ASK WHAT IF BEFORE COMMITTING: COUNTERFACTUAL STABILITY FOR VISION-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language navigation in continuous environments often employs online topological planners to rank candidate goals on an evolving map built from partial observations. Yet a high policy score does not reveal whether a goal remains supported when the computation behind that score changes. We propose CTPNav, a framework that measures decision-time counterfactual stability by combining mean support with a penalty for response dispersion under controlled internal interventions. At the representation level, standardized edge responses gate the learned attention residual, while their dispersion adjusts the association radius for the next map update. At the action level, CTPNav probes goal competition, geometric bias, and instruction grounding to assess the stability of candidate goals. The resulting responses provide detached targets for policy learning and guide a bounded inference-time correction of action scores. On R2R-CE Val-Unseen, CTPNav achieves 61% SR and 51% SPL. On RxR-CE Val-Unseen, it achieves 54.53% SR, 45.98% SPL, and 63.19% nDTW. These results demonstrate the value of intervention-response stability for learning and topological planning. Our code is available at `https://anonymous.4open.science/r/CTPNav-AD2F`.

1 INTRODUCTION

Vision-language navigation in continuous environments (VLN-CE) requires an agent to follow natural-language instructions through sequential decisions from partial observations (Anderson et al., 2018b; Ku et al., 2020; Krantz et al., 2020). Online topological planners organize explored locations and candidate frontiers into an evolving graph, select high-level goals, and rely on a local controller for execution (Krantz et al., 2021; Hong et al., 2022; An et al., 2025). At each decision step, the graph policy assigns scores to candidate goals and selects an action accordingly.

Policy scores, however, only describe the current ranking. Two candidate goals may receive similar scores yet respond differently when geometric or instruction-conditioned information is perturbed: one may retain its support, while the other may depend heavily on a particular computational component. Such differences are not captured by the original scores alone. This distinction matters in online navigation, where a selected goal influences not only the current movement but also the observations and graph updates available for subsequent decisions. Examining how candidate preferences respond to controlled internal interventions therefore provides information beyond a single policy evaluation.

Motivated by this observation, we propose *decision-time counterfactual stability*, a criterion that evaluates action support under controlled changes to the underlying computation. At a fixed navigation state, we compare the original policy response with responses obtained under a specified set of internal interventions. The criterion combines mean support with a penalty for response dispersion (Figure 1). The mean captures average support across the tested conditions, while dispersion measures its variation across interventions. Together, these statistics provide a stability signal that complements the original policy scores.

Building on this principle, we develop CTPNav to incorporate intervention responses into both representation-level reasoning and action-level decision making. At the representation level, standardized edge responses regulate the learned attention residual during graph propagation. Their dispersion further controls the subsequent waypoint-association update, allowing finer association when structural responses vary across interventions and preserving more candidate frontiers. At

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.

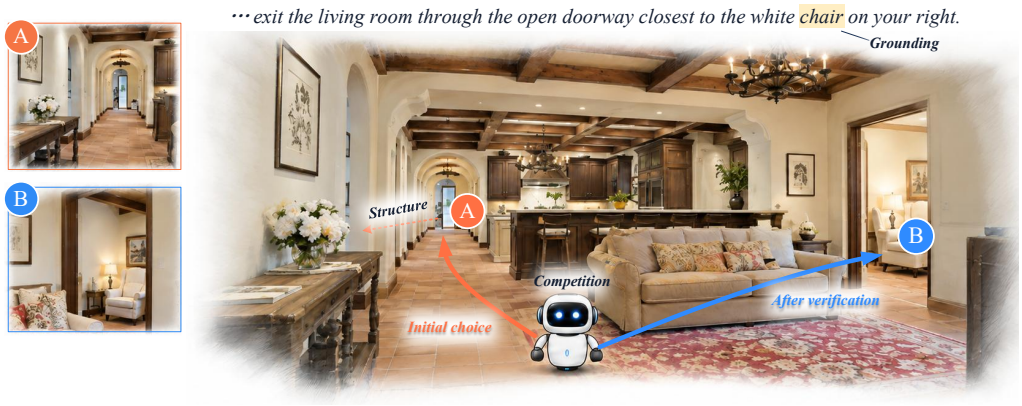

Figure 1: **Motivation.** A policy score ranks candidate goals but does not reveal whether their support persists under changes to the underlying computation. CTPNav probes this dependence through three controlled internal interventions: neutralizing the top-two score margin (Competition), attenuating geometric attention bias (Structure), and masking an aligned instruction token (Grounding). Mean support and response dispersion characterize the resulting responses and guide a bounded refinement of action preferences before execution.

