CREDIT-GUIDED POLICY IMPROVEMENT FOR TEST-TIME ADAPTIVE VISION-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

To advance the development of embodied general navigation, Test-time Adaptation for Vision-Language Navigation (TTA-VLN) has attracted increasing attention, aiming to adapt pretrained policies online to previously unseen environments using only test-time observations and interaction history. However, distribution shifts in unseen environments can distort the pretrained policy's local action preferences and lead to off-course decisions. Existing methods seek to correct such deviations using test-time signals, such as predictive uncertainty, trajectory-level feedback, or accumulated adaptation experience. Yet these signals do not directly establish whether an executed behavior actually contributes to instruction-guided progress toward the goal. Moreover, even when a test-time signal suggests a plausible corrective direction, the resulting policy change may still be unreliable and should not necessarily persist in subsequent decisions. The central challenge is therefore twofold: how to identify whether an interaction supports goal-directed improvement, and how to determine whether the resulting policy update is worth retaining. We observe that every executed action induces an immediate observation transition that exposes evidence of its local consequence. Based on this observation, we propose Credit-Guided Policy Improvement (CGPI), which uses action-induced observation transitions to recover signed, reference-relative decision credit without external outcome feedback. The recovered credit proposes a lightweight policy update, which is verified against prior credit-supported interactions and retained only when supported; otherwise, it is rolled back, while the pretrained navigation policy remains frozen. CGPI achieves consistent gains across the evaluated VLN benchmarks and navigation backbones, while qualitative robot trials further illustrate zero-shot sim-to-real feasibility.

1 INTRODUCTION

As an important capability of embodied robots, Vision-Language Navigation (VLN) enables an agent to follow natural language instructions and navigate to target locations based on visual observations [3, 7, 9, 10]. Despite substantial progress on established VLN benchmarks, pretrained navigation policies can still suffer significant performance degradation when deployed with previously unseen environments and instructions. Test-Time Adaptation for VLN (TTA-VLN) [8, 13, 14, 12] has therefore attracted increasing attention, allowing a pretrained policy to adapt online from observations and interactions encountered during deployment. Such capability is important for bringing VLN from benchmark generalization toward practical navigation in open-world environments.

The difficulty of TTA-VLN stems from the mismatch between training and deployment distributions in both vision and language. Unseen environments may introduce new appearances, layouts, and local navigation contexts, while unseen instructions may contain unfamiliar expressions. Such shifts can make the pretrained policy's learned cross-modal decision preferences unreliable, leading to off-course behavior. Since the required correction is not explicitly available at test time, existing TTA-VLN methods infer how to adapt from predictive uncertainty, trajectory-level feedback, or accumulated adaptation experience. However, these signals do not directly evaluate the actual consequence of the current executed action, providing limited evidence of whether that interaction truly contributes to instruction-guided progress. Moreover, even when a signal suggests a plausible correction, the resulting policy change may still be unreliable and should not necessarily persist.

The key issue is therefore that TTA enables *policy change*, but policy change does not necessarily imply *policy improvement*. As illustrated in Fig. 1, reliable TTA-VLN requires answering two coupled questions: *how to identify whether the current interaction supports goal-directed improvement, and*

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.

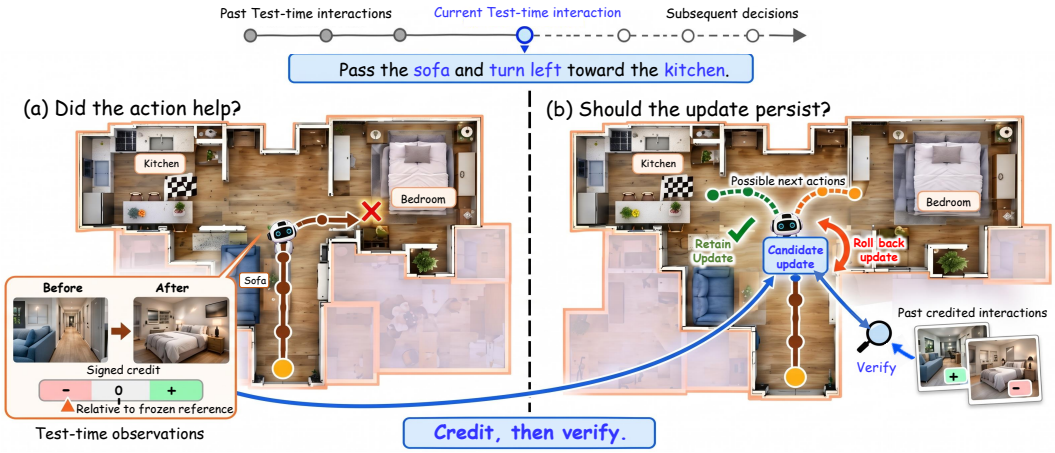

Figure 1: **Motivation for CGPI.** TTA enables online policy change, but change does not necessarily imply improvement. Decision credit first identifies whether the current interaction provides local evidence of goal-directed improvement; the resulting candidate update is then verified against past credit-supported interactions before being retained or rolled back.

how to determine whether the resulting candidate policy update is worth retaining. We refer to these two challenges as *improvement identification* and *improvement retention*, respectively. We observe that every executed action naturally produces an immediate observation transition that exposes its local consequence. This provides interaction-grounded evidence for identifying a supported direction of change, while the resulting policy modification should still be treated as a candidate that must be verified before it is allowed to persist.

Based on this insight, we propose Credit-Guided Policy Improvement (CGPI), a framework for TTA-VLN that explicitly separates *improvement identification* from *improvement retention*. For improvement identification, CGPI grounds each executed action with its actual resulting observation and compares the observed consequence against the frozen pretrained policy's local reference expectation, yielding signed, reference-relative decision credit without external outcome feedback. The recovered credit provides directional guidance for a lightweight action reranker, producing a candidate policy update rather than immediately committing the change. For improvement retention, CGPI verifies the candidate against previously observed credit-supported interactions and retains it only when the modification remains sufficiently supported and satisfies the proposal-state reference constraint; otherwise, the policy rolls back to the last accepted state. The pretrained backbone remains frozen throughout adaptation, forming an *identify–propose–verify–retain* loop for continual test-time policy improvement. Experiments across three VLN benchmarks and four navigation backbones evaluate CGPI, while qualitative trials on a physical robot further illustrate sim-to-real feasibility.

Our contributions are threefold. ***First***, we formulate reliable TTA-VLN from the perspective of *policy improvement* rather than policy change, identifying two coupled problems: *improvement identification* and *improvement retention*. ***Second***, we propose Credit-Guided Policy Improvement (CGPI), which recovers signed, reference-relative decision credit from action-induced observation transitions and verifies credit-guided candidate updates before selectively retaining or rolling them back. ***Third***, experiments across REVERIE, R2R, and R2R-CE with HAMT, DUET, BEVBert, and ETPNav evaluate CGPI across discrete and continuous VLN settings; qualitative trials on a Unitree Go2 further illustrate zero-shot sim-to-real feasibility.

2 Method

2.1 Problem Setup and Overview

We consider vision-language navigation in a streaming test-time setting. The agent is given a pretrained policy π_{θ_0} and receives a stream of navigation tasks $\mathcal{X} = \{X_1, \ldots, X_N\}$. Each task $X_i = (I_i, o_0^i)$ contains a natural-language instruction I_i and an initial observation o_0^i. At step t, the state is $s_t = (I_i, o_t, \tau_{<t}, \mathcal{G}_t)$, where $\tau_{<t}$ denotes the navigation history and $\mathcal{G}_t$ denotes the currently available local candidate structure. The agent selects $a_t \in \mathcal{A}_t$ until it chooses STOP.

We study *feedback-free* test-time adaptation, by which we mean that no external task-level correctness or outcome signal is available during deployment. The agent cannot access expert actions, target

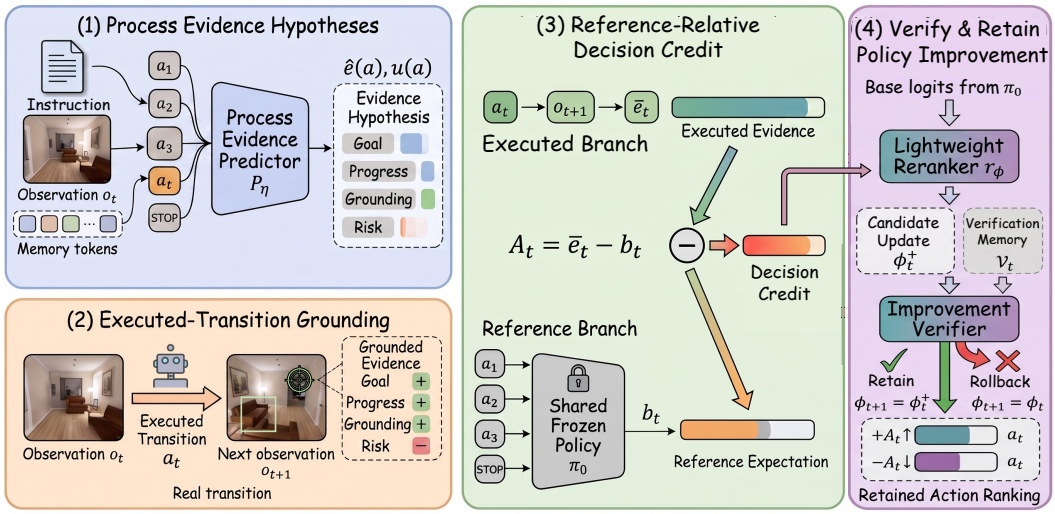

Figure 2: Overview of CGPI. The method constructs local process-evidence hypotheses and grounds the executed decision using its actual resulting observation. Reference-relative comparison recovers signed decision credit, which proposes a candidate modification to an action reranker. The candidate is then verified against past credit-supported interactions and retained only when sufficiently supported and satisfying the proposal-state reference constraint; otherwise, it is rolled back.

locations, dense rewards, trajectory-level success labels, or human feedback. For a movement action, the only new environment information obtained after execution is its resulting observation o_{t+1}. Candidate evidence for unexecuted actions is constructed only from information already observable at s_t; no future observation of an unexecuted action is accessed.

The frozen backbone produces base logits $z_t^0(a)$ and the reference policy $\pi_0(a|s_t)$. We keep all backbone parameters fixed and adapt only a lightweight residual reranker r_ϕ. Let ϕ_t denote the currently retained reranker parameters. The adapted policy is

$$\tilde{z}_t(a;\phi_t) = z_t^0(a) + \alpha \tanh(r_{\phi_t}(s_t,a)), \qquad \tilde{\pi}_{\phi_t}(a|s_t) = \mathrm{softmax}_{a\in\mathcal{A}_t}(\tilde{z}_t(a;\phi_t)). \tag{1}$$

Here, α bounds the maximum residual-logit correction.

Our goal is to identify interaction-induced policy changes that are locally supported and to determine which resulting modifications should persist. We denote a candidate modification proposed from the current interaction by ϕ_t^+. A candidate is committed only after verification; otherwise, the reranker remains at the previously retained state ϕ_t.

As illustrated in Fig. 2, CGPI follows an *identify–propose–verify–retain* loop. First, the consequence of an executed interaction is grounded by the actual next observation and evaluated relative to the frozen policy's local expectation, yielding reference-relative decision credit. Second, sufficiently supported credit proposes a candidate modification to the lightweight reranker. Third, the candidate is evaluated against previously observed credit-supported interactions. Only a verified candidate is retained; otherwise, CGPI rolls back to the last accepted reranker state. The pretrained VLN backbone remains frozen throughout this process.

2.2 Interaction-Grounded Improvement Identification

For movement-action credit, we define the movement candidate set as $\mathcal{A}_t^{\mathrm{mov}} = \mathcal{A}_t \setminus \{\mathrm{STOP}\}$. We renormalize the frozen policy over movement actions as $\pi_0^{\mathrm{mov}}(a|s_t) = \pi_0(a|s_t)/\sum_{a'\in\mathcal{A}_t^{\mathrm{mov}}}\pi_0(a'|s_t)$, so that $\sum_{a\in\mathcal{A}_t^{\mathrm{mov}}}\pi_0^{\mathrm{mov}}(a|s_t) = 1$. This renormalization is used only to construct the movement-action reference; the original policy over $\mathcal{A}_t$, including STOP, is still used for navigation and action selection.

For each evidence view k, we construct a current-state hypothesis for every locally available movement action. After executing a non-STOP action a_t, the actual next observation o_{t+1} grounds the evidence for the executed decision:

$$\hat{e}_t^k(a) = \hat{E}^k(s_t,a), \quad a\in\mathcal{A}_t^{\mathrm{mov}}, \qquad e_t^k = E^k(s_t,a_t,o_{t+1}), \quad a_t \neq \mathrm{STOP}. \tag{2}$$

Here, $\hat{E}^k$ is computed entirely from information available at s_t, while e_t^k is grounded by the actual observation obtained after executing a_t. We instantiate four lightweight evidence terms: **Goal**, **Progress**, **Grounding**, and **Risk**; their concrete definitions are provided in Appendix D.

Because different evidence terms have different numerical scales, we standardize them using source-training statistics computed once and fixed throughout deployment:

$$\bar{e}_t^k = \frac{e_t^k - \mu_k}{\sigma_k + \epsilon}, \qquad \bar{\hat{e}}_t^k(a) = \frac{\hat{e}_t^k(a) - \mu_k}{\sigma_k + \epsilon}. \tag{3}$$

No target labels, target locations, or test-time outcome signals are used to construct these statistics.

