context-Think generates a state-conditioned language plan from the agent's current state to the original destination. The plan removes completed stages while retaining unresolved subgoals, their ordering, relevant landmarks, and the final stopping condition. It thus maintains the concise and action-oriented nature of subgoal planning while allowing future route requirements to inform current decisions.

To construct supervision, a fixed vision-language teacher receives the original instruction, the demonstrated trajectory prefix up to the current state, and the remaining expert trajectory. It then rewrites the instruction as a plan covering only the unfinished route:

$$G^*_{b_k} = \mathcal{T}_{\text{goal}}\big(I,\, O^*_{0:b_k}, A^*_{0:b_k},\, O^*_{b_k:e}, A^*_{b_k:e}\big)\,, \tag{4}$$

where $\mathcal{T}_{\text{goal}}$ denotes the teacher, O^* and A^* denote expert observation and action sequences, respectively, and e denotes the trajectory endpoint. Completed steps are omitted, whereas the remaining route structure, critical landmarks, and final stopping condition are preserved. Future trajectory information is used solely to construct offline supervision and is unavailable during deployment.

Context-Think uses $G^*_{b_k}$ as both the reasoning target and the persistent context for subsequent action prediction. At inference, the policy generates the remaining-route plan solely from the original instruction and observed history, retains it across actions, and regenerates it as navigation progresses. Relative to subgoal planning, the policy architecture, input interface, training reasoning-update positions, execution procedure, and optimization settings remain unchanged. The central distinction is the planning horizon: Context-Think plans toward the final goal rather than only the next navigation phase.

4 Experiments

4.1 Experimental setup

Benchmarks. We evaluate on R2R-CE and RxR-CE, which adapt R2R (Anderson et al., 2018) and RxR (Ku et al., 2020) to continuous environments (Krantz et al., 2020; Savva et al., 2019). Both benchmarks use Matterport3D indoor scenes (Chang et al., 2017) and are evaluated on *val-unseen* environments. R2R-CE contains English route instructions, while RxR-CE contains longer and more detailed multilingual instructions, providing a complementary test of long-horizon navigation.

Training data. We train Context-Think on a mixture of R2R-CE, RxR-CE, R2RFollow, RxRFollow Guo et al. (2026), and human demonstrations Cheng et al. (2024), while using R2RFollow and RxRFollow for controlled studies with shared trajectory sources and training settings.

Evaluation metrics. Following prior VLN work (Cheng et al., 2024; Guo et al., 2026), we report Success Rate (SR) and Success weighted by Path Length (SPL) as the primary metrics. SR measures the fraction of episodes that stop within 3 m of the goal, while SPL additionally penalizes inefficient paths. We also report Navigation Error (NE) and Oracle Success (OS) on R2R-CE. For the controlled RxR-CE studies, we additionally report normalized Dynamic Time Warping (nDTW) to measure alignment with the reference trajectory.

4.2 Main results

We compare Context-Think with a broad range of state-of-the-art VLN methods on R2R-CE and RxR-CE, with the results reported in Table 1. Context-Think achieves the highest SR on both benchmarks. On R2R-CE, it obtains the lowest NE (3.72 m) and the highest OS (77.1%) and SR (66.5%), while achieving a competitive SPL of 57.0%. On the more challenging RxR-CE benchmark, Context-Think ranks first across all metrics, reaching an NE of 3.92 m, an SR of 70.3%, and an SPL of 57.8%. It also consistently outperforms AwareVLN, a representative reasoning-augmented method, by 1.1/1.9 SR/SPL points on R2R-CE and 2.7/1.7 points on RxR-CE.

We attribute these improvements to the persistent, final-goal-oriented planning context of Context-Think. By repeatedly reformulating the unfinished route from the current state, Context-Think

Table 1: Navigation performance on the val-unseen splits of R2R-CE (Anderson et al., 2018; Krantz et al., 2020) and RxR-CE (Ku et al., 2020). NE is in meters; other metrics are percentages. Bold marks the best value in each metric column.