the action level, CTPNav probes goal competition, geometric bias, and instruction grounding to characterize the sensitivity of candidate preferences. The resulting responses provide detached stability targets for policy learning and guide a bounded inference-time correction of action scores. The training targets and inference-time correction share the response construction but operate through separate mechanisms.

On R2R-CE Val-Unseen, CTPNav achieves 61% SR and 51% SPL. On RxR-CE Val-Unseen, it reaches 54.53% SR and 45.98% SPL. Our contributions are summarized as follows:

- We formulate decision-time counterfactual stability for online topological navigation, combining mean support with a penalty for response dispersion to characterize action preferences under controlled internal interventions.
- We develop CTPNav, which incorporates intervention responses into graph propagation, adaptive waypoint association, and policy learning, together with a bounded inference-time correction for action selection.
- Experiments on R2R-CE and RxR-CE demonstrate improvements in navigation performance. Further analyses show that response dispersion provides information beyond mean support and that combining the three intervention channels improves performance over individual-channel configurations. The full configuration improves RxR-CE Val-Unseen SPL by 2.16 points over the within-pipeline base.

2 Related Work

2.1 Continuous VLN and Topological Planning

R2R and RxR establish instruction-following navigation with English and multilingual route descriptions (Anderson et al., 2018b; Ku et al., 2020). VLN-CE replaces predefined graph transitions with continuous movement in Habitat, making waypoint prediction and local execution part of the task (Krantz et al., 2020; Savva et al., 2019; Krantz et al., 2021; Hong et al., 2022). History-aware encoders retain cross-modal context along a trajectory (Hong et al., 2021; Chen et al., 2021b), while topological memory exposes explored locations and frontiers beyond the current view (Chen et al., 2021a; 2022). Spatial representations organize observations into cross-modal maps, bird's-eye-view features, or grid memories (Georgakis et al., 2022; An et al., 2023; Wang et al., 2023a). GA-VLN builds BEV representations from depth-projected and learned geometric features (Yang et al., 2026), and ScaleVLN studies large-scale data generation for navigation (Wang et al., 2023b).

ETPNav combines online topological planning with waypoint prediction and local control (An et al., 2025), while DGNav adapts waypoint granularity and multimodal graph relations (Peng et al., 2026). CTPNav studies response sensitivity within this planning setting, using internal interventions to regulate graph representations, waypoint association, and action preferences, and to construct policy-learning targets.

2.2 Decision Sensitivity and Counterfactual Reasoning

Recent embodied-navigation methods model spatial uncertainty (Gao et al., 2026), reason over alternative environmental hypotheses (Seo et al., 2026), and assess decision risk under uncertain outcomes (Zhou et al., 2026). Structured navigation further uses plan refinement, memory, and feedback to improve sequential decisions (Qiao et al., 2025; Liang et al., 2024; Luo et al., 2026; Deng et al., 2026). Counterfactual VLN methods explore adversarial path sampling, alternative observations, and causal feature learning (Fu et al., 2020; Parvaneh et al., 2020; Wang et al., 2024), while cycle-consistent learning couples instruction following and generation with synthesized environments (Wang et al., 2022). These approaches introduce counterfactual variation through paths, observations, features, or environments.

CTPNav instead holds the observation and instruction fixed at a decision state and perturbs the internal computation supporting action preferences. Response dispersion measures variation across these interventions rather than concentration within a single action distribution. The resulting responses regulate graph propagation, waypoint association, and action preferences, and provide a detached training target. Section 4.3 compares the contributions of training targets, mapping feedback, and online correction, while Section 4.4 examines intervention design and the diagnostic value of response statistics.