Observation grounding indicates what happened after the executed action, but an absolute evidence value does not reveal whether that consequence is locally favorable relative to the movement alternatives available at the same state. We therefore compare the grounded consequence against the frozen policy's movement-only local reference:

$$b_t^k = \sum_{a \in \mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t)\, \bar{\hat{e}}_t^k(a), \qquad C_t^k = \bar{e}_t^k - b_t^k, \qquad A_t = \frac{1}{K}\sum_{k=1}^{K} C_t^k, \;\; a_t \neq \mathrm{STOP}. \tag{4}$$

Here, b_t^k is the frozen-policy local reference over movement actions, C_t^k is the view-wise reference-relative credit, and A_t is the final movement-action credit.

The resulting A_t is a *local, reference-relative* estimate of the executed action's consequence. Positive credit indicates that the observed consequence is more favorable than the frozen policy's movement-only local reference expectation under the chosen evidence functions, while negative credit indicates the opposite. As shown in Appendix G, the normalized movement reference removes action-independent state shifts under an additive-error model and, under bounded evidence error, preserves the sign of the latent local advantage when the advantage margin exceeds the estimation error. Since STOP does not induce a new post-action observation, it is not included in the movement-action reference above. We handle STOP separately using a conservative completion-versus-movement credit, whose full construction is provided in Appendix D.

2.3 Credit-Guided Candidate Policy Update

Decision credit provides the direction for a candidate policy change. We verify this candidate before commitment, separating *update proposal* from *update retention*.

Let $D_t(\phi) = D_{\mathrm{KL}}(\tilde{\pi}_\phi(\cdot|s_t)\|\pi_0(\cdot|s_t))$ denote deviation from the frozen navigation prior. We clip extreme credits and propose an update only when the credit is sufficiently strong and the retained policy has not already drifted excessively:

$$\hat{A}_t = \mathrm{clip}(A_t, -A_{\max}, A_{\max}), \qquad m_t^{\mathrm{prop}} = \mathbf{1}[|A_t| \geq \delta_A]\, \mathbf{1}[D_t(\phi_t) \leq \delta_{\mathrm{KL}}]. \tag{5}$$

The credit-magnitude condition filters weak signals whose directions are most sensitive to evidence-estimation error; its interpretation is provided in Appendix G.

For an eligible interaction, the candidate-proposal objective is

$$\mathcal{L}_{\mathrm{prop}}(t) = -\mathrm{sg}(\hat{A}_t) \log \tilde{\pi}_{\phi_t}(a_t|s_t) + \lambda_{\mathrm{KL}} D_t(\phi_t) + \lambda_{\mathrm{reg}}\|\phi_t\|_2^2. \tag{6}$$

Positive credit therefore proposes increasing the corresponding decision tendency, whereas negative credit proposes suppressing it. The KL term regularizes the proposal toward the frozen navigation prior. One optimizer step on $\mathcal{L}_{\mathrm{prop}}(t)$ produces candidate model and optimizer states $(\phi_t^+, \omega_t^+) = \mathrm{OptStep}((\phi_t, \omega_t), \mathcal{L}_{\mathrm{prop}}(t))$, where ω_t is the optimizer state associated with the currently retained reranker. We pass (ϕ_t^+, ω_t^+) to the verification stage as the candidate model and optimizer states.

2.4 Verify-and-Retain Policy Improvement

To determine whether a candidate modification deserves to persist, we maintain a bounded verification memory $\mathcal{V}_t = \{(s_i, a_i, A_i) \mid i \in \mathcal{I}_t\}$ of recent credit-supported interactions, where $|A_i| \geq \delta_A$ and $i < t$. The current interaction that produces ϕ_t^+ is excluded during its own verification and inserted only after the retain-or-rollback decision, preventing self-validation. Memory construction, caching, and warm-up details are provided in Appendix E.

For any reranker parameters ϕ, we measure consistency with stored decision credits and define the candidate verification gain as

$$J_t(\phi) = \frac{1}{|\mathcal{V}_t|} \sum_{i \in \mathcal{I}_t} \mathrm{sg}(A_i) \log \tilde{\pi}_\phi(a_i|s_i), \qquad \Delta_t^{\mathrm{ver}} = J_t(\phi_t^+) - J_t(\phi_t). \tag{7}$$

For positive stored credit, increasing the corresponding action probability raises J_t; for negative stored credit, decreasing it also raises J_t. Hence Δ_t^{ver} measures whether the candidate is more consistent with previously supported decision directions than the currently retained policy.

We retain a candidate only when it improves the verification objective while remaining sufficiently close to the frozen navigation prior:

$$r_t = \mathbf{1}[\Delta_t^{\text{ver}} \geq \delta_{\text{keep}}]\, \mathbf{1}[D_t(\phi_t^+) \leq \delta_{\text{KL}}]. \tag{8}$$

If $r_t = 1$, both candidate states (ϕ_t^+, ω_t^+) are retained; otherwise, the policy and optimizer remain at (ϕ_t, ω_t). Thus, decision credit determines what change should be proposed, whereas verification determines whether that change should persist.

If no candidate is proposed or the verification memory has not reached its warm-up size, the retained state remains unchanged. After the retain-or-rollback decision, an eligible current interaction is inserted into the bounded verification memory and becomes available for evaluating future candidates.

The verification criterion also admits a local improvement interpretation. As shown in Appendix G, under bounded credit-estimation error, a sufficiently positive verification margin implies positive improvement of the corresponding latent local surrogate on the verification memory. This interpretation is local to credit-supported surrogate improvement on the verification memory; long-horizon navigation guarantees lie outside this analysis. The complete online procedure is provided in Appendix I.

3 Experiments

Experimental setup. We evaluate CGPI on REVERIE [24], R2R [3], and R2R-CE [15] with four pretrained navigation policies: HAMT [5], DUET [6], BEVBert [1], and ETPNav [2]. Following prior TTA-VLN evaluation protocols [12], we use the standard benchmark splits, pretrained policies, and official episode order under continual test-time adaptation. Implementation details and stream-order robustness are provided in Appendix C and Appendix B.5.

3.1 Main Results

Evaluation on REVERIE. Table 1 compares CGPI with existing TTA methods on REVERIE using HAMT and DUET as navigation backbones. CGPI consistently achieves the best performance across Val Seen, Val Unseen, and Test Unseen. On DUET, CGPI improves SR/SPL from 56.92/38.03 to 58.76/40.15 on Val Unseen and from 55.12/39.84 to 57.31/41.87 on Test Unseen compared with IDEA. Similar improvements are observed with HAMT, where SR/SPL increases from 34.92/31.52 to 36.97/33.26 on Val Unseen. These results show that combining observation-grounded, reference-relative improvement identification with selective verify-and-retain adaptation provides an effective test-time policy improvement mechanism across navigation backbones.

Evaluation on R2R & R2R-CE. Tables 2 and 3 further evaluate CGPI on instruction-following navigation in both discrete and continuous environments. On R2R Val Unseen, CGPI improves SR/SPL from 76/67 to 79/68 with DUET and from 76/68 to 79/70 with BEVBert compared with

Table 1: Experimental results for different TTA strategies on the REVERIE dataset.

Methods+Model	REVERIE Val Seen				REVERIE Val Unseen				REVERIE Test Unseen			
	OSR↑	SR↑	SPL↑	RGSPL↑	OSR↑	SR↑	SPL↑	RGSPL↑	OSR↑	SR↑	SPL↑	RGSPL↑
HAMT [5]	47.65	43.29	40.19	25.18	36.84	32.95	30.20	17.28	33.41	30.40	26.67	13.08
+ Tent [26]	46.03	43.43	40.78	25.81	32.60	30.56	28.23	14.48	25.06	23.73	21.78	10.82
+ SAR [22]	47.12	43.85	39.97	25.44	32.86	31.12	29.15	15.23	27.94	25.86	23.08	11.58
+ ViDA [18]	47.67	43.55	41.23	25.27	32.74	30.97	28.82	14.97	27.03	24.81	22.45	11.23
+ FSTTA [8]	48.21	42.87	39.56	24.58	36.78	32.89	30.51	17.20	33.39	30.39	26.65	13.61
+ ReCAP [11]	48.49	44.06	40.69	25.46	37.04	33.06	30.28	17.37	34.11	30.51	24.27	13.11
+ IDEA [12]	50.67	47.33	42.13	26.82	39.87	34.92	31.52	17.76	38.14	32.81	28.52	14.45
+ Ours	**53.01**	**49.19**	**44.22**	**28.78**	**42.05**	**36.97**	**33.26**	**19.98**	**40.25**	**34.73**	**30.78**	**15.48**
DUET [6]	73.86	71.75	63.94	51.14	51.07	46.98	33.73	23.03	56.91	52.51	36.06	22.06
+ Tent [26]	73.72	71.89	64.06	50.41	51.43	47.55	33.99	23.32	57.12	52.61	36.17	22.16
+ SAR [22]	74.84	71.75	64.43	51.70	53.26	48.00	33.92	23.09	57.11	53.04	36.07	22.27
+ ViDA [18]	73.99	72.49	63.49	50.89	52.53	48.14	32.45	21.92	56.78	52.74	35.10	21.77
+ FSTTA [8]	75.59	75.48	65.84	52.23	56.26	54.15	36.41	23.56	58.44	53.40	36.43	22.40
+ ReCAP [11]	75.06	74.72	65.87	53.42	56.67	54.74	36.22	23.74	57.72	53.07	36.52	22.47
+ IDEA [12]	78.45	78.24	67.74	55.07	58.51	56.92	38.03	25.47	58.91	55.12	39.84	24.52
+ Ours	**79.62**	**79.22**	**68.05**	**56.96**	**60.75**	**58.76**	**40.15**	**27.54**	**60.88**	**57.31**	**41.87**	**24.67**

Table 2: Experiments on the R2R dataset.

Methods	R2R Val Seen				R2R Val Unseen			
	TL↓	NE↓	SR↑	SPL↑	TL↓	NE↓	SR↑	SPL↑
DUET [6]	12.33	2.28	79	73	13.94	3.31	72	60
+ Tent	12.17	2.38	78	72	13.78	3.42	72	60
+ SAR	12.05	2.28	78	72	13.59	3.28	72	61
+ ViDA	12.11	2.31	79	73	13.63	3.34	72	61
+ FSTTA	13.39	2.25	79	73	14.64	3.03	75	62
+ ReCAP	12.02	2.27	78	73	13.39	3.28	72	61
+ IDEA	11.23	2.03	81	76	12.47	2.91	76	67
+ **Ours**	**10.91**	**1.89**	**83**	**77**	**12.08**	**2.73**	**79**	**68**
BEVBert [1]	13.56	2.17	81	74	14.55	2.81	75	64
+ Tent	12.68	2.36	80	74	13.14	2.93	74	63
+ SAR	12.49	2.28	80	74	12.98	2.79	75	64
+ ViDA	12.53	2.31	81	74	13.11	2.83	75	64
+ FSTTA	12.28	2.31	80	75	13.96	2.89	74	63
+ ReCAP	12.31	2.27	81	74	12.94	2.78	75	64
+ IDEA	10.89	2.15	83	79	12.03	2.53	76	68
+ **Ours**	**10.61**	**2.02**	**84**	**81**	**11.69**	**2.38**	**79**	**70**

Table 3: Experiments on the R2R-CE dataset.

Methods	R2R-CE Val Seen					R2R-CE Val Unseen				
	TL↓	NE↓	OSR↑	SR↑	SPL↑	TL↓	NE↓	OSR↑	SR↑	SPL↑
BEVBert [1]	13.98	3.77	73	68	60	13.27	4.57	67	59	50
+ Tent	12.74	3.35	76	70	62	13.29	4.61	65	59	49
+ SAR	12.53	3.28	76	70	63	13.23	4.54	66	59	50
+ ViDA	12.61	3.31	76	71	62	13.26	4.57	67	59	49
+ FSTTA	14.07	4.11	74	69	60	13.11	4.39	65	60	51
+ ReCAP	12.40	3.31	76	71	63	13.01	4.57	66	60	50
+ IDEA	12.27	3.04	78	73	64	12.67	4.26	69	62	52
+ **Ours**	**11.93**	**2.92**	**81**	**75**	**65**	**12.33**	**4.13**	**70**	**65**	**54**
ETPNav [2]	11.78	3.95	72	66	59	11.99	4.71	65	57	49
+ Tent	11.36	3.94	72	66	59	11.57	4.74	64	57	49
+ SAR	11.31	3.89	72	66	60	11.55	4.67	65	57	49
+ ViDA	11.58	3.91	72	66	60	11.62	4.72	64	57	49
+ FSTTA	11.35	3.93	72	66	59	11.57	4.77	64	57	49
+ ReCAP	11.31	3.92	72	66	60	11.56	4.74	65	57	50
+ IDEA	10.87	3.82	73	68	62	11.17	4.47	67	59	51
+ **Ours**	**10.59**	**3.70**	**75**	**69**	**65**	**10.90**	**4.34**	**70**	**61**	**52**



Figure 3: Real-world qualitative results of CGPI on a Unitree Go2 robot. We show third-person views and egocentric observations for three executed trajectories.

IDEA. On R2R-CE Val Unseen, CGPI also improves SR/SPL from 62/52 to 65/54 with BEVBert and from 59/51 to 61/52 with ETPNav. These results establish gains across the four evaluated navigation architectures in both discrete and continuous VLN settings.

Real-world qualitative results. We further deploy CGPI on a Unitree Go2 using a navigation policy trained only in simulated environments, without real-world fine-tuning or external outcome feedback. The three trajectories in Fig. 3 illustrate successful multi-stage instruction following and sim-to-real feasibility in unseen environments.