Method	Publication	R2R-CE				RxR-CE		
		NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑
WS-MGMap (Chen et al., 2022)	NeurIPS'22	6.28	47.0	38.0	34.0	9.83	15.0	12.1
NaVid (Zhang et al., 2024b)	RSS'24	5.47	49.0	37.0	35.0	6.24	48.7	40.9
InstructNav (Long et al., 2024a)	CoRL'24	6.89	44.0	31.0	24.0	10.90	8.0	8.0
sim2real (Wang et al., 2024)	CoRL'24	5.95	55.8	44.9	30.4	8.79	25.5	18.1
AO-Planner (Chen et al., 2025)	AAAI'25	5.55	59.0	47.0	33.0	7.06	43.3	30.5
NaVid-4D (Liu et al., 2025)	ICRA'25	5.99	55.7	43.8	37.1	—	—	—
g3D-LF (Wang & Lee, 2025)	CVPR'25	5.70	59.5	47.2	34.6	—	—	—
MapNav (Zhang et al., 2025a)	ACL'25	4.93	53.0	39.7	37.2	7.62	32.6	27.7
Uni-NaVid (Zhang et al., 2024a)	RSS'25	5.58	53.5	47.0	42.7	6.77	49.3	44.0
NaVILA (Cheng et al., 2024)	RSS'25	5.22	62.5	54.0	49.0	6.77	49.3	44.0
monoVLN (Lu et al., 2025)	ICCV'25	4.61	62.4	54.8	44.4	8.29	31.8	26.8
NavMorph (Yao et al., 2025)	ICCV'25	5.75	56.9	47.9	33.2	6.67	30.8	22.8
Aux-Think (Wang et al., 2026a)	NeurIPS'25	5.49	62.9	55.7	48.7	6.24	52.2	40.2
Dynam3D (Wang et al., 2026c)	NeurIPS'25	5.34	62.1	52.9	45.7	—	—	—
NavFoM (4-view) (Zhang et al., 2026)	ICLR'26	4.61	72.1	61.7	55.3	4.74	64.4	56.2
DualVLN (Wei et al., 2026)	ICLR'26	4.05	70.7	64.3	**58.5**	4.58	61.4	51.8
StreamVLN (Wei et al., 2025)	ICRA'26	4.98	64.2	56.9	51.9	6.22	52.9	46.0
ActiveVLN (Zhang et al., 2025d)	ICRA'26	5.31	58.3	50.1	43.7	5.84	50.7	41.2
AwareVLN (Guo et al., 2026)	CVPR'26	4.02	73.5	65.4	55.1	3.95	67.6	56.1
GA-VLN (Yang et al., 2026)	CVPR'26	4.80	67.6	61.0	55.2	5.88	55.4	45.2
Progress-Think (Wang et al., 2026b)	CVPR'26	4.68	63.6	60.1	53.6	8.30	27.5	22.7
Context-Think (ours)	—	**3.72**	**77.1**	**66.5**	57.0	**3.92**	**70.3**	**57.8**

retains downstream route constraints and the final stopping objective while discarding completed stages and extraneous descriptive information. This design promotes globally coherent decision-making rather than locally myopic action selection. The gains are particularly pronounced on RxR-CE, whose longer and more detailed instructions demand sustained task-level planning across multiple navigation stages.

4.3 Ablation and Analysis

The controlled reasoning-context composition study is reported in Section 3.3, where it serves to motivate the design of Context-Think. Here we evaluate the subsequent planning decisions: how far the retained plan should extend, how its supervision should be generated, and which information should remain available while the plan is updated.

Planning horizon. In panel (a), final-goal replanning improves SR by 0.98 points on R2R-CE and 2.27 points on RxR-CE, with corresponding SPL gains of 0.96 and 1.78 points. The two conditions share the same policy initialization, instruction/history interface, and optimization settings. Appendix B.4 describes target replacement and context propagation. The larger gain on RxR-CE is consistent with the hypothesis that downstream route constraints become more useful when the remaining task spans more navigation stages. On RxR-CE, final-goal replanning reduces NE from 6.44 to 6.21 m and increases nDTW from 59.99 to 60.53. On R2R-CE, NE rises slightly despite improved SR and SPL, showing that task success and mean endpoint error need not move together.