3 Method

3.1 Overview and Counterfactual Stability Principle

At step t, a waypoint predictor processes the panoramic RGB-D observation O_t and updates the online graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ with predicted waypoints. Given instruction W, the graph policy scores unvisited frontiers (ghost nodes) $\mathcal{U}_t$ and STOP, with valid actions $\mathcal{A}_t = \{\mathrm{STOP}\} \cup \mathcal{U}_t$. Visited nodes provide graph context.

CTPNav computes edge- and action-level responses under separate interventions. Edge responses probe geometric, semantic, and instruction-conditioned attention terms to regulate graph propagation and the next map update. Competition, Structure, and Grounding produce action responses for detached training targets (Distill) and bounded inference-time score correction (Verify). Figure 2 illustrates the navigation pipeline, while Algorithm 1 gives the complete inference loop.

For an edge or action i, let $z_i^{(m)}$ denote its normalized response under condition m. Edge responses are standardized within source rows, whereas action logits are softmax-normalized. We define

$$q_i = \underbrace{\mathrm{Mean}_m\, z_i^{(m)}}_{\text{support}} - \lambda \underbrace{\mathrm{Std}_m\, z_i^{(m)}}_{\text{dispersion}} = \mu_i - \lambda \sigma_i. \tag{1}$$

Here $\lambda \geq 0$ controls the dispersion penalty, instantiated as λ_E for edges and λ_D for actions. Each level compares the original response with three interventions at a fixed navigation state.

3.2 Representation-Level Stability

Edge responses regulate graph attention, while their dispersion controls the association radius for the next map update.

3.2.1 Edge Responses and Graph Reasoning

Graph attention combines a geometric bias E_{geo} with learned semantic and instruction-conditioned terms S and I. At each layer, we form three variants by attenuating the geometric, semantic, or

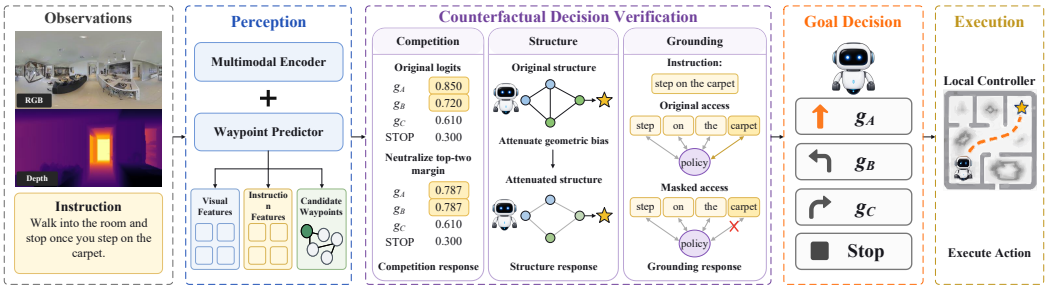

Figure 2: **Pipeline overview.** Panoramic RGB-D observations and the instruction are encoded to propose candidate waypoints and score navigation goals. Competition, Structure, and Grounding produce counterfactual action responses. During training, these responses provide detached targets for policy learning; at inference, they guide a bounded correction of action scores before goal selection and local execution.

instruction-conditioned term in turn. The original and perturbed biases are standardized across targets of the same source node. This compares each edge with the alternatives of the same source node, rather than relying on raw bias magnitudes across interventions. Their mean μ_{ij}^{E} and dispersion σ_{ij}^{E} then gate the learned residual:

$$R_{ij} = \mathrm{sigmoid}\left(\frac{\mu_{ij}^{E} - \lambda_E \sigma_{ij}^{E}}{T_E}\right), \qquad (2)$$
$$E_{ij}^{\mathrm{ver}} = E_{\mathrm{geo},ij} + [1 - \eta_E(1 - R_{ij})]\,(S_{ij} + I_{ij}).$$

Here $T_E > 0$ sets the gate temperature and $\eta_E \in [0, 1]$ controls its strength. Higher mean response retains more of $S + I$, whereas higher dispersion attenuates it. Keeping E_{geo} unchanged preserves the geometric contribution, while the gate controls how strongly the learned semantic and instruction-conditioned terms enter graph attention. The resulting E^{ver} is used at every layer. Mapping uses dispersion from the base pass, while auxiliary passes operate on the same map. Appendix A.2 specifies the construction.