3.2 Case Studies of Credit-Guided Policy Improvement

Figure 4 visualizes representative test-time policy improvements in unseen environments, where CGPI trajectories mark key decision points with recovered decision credit A_t and associated verification outcomes. For the instruction *"Go to the small spa room with the white painted walls. Bring me the hand soap from the sink,"* CGPI first obtains a positive credit $A_t = +0.36$; the resulting candidate update is retained and shifts the policy toward the spa room. At the following decision point, a negative credit $A_t = -0.28$ produces a candidate update that is rejected by verification, preventing the policy from drifting toward the wrong side room. A later positive credit $A_t = +0.29$ is retained, further reinforcing movement toward the target sink. In contrast, IDEA continues along an off-course

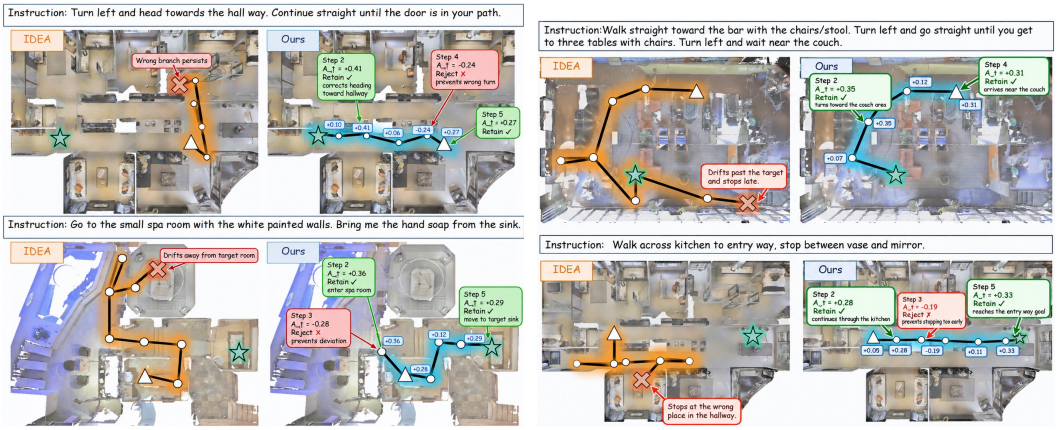

Figure 4: Qualitative comparison of IDEA and CGPI in unseen environments.

Table 4: Ablation of CGPI's two core improvement-identification operations on REVERIE Val Unseen with DUET.

Variant	Grounding	Reference	SR↑	SPL↑	CPC↑	WUR↓
Absolute Hypothesis	–	–	54.32	36.71	0.297	0.312
Grounded Evidence	✓	–	56.14	38.06	0.414	0.244
Relative Hypothesis	–	✓	56.82	38.67	0.463	0.221
CGPI	✓	✓	**58.76**	**40.15**	**0.572**	**0.154**

branch and stops away from the target. This visualizes how decision credit determines the update direction and verification controls whether that update is retained.

3.3 Ablation of Improvement Identification

We isolate the two operations used for improvement identification: *observation grounding*, which evaluates the executed action using its actual next observation, and *reference-relative comparison*, which evaluates the resulting evidence relative to the frozen policy's local expectation. All other CGPI components, including candidate proposal, verification, and retention, are kept fixed.

For this ablation, we use the view-averaged grounded evidence $\bar{e}_t = \frac{1}{K}\sum_k \bar{e}_t^k$, the view-averaged pre-action hypothesis $\bar{\hat{e}}_t(a) = \frac{1}{K}\sum_k \bar{\hat{e}}_t^k(a)$, and the view-averaged reference $b_t = \frac{1}{K}\sum_k b_t^k$. The four variants differ only in the signal used for candidate proposal: **Absolute Hypothesis** uses $\bar{\hat{e}}_t(a_t)$; **Grounded Evidence** uses $\bar{e}_t$; **Relative Hypothesis** uses $\bar{\hat{e}}_t(a_t) - b_t$; and **CGPI** uses $\bar{e}_t - b_t$, which is equivalent to Eq. 4 for movement actions.

To assess the quality of the proposal signal, we additionally report two post-hoc diagnostics. **Credit–Progress Correlation (CPC)** measures the Spearman correlation between the test-time signal and actual one-step geodesic progress, while **Wrong Update Rate (WUR)** measures the fraction of proposal-triggering interactions whose signal direction disagrees with oracle local progress. Geodesic information is used only for post-hoc evaluation and is never available during adaptation; formal definitions are provided in Appendix F.

Table 4 shows complementary contributions from both operations. Observation grounding improves SR, SPL, and CPC over the pre-action hypothesis, while reference-relative comparison further improves the quality of the proposal signal. Combining both yields the highest navigation performance and CPC together with the lowest WUR. This supports using the observed transition to determine what happened and the frozen-policy reference to determine whether that consequence is locally favorable. Additional diagnostics of the recovered credit against one-step geodesic progress are provided in Appendix B.4.

Contribution of STOP credit. The core ablation above keeps STOP handling fixed in order to isolate the two operations used for movement-action credit. We therefore further separate the contributions of observation-grounded movement credit and the dedicated STOP credit in Eq. 28. When movement credit is disabled, movement actions do not generate credit-guided candidate policy

Table 5: Contribution of movement-action credit and STOP-specific credit in CGPI on REVERIE.

Movement Credit	STOP Credit	SR↑	SPL↑
–	–	30.40	26.67
✓	–	33.54	29.61
–	✓	31.46	27.55
✓	✓	**34.73**	**30.78**

modifications. When STOP credit is disabled, STOP remains a valid action of the navigation policy, but a STOP decision does not generate a candidate modification. All other CGPI components, including verification and retention, remain unchanged.

As shown in Table 5, enabling movement-action credit alone improves SR and SPL from 30.40/26.67 to 33.54/29.61, accounting for most of the gain over the frozen HAMT policy. Using only STOP credit also improves performance to 31.46/27.55, showing that learning from stopping decisions provides an additional adaptation signal. Combining movement and STOP credit yields the strongest result of 34.73 SR and 30.78 SPL, indicating that the two sources of credit are complementary. Because STOP terminates the current navigation episode, a retained update induced by STOP cannot affect later decisions within the same trajectory. Under the continual protocol, however, the retained policy modification can influence subsequent episodes.

3.4 Effect of Improvement Verification and Retention

We next examine the second stage of CGPI: whether a candidate policy change should be retained after it has been proposed by decision credit. All variants use the same observation-grounded, reference-relative credit, lightweight reranker, and candidate-proposal objective. They differ only in how candidate modifications are evaluated before being committed. Table 6 reports this ablation on REVERIE Val Unseen with DUET.

We evaluate retention behavior using three complementary diagnostics. **Policy Drift (PD)** measures the average KL divergence between the adapted and frozen action distributions. **Accepted Improvement Rate (AIR)** is the fraction of retained movement-action candidates that yield positive short-horizon counterfactual progress, while **Retention Rate (RR)** is the fraction of proposed candidates that are committed. These diagnostics are used only for analysis; their formal definitions are provided in Appendix F.

Table 6: Ablation of improvement verification and retention on REVERIE Val Unseen with DUET.

Variant	Verify	Cand. KL	SR↑	SPL↑	PD↓	AIR↑	RR(%)
CGPI w/o Verification	–	–	57.41	38.96	0.071	0.548	100.0
CGPI w/o Candidate KL	✓	–	58.29	39.72	0.057	0.704	67.8
Full CGPI	✓	✓	**58.76**	**40.15**	**0.043**	**0.741**	**61.3**

The comparison isolates whether evaluating a proposed policy change before committing it improves the quality of retained adaptations. Without verification, every eligible candidate is committed, resulting in a higher retention rate but also larger policy drift and a lower AIR. Adding credit-based verification rejects unsupported modifications and improves both navigation performance and retained-candidate quality. The full method further applies the candidate-level KL condition, reducing policy drift from 0.057 to 0.043 while increasing AIR from 0.704 to 0.741.

3.5 Are Retained Policy Changes Actually Better?

To directly examine whether the verify-and-retain mechanism distinguishes useful policy changes from harmful ones, we perform a post-hoc counterfactual analysis on REVERIE Val Unseen with DUET. For every movement-action candidate, we clone the complete navigation state after the executed transition, including the simulator state, trajectory history, and accumulated navigation structure. The retained policy ϕ_t and candidate policy ϕ_t^+ are then rolled out independently from identical decision contexts for a short horizon H. Geodesic information is used only for this post-hoc diagnosis and is never available to CGPI during adaptation.

Table 7: Policy-improvement analysis on REVERIE. Left: post-hoc candidate-update quality (DUET, Val Unseen). Right: within-episode versus continual adaptation (HAMT, Test Unseen).

Method	AIR↑	AIG (m)↑	RP↑
Credit-Guided Direct Commit	0.548	0.19	–
CGPI	**0.741**	**0.63**	**0.716**

Setting	SR↑	SPL↑
Frozen HAMT	30.40	26.67
CGPI, episode reset	32.62	28.41
CGPI, continual	**34.73**	**30.78**

Let $\Gamma_t^{(H)}$ denote the difference in short-horizon oracle progress between the candidate and the currently retained policy. We report: (i) **Accepted Improvement Rate (AIR)**, the fraction of retained candidates with $\Gamma_t^{(H)} > 0$; (ii) **Accepted Improvement Gain (AIG)**, the average $\Gamma_t^{(H)}$ among retained candidates; and (iii) **Rejection Precision (RP)**, the fraction of rejected candidates with $\Gamma_t^{(H)} < 0$. Formal definitions are provided in Appendix F.

As shown in Table 7 (left), compared with direct commitment, CGPI increases AIR from 0.548 to 0.741 and the mean accepted improvement gain from 0.19 to 0.63 m. Moreover, an RP of 0.716 indicates that 71.6% of rejected candidates would have produced negative short-horizon oracle progress under the post-hoc counterfactual evaluation. These results show that verification both improves the quality of retained candidates and filters a substantial fraction of harmful policy changes.

3.6 Within-Episode versus Continual Adaptation

Since the decision credit for an executed action is obtained only after observing its resulting transition, a natural question is whether the recovered credit already benefits subsequent decisions within the same episode or mainly contributes through accumulation across episodes. We therefore compare CGPI under episode-reset and continual adaptation protocols on REVERIE Test Unseen with HAMT. In the *episode-reset* setting, the reranker parameters, optimizer state, and verification memory are restored to the same initialization before every navigation episode. All CGPI operations, including credit recovery, candidate proposal, verification, and retention, are performed normally within the episode. Thus, any improvement over the frozen policy can only arise from within-episode policy improvement. In the *continual* setting, the retained reranker parameters, optimizer state, and verification memory are preserved across episodes, following our default evaluation protocol.

As shown in Table 7 (right), episode-reset CGPI improves SR/SPL from 30.40/26.67 to 32.62/28.41 over the frozen HAMT policy. Since the reranker and optimizer state are reset before every episode, this improvement can only arise from CGPI updates affecting subsequent decisions within the same trajectory. This isolates a clear within-episode adaptation benefit from post-action decision credit. Retaining the adaptation state across episodes further improves performance to 34.73/30.78, indicating an additional benefit from cross-episode accumulation under the continual test-time setting.

4 Conclusion

We introduced CGPI, a feedback-free test-time adaptation framework that formulates online VLN adaptation from the perspective of *policy improvement* rather than policy change. CGPI uses action-induced observation transitions to recover local, reference-relative decision credit and separates credit-guided update proposal from subsequent verification and retention. Across the evaluated discrete and continuous VLN settings, CGPI improves navigation performance with multiple pretrained backbones while adapting only a lightweight residual reranker. Post-hoc analyses further show that the recovered credit is aligned with local navigation progress and that selective verification improves the quality of retained candidate updates. Qualitative robot trials additionally illustrate zero-shot sim-to-real feasibility in the shown real-world settings.

AI Use Statement

Generative AI tools, including ChatGPT, were used during this work to support brainstorming and refinement of research ideas, discussion of methodological and experimental design choices, and revision of the manuscript for clarity and presentation. They were also used as auxiliary assistance for experiment planning, implementation-related discussions, and analysis of experimental results.

Ethics Statement

This work adheres to the ICLR Code of Ethics. It involves no human-subject or animal experiments and introduces no new collection of personally identifiable information. All experiments are conducted on publicly available benchmark datasets in accordance with their applicable usage terms. We are not aware of additional privacy, security, or ethical concerns arising from this work.

Reproducibility Statement

We provide implementation details, hyperparameter settings, evaluation protocols, diagnostic metric definitions, and the complete online algorithm in the appendix. Hardware and test-time adaptation settings are specified for reproducibility.

References

[1] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[2] Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[3] Peter Anderson, 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*, pp. 3674–3683, 2018.

[4] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15791–15801, 2025.

[5] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 5834–5847, 2021.

[6] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16537–16547, 2022.

[7] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[8] Junyu Gao, Xuan Yao, and Changsheng Xu. Fast-slow test-time adaptation for online vision-and-language navigation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 14902–14919. PMLR, 2024.

[9] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13137–13146, 2020.

[10] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. VLN-BERT: A recurrent vision-and-language BERT for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1643–1653, 2021.

[11] Zixuan Hu, Yichun Hu, Xiaotong Li, Shixiang Tang, and Ling-Yu Duan. Beyond entropy: Region confidence proxy for wild test-time adaptation. *ICML*, 2025.

[12] Zixuan Hu, Xuantuo Huang, Yancheng Li, Yichun Hu, Shengyong Xu, and Ling-Yu Duan. Turning adaptation into assets: Cross-domain bridging for online vision-language navigation. *arXiv preprint arXiv:2605.23257*, 2026.