Long-horizon navigation. Fig. 1(b,c) reports RxR-CE SR across overlapping 60-token instruction windows and 5 m reference-trajectory intervals. Although SR decreases with task length, final-goal replanning outperforms subgoal planning and full context across the displayed ranges. For 140–199-token instructions, it achieves 53.02% SR, compared with 50.89% and 46.26%, respectively.

Table 2: Planning-horizon and implementation ablations on the val-unseen splits. (a) compares next-subgoal planning with final-goal replanning; (b) changes how remaining-route supervision is generated while keeping runtime inference unchanged; (c) jointly changes instruction and history handling during execution, so their individual effects are not isolated. The final-goal reference condition is repeated across panels. Bold marks the best value for each metric within each panel.

Condition	R2R-CE				RxR-CE			
	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
(a) Planning horizon								
Subgoal planning	**5.82**	60.03	52.15	46.45	6.44	56.30	47.93	59.99
Final-goal replanning	5.92	**60.96**	**53.13**	**47.41**	**6.21**	**58.57**	**49.71**	**60.53**
(b) Remaining-route target generation								
Teacher direct generation	5.92	**60.96**	**53.13**	**47.41**	**6.21**	**58.57**	**49.71**	60.53
Teacher with explicit reasoning	**5.78**	59.43	51.60	46.51	6.36	57.98	49.07	**60.95**
(c) Instruction and history handling								
Original instruction/history retained	**5.92**	60.96	**53.13**	**47.41**	**6.21**	**58.57**	**49.71**	**60.53**
Instruction/history jointly updated	5.94	**61.94**	50.73	45.36	6.82	55.04	46.55	57.76

For reference trajectories in $[20, 25)$ m, the corresponding values are 45.16%, 41.29%, and 36.13%. These results support preserving downstream route constraints and the final stopping objective beyond the immediate subgoal when navigating longer routes and following longer instructions.

Remaining-route target generation. Panel (b) tests whether more elaborate teacher reasoning produces better final-goal supervision. Both variants use the same navigation training configuration and runtime policy prompt; only the offline remaining-route targets differ. The teacher's intermediate reasoning is confined to annotation and is excluded from the deployed policy context. The comparison therefore evaluates whether this target-generation procedure provides more useful supervision for learning executable remaining-route plans. Adding an explicit teacher reasoning process reduces SR from 53.13% to 51.60% on R2R-CE and from 58.57% to 57.98% on RxR-CE, with corresponding decreases in SPL. RxR nDTW increases slightly, showing that greater reference-path similarity does not necessarily translate into higher task success. We therefore use direct remaining-route generation as the default. This result is consistent with our broader finding that additional reasoning content or reasoning steps are not automatically useful for downstream action selection.

Retaining the task reference and execution history. Context-Think replans the unfinished route but keeps the original instruction and accumulated observation history available. Panel (c) jointly changes these two sources of information. Updating them together lowers SR by 2.40 points on R2R-CE and 3.53 points on RxR-CE. On R2R-CE, OS rises from 60.96% to 61.94% while SR falls, indicating that more episodes enter the goal region but fewer terminate successfully. The ablation therefore supports retaining the original task reference and execution history while only refreshing the planning context. Because instruction replacement and history refresh are changed jointly, the experiment does not identify their individual effects.

Qualitative analysis. Fig. 4 compares the two planning horizons in Table 2 on the same RxR-CE episode. Both rollouts use plan-only reasoning context and start from the same instruction and pose. Subgoal planning repeatedly targets the kitchen: at $t = 122$ it plans to enter through the open doorway, and at $t = 202$ it plans to leave the kitchen, without naming the final stopping destination. Final-goal replanning names the sliding glass door and the stopping condition at $t = 100$, then contracts to a direct stopping plan at $t = 117$ and succeeds. Aggregate results are reported in Table 2. In this case, subgoal planning makes six updates and travels 39.50 m, ending 5.97 m from the goal; final-goal replanning makes four updates and travels 18.50 m, ending 0.89 m from the goal. The latter retains the destination as its remaining plan shortens.