3.2.2 Adaptive Topology

The association radius determines whether new waypoints merge with existing map nodes. We aggregate edge-response dispersion from the current visited node across targets and layers into $F_t^{\mathrm{map}} \in [0, 1]$. Its smoothed value sets the next association radius:

$$\bar{F}_t^{\mathrm{map}} = \gamma \bar{F}_{t-1}^{\mathrm{map}} + (1 - \gamma) F_t^{\mathrm{map}}, \qquad (3)$$
$$\rho_{t+1} = \rho_{\min} + (\rho_{\mathrm{base}} - \rho_{\min}) \exp(-\beta \bar{F}_t^{\mathrm{map}}).$$

Here $0 \le \gamma < 1$ controls smoothing and $\beta > 0$ controls sensitivity. Higher dispersion reduces the radius, making nearby waypoints less likely to merge and preserving more candidate frontiers. For example, two nearby waypoints that would merge at the base radius may remain separate when edge responses vary across interventions. The feedback applies to the next map update.

Each new waypoint is first matched to a visited location, then to an existing ghost. An unmatched waypoint creates a new ghost. Training uses a fixed base radius to avoid feedback-driven changes to graph construction and pseudo-expert targets, while evaluation enables adaptive mapping. Appendix A.3 provides further details.

3.3 Action-Level Stability

The graph policy supplies a base logit for each valid action. We construct three intervention responses on the current graph and combine their normalized support to refine that preference before execution.

3.3.1 Counterfactual Decision Verification

Verification compares action preferences under controlled changes to their supporting computation. Let $\mathcal{H}_t$ contain the encoded instruction, node features, and geometric attention biases. The graph

policy f_θ first produces base scores using the edge reasoning described in Section 3.2.1. Structure and grounding interventions then produce two additional score vectors

$$\mathbf{s}_t^{(0)} = f_\theta(\mathcal{H}_t), \qquad \mathbf{s}_t^{(m)} = f_\theta(\mathcal{I}_m(\mathcal{H}_t)), \quad m \in \{e, g\}, \tag{4}$$

The observations and parameters θ remain fixed across these passes. A competition response c is computed directly from the base logits, giving the four channels $\mathcal{M} = \{0, c, e, g\}$. For online verification, all channels are normalized over the same shortlist $\mathcal{K}_t$, formed from actions within a score margin of the best base action, capped by score, and augmented with STOP.

Competition. We neutralize the margin between the two highest-scoring shortlisted actions by replacing their logits with their log-mean-exp, yielding a shared competition response $\mathbf{s}_t^{(c)}$. STOP participates in the same ranking. Appendix A.4 gives the full construction.

Structure. We attenuate the projected geometric attention bias and rerun the graph encoder and action head from the same node and language features, recomputing dynamic edge terms at each layer.

Grounding. We mask the interior instruction token most aligned with the highest-scoring movement action during graph reasoning and instruction-edge pooling. The shared mask produces responses for all actions, including STOP. Appendix A.4 specifies token selection and masking.

Competition probes the current score margin, while structure and grounding probe dependence on geometric bias and token access. Their logits are normalized over the same shortlist. Equal weighting assigns each tested condition the same contribution to the mean and dispersion. The resulting criterion measures support across this intervention set.