[13] Sungjune Kim, Gyeongrok Oh, Heeju Ko, Daehyun Ji, Dongwook Lee, Byung-Jun Lee, Sujin Jang, and Sangpil Kim. Test-time adaptation for online vision-language navigation with feedback-based reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

[14] Heeju Ko, Sung June Kim, Gyeongrok Oh, Jeongyoon Yoon, Honglak Lee, Sujin Jang, Seungryong Kim, and Sangpil Kim. Active test-time vision-language navigation. *Advances in Neural Information Processing Systems*, 38:44756–44775, 2026.

[15] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the navgraph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120, 2020.

[16] Yang Li, Aming Wu, Zihao Zhang, and Yahong Han. Towards open environments and instructions: General vision-language navigation via fast-slow interactive reasoning. *arXiv preprint arXiv:2601.09111*, 2026.

[17] 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*, 2025. doi: 10.1109/TPAMI.2025.3554559.

[18] Jiaming Liu, Senqiao Yang, Peidong Jia, Renrui Zhang, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang. ViDA: Homeostatic visual domain adapter for continual test time adaptation. In *The Twelfth International Conference on Learning Representations*, 2024.

[19] Yujie Lu, Huiliang Zhang, Ping Nie, Weixi Feng, Wenda Xu, Xin Eric Wang, and William Yang Wang. Anticipating the unseen discrepancy for vision and language navigation. *arXiv preprint arXiv:2209.04725*, 2022.

[20] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. URL `https://arxiv.org/abs/1903.01602`.

[21] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Songlin Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*, pp. 16888–16905, 2022.

[22] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.

[23] Akhil Perincherry, Jacob Krantz, and Stefan Lee. Do visual imaginations improve vision-and-language navigation agents? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3846–3855, 2025.

[24] Yuankai Qi, Zizheng Pan, Yicong Hong, Ming-Hsuan Yang, Anton van den Hengel, and Qi Wu. REVERIE: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9982–9991, 2020.

[25] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pp. 9229–9248, 2020.

[26] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

[27] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

[28] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022.

[29] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6629–6638, 2019.

[30] Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025. doi: 10.48550/arXiv.2507.05240.

[31] Xinda Xue, Junjun Hu, Minghua Luo, Shichao Xie, Jintao Chen, Zixun Xie, Kuichen Quan, Wei Guo, Mu Xu, and Zedong Chu. Omninav: A unified framework for prospective exploration and visual-language navigation. *arXiv preprint arXiv:2509.25687*, 2025. doi: 10.48550/arXiv.2509.25687.

[32] Xuan Yao, Junyu Gao, and Changsheng Xu. Navmorph: A self-evolving world model for vision-and-language navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5536–5546, 2025.

[33] Shuai Zhao, Xiaohan Wang, Linchao Zhu, and Yi Yang. Test-time adaptation with clip reward for zero-shot generalization in vision-language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 3597–3613, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/0faa4bc5f522076947a030273629d4fe-Paper-Conference.pdf.

APPENDIX

A Related Work **13**

B Additional Experimental Analysis **14**
B.1 Comparison with Feedback-Driven Methods 14
B.2 Comparison under Different External-Feedback Settings 15
B.3 Adaptation Efficiency 16
B.4 Credit-Signal Diagnostics 16
B.5 Robustness to Test-Stream Ordering 17
B.6 Proposal-Side Stability 17
B.7 Robustness to Noisy Improvement Signals 18
B.8 Sensitivity to Verification Memory 18

C Implementation Details **19**

D Details of Evidence Views **21**

E Verification and Retention Details **23**

F Diagnostic Metrics **24**

G Theoretical Analysis and Proofs **25**
G.1 Reference Centering Removes Shared State Bias 25
G.2 View-Level Credit Error 26
G.3 Sign Consistency of Improvement Identification 26
G.4 Interpretation of the Proposal Gate 27
G.5 Verified Local-Surrogate Improvement 27
G.6 Reference-Deviation Control 28

H More Visualizations **28**

I Full Online Algorithm **30**

J Scope and Limitations **30**

A RELATED WORK

Vision-Language Navigation. Vision-Language Navigation (VLN) requires an embodied agent to ground natural-language instructions in visual observations and make sequential decisions in unseen environments. Early instruction-following and speaker–follower models established the basic learning paradigm [3, 7], while subsequent work improved cross-modal grounding and generalization through pretraining and Transformer-based architectures [9, 10]. Later approaches further strengthened long-horizon history modeling and spatial reasoning, exemplified by HAMT [5], DUET [6], and BEVBert [1]. More recent studies incorporate LLM/VLM reasoning [17, 16], fast–slow decision making [30, 31], and visual imagination or world modeling [23, 4, 32]. These methods primarily improve how navigation policies represent instructions, observations, history, and future possibilities. However, these methods are generally trained before deployment and evaluated with fixed parameters, which limits their ability to adapt to distribution shifts encountered during online navigation.

Test-Time Adaptation. Test-Time Adaptation (TTA) updates pretrained models using unlabeled or weakly supervised deployment data without revisiting the original training set. Test-time training introduces auxiliary self-supervised objectives, while Tent adapts models by minimizing prediction entropy [25, 27]. Subsequent approaches improve robustness under continual distribution shifts through sample selection and regularization in EATA [21], sharpness-aware optimization and recovery in SAR [22], and teacher averaging with stochastic restoration in CoTTA [28]. ReCAP further replaces conventional pointwise confidence objectives with a region-confidence proxy [11]. While effective for static recognition tasks, these approaches are less suitable for VLN, where each action changes future observations and affects the final trajectory. In this sequential setting, increasing prediction

confidence does not necessarily imply navigation progress and may reinforce incorrect turns, repeated exploration, or premature stopping.

Online Adaptation for VLN. Recent studies have extended test-time adaptation to sequential VLN. DAVIS [19] exploits visual consistency and contrastive objectives to anticipate discrepancies in unseen environments, while FSTTA develops fast–slow online adaptation based primarily on entropy minimization and gradient/parameter decomposition [8]. FeedTTA introduces reinforcement learning from binary episodic feedback, and ATENA selectively requests external feedback for uncertain navigation episodes and combines feedback-conditioned adaptation with self-active learning [13, 14]. IDEA instead accumulates and reuses historical adaptation assets through distribution alignment and cross-domain composition while keeping the navigation backbone frozen [12]. These methods adapt VLN policies using uncertainty, episode-level feedback, or accumulated experience, but they do not directly assess whether an individual executed action warrants a persistent policy change. In contrast, CGPI derives signed decision credit from the action-induced observation transition, uses it to propose a local policy update, and explicitly verifies whether that update should be retained. This enables fine-grained, feedback-free adaptation while filtering unsupported policy changes.

Self-Supervised Navigation Improvement. Prior VLN studies have explored improving navigation policies from self-generated experience. RCM introduces a cross-modal matching critic to provide intrinsic rewards for instruction–trajectory consistency, while its Self-Supervised Imitation Learning (SIL) extension improves generalization by imitating previously discovered successful behaviors in unseen environments [29]. Unlike trajectory-level self-imitation, CGPI operates at the decision level: it assigns signed credit to individual executed interactions and uses this credit to update future action preferences online. Moreover, each proposed update is explicitly verified before being retained, reducing the risk of propagating unreliable interaction signals.

Credit Assignment and Process-Level Decision Evaluation. Prior navigation methods have used intermediate progress signals to correct local decisions. Self-Monitoring learns a progress estimator to track instruction completion, while the Regretful Agent uses progress estimates and progress markers to trigger backtracking and redirect navigation [20]. In contrast, CGPI uses the actual post-action observation to assign signed credit to the executed decision and converts this credit into an online policy update. Thus, prior methods primarily use progress estimates to revise the navigation trajectory, whereas CGPI uses interaction consequences to modify future action preferences. More broadly, CGPI is related to advantage-based policy optimization, counterfactual credit assignment, and process-level reward modeling, where intermediate actions are evaluated relative to a reference or expected outcome. RLCF [33] similarly uses centered semantic rewards for test-time policy optimization. CGPI differs in two aspects: it constructs credit from the realized navigation transition relative to the frozen policy's local candidate expectation, without learning an additional value or reward model; and it separates *update proposal* from *update retention*. A credit-supported update is retained only if it remains supported by previously observed credited interactions while satisfying a local proposal-state deviation constraint relative to the frozen policy. This enables CGPI to not only identify how the policy should change, but also filter unsupported policy changes before they persist.

B Additional Experimental Analysis

B.1 Comparison with Feedback-Driven Methods

Feedback-driven adaptation. Unlike standard test-time adaptation methods that operate without external task-level supervision, feedback-driven approaches introduce additional guidance from foundation models or human interaction to correct model updates or navigation behavior. We compare CGPI against three representative feedback-driven approaches: **RLCF** [33], which uses CLIP-based vision-language alignment scores as reward signals; **ATENA** [14], which follows an active-learning paradigm and requests human feedback for uncertain decisions; and **FeedTTA** [13], which performs online reinforcement learning from binary episodic success/failure feedback. In contrast, CGPI does not access human annotations, foundation-model judgments, trajectory success labels, or other external outcome feedback during adaptation.

Experimental results. Tables 8, 9, and 10 compare feedback-driven and feedback-free adaptation across REVERIE, R2R, and R2R-CE. The results of the feedback-driven baselines are adopted from the corresponding comparison reported by IDEA [12] under the same navigation backbones and benchmark settings.

Table 8: Comparison with feedback-driven methods on REVERIE.

	Model+Method	REVERIE Val Seen				REVERIE Val Unseen				REVERIE Test Unseen			
		OSR↑	SR↑	SPL↑	RGSPL↑	OSR↑	SR↑	SPL↑	RGSPL↑	OSR↑	SR↑	SPL↑	RGSPL↑
With Feedback	HAMT [5]	47.65	43.29	40.19	25.18	36.84	32.95	30.20	17.28	33.41	30.40	26.67	13.08
	+ RLCF [33]	47.92	44.21	41.13	25.97	37.37	33.11	30.23	17.24	33.11	30.21	26.13	12.97
	+ FeedTTA [13]	62.97	55.80	49.70	31.80	40.73	35.05	31.60	17.83	38.62	34.14	29.07	14.36
	+ ATENA [14]	62.92	57.34	48.08	29.60	38.85	34.00	30.96	17.51	38.19	32.55	28.38	14.32
	DUET [6]	73.86	71.75	63.94	51.14	51.07	46.98	33.73	23.03	56.91	52.51	36.06	22.06
	+ RLCF [33]	74.67	73.84	64.97	52.47	53.17	49.23	34.76	23.46	57.32	52.92	36.40	22.23
	+ FeedTTA [13]	86.16	84.19	75.54	60.32	71.60	66.49	45.38	30.75	58.76	53.58	37.66	24.10
	+ ATENA [14]	85.52	84.33	74.31	59.99	71.88	68.11	45.82	31.26	57.74	54.28	40.70	25.01
Without Feedback	HAMT [5]	47.65	43.29	40.19	25.18	36.84	32.95	30.20	17.28	33.41	30.40	26.67	13.08
	+ Tent [26]	46.03	43.43	40.78	25.81	32.60	30.56	28.23	14.48	25.06	23.73	21.78	10.82
	+ FSTTA [8]	48.21	42.87	39.56	24.58	36.78	32.89	30.51	17.20	33.39	30.39	26.65	13.61
	+ ReCAP [11]	48.49	44.06	40.69	25.46	37.04	33.06	30.28	17.37	34.11	30.51	24.27	13.11
	+ IDEA [12]	50.67	47.33	42.13	26.82	39.87	34.92	31.52	17.76	38.14	32.81	28.52	14.45
	+ **Ours**	53.01	49.19	44.22	28.78	42.05	36.97	33.26	19.98	40.25	34.73	30.78	15.48
	DUET [6]	73.86	71.75	63.94	51.14	51.07	46.98	33.73	23.03	56.91	52.51	36.06	22.06
	+ Tent [26]	73.72	71.89	64.06	50.41	51.43	47.55	33.99	23.32	57.12	52.61	36.17	22.16
	+ FSTTA [8]	75.59	75.48	65.84	52.23	56.26	54.15	36.41	23.56	58.44	53.40	36.43	22.40
	+ ReCAP [11]	75.06	74.72	65.87	53.42	56.67	54.74	36.22	23.74	57.72	53.07	36.52	22.47
	+ IDEA [12]	78.45	78.24	67.74	55.07	58.51	56.92	38.03	25.47	58.91	55.12	39.84	24.52
	+ **Ours**	79.62	79.22	68.05	56.96	60.75	58.76	40.15	27.54	60.88	57.31	41.87	24.67

Table 9: Experimental results on R2R.