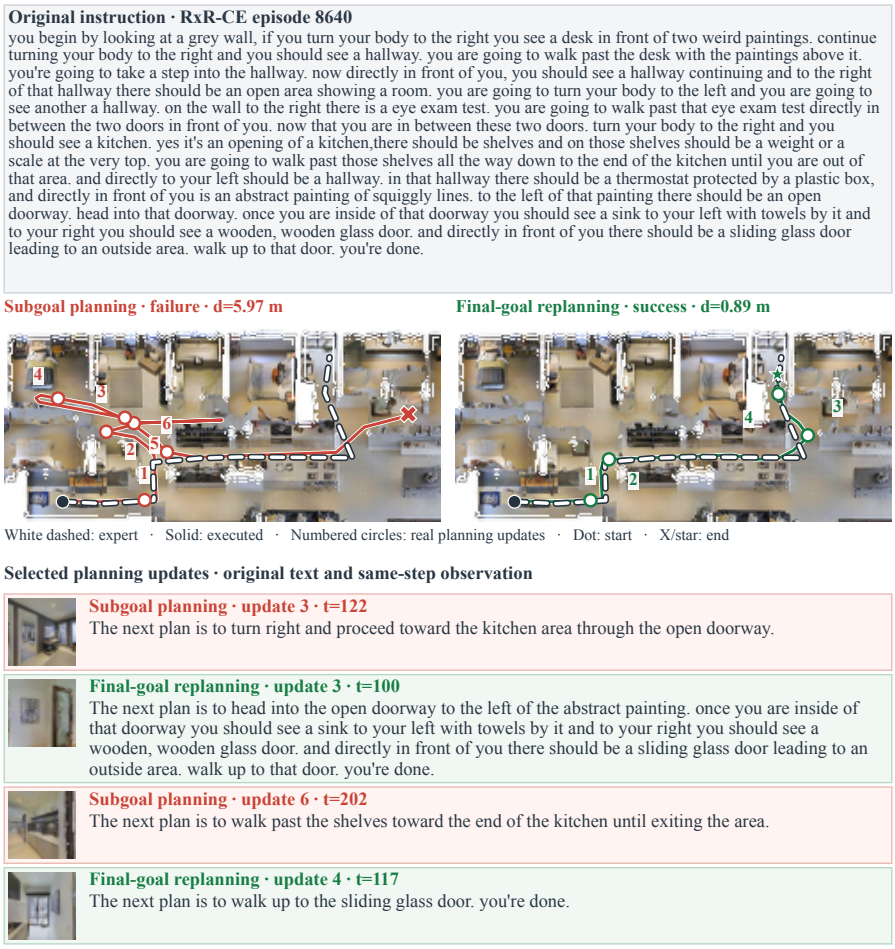

Fig. 4: **Qualitative comparison of planning horizons** on RxR-CE episode 8640, with plan-only context and a shared start pose. White dashed lines show the expert route; numbered circles mark planning updates. Each card pairs a selected verbatim plan with its same-step observation.

5 Conclusion

We have studied how language reasoning should be retained as policy context for vision-language navigation. Through controlled comparisons of scene, progress, and planning contexts, we show that more comprehensive reasoning is not necessarily more useful, and that planning provides the most effective guidance for future actions. Building on this finding, we propose **Context-Think**, which extends planning from the next subgoal to the complete remaining route through final-goal replanning. Experiments on R2R-CE and RxR-CE demonstrate that Context-Think improves both navigation success and efficiency over existing reasoning contexts and local planning. Our results suggest a simple principle for reasoning context: retain decision-oriented information and plan far enough to preserve future constraints.

Limitations. Our study is conducted on simulated indoor navigation benchmarks, where environments, instructions, and trajectories are constrained by existing datasets. While Context-Think demonstrates the benefit of long-horizon planning context in these settings, extending the analysis to physical robots, open-world environments, and more diverse embodied tasks remains an important direction. In addition, our study focuses on language-based reasoning context; exploring richer forms of multimodal planning context beyond textual representations is left for future work.