3.3.2 Stability-Guided Goal Selection

Action stability adjusts commitment among competitive choices. A temperature $T_D > 0$ converts each channel to support weights over $\mathcal{K}_t$. Equation (1) gives $q_{t,i} = \mu_{t,i}^D - \lambda_D \sigma_{t,i}^D$, which is converted to normalized support $r_{t,i}$

$$p_{t,i}^{(m)} = \mathrm{Softmax}_{i \in \mathcal{K}_t}\left(\mathbf{s}_t^{(m)}/T_D\right), \qquad r_{t,i} = \frac{\max(q_{t,i}, \epsilon)}{\sum_{j \in \mathcal{K}_t} \max(q_{t,j}, \epsilon)}. \tag{5}$$

Here $\mu_{t,i}^D$ is average probability support within the shortlist, and $\sigma_{t,i}^D$ measures its redistribution across interventions. The constant ϵ regularizes normalization and logarithms. Comparing $r_{t,i}$ with the base probability gives a bounded score update

$$\begin{aligned} \Delta_{t,i} &= \mathrm{clip}\Big(\log r_{t,i} - \log \max(p_{t,i}^{(0)}, \epsilon), -\Delta_{\max}, \Delta_{\max}\Big), \\ s_{t,i}^{\mathrm{final}} &= s_{t,i}^{(0)} + \eta_D \Delta_{t,i}, \qquad i \in \mathcal{K}_t. \end{aligned} \tag{6}$$

The scale $\eta_D \geq 0$ controls the correction, and $\Delta_{\max} > 0$ bounds it. A softmax over the complete action set gives the final policy, retaining the base logits for actions outside the shortlist. STOP receives the same verification and correction as movement actions. The agent selects the highest-probability action and either terminates or executes the route to the chosen ghost through its local controller.

Appendix A.5 gives the corresponding correction bound.

3.4 Learning from Intervention Responses

The learning objective combines pseudo-expert supervision with a detached response-derived target over all valid actions. This target encourages the uncorrected policy to favor actions that retain support across the tested interventions. The base, competition, structure, and grounding logits each define a softmax distribution over the full action set. Their per-action mean is penalized by λ_D times the population standard deviation, clipped at zero, and renormalized to obtain $\mathbf{r}_t^{\mathrm{train}}$. The objective is

$$\mathcal{L} = \sum_t \left[-\log \pi_t^{(0)}(a_t^*) - \lambda_{\mathrm{stab}} \sum_{i \in \mathcal{A}_t} \mathrm{sg}(r_{t,i}^{\mathrm{train}}) \log \pi_t^{(0)}(i) \right], \tag{7}$$

Table 1: Comparison on R2R-CE.

Method	Val-Seen					Val-Unseen				
	TL	NE ↓	OSR ↑	SR ↑	SPL ↑	TL	NE ↓	OSR ↑	SR ↑	SPL ↑
End-to-end										
NaVid (Zhang et al., 2024)	–	–	–	–	–	7.63	5.47	49	37	36
Uni-NaVid (Zhang et al., 2025)	–	–	–	–	–	9.71	5.58	53	47	43
StreamVLN (Wei et al., 2025)	–	–	–	–	–	–	5.43	63	53	47
Explicit map-based										
GridMM (Wang et al., 2023a)	12.69	4.21	69	59	51	13.36	5.11	61	49	41
Safe-VLN (Yue et al., 2024)	13.71	**3.35**	**79**	**71**	**60**	15.00	4.48	**68**	60	47
MossVLN (Yu et al., 2025)	12.43	3.47	75	68	**60**	12.79	4.40	66	60	**51**
ENP-ETPNav (Liu et al., 2024)	11.82	3.90	73	68	59	11.45	4.69	65	58	50
ETPNav (An et al., 2025)	11.78	3.95	72	66	59	11.99	4.71	65	57	49
DGNav (Peng et al., 2026)	10.79	3.78	72	66	**60**	11.64	4.66	65	59	50
CTPNav (ours)	11.27	3.75	73	67	**60**	9.81	**4.22**	**68**	**61**	**51**

Here $\pi_t^{(0)} = \mathrm{Softmax}_{\mathcal{A}_t}(\mathbf{s}_t^{(0)})$ is the uncorrected policy, a_t^* is the pseudo-expert action, λ_{stab} weights the auxiliary term, and sg denotes stop-gradient. Structure and grounding passes reuse perceptual features, and graph passes share dropout realizations during training so that response differences reflect the specified interventions. Distill uses the auxiliary loss in Eq. (7), while Verify applies the bounded score correction in Eq. (6) at inference. They share the response construction but have separate switches. The bounded correction is excluded from both the training loss and rollout sampling. Appendix A.6 details target construction and rollout supervision.