	Model+Method	R2R Val Seen				R2R Val Unseen			
		TL↓	NE↓	SR↑	SPL↑	TL↓	NE↓	SR↑	SPL↑
With Feedback	DUET [6]	12.33	2.28	79	73	13.94	3.31	72	60
	+ RLCF [33]	12.28	2.25	79	74	13.87	3.12	73	61
	+ FeedTTA [13]	11.49	2.09	80	75	13.52	2.95	75	65
	+ ATENA [14]	11.27	2.18	80	75	12.31	2.90	75	66
	BEVBert [1]	13.56	2.17	81	74	14.55	2.81	75	64
	+ RLCF [33]	12.94	2.23	80	74	14.11	2.76	75	64
	+ FeedTTA [13]	11.88	2.17	82	77	12.24	2.77	75	66
	+ ATENA [14]	10.79	2.26	82	78	12.22	2.78	76	68
Without Feedback	DUET [6]	12.33	2.28	79	73	13.94	3.31	72	60
	+ FSTTA [8]	13.39	2.25	79	73	14.64	3.03	75	62
	+ ReCAP [11]	12.02	2.27	78	73	13.39	3.28	72	61
	+ IDEA [12]	11.23	2.03	81	76	12.47	2.91	76	67
	+ **Ours**	**10.91**	**1.89**	**83**	**77**	**12.08**	**2.73**	**79**	**68**
	BEVBert [1]	13.56	2.17	81	74	14.55	2.81	75	64
	+ FSTTA [8]	12.28	2.31	80	75	13.96	2.89	74	63
	+ ReCAP [11]	12.31	2.27	81	74	12.94	2.78	75	64
	+ IDEA [12]	10.89	2.15	83	79	12.03	2.53	76	68
	+ **Ours**	**10.61**	**2.02**	**84**	**81**	**11.69**	**2.38**	**79**	**70**

Table 10: Experimental results on R2R-CE.

	Model+Method	R2R-CE Val Seen					R2R-CE Val Unseen				
		TL↓	NE↓	OSR↑	SR↑	SPL↑	TL↓	NE↓	OSR↑	SR↑	SPL↑
With Feedback	BEVBert [1]	13.98	3.77	73	68	60	13.27	4.57	67	59	50
	+ RLCF [33]	13.74	3.56	74	69	61	13.15	4.53	66	60	50
	+ FeedTTA [13]	13.54	3.08	79	73	63	16.15	4.33	69	61	50
	+ ATENA [14]	11.31	3.24	75	71	64	13.48	4.50	67	60	51
	ETPNav [2]	11.78	3.95	72	66	59	11.99	4.71	65	57	49
	+ RLCF [33]	11.45	3.91	72	66	59	11.71	4.67	65	57	49
	+ FeedTTA [13]	10.88	3.85	72	67	61	11.99	4.47	66	58	50
	+ ATENA [14]	10.81	3.86	72	67	61	12.89	4.53	66	58	49
Without Feedback	BEVBert [1]	13.98	3.77	73	68	60	13.27	4.57	67	59	50
	+ FSTTA [8]	14.07	4.11	74	69	60	13.11	4.39	65	60	51
	+ ReCAP [11]	12.40	3.31	76	71	63	13.01	4.57	66	60	50
	+ IDEA [12]	12.27	3.04	78	73	64	12.67	4.26	69	62	52
	+ **Ours**	**11.93**	**2.92**	**81**	**75**	**65**	**12.33**	**4.13**	**70**	**65**	**54**
	ETPNav [2]	11.78	3.95	72	66	59	11.99	4.71	65	57	49
	+ FSTTA [8]	11.35	3.93	72	66	59	11.57	4.77	64	57	49
	+ ReCAP [11]	11.31	3.92	72	66	60	11.56	4.74	65	57	50
	+ IDEA [12]	10.87	3.82	73	68	62	11.17	4.47	67	59	51
	+ **Ours**	**10.59**	**3.70**	**75**	**69**	**65**	**10.90**	**4.34**	**70**	**61**	**52**

1) Strong performance without external outcome feedback. Despite operating under a more restrictive test-time information budget, CGPI remains competitive with or outperforms feedback-driven approaches on challenging unseen environments. On REVERIE Test Unseen, CGPI achieves an SR/SPL of $34.73/30.78$ with HAMT and $57.31/41.87$ with DUET, exceeding the corresponding feedback-driven baselines in SR and SPL. On R2R Val Unseen, CGPI reaches $79/68$ SR/SPL with DUET and $79/70$ with BEVBert, compared with $75/66$ and $76/68$ for the strongest feedback-driven counterparts. Similar advantages are observed on R2R-CE Val Unseen, where CGPI obtains $65/54$ with BEVBert and $61/52$ with ETPNav.

2) Effective adaptation under a more restrictive information budget. Feedback-driven methods receive additional test-time information from external vision-language reward models, human interaction, or multimodal foundation-model judgments. In contrast, CGPI derives both its candidate-update direction and its retention criterion from the agent's own test-time interactions, without external outcome supervision.

B.2 Comparison under Different External-Feedback Settings

We group representative TTA methods according to the type of external guidance available during test-time adaptation.

Table 11: Comparison under different external-feedback settings. We report SR/SPL on REVERIE Test Unseen with HAMT, R2R Val Unseen with BEVBert, and R2R-CE Val Unseen with BEVBert. Feedback-driven baseline results follow the corresponding comparison reported by IDEA [12].

Method	Ext. Feedback	Source	REVERIE Test Unseen		R2R Val Unseen		R2R-CE Val Unseen	
			SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
FSTTA [8]	No	–	30.39	26.65	74	63	60	51
IDEA [12]	No	–	32.81	28.52	76	68	62	52
CGPI (Ours)	**No**	**–**	**34.73**	**30.78**	**79**	**70**	**65**	**54**
RLCF [33]	Yes	CLIP reward	30.21	26.13	75	64	60	50
FeedTTA [13]	Yes	Episodic feedback	34.14	29.07	75	66	61	50
ATENA [14]	Yes	Human query	32.55	28.38	76	68	60	51

Table 12: Computational overhead of different test-time adaptation methods using DUET. Latency denotes the average additional computation per navigation step over frozen-policy inference.

Method	Updated Params. (%)	Extra Latency (ms)	Peak Memory (GB)
Frozen Policy	0.00	0.0	8.41
TENT [26]	1.47	11.8	8.73
FSTTA [8]	3.26	24.6	9.21
IDEA [12]	2.18	21.3	9.06
CGPI	**0.82**	**16.7**	**8.79**

For compactness, Table 11 reports one representative unseen setting from each benchmark. Complete results under the corresponding feedback and feedback-free settings are provided in Tables 8, 9, and 10.

Table 11 shows that CGPI achieves the strongest SR and SPL across all three representative unseen settings without accessing external feedback. On REVERIE Test Unseen with HAMT, CGPI reaches 34.73 SR and 30.78 SPL, compared with 34.14 SR and 29.07 SPL for FeedTTA, the strongest feedback-driven competitor in SR, and 32.55 SR and 28.38 SPL for ATENA. On R2R Val Unseen with BEVBert, CGPI achieves 79 SR and 70 SPL, outperforming the strongest feedback-driven result of 76 SR and 68 SPL. The same trend holds on R2R-CE Val Unseen with BEVBert, where CGPI obtains 65 SR and 54 SPL, compared with 61 SR and 50 SPL for FeedTTA and 60 SR and 51 SPL for ATENA.

Table 11 compares methods under different test-time information budgets. Feedback-driven approaches additionally use external reward models, episodic feedback, or human interaction, whereas CGPI operates without these signals. Under this feedback-free setting, the results show that observation-grounded, reference-relative credit remains effective across the evaluated benchmarks.

B.3 Adaptation Efficiency

Finally, we evaluate the additional computational cost introduced by test-time adaptation. Since CGPI freezes the pretrained navigation backbone and updates only a lightweight action reranker, its number of trainable parameters remains small. We report the fraction of parameters updated during testing, additional latency per navigation step, and peak GPU memory relative to standard inference.

Table 12 shows that CGPI introduces moderate test-time overhead while updating only the lightweight reranker. In addition to transition-evidence computation and reference-relative credit recovery, CGPI evaluates candidate reranker states on a small cached verification memory. Because the VLN backbone remains frozen and verification reuses cached state-action representations and base logits, this additional computation requires only lightweight reranker forward passes rather than repeated backbone inference. These results demonstrate that the verify-and-retain mechanism can be implemented with limited additional parameter and memory overhead.

B.4 Credit-Signal Diagnostics

Beyond final navigation performance, we evaluate whether the recovered signal is aligned with actual local navigation progress. We conduct this analysis on REVERIE Val Unseen using DUET [6]. All signals are evaluated on the same set of navigation transitions. Geodesic distance is used only for post-hoc diagnostics and is never available to the adaptation algorithm.

Table 13: Quality of different test-time adaptation signals on REVERIE Val Unseen. All signals are evaluated on the same transitions, and geodesic information is used only for post-hoc diagnostics.

Signal	CPC↑	WUR↓
Policy Confidence	0.118	0.412
Raw Grounded Evidence	0.286	0.328
Normalized Grounded Evidence	0.401	0.251
CGPI Credit	**0.572**	**0.154**

Table 14: Robustness to different test-stream orders on REVERIE Val Unseen. Results are averaged over five randomly shuffled episode orders.

Method	SR↑	SPL↑
FSTTA [8]	54.06 ± 0.37	36.28 ± 0.31
IDEA [12]	56.79 ± 0.30	37.91 ± 0.27
CGPI	$\mathbf{58.63 \pm 0.28}$	$\mathbf{40.02 \pm 0.24}$

We use two diagnostic metrics. **Credit–Progress Correlation (CPC)** is the Spearman correlation between the test-time signal and the actual one-step geodesic progress $\Delta d_t = d(o_t, g) - d(o_{t+1}, g)$. **Wrong Update Rate (WUR)** measures the fraction of proposal-triggering interactions whose recovered credit direction disagrees with the oracle local-progress direction. Thus, WUR measures agreement between the proposal direction and *absolute one-step geodesic progress* before candidate verification. CPC and WUR quantify post-hoc alignment with absolute one-step geodesic progress; the reference-relative advantage is characterized separately by the theoretical analysis.

As shown in Table 13, policy confidence is only weakly aligned with actual navigation progress and produces a high wrong-update rate. Grounding the executed action with the true next observation improves both metrics, while source-only normalization further reduces scale mismatch across evidence views. Reference-relative decision credit achieves the highest CPC and the lowest WUR among the evaluated signals, showing stronger alignment with absolute one-step geodesic progress under these post-hoc diagnostics.

B.5 Robustness to Test-Stream Ordering

Since test-time adaptation is performed continuously, the sequence in which test episodes are encountered may affect the evolving adaptation parameters. To examine whether the improvement of CGPI depends on a particular episode ordering, we randomly shuffle the REVERIE Val Unseen test stream using five independent seeds. For each run, the reranker, optimizer state, and verification memory are initialized once and retained across episodes. We report the mean and standard deviation over the five stream orders.

As shown in Table 14, CGPI consistently outperforms competing TTA strategies across different episode orders and exhibits relatively small variance. In particular, its average SR and SPL remain close to the results obtained using the default evaluation order. CGPI therefore remains stable across different orderings of unseen navigation episodes, with average SR and SPL close to the default-order results.

B.6 Proposal-Side Stability

We next examine whether continual credit-guided policy improvement causes excessive policy drift. The retained reranker, optimizer state, and verification memory are initialized once and preserved throughout REVERIE Val Unseen without resetting between episodes. The verify-and-retain mechanism is enabled for all variants in this experiment, while we ablate only the proposal-side KL regularization in Eq. 6 and credit-magnitude selection in Eq. 5.

We measure **Policy Drift (PD)** as

$$\mathrm{PD} = \frac{1}{T} \sum_{t=1}^{T} D_{\mathrm{KL}} \left(\tilde{\pi}_\phi(\cdot|s_t) \| \pi_0(\cdot|s_t) \right). \tag{9}$$

Table 15 shows that removing proposal-side stabilization increases both policy drift and wrong proposal directions. Proposal KL regularization primarily reduces deviation from the pretrained

Table 15: Stability analysis of CGPI on REVERIE Val Unseen. The verify-and-retain mechanism is enabled for all variants; we ablate only proposal-side KL regularization and credit-magnitude selection.

Variant	KL	Sel.	SR↑	SPL↑	PD↓	WUR↓
CGPI w/o Both	–	–	56.45	37.82	0.119	0.281
+ KL Regularization	✓	–	57.10	38.46	0.061	0.259
+ Selective Update	–	✓	57.79	39.35	0.082	0.187
Full CGPI	✓	✓	**58.76**	**40.15**	**0.043**	**0.154**

Table 16: Robustness to synthetic credit noise on REVERIE Val Unseen with DUET. Gaussian perturbations are injected into movement-action credits before candidate proposal. SR and SPL are measured from closed-loop navigation, while AIR and RP characterize candidate quality.

σ_{noise}	Method	SR↑	SPL↑	AIR↑	RP↑
0.10	Direct Commit	56.88	38.21	0.512	–
0.10	Full CGPI	**58.08**	**39.54**	**0.704**	**0.689**
0.20	Direct Commit	55.74	37.18	0.443	–
0.20	Full CGPI	**57.26**	**38.65**	**0.631**	**0.731**

policy, whereas selective proposal reduces WUR by filtering unreliable candidate updates. Their combination provides the strongest proposal stage before verify-and-retain evaluation.

B.7 Robustness to Noisy Improvement Signals

The verify-and-retain mechanism is motivated by the fact that a candidate policy change proposed from a noisy interaction signal need not constitute a useful improvement. We therefore conduct a controlled stress test by perturbing the recovered movement-action credits before candidate proposal.

For each evidence view, we add zero-mean Gaussian perturbation

$$\tilde{C}_t^k = C_t^k + \eta_t^k, \qquad \eta_t^k \sim \mathcal{N}(0, \sigma_{\text{noise}}^2), \tag{10}$$

and aggregate $\tilde{C}_t^k$ in the same way as the standard decision credit. The perturbation is introduced only for this stress test. For each noise level, Direct Commit and Full CGPI are evaluated in their respective closed-loop navigation runs under the same noise magnitude. SR and SPL measure the resulting navigation performance, while AIR and RP use the counterfactual candidate-quality diagnostics defined in Appendix F.