AI USE STATEMENT

In this work, we used generative AI tools to generate offline reasoning supervision with a vision-language teacher, as described in Section 3.2 and Appendix B. Additionally, we used generative AI tools for writing assistance and literature organization. Reported navigation metrics are computed from the benchmark evaluations described in Section 4.1. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our experiments use public R2R-CE and RxR-CE benchmarks and released navigation models. Section 4.1 describes the datasets, evaluation protocol, and metrics. Appendix A documents model initialization, training hyperparameters, visual-history sampling, and action execution for the controlled planning study. Appendix B describes component annotation, runtime policy prompts, remaining-route target construction, and context propagation. Together, these details specify how the reasoning contexts and planning horizons are constructed, trained, and evaluated.

REFERENCES

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 *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 3674–3683. IEEE, 2018.

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. *arXiv preprint arXiv:1709.06158*, 2017.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9796–9810, 2024.

Jiaqi Chen, Bingqian Lin, Xinmin Liu, Lin Ma, Xiaodan Liang, and Kwan-Yee K Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23568–23576, 2025.

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

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

Wenxuan Guo, Xiuwei Xu, Yichen Liu, Xiangyu Li, Hang Yin, Huangxing Chen, Wenzhao Zheng, Jianjiang Feng, Jie Zhou, and Jiwen Lu. Awarevln: Reasoning with self-awareness for vision-language navigation. *arXiv preprint arXiv:2605.22816*, 2026.

Xiangyun Huang, Xiangchen Wang, Runfeng Lin, Yihao Xu, Kangyu Huang, Jiang Hengchen, Xiwang Dong, and Lin Jiarong. From routes to steps: Separating semantic progress from local execution in vision-and-language navigation. *arXiv preprint arXiv:2608.03143*, 2026.

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*, pp. 104–120. Springer, 2020.

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.

Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5945–5957, 2025.

Haoran Liu, Weikang Wan, Xiqian Yu, Minghan Li, Jiazhao Zhang, Bo Zhao, Zhibo Chen, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Na vid-4d: Unleashing spatial intelligence in egocentric rgb-d videos for vision-and-language navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10607–10615. IEEE, 2025.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024a.

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

Renjie Lu, Yu Zhou, Hao Cheng, Jingke Meng, and Wei-Shi Zheng. monovln: Bridging the observation gap between monocular and panoramic vision and language navigation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9477–9486. IEEE, 2025.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. *arXiv preprint arXiv:1901.03035*, 2019.

Bowen Pan, Rameswar Panda, SouYoung Jin, Rogerio Feris, Aude Oliva, Phillip Isola, and Yoon Kim. Langnav: Language as a perceptual representation for navigation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 950–974, 2024.

Yanyuan Qiao, Qianyi Liu, Jiajun Liu, Jing Liu, and Qi Wu. Llm as copilot for coarse-grained vision-and-language navigation. In *European Conference on Computer Vision*, pp. 459–476. Springer, 2024.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346. IEEE, 2019.

Shuo Wang, Yongcai Wang, Wanting Li, Xudong Cai, Yucheng Wang, Maiyue Chen, Zhizhong Su, Deying Li, and Zhaoxin Fan. Aux-think: Exploring reasoning strategies for data-efficient vision-language navigation. *Advances in Neural Information Processing Systems*, 38:30892–30916, 2026a.

Shuo Wang, Yucheng Wang, Guoxin Lian, Yongcai Wang, Maiyue Chen, Kaihui Wang, Bo Zhang, Zhizhong Su, Yutian Zhou, Wanting Li, et al. Progress-think: Semantic progress reasoning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4076–4086, 2026b.

Zihan Wang and Gim Hee Lee. g3d-lf: Generalizable 3d-language feature fields for embodied tasks. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14191–14202. IEEE, 2025.

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

Zihan Wang, Seungjun Lee, and Gim Hee Lee. Dynam3d: Dynamic layered 3d tokens empower vlm for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 38:153522–153544, 2026c.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

Meng Wei, Chenyang Wan, Peng Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation. In *International Conference on Learning Representations*, volume 2026, pp. 12380–12396, 2026.