4 Experiments

4.1 Experimental Setup

Datasets and metrics. We consider R2R-CE and RxR-CE in continuous Habitat environments (Krantz et al., 2020; Ku et al., 2020; Savva et al., 2019). R2R-CE evaluates English instruction following, whereas RxR-CE includes longer routes and instructions in English, Hindi, and Telugu. Val-Seen evaluates familiar environments and Val-Unseen measures generalization to new scenes. We report Success Rate (SR), Success weighted by Path Length (SPL), Navigation Error (NE), Oracle Success Rate (OSR), and Trajectory Length (TL), together with the route-fidelity metrics nDTW and SDTW that are emphasized on RxR-CE (Anderson et al., 2018a; Ilharco et al., 2019).

Implementation and evaluation. We fine-tune CTPNav's navigation modules for 20,000 iterations on each of R2R-CE and RxR-CE, using imitation learning and stability-target distillation. Training uses two NVIDIA RTX 2080 Ti GPUs, and all runs are evaluated at their final checkpoint. Table 2 reports this checkpoint from a run with a prespecified training seed, while Table 4(b) reports the mean and sample standard deviation across three training seeds. Both datasets are evaluated on their full validation splits with no episode sampling: R2R-CE on all 778 seen and 1,839 unseen episodes, and RxR-CE on all 6,746 seen and 11,006 unseen episodes. Inference uses structural-feedback mapping and selects the highest-scoring final action. Appendices A.6–A.9 detail training, initialization, and execution settings.

4.2 Main Results

Tables 1 and 2 report distances in meters and success and fidelity metrics as percentages. Dashes denote metrics not reported in the source and bold indicates the best value per column. Table 1 groups R2R-CE methods by their use of explicit spatial memory.

R2R-CE. CTPNav achieves 61% SR, 51% SPL, and $4.22\,\mathrm{m}$ NE on Val-Unseen: the highest SR, the lowest NE, and SPL tied with MossVLN. The SR margin over DGNav is 2 points on this split. On Val-Seen, it matches the best SPL, while Safe-VLN attains higher SR (Table 1).

Table 2: Comparison on RxR-CE.

Method	Val-Seen					Val-Unseen				
	NE ↓	SR ↑	SPL ↑	nDTW ↑	SDTW ↑	NE ↓	SR ↑	SPL ↑	nDTW ↑	SDTW ↑
StreamVLN (Wei et al., 2025)	–	–	–	–	–	6.72	48.60	42.50	60.20	–
ETPNav (An et al., 2025)	5.55	57.26	47.67	64.15	47.57	**5.80**	53.07	44.16	61.49	43.92
DGNav (Peng et al., 2026)	5.43	58.48	48.54	65.49	48.60	6.00	53.78	44.37	62.04	44.49
CTPNav (ours)	**5.39**	**59.53**	**50.57**	**67.31**	**50.65**	5.91	**54.53**	**45.98**	**63.19**	**45.86**



Figure 3: RxR-CE Val-Unseen trajectories: ETPNav (top) and CTPNav (bottom). Dashed lines denote reference routes. The first two cases favor CTPNav. In the last two, CTPNav terminates early while ETPNav reaches the goal.

RxR-CE. On the RxR-CE validation splits, CTPNav holds the highest SR, SPL, nDTW, and SDTW in the table, with Val-Unseen margins over DGNav of 0.75, 1.61, 1.15, and 1.37 points respectively. ETPNav retains the lowest Val-Unseen NE (Table 2). Paired evaluation matches the locally evaluated DGNav checkpoint and CTPNav on the same 11,006 episode IDs, instructions, initial conditions, and evaluation settings. CTPNav gains 400 successes and loses 317, giving $p = 0.0022$ under continuity-corrected McNemar testing. Two-sided Wilcoxon signed-rank tests, discarding zero differences, yield $p < 0.001$ for both SPL and nDTW. The 95% BCa bootstrap intervals for mean paired differences (10,000 resamples) are $[1.35, 1.87]$ for SPL and $[0.92, 1.38]$ for nDTW. The paired results show gains in both success and route fidelity on this split.