As shown in Table 16, increasing the injected credit noise degrades both navigation performance and candidate quality. However, Full CGPI consistently maintains higher SR, SPL, and AIR than direct commitment at both noise levels. This result indicates that the verify-and-retain stage remains useful when the proposal signal is corrupted, since it can prevent a subset of harmful candidate modifications from being committed. Moreover, the higher RP under stronger corruption shows that a larger fraction of rejected candidates correspond to negative short-horizon oracle progress under the post-hoc counterfactual diagnostic. Together, these results support the role of verification in improving the robustness of credit-guided adaptation under noisy proposal signals.

B.8 Sensitivity to Verification Memory

We analyze how the amount of historical interaction evidence affects candidate verification. All other adaptation settings are kept fixed.

As shown in Table 17, a very small memory provides limited historical evidence for evaluating candidate changes, leading to a higher retention rate but lower AIR. Increasing the capacity to $M_{\text{mem}} = 32$ improves both navigation performance and retained-candidate quality. Further increasing the memory to 64 yields no additional benefit, suggesting that excessively old interactions can become less informative under a continually evolving test stream. We therefore use $M_{\text{mem}} = 32$ in all main experiments.

Table 17: Sensitivity to verification-memory capacity on REVERIE Val Unseen with DUET.

M_{mem}	SR↑	SPL↑	AIR↑	RP↑	RR(%)
16	58.22	39.61	0.716	0.691	65.8
32	**58.76**	**40.15**	**0.741**	**0.716**	61.3
64	58.47	39.88	0.729	0.703	59.6

C Implementation Details

Backbone-compatible adaptation architecture. CGPI is implemented as a lightweight test-time adaptation module on top of the original VLN policy. For all navigation backbones, including HAMT, DUET, BEVBert, and ETPNav, we preserve the original tokenizer, visual feature pipeline, language encoder, navigation representation, candidate construction, action space, and backbone-specific inference procedure. All pretrained backbone parameters remain frozen throughout test-time adaptation. CGPI reuses the original backbone representation pipeline and optimizes only the lightweight residual reranker r_ϕ online.

Backbone-native evidence representations. The evidence functions reuse frozen representations that are already available in the corresponding pretrained VLN backbone. The evidence functions operate on textual and visual representations after the backbone-native frozen projections into the common hidden space consumed by the pretrained multimodal navigation module.

For notational simplicity, let $T(\cdot)$ denote the frozen language pathway, including the backbone's native textual projection into this navigation hidden space. For an instruction or instruction segment, we mean-pool the valid projected token representations produced by $T(\cdot)$ and then apply ℓ_2 normalization. Likewise, let $v(\cdot)$ denote the corresponding frozen visual or candidate representation after the backbone-native projection into the same navigation hidden space.

For a current observation o_t, $v(o_t)$ is obtained by mean pooling the valid panoramic or current-node visual representations provided by the original navigation backbone, followed by ℓ_2 normalization. For a movement candidate a, $v(\xi_t(a))$ is the normalized native candidate representation associated with that action before the final navigation decision stage. For graph-based policies, this corresponds to the candidate-node representation already constructed by the backbone; for continuous navigation, we use the corresponding waypoint-candidate representation.

All similarities are computed between ℓ_2-normalized representations after the pretrained backbone's frozen modality-specific projections. These representations are detached from the computation graph before evidence computation. The pretrained cross-modal representation pipeline remains frozen, with the residual reranker as the only trainable component of CGPI. Candidate evidence is constructed exclusively from information already available at the current navigation state, and no future observation of an unexecuted action is accessed.

Residual action reranker. Let $h_t(a) \in \mathbb{R}^{d_b}$ denote the frozen candidate-specific navigation representation immediately before the original action-scoring stage of backbone b. For backbones with multiple navigation branches, we use the candidate-specific representation associated with the final fused action logits. For STOP, we use the corresponding native STOP representation provided by the backbone.

We instantiate the residual term in Eq. 1 as a two-layer MLP,

$$r_\phi(s_t, a) = W_2\,\mathrm{GELU}\left(W_1 h_t(a) + b_1\right) + b_2, \tag{11}$$

where W_1 maps the backbone-specific input dimension d_b to a hidden dimension of 256, and W_2 maps the hidden representation to a scalar residual logit. The same reranker architecture is used for all backbones; only its input dimension follows the native hidden dimension of the corresponding navigation policy. No additional dropout or trainable normalization layer is introduced.

The first linear layer is initialized with Xavier-uniform initialization and zero bias. The output layer is zero-initialized, i.e., $W_2 = 0$ and $b_2 = 0$ at the beginning of each evaluation stream. Consequently, $r_{\phi_0}(s_t, a) = 0$ and $\tilde{\pi}_{\phi_0} = \pi_0$ before the first accepted test-time update. This initialization makes the initial adapted policy exactly match the pretrained navigation policy.

Online optimization. We optimize only the reranker parameters using AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We set the optimizer weight decay to zero because the proposal objective

in Eq. 6 already contains the explicit ℓ_2 regularization term $\lambda_{\text{reg}}\|\phi\|_2^2$. Gradients of the reranker are clipped to a maximum global norm of 1.0.

Each eligible test-time interaction produces at most one candidate optimizer step with batch size 1. The resulting candidate parameters are not immediately committed; they are first evaluated by the verify-and-retain mechanism. The learning rate is 5×10^{-6} for REVERIE and 1×10^{-6} for R2R and R2R-CE.

Evidence normalization and STOP margin. For each movement evidence term, the normalization statistics (μ_k, σ_k) are computed once on the source training split for each backbone–benchmark configuration and remain fixed during testing.

For STOP, we first compute the pre-margin score

$$u_t^{\text{stop}} = S_t^{\text{comp}} - M_t^{\text{move}}.$$

The statistics $(\mu_{\text{stop}}, \sigma_{\text{stop}})$ are computed from this pre-margin source score. The conservative margin is applied only after standardization, as defined in Eq. 28, with $\delta_{\text{stop}} = 0.25$ for all benchmarks. No target locations, shortest paths, success labels, or target-domain outcomes are used for evidence normalization or online adaptation.

Candidate-state management. At test time, we distinguish the currently retained reranker parameters ϕ_t from candidate parameters ϕ_t^+ proposed by the current interaction. For every candidate proposal, we preserve both the currently retained reranker parameters and the corresponding optimizer state ω_t. A proposal step produces (ϕ_t^+, ω_t^+). If the candidate satisfies the verification and proposal-state reference-deviation conditions in Eq. 8, both candidate states are committed. Otherwise, both the reranker and optimizer states remain at (ϕ_t, ω_t). Rolling back the optimizer state together with the model parameters prevents rejected proposals from influencing subsequent adaptation through optimizer momentum or adaptive moments.

Verification memory. We maintain a bounded FIFO verification memory of capacity M_{mem}. A transition is eligible for insertion when its recovered credit satisfies $|A_t| \geq \delta_A$. The current transition is always inserted *after* the retain-or-rollback decision and therefore cannot be used to verify the candidate update that it generated. Candidate verification begins only after the memory contains at least $N_{\min}$ eligible past interactions. Before this warm-up is completed, eligible interactions are accumulated while the retained policy remains unchanged.

Under the default continual-stream protocol, the verification memory, retained reranker, and optimizer state are preserved across episodes. For efficiency, the memory stores frozen candidate-specific state-action representations, base logits, valid-action masks, executed-action indices, and recovered credits. These cached quantities are sufficient to recompute the lightweight reranker distribution during verification, avoiding repeated backbone inference.

Hyperparameter selection. We use the corresponding source-side Val Seen split for hyperparameter selection and fix all settings before evaluation on unseen environments. SPL is used as the source-side selection criterion. Within each benchmark, the selected hyperparameters are shared across navigation backbones rather than tuned separately for each backbone.

We use a limited search over learning rate $\{5 \times 10^{-7}, 1 \times 10^{-6}, 5 \times 10^{-6}\}$, credit threshold $\delta_A \in \{0.15, 0.20, 0.25\}$, reference-deviation threshold $\delta_{\text{KL}} \in \{0.04, 0.05, 0.10\}$, KL coefficient $\lambda_{\text{KL}} \in \{0.05, 0.10\}$, verification-memory capacity $M_{\text{mem}} \in \{16, 32, 64\}$, warm-up size $N_{\min} \in \{4, 8, 16\}$, and retention threshold $\delta_{\text{keep}} \in \{0, 1 \times 10^{-4}, 5 \times 10^{-4}\}$. The structural settings $\alpha = 0.5$, $A_{\max} = 2.0$, $\lambda_{\text{reg}} = 1 \times 10^{-4}$, reranker hidden dimension 256, gradient clipping norm 1.0, and $\delta_{\text{stop}} = 0.25$ are fixed across benchmarks. No target-domain success labels, target locations, shortest paths, or external outcome feedback are used for hyperparameter selection. Table 18 summarizes the final hyperparameter settings used for all benchmarks.

Hardware and reproducibility. All experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU. Reported results are averaged over three independent random seeds. For each seed, the reranker parameters, optimizer state, and verification memory are initialized at the beginning of the evaluation split. Under the default continual-stream protocol, all three are subsequently retained across episodes, while candidate modifications are committed only after passing the verify-and-retain criterion.

Apart from the CGPI components described above, all dataset preprocessing, candidate construction, pretrained visual and language representations, action spaces, and backbone-specific inference settings strictly follow the original configurations of the corresponding pretrained navigation policies.

Table 18: Main hyperparameters used for test-time adaptation.

Hyperparameter	**REVERIE**	**R2R**	**R2R-CE**
Reranker hidden dim.	256	256	256
Learning rate	5×10^{-6}	1×10^{-6}	1×10^{-6}
Gradient clip	1.0	1.0	1.0
α	0.5	0.5	0.5
$A_{\max}$	2.0	2.0	2.0
λ_{KL}	0.05	0.05	0.10
λ_{reg}	1×10^{-4}	1×10^{-4}	1×10^{-4}
δ_A	0.20	0.20	0.25
δ_{KL}	0.05	0.05	0.04
δ_{stop}	0.25	0.25	0.25
M_{mem}	32	32	32
$N_{\min}$	8	8	8
δ_{keep}	1×10^{-4}	1×10^{-4}	1×10^{-4}

D Details of Evidence Views

This appendix provides concrete instantiations of the four movement evidence views used by CGPI: Goal, Progress, Grounding, and Risk. Each view produces a current-state candidate hypothesis $\hat{e}_t^k(a)$ and, after executing a movement action, an observation-grounded value e_t^k. Candidate hypotheses use only information already available at state s_t and never access future observations of unexecuted actions. All evidence values are standardized using the source-training statistics defined in Eq. 3. STOP does not produce a post-action observation and is therefore handled separately using the dedicated completion-versus-movement construction given at the end of this section.

Fixed instruction contexts. As described in Appendix C, the evidence functions reuse the frozen representations of the underlying navigation backbone. Let $T(x) = \{t_j(x)\}_{j=1}^{L_x}$ denote the valid token representations obtained after passing text x through the backbone's frozen language pathway and its native projection into the navigation hidden space. We define

$$\mathrm{Pool}(T(x)) = \frac{\frac{1}{L_x}\sum_{j=1}^{L_x} t_j(x)}{\left\| \frac{1}{L_x}\sum_{j=1}^{L_x} t_j(x) \right\|_2 + \epsilon}, \tag{12}$$

i.e., mean pooling over valid projected tokens followed by ℓ_2 normalization.

The full-instruction representation is therefore

$$q_I = \mathrm{Pool}(T(I)). \tag{13}$$

To obtain a small set of local language contexts without introducing an online parser or progress tracker, we deterministically split the raw instruction at punctuation boundaries (comma, semicolon, and sentence boundary), remove empty spans, and preserve the original order. If no boundary is present, the complete instruction forms a single segment. For the resulting text spans $\{\tilde{c}_m\}_{m=1}^{M}$, we define

$$c_m = \mathrm{Pool}(T(\tilde{c}_m)), \qquad m = 1, \ldots, M. \tag{14}$$

These contexts are computed once from the instruction and remain fixed throughout the episode; in particular, we do not infer whether a segment has been completed and do not maintain a learned or threshold-based active sub-instruction state.

Let $v(o)$ denote the ℓ_2-normalized backbone-native representation of observation o after the corresponding frozen visual projection, and let $v(\xi_t(a))$ denote the normalized candidate representation already available for movement action a at state s_t. Thus, all similarities below compare textual and visual representations after the pretrained backbone's frozen projections into the navigation hidden space.

We use a fixed goal-bearing context q_{goal}. For REVERIE, whose instruction is already goal-oriented, we use the full-instruction representation q_I. For R2R and R2R-CE, we use the final fixed instruction segment c_M, which corresponds to the terminal portion of the route description.

For REVERIE, $q_{\mathrm{goal}} = q_I$, so Goal and Grounding coincide by construction. Their joint inclusion assigns additional weight to full-instruction compatibility while preserving the same evidence-aggregation rule across benchmarks. For R2R and R2R-CE, the two terms remain distinct because

Goal uses the terminal instruction segment c_M, whereas Grounding uses the complete instruction representation q_I.

The observation history is $\mathcal{H}_t = \{o_0, \ldots, o_t\}$. All similarities below use cosine similarity between the normalized representations defined above.

Goal evidence. Goal evidence measures whether the transition strengthens visual support for the fixed goal-bearing context. The grounded evidence is

$$e_t^{\text{goal}} = \text{sim}(q_{\text{goal}}, v(o_{t+1})) - \text{sim}(q_{\text{goal}}, v(o_t)). \tag{15}$$

Before execution, the corresponding candidate hypothesis is

$$\hat{e}_t^{\text{goal}}(a) = \text{sim}(q_{\text{goal}}, v(\xi_t(a))) - \text{sim}(q_{\text{goal}}, v(o_t)). \tag{16}$$

This term captures destination-oriented semantic change without requiring future observations or target-location supervision.