Jiahao Yang, Zihan Wang, Xiangyang Li, Xing Zhu, Yujun Shen, Yinghao Xu, and Shuqiang Jiang. Ga-vln: Geometry-aware bev representation for efficient vision-language navigation. *arXiv preprint arXiv:2605.22036*, 2026.

Xuan Yao, Junyu Gao, and Changsheng Xu. Navmorph: A self-evolving world model for vision-and-language navigation in continuous environments. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5536–5546. IEEE, 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. *arXiv preprint arXiv:2402.15852*, 2024b.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. In *International Conference on Learning Representations*, volume 2026, pp. 127293–127322, 2026.

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, 2025a.

Siqi Zhang, Yanyuan Qiao, Qunbo Wang, Longteng Guo, Zhihua Wei, and Jing Liu. Flexvln: Flexible adaptation for diverse vision-and-language navigation tasks. *IEEE Transactions on Multimedia*, 27:6307–6318, 2025b.

Yue Zhang, Quan Guo, and Parisa Kordjamshidi. Navhint: Vision and language navigation agent with a hint generator. Association for Computational Linguistics, 2024c.

Yue Zhang, Tianyi Ma, Zun Wang, Yanyuan Qiao, and Parisa Kordjamshidi. Vision-and-language navigation with analogical textual descriptions in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 15028–15036, 2025c.

Zekai Zhang, Weiye Zhu, Hewei Pan, Xiangchen Wang, Rongtao Xu, Xing Sun, and Feng Zheng. Activevln: Towards active exploration via multi-turn rl in vision-and-language navigation. *arXiv preprint arXiv:2509.12618*, 2025d.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, pp. 260–278. Springer, 2024a.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024b.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10009–10019. IEEE, 2020.

Jing Zuo, Lingzhou Mu, Fan Jiang, Chengcheng Ma, Mu Xu, and Yonggang Qi. Fantasyvln: Unified multimodal chain-of-thought reasoning for vision-language navigation. *arXiv preprint arXiv:2601.13976*, 2026.

A IMPLEMENTATION DETAILS

A.1 TRAINING CONFIGURATION

The controlled reasoning-context and planning-horizon studies use the R2RFollow and RxRFollow training subsets and initialize the policy from the released NaVILA Llama-3 8B model with eight-frame visual input (Cheng et al., 2024). The language model is fine-tuned while the vision encoder and multimodal projector remain frozen. Table 3 reports the configuration used for the final-goal replanning condition in Table 2.

The teacher-reasoning ablation uses the same navigation training configuration and runtime policy prompt; only the offline remaining-route targets are replaced by targets generated with explicit teacher reasoning.

Table 3: Training configuration for the controlled final-goal replanning study.

Training setting	**Value**
Maximum sequence length	4,096 tokens
Effective batch size	1,024
Training epochs	1
Learning rate	5×10^{-5}
Learning-rate schedule	Cosine decay
Warmup ratio	0.03
Weight decay	0
Precision	Bfloat16

A.2 VISUAL HISTORY AND EXECUTION

Each policy query contains the current observation and seven sampled historical frames. The current frame is always retained, while historical frames are sampled at approximately uniform temporal intervals; short histories are padded with black frames. The original instruction remains available throughout navigation. After a reasoning update is produced, its generated text and the number of environment steps since that update are included in subsequent policy queries.

The policy generates either a reasoning update or an action. A reasoning update replaces the retained context without moving the agent. An action is translated into simulator commands: forward motion is executed in 0.25 m increments and turns in 15° increments. Multi-step commands are queued before the next policy query. The controlled planning study uses greedy decoding with at most 256 newly generated tokens per query. To prevent repeated reasoning from stalling an episode, execution terminates after more than three consecutive reasoning outputs without an intervening action.

Execution procedure.

1. Initialize the reasoning context as empty while retaining the original instruction I.
2. At decision step t, construct the visual input from H_t and generate Y_t conditioned on I, H_t, and C_t.
3. If Y_t is a reasoning update, replace C_t with the generated text and query the policy again from the same physical state.
4. Otherwise, execute the decoded action or action sequence, append new observations to the history, and retain the current reasoning context until the next update.