Qualitative cases. Figure 3 shows representative successes and termination failures, Appendix Figure 4 summarizes episode-level SPL changes on the profiled subset, and Appendix A.10.6 traces two further episodes decision by decision, including a case where the two methods differ in goal attainment.

4.3 Contributions of Training and Planning

Component contributions. The all-off row in Table 3(a) is our within-pipeline baseline, using $E_{\text{geo}} + S + I$ without edge gating, fixed-radius mapping, pseudo-expert supervision alone, and uncorrected policy scores at inference. All configurations use the same initialization, data, optimizer, controller, and final-checkpoint protocol.

Table 3: CTPNav component ablation on RxR-CE Val-Unseen. Verify is inference-time correction; Distill is the auxiliary training loss. Panel (a) uses separately trained configurations; its final four rows fix Edge and Map. Panel (b) changes only mapping at a fixed full-model checkpoint. Metric dashes indicate unreported values.

(a) Component contributions and training/inference switches							
Edge	Map	Verify	Distill	SR ↑	SPL ↑	nDTW ↑	SDTW ↑
–	–	–	–	52.63	43.82	60.91	43.20
✓	–	–	–	53.05	44.11	61.34	43.85
✓	✓	–	–	53.48	44.75	61.92	44.40
✓	✓	✓	–	53.72	44.95	–	–
✓	✓	–	✓	54.30	45.78	63.02	45.65
✓	✓	✓	✓	**54.53**	**45.98**	**63.19**	**45.86**
(b) Mapping controls at a fixed checkpoint (radii in meters)							
Fixed 0.50				53.96	45.19	62.48	45.02
Fixed 0.45				54.08	45.43	62.69	45.24
Fixed matched-mean (0.48852)				54.12	45.49	62.75	45.31
Within-episode shuffled replay				54.10	45.46	62.72	45.28
Full adaptive feedback				**54.53**	**45.98**	**63.19**	**45.86**

Adding Edge and Map raises SPL from 43.82 to 44.75 before either action-response application is enabled. The final four rows then hold Edge and Map fixed to distinguish learning from online correction. Distill alone adds 0.82 SR and 1.03 SPL, whereas Verify alone adds 0.24 SR and 0.20 SPL. Thus, the larger observed contribution in this comparison comes from using responses as a training target. The full configuration combines both uses and reaches 54.53 SR and 45.98 SPL.

State-dependent mapping. A smaller association radius can preserve more frontiers even without using response dispersion. Table 3(b) therefore compares adaptive feedback with fixed radii and replay controls at the same checkpoint. The matched-mean radius, $0.48852\,\mathrm{m}$, isolates the average scale of adaptation. Shuffling the recorded radii within each episode changes their temporal assignment. Full feedback exceeds these controls by 0.49 and 0.52 SPL, respectively, supporting the value of aligning map resolution with the evolving structural response. These evaluation-time effects are distinct from the retrained configurations in panel (a). Appendix A.3 specifies association, and Appendix A.10 details replay.

4.4 Response Design and Diagnostic Value

Intervention selection. Table 4(a) compares the response channels used for training targets and inference correction. Combining all three channels achieves 45.98 SPL, compared with 44.75 without action-level interventions.

Panel (b) compares selected perturbations with random alternatives across three training seeds. The selected configuration improves SPL by 0.78–0.94 points. Unlike the rival and token controls, the structural control changes the perturbation scope (one eligible edge versus the full geometric bias), rather than isolating edge selection.

Information beyond mean support. We examine whether response statistics improve the ranking of pseudo-expert agreement beyond single-pass confidence. Table 5(a) evaluates agreement with the pseudo-expert on recorded states. Mean response improves AUROC over maximum probability, margin, and entropy. Subtracting dispersion further increases AUROC from 0.781 to 0.814 and reduces AURC from 0.161 to 0.128. The dispersion term therefore adds information for ranking pseudo-expert agreement beyond the average response.