Progress evidence. Progress evidence measures whether a transition increases compatibility with the most strongly supported fixed instruction segment. Because the segment set is fixed for the whole episode, this term does not require a separate clause-completion detector. The grounded evidence is

$$e_t^{\text{progress}} = \max_{1 \le m \le M} \text{sim}(c_m, v(o_{t+1})) - \max_{1 \le m \le M} \text{sim}(c_m, v(o_t)). \tag{17}$$

The current-state candidate hypothesis is

$$\hat{e}_t^{\text{progress}}(a) = \max_{1 \le m \le M} \text{sim}(c_m, v(\xi_t(a))) - \max_{1 \le m \le M} \text{sim}(c_m, v(o_t)). \tag{18}$$

Thus, Progress is defined as a lightweight semantic-change signal over the fixed instruction segments.

Grounding evidence. Grounding evidence measures whether the transition becomes more compatible with the complete navigation instruction. Using the fixed full-instruction representation q_I from Eq. 13, we define

$$e_t^{\text{grounding}} = \text{sim}(q_I, v(o_{t+1})) - \text{sim}(q_I, v(o_t)), \tag{19}$$

with candidate hypothesis

$$\hat{e}_t^{\text{grounding}}(a) = \text{sim}(q_I, v(\xi_t(a))) - \text{sim}(q_I, v(o_t)). \tag{20}$$

For R2R and R2R-CE, Grounding complements Goal by measuring compatibility with the complete instruction rather than only the terminal goal-bearing segment. For REVERIE, the two terms coincide because $q_{\text{goal}} = q_I$, as noted above.

Risk evidence. Risk evidence penalizes transitions that return to observations highly similar to previously visited states. For the executed movement action, we define

$$e_t^{\text{risk}} = -\max_{o_j \in \mathcal{H}_{t-1}} \text{sim}(v(o_{t+1}), v(o_j)). \tag{21}$$

The corresponding candidate hypothesis is

$$\hat{e}_t^{\text{risk}}(a) = -\max_{o_j \in \mathcal{H}_{t-1}} \text{sim}(v(\xi_t(a)), v(o_j)). \tag{22}$$

When $\mathcal{H}_{t-1} = \emptyset$, no historical observation is available for comparison. We therefore set

$$e_t^{\text{risk}} = 0, \qquad \hat{e}_t^{\text{risk}}(a) = 0. \tag{23}$$

A candidate that is highly similar to previously visited observations therefore receives less favorable evidence, discouraging repeated or locally unproductive behavior.

STOP-specific credit. STOP does not produce a new observation and therefore cannot be grounded using Eq. 2. We use the same fixed goal-bearing context to instantiate the current-state completion score as

$$S_t^{\text{comp}} = \text{sim}(q_{\text{goal}}, v(o_t)). \tag{24}$$

For each non-STOP candidate, the view-averaged movement score is

$$M_t(a) = \frac{1}{K} \sum_{k=1}^{K} \bar{\hat{e}}_t^k(a), \qquad a \in \mathcal{A}_t^{\text{mov}}, \tag{25}$$

and the strongest continue-moving alternative is

$$M_t^{\text{move}} = \max_{a \in \mathcal{A}_t^{\text{mov}}} M_t(a). \tag{26}$$

We first define the pre-margin STOP score

$$u_t^{\text{stop}} = S_t^{\text{comp}} - M_t^{\text{move}}. \tag{27}$$

We standardize this score using source-training statistics and apply the conservative stopping margin on the standardized scale:

$$A_t = \frac{u_t^{\text{stop}} - \mu_{\text{stop}}}{\sigma_{\text{stop}} + \epsilon} - \delta_{\text{stop}}, \qquad a_t = \text{STOP}. \tag{28}$$

The statistics $(\mu_{\text{stop}}, \sigma_{\text{stop}})$ are computed once from the pre-margin source score u_t^{stop} and remain fixed during deployment. Applying $\delta_{\text{stop}} > 0$ after standardization preserves its conservative effect under source centering.

STOP uses the dedicated completion-versus-movement construction: Eq. 27 compares current completion evidence against the strongest continue-moving alternative, and Eq. 28 converts this comparison into a conservative local credit.

E Verification and Retention Details

Memory construction. Let $\mathcal{V}_t$ denote the verification memory available before processing the candidate generated at step t. An interaction is eligible for insertion when its recovered credit satisfies

$$|A_i| \geq \delta_A. \tag{29}$$

The memory follows a FIFO policy with maximum capacity M_{mem}, such that verification emphasizes relatively recent interaction evidence under the current test-time distribution.

Crucially, the current tuple (s_t, a_t, A_t) is inserted only after the candidate generated by this same interaction has already been accepted or rejected. Therefore,

$$(s_t, a_t, A_t) \notin \mathcal{V}_t \tag{30}$$

during verification of ϕ_t^+. This leave-current-interaction-out construction prevents a candidate from being evaluated on the same interaction that produced its gradient.

Warm-up. Verification starts after the memory contains at least $N_{\min}$ eligible past interactions. Before this point, credit-supported interactions are accumulated but no candidate modification is committed. Once $|\mathcal{V}_t| \geq N_{\min}$, an interaction satisfying the proposal condition in Eq. 5 may generate a candidate update. Under the default continual-stream protocol, $\mathcal{V}_t$ is retained across episodes.

Candidate evaluation. Given retained parameters ϕ_t and candidate parameters ϕ_t^+, both rerankers are evaluated on exactly the same verification memory. For each stored interaction i, the contribution

$$A_i \log \tilde{\pi}_\phi(a_i|s_i) \tag{31}$$

has the desired directional behavior. When $A_i > 0$, increasing the probability of the stored action raises the score; when $A_i < 0$, decreasing that probability also raises the score. The aggregate difference between the candidate and retained policy is therefore the verification gain defined in Eq. 7.

Because the VLN backbone is frozen, verification does not require replaying the environment or recomputing backbone features. For every stored interaction, we cache the frozen state-action representations, base logits, valid-action mask, executed-action index, and recovered credit. The verifier therefore only recomputes the lightweight reranker distribution.

Retaining model and optimizer states. The proposal step jointly produces candidate model and optimizer states (ϕ_t^+, ω_t^+). If Eq. 8 is satisfied, both are retained:

$$(\phi_{t+1}, \omega_{t+1}) = (\phi_t^+, \omega_t^+). \tag{32}$$

Otherwise,

$$(\phi_{t+1}, \omega_{t+1}) = (\phi_t, \omega_t). \tag{33}$$

Rolling back the optimizer state together with the reranker parameters prevents a rejected candidate from influencing later proposals through momentum or adaptive optimizer statistics.

Scope of verification. The verifier is defined solely from the agent's previously observed interactions and recovered decision credits. Its criterion measures consistency with the finite local credit-based surrogate stored in $\mathcal{V}_t$. Geodesic distance, target locations, benchmark success conditions, expert actions, human feedback, and external reward models are used neither by the verifier nor by the adaptation procedure.

F Diagnostic Metrics

The diagnostic quantities in this section are used only for post-hoc evaluation and never participate in test-time adaptation. In particular, geodesic distance and goal location are not available to CGPI during navigation.

Credit–Progress Correlation. For each executed movement action, define one-step geodesic progress as

$$\Delta d_t = d(o_t, g) - d(o_{t+1}, g). \tag{34}$$

We report

$$\mathrm{CPC} = \mathrm{Spearman}(A_t, \Delta d_t), \tag{35}$$

computed on the same fixed set of transitions for all compared signals.

Wrong Update Rate. Let $\mathcal{U}$ denote the set of steps that trigger a candidate proposal before the verify-and-retain stage. For CGPI and its credit ablations, this corresponds to interactions that satisfy the proposal-selection criterion associated with the evaluated signal. For a movement step, define the oracle local direction as

$$y_t = \mathrm{sign}(\Delta d_t). \tag{36}$$

For STOP, $y_t = +1$ when the benchmark success condition is satisfied at the current state and $y_t = -1$ otherwise. We define

$$\mathrm{WUR} = \frac{\sum_{t \in \mathcal{U}} \mathbf{1}[\mathrm{sign}(A_t) \neq y_t]}{|\mathcal{U}| + \epsilon}. \tag{37}$$

Lower WUR indicates that fewer candidate proposals are generated from credit directions inconsistent with oracle local progress.

Policy Drift. Policy drift is the average deviation of the adapted action distribution from the frozen policy:

$$\mathrm{PD} = \frac{1}{T} \sum_{t=1}^{T} D_{\mathrm{KL}} \left(\tilde{\pi}_\phi(\cdot|s_t) \| \pi_0(\cdot|s_t) \right). \tag{38}$$

Post-hoc candidate improvement gain. To directly diagnose whether a retained candidate constitutes a useful policy change, we perform simulator-only counterfactual evaluation. This diagnostic is never used by the adaptation algorithm.

For a movement-action candidate generated after reaching state s_{t+1}, we clone the complete navigation state, including the simulator state, trajectory history, and accumulated navigation structure, and independently roll out the retained policy ϕ_t and candidate policy ϕ_t^+ for the same horizon H. We use a fixed horizon of $H = 5$ navigation steps for all reported counterfactual analyses. Both policies are evaluated from the identical cloned decision context using greedy action selection. Let

$$R_H(\phi; s_{t+1}) = d(s_{t+1}, g) - d(s^{\phi}_{t+H+1}, g) \tag{39}$$

denote short-horizon geodesic progress, where g is the benchmark goal used only for post-hoc evaluation. The candidate improvement gain is

$$\Gamma_t^{(H)} = R_H(\phi_t^+; s_{t+1}) - R_H(\phi_t; s_{t+1}). \tag{40}$$

A positive $\Gamma_t^{(H)}$ indicates that the candidate yields better short-horizon oracle progress than the previously retained policy from the same state. STOP proposals are excluded from this diagnostic because they do not produce a subsequent state within the same episode.

Accepted Improvement Rate. Let $\mathcal{R}$ denote the set of retained movement-action candidates. We define

$$\mathrm{AIR} = \frac{\sum_{t\in\mathcal{R}} \mathbf{1}\left[\Gamma_t^{(H)} > 0\right]}{|\mathcal{R}| + \epsilon}. \tag{41}$$

Higher AIR indicates that a larger fraction of retained candidate modifications correspond to positive post-hoc short-horizon improvement.

Accepted Improvement Gain. We additionally report the mean oracle gain among retained candidates:

$$\mathrm{AIG} = \frac{1}{|\mathcal{R}| + \epsilon} \sum_{t\in\mathcal{R}} \Gamma_t^{(H)}. \tag{42}$$

Rejection Precision. Let $\mathcal{B}$ denote the set of rejected movement-action candidates. We define

$$\mathrm{RP} = \frac{\sum_{t\in\mathcal{B}} \mathbf{1}\left[\Gamma_t^{(H)} < 0\right]}{|\mathcal{B}| + \epsilon}. \tag{43}$$

Higher RP indicates that rejected candidates are more frequently those that would have degraded short-horizon oracle progress.

Retention Rate. Let $\mathcal{P}$ denote all movement-action candidate proposals. The retention rate is

$$\mathrm{RR} = \frac{\sum_{t\in\mathcal{P}} r_t}{|\mathcal{P}| + \epsilon}. \tag{44}$$

This metric measures how selectively the verifier commits candidate modifications.

G Theoretical Analysis and Proofs

Our analysis characterizes two aspects of CGPI: the reliability of local improvement identification and the local-surrogate interpretation of the verify-and-retain criterion.

G.1 Reference Centering Removes Shared State Bias

Proposition G.1 (Reference centering removes shared state bias)**.** *Let $g_t^k(a)$ denote the latent local utility of movement action $a \in \mathcal{A}_t^{\mathrm{mov}}$ under evidence view k. Suppose the normalized grounded evidence and movement-candidate hypotheses admit*

$$\bar{e}_t^k = g_t^k(a_t) + h_t^k + \varepsilon_t^k, \tag{45}$$

$$\bar{\tilde{e}}_t^k(a) = g_t^k(a) + h_t^k + \xi_t^k(a), \qquad a \in \mathcal{A}_t^{\mathrm{mov}}, \tag{46}$$

where h_t^k is an action-independent state-dependent shift, ε_t^k is grounded-evidence error, and $\xi_t^k(a)$ is candidate-evidence error. Then

$$C_t^k = G_t^k + \varepsilon_t^k - \sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t)\xi_t^k(a), \tag{47}$$

where

$$G_t^k = g_t^k(a_t) - \sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t) g_t^k(a) \tag{48}$$

is the ideal local advantage of the executed movement action relative to the renormalized frozen-policy movement reference. In particular, the shared state-dependent shift h_t^k cancels exactly.

Proof. Substituting Eqs. 45 and 46 into Eq. 4 gives

$$\begin{aligned} C_t^k = {} & g_t^k(a_t) + h_t^k + \varepsilon_t^k \\ & - \sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t)\left(g_t^k(a) + h_t^k + \xi_t^k(a)\right). \end{aligned} \tag{49}$$

Since the movement-only frozen-policy reference is normalized,

$$\sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t) = 1, \tag{50}$$

the common state term cancels:

$$\begin{aligned} C_t^k = g_t^k(a_t) &- \sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t) g_t^k(a) \\ &+ \varepsilon_t^k - \sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t)\xi_t^k(a). \end{aligned} \tag{51}$$

Using Eq. 48 proves Eq. 47. □

The proposition establishes shift cancellation for normalized reference weights. We instantiate the reference with the frozen-policy movement distribution, which provides a stationary, non-self-referential local baseline throughout adaptation.