B REASONING-CONTEXT SUPERVISION

B.1 INDEPENDENT COMPONENT ANNOTATION

As described in Section 3.2, scene, progress, and subgoal-plan supervision are generated independently at matched reasoning-update positions. They are not extracted from a previously generated full reasoning trace.

For each update position, the teacher receives the same original instruction and state-aligned episode information. Only the requested reasoning role changes. The scene prompt requests a concise description of the current navigation-relevant environment and explicitly excludes progress assessment and future planning. The progress prompt requests the completed portion and current stage of the instructed task and excludes future navigation actions. The subgoal-plan prompt requests one high-level objective for the next navigation phase and explicitly excludes both completed-progress summaries and the complete remaining route.

Table 4: Teacher requests used to independently construct the three reasoning-context components.

Component	**Teacher request**
Scene description	Describe only the agent's current navigation-relevant environment. Mention visible landmarks, room type, and spatial relations when useful. Do not assess completed instruction steps or propose future actions.
Progress assessment	Using the original instruction and executed trajectory so far, state which part of the task has been completed and the agent's current task stage. Do not propose the next navigation action or route.
Subgoal plan	From the current state, provide one concise high-level objective for the next navigation phase. Do not summarize completed progress and do not describe the complete remaining route to the final destination.

The full-context target is created only after generation by concatenating the three matched annotations in the fixed order scene–progress–plan. Thus, its components are exactly the same supervision items used by the corresponding single-component conditions; only their composition differs.

B.2 RUNTIME POLICY PROMPTS

Teacher annotation prompts and runtime policy prompts serve different purposes. The former create supervision offline; the latter specify which type of reasoning the navigation policy should produce during training and deployment. Table 5 lists the policy-side instructions.

Table 5: Reasoning request supplied to the navigation policy. The planning request is shared by subgoal planning and final-goal replanning; their different horizons are learned from different supervision targets rather than from different runtime instructions.

Context	**Policy request**
Full context	If reasoning, describe the current observations, assess task progress, and provide a high-level plan for the next navigation phase.
Scene only	If reasoning, describe the current navigation-relevant observations.
Progress only	If reasoning, assess progress through the original instruction.
Planning	If reasoning, provide a high-level navigation plan.

B.3 OFFLINE GENERATION OF FINAL-GOAL TARGETS

Final-goal supervision is generated separately from the single-component study. At candidate reasoning-update positions, we construct a state-conditioned remaining-route target that starts from the current state and terminates at the original destination.

We use a two-stage annotation procedure. The first stage provides the teacher with the original instruction and sampled observations from the demonstrated episode so that it can recover the overall route structure and landmark transitions. The second stage marks the selected current state and provides the demonstrated suffix from that state to the destination. The teacher is then asked to rewrite the unfinished navigation task.

The teacher is instructed to:

- remove navigation stages already completed;
- preserve the order of the remaining route;
- retain navigation-relevant landmarks and spatial relations;
- preserve the original destination and stopping condition; and
- output only an executable remaining-route instruction.

The first-stage route interpretation is annotation-time scaffolding only and is never used as policy context. Likewise, future trajectory observations are available only while constructing supervision. At deployment, the policy must infer the remaining-route plan from the original instruction and its observed execution history.

Before constructing the supervision targets, we filter the generated responses for output-format compliance. We exclude empty responses, responses containing reasoning or structured-data prefixes, and responses explicitly referring to teacher-only information, including demonstrations, annotations, future observations, or progress metadata. The retained responses form the final-goal target annotations, each linked to its source episode and reasoning-update position.

B.4 TARGET FORMATTING AND CONTEXT PROPAGATION

Subgoal-plan and final-goal targets use the same textual planning format and runtime interface. For each retained final-goal annotation, we replace the reasoning target at the corresponding update position with the remaining-route instruction, without adding separate scene or progress fields. The corresponding records retain their source trajectory, action labels, and observation references.

The inserted target serves as policy context for subsequent action samples until the next reasoning update. Segments preceding the first update use an empty reasoning context. Thus, target insertion updates both the reasoning supervision and the context presented to the action policy.