G.2 View-Level Credit Error

From Proposition G.1,

$$C_t^k - G_t^k = \varepsilon_t^k - \sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t)\xi_t^k(a). \tag{52}$$

Suppose

$$|\varepsilon_t^k| \le \epsilon_{\mathrm{g}}^k, \qquad |\xi_t^k(a)| \le \epsilon_{\mathrm{c}}^k(a), \quad a \in \mathcal{A}_t^{\mathrm{mov}}. \tag{53}$$

Then

$$|C_t^k - G_t^k| \le |\varepsilon_t^k| + \sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t)|\xi_t^k(a)| \tag{54}$$

$$\le \epsilon_{\mathrm{g}}^k + \sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t)\epsilon_{\mathrm{c}}^k(a). \tag{55}$$

Hence a valid view-level error bound is

$$B_t^k = \epsilon_{\mathrm{g}}^k + \sum_{a\in\mathcal{A}_t^{\mathrm{mov}}} \pi_0^{\mathrm{mov}}(a|s_t)\epsilon_{\mathrm{c}}^k(a). \tag{56}$$

G.3 Sign Consistency of Improvement Identification

Proposition G.2 (Sign consistency under bounded evidence error)**.** *Suppose each movement evidence view satisfies*

$$|C_t^k - G_t^k| \le B_t^k. \tag{57}$$

Let

$$A_t = \frac{1}{K}\sum_{k=1}^{K} C_t^k, \qquad G_t = \frac{1}{K}\sum_{k=1}^{K} G_t^k, \tag{58}$$

and define

$$B_t = \frac{1}{K}\sum_{k=1}^{K} B_t^k. \tag{59}$$

Then

$$|A_t - G_t| \le B_t. \tag{60}$$

Consequently, whenever

$$|G_t| > B_t, \tag{61}$$

the recovered credit preserves the direction of the ideal local advantage:

$$\mathrm{sign}(A_t) = \mathrm{sign}(G_t). \tag{62}$$

Proof. By the triangle inequality,

$$|A_t - G_t| = \left| \frac{1}{K} \sum_{k=1}^{K} (C_t^k - G_t^k) \right| \tag{63}$$

$$\leq \frac{1}{K} \sum_{k=1}^{K} |C_t^k - G_t^k| \tag{64}$$

$$\leq \frac{1}{K} \sum_{k=1}^{K} B_t^k = B_t. \tag{65}$$

If $G_t > B_t$, then

$$A_t \geq G_t - |A_t - G_t| \geq G_t - B_t > 0. \tag{66}$$

The case $G_t < -B_t$ is symmetric. □

G.4 Interpretation of the Proposal Gate

The credit-magnitude component of Eq. 5 requires

$$|A_t| \geq \delta_A. \tag{67}$$

Suppose

$$|A_t - G_t| \leq B_t \tag{68}$$

and

$$\delta_A > B_t. \tag{69}$$

If $A_t \geq \delta_A$, then

$$G_t \geq A_t - |A_t - G_t| \geq \delta_A - B_t > 0. \tag{70}$$

Likewise, if $A_t \leq -\delta_A$, then $G_t < 0$. Therefore,

$$|A_t| \geq \delta_A \quad \text{and} \quad \delta_A > B_t \quad \Longrightarrow \quad \operatorname{sign}(A_t) = \operatorname{sign}(G_t). \tag{71}$$

The bound motivates filtering weak-magnitude credits, whose directions are most sensitive to evidence error. CGPI implements this principle through the fixed threshold δ_A without estimating B_t online.

G.5 Verified Local-Surrogate Improvement

We next analyze the verify-and-retain criterion. For every interaction i in the verification memory, define the log-probability change induced by a candidate as

$$\Delta_i^t = \log \tilde{\pi}_{\phi_t^+}(a_i|s_i) - \log \tilde{\pi}_{\phi_t}(a_i|s_i). \tag{72}$$

From Eq. 7, the verification gain can be written as

$$\Delta_t^{\text{ver}} = \frac{1}{|\mathcal{V}_t|} \sum_{i \in \mathcal{V}_t} A_i \Delta_i^t. \tag{73}$$

Let G_i denote the corresponding latent local advantage. We define the latent local-surrogate change

$$\Delta_t^{\text{lat}} = \frac{1}{|\mathcal{V}_t|} \sum_{i \in \mathcal{V}_t} G_i \Delta_i^t. \tag{74}$$

Proposition G.3 (Verified local-surrogate improvement)**.** *Suppose every verification-memory credit satisfies*

$$|A_i - G_i| \leq B_i. \tag{75}$$

Then

$$\left|\Delta_t^{\text{ver}} - \Delta_t^{\text{lat}}\right| \leq \frac{1}{|\mathcal{V}_t|} \sum_{i \in \mathcal{V}_t} B_i |\Delta_i^t|. \tag{76}$$

Consequently, if

$$\Delta_t^{\text{ver}} > \frac{1}{|\mathcal{V}_t|} \sum_{i \in \mathcal{V}_t} B_i |\Delta_i^t|, \tag{77}$$

then

$$\Delta_t^{\mathrm{lat}} > 0. \tag{78}$$

Proof. Using Eqs. 73 and 74,

$$\left|\Delta_t^{\mathrm{ver}} - \Delta_t^{\mathrm{lat}}\right| = \left|\frac{1}{|\mathcal{V}_t|}\sum_i (A_i - G_i)\Delta_i^t\right| \tag{79}$$

$$\leq \frac{1}{|\mathcal{V}_t|}\sum_i |A_i - G_i||\Delta_i^t| \tag{80}$$

$$\leq \frac{1}{|\mathcal{V}_t|}\sum_i B_i|\Delta_i^t|. \tag{81}$$

Therefore,

$$\Delta_t^{\mathrm{lat}} \geq \Delta_t^{\mathrm{ver}} - \left|\Delta_t^{\mathrm{ver}} - \Delta_t^{\mathrm{lat}}\right| \tag{82}$$

$$> 0 \tag{83}$$

whenever Eq. 77 holds. □

Proposition G.3 provides a local interpretation of the verify-and-retain mechanism. Under bounded credit-estimation error, a sufficiently positive credit-weighted verification gain implies improvement of the corresponding latent local-advantage surrogate. In deployment, CGPI uses the fixed empirical margin δ_{keep} in Eq. 8 because the error bounds B_i are unavailable online. This provides a practical criterion for retaining locally supported candidate updates.

G.6 Reference-Deviation Control

For a candidate satisfying the second condition in Eq. 8,

$$D_{\mathrm{KL}}\left(\tilde{\pi}_{\phi_t^+}(\cdot|s_t)\|\pi_0(\cdot|s_t)\right) \leq \delta_{\mathrm{KL}}. \tag{84}$$

Pinsker's inequality gives

$$\left\|\tilde{\pi}_{\phi_t^+}(\cdot|s_t) - \pi_0(\cdot|s_t)\right\|_{\mathrm{TV}} \leq \sqrt{\frac{\delta_{\mathrm{KL}}}{2}}. \tag{85}$$

Thus, verification evaluates whether a candidate is supported by previous interaction evidence, while the KL condition bounds its deviation from the frozen policy at the proposal state. Accordingly, the analysis characterizes local policy deviation rather than long-horizon navigation performance.

H More Visualizations

Figure 5 presents additional real-world trajectories executed by the Unitree Go2 robot. For each instruction, we show synchronized third-person views and egocentric observations along the trajectory. The instructions involve multiple sequential navigation steps, including changing direction, passing specified landmarks, and stopping at the instructed target. Despite being trained only in simulated environments, the navigation policy successfully executes these multi-step instructions in the shown real-world scenes without real-world fine-tuning or external outcome feedback. These examples provide additional qualitative evidence of the zero-shot sim-to-real feasibility of CGPI.

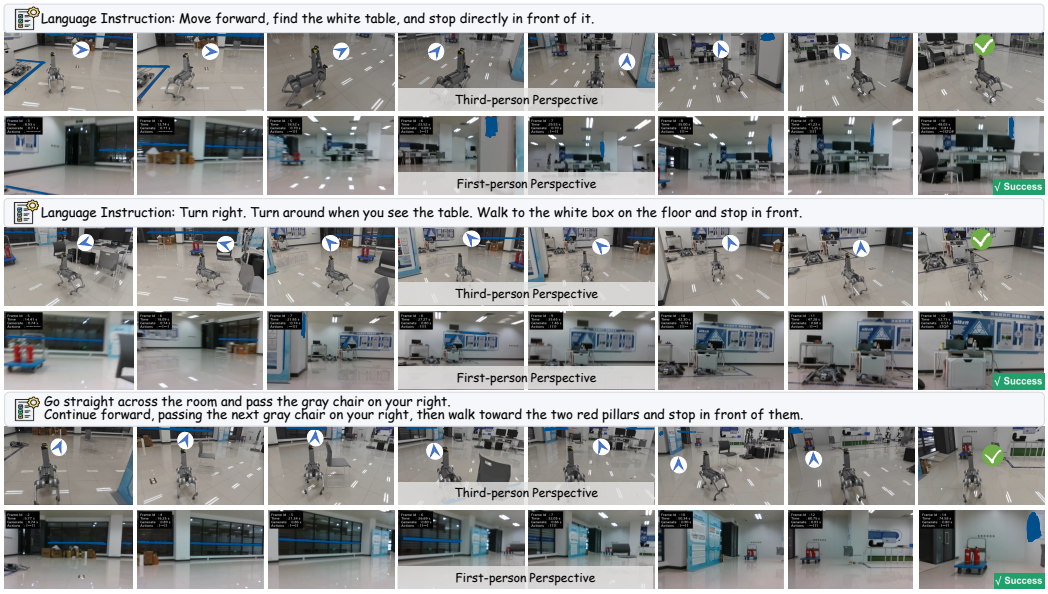

Figure 5: **Additional real-world qualitative results of CGPI on a Unitree Go2 robot.** We show third-person views and corresponding egocentric observations along three executed trajectories. The agent follows multi-step language instructions involving directional changes and landmark-based navigation, and reaches the instructed destinations in the shown real-world environments.

I FULL ONLINE ALGORITHM

Algorithm 1 Credit-Guided Policy Improvement with Verify-and-Retain Adaptation

Input: frozen policy π_0, fixed evidence functions $\{\hat{E}^k, E^k\}_{k=1}^K$, source normalization statistics, lightweight reranker r_ϕ, optimizer state ω, verification-memory capacity M_{mem}, warm-up size $N_{\min}$, and thresholds δ_A, δ_{keep}, δ_{KL}.
Initialize retained reranker parameters ϕ and optimizer state ω.
Initialize verification memory $\mathcal{V} \leftarrow \emptyset$.
for each test episode X_i **do**
  Initialize only the episode trajectory history.
  for $t = 0, 1, \ldots$ **do**
    Construct state s_t and candidate set $\mathcal{A}_t$.
    Compute frozen policy $\pi_0(\cdot|s_t)$.
    Compute adapted policy $\tilde{\pi}_\phi(\cdot|s_t)$ using Eq. 1.
    Compute current-state candidate evidence $\hat{e}_t^k(a)$ for available movement actions.
    Select and execute action a_t.
    if $a_t = \text{STOP}$ **then**
      Compute STOP credit A_t using Eq. 28.
    else
      Observe actual next observation o_{t+1}.
      Compute grounded evidence $e_t^k = E^k(s_t, a_t, o_{t+1})$.
      Normalize candidate and grounded evidence using Eq. 3.
      Compute frozen-policy references b_t^k and aggregate decision credit A_t using Eq. 4.
    end if
    Compute proposal mask m_t^{prop} using Eq. 5.
    if $m_t^{\text{prop}} = 1$ **and** $|\mathcal{V}| \geq N_{\min}$ **then**
      Preserve currently retained (ϕ, ω).
      Compute proposal loss $\mathcal{L}_{\text{prop}}(t)$ using Eq. 6.
      Generate candidate (ϕ^+, ω^+) by one optimizer step on $\mathcal{L}_{\text{prop}}(t)$.
      Evaluate $J(\phi^+)$ and $J(\phi)$ on the same verification memory $\mathcal{V}$.
      Compute $\Delta_t^{\text{ver}} = J(\phi^+) - J(\phi)$.
      Compute candidate deviation $D_t(\phi^+)$.
      if $\Delta_t^{\text{ver}} \geq \delta_{\text{keep}}$ **and** $D_t(\phi^+) \leq \delta_{\text{KL}}$ **then**
        Retain candidate: $(\phi, \omega) \leftarrow (\phi^+, \omega^+)$.
      else
        Roll back by leaving (ϕ, ω) unchanged.
      end if
    end if
    if $|A_t| \geq \delta_A$ **then**
      Cache the current frozen reranker inputs, executed-action index, and credit A_t.
      Insert the current record into $\mathcal{V}$ *after* candidate verification.
      If $|\mathcal{V}| > M_{\text{mem}}$, remove the oldest record.
    end if
    if $a_t = \text{STOP}$ **then**
      Terminate the episode.
    else
      Append (s_t, a_t, o_{t+1}) to the trajectory history.
    end if
  end for
end for

J SCOPE AND LIMITATIONS

CGPI is designed for feedback-free online adaptation using local interaction evidence, and its effectiveness depends on the quality of the backbone-native candidate representations and recovered decision credit. The current verification mechanism uses a finite memory of recent credit-supported interactions and focuses on local policy improvement. Extending CGPI to longer-horizon interaction evidence and more adaptive verification mechanisms is a promising direction for future work.