The training comparison in panel (b) tests a different use of the same statistic. With Verify enabled, the mean-only target ($\lambda_D = 0$) reaches 53.88 SR versus 53.72 without Distill. Adding dispersion at $\lambda_D = 1$ raises SR to 54.53 and SPL from 45.22 to 45.98. Increasing the weight to 2 reduces SPL to 45.68: stronger sensitivity penalties do not monotonically improve navigation. The results favor balancing retained support and dispersion over the tested intervention set.

Table 4: Response-channel design on RxR-CE Val-Unseen. Panel (a) retrains each channel combination (single seed). Panel (b) replaces one selected intervention at a time (three-seed mean ± sample standard deviation).

(a) Response-channel combinations

Competition	Structure	Grounding	SR ↑	SPL ↑	nDTW ↑	SDTW ↑
–	–	–	53.48	44.75	61.92	44.40
✓	–	–	53.81	45.02	62.24	44.78
–	✓	–	53.75	44.93	62.15	44.65
–	–	✓	53.72	44.89	62.11	44.59
✓	✓	–	54.12	45.44	62.61	45.18
✓	–	✓	54.08	45.39	62.58	45.12
–	✓	✓	54.01	45.32	62.47	45.05
✓	✓	✓	**54.53**	**45.98**	**63.19**	**45.86**

(b) Perturbation controls (three seeds)

Perturbation	SR ↑	SPL ↑	AUROC ↑
Random rival	53.83 ± 0.12	45.13 ± 0.10	0.789 ± 0.006
Random eligible edge	53.78 ± 0.13	44.97 ± 0.11	0.779 ± 0.004
Random interior token	53.86 ± 0.16	45.13 ± 0.09	0.783 ± 0.005
Selected perturbations	$\mathbf{54.46 \pm 0.12}$	$\mathbf{45.91 \pm 0.11}$	$\mathbf{0.812 \pm 0.004}$

Table 5: Response dispersion as a decision score and a training signal. Panel (a) ranks pseudo-expert agreement (three-seed means). Panel (b) varies only the training-target weight (single seed); inference correction is fixed.

(a) Decision-score ranking

Score	AUROC ↑	AUPRC ↑	AURC ↓
Max. probability	0.748	0.719	0.192
Top-1–Top-2 margin	0.759	0.728	0.185
Negative entropy	0.755	0.724	0.188
Mean response μ	0.781	0.756	0.161
Stability $\mu - \lambda\sigma$	**0.814**	**0.795**	**0.128**

(b) Training-target weight

λ_D	SR ↑	SPL ↑
0	53.88	45.22
0.5	54.15	45.61
1	**54.53**	**45.98**
2	54.21	45.68

Online effect and computational cost. At the Distill-trained checkpoint, Verify gains 125 successes and loses 99, a net 0.24 SR points; the paired difference is not significant (McNemar $p = 0.095$). It changes 2,139 of 142,656 recorded actions (1.50%), all at states with original top-1–top-2 margins below $2b$, where $b = \eta_D \Delta_{\max}$. This matches the predicted restriction on action changes. The extra graph-policy passes increase latency from 78.8 to 119.6 ms per decision on the 33-episode profiling subset, with the same 479.5M policy parameters as DGNav. Appendix A.10 provides paired outcomes, resource measurements, sensitivity analyses, and language breakdowns; Appendix A.11 discusses limitations.

5 Conclusion

We propose CTPNav to assess decision-time counterfactual stability by combining mean support with a penalty for response dispersion under controlled internal interventions. At the representation level, edge responses regulate learned attention residuals, while their dispersion adapts subsequent waypoint association. At the action level, Competition, Structure, and Grounding probe how candidate-goal preferences respond to changes in score margins, geometric bias, and instruction-token access. The resulting responses provide detached targets for policy learning and guide a bounded inference-time correction of action scores. Experiments on R2R-CE and RxR-CE demonstrate improvements in navigation performance, while further analyses show that dispersion provides information beyond mean support alone. These findings support using intervention responses alongside the original policy scores for learning and online topological planning.

AI USE STATEMENT

Generative AI tools assisted with language editing, literature lookup, and feedback on methodological presentation, experimental analysis, and revision priorities. The authors are responsible for checking AI-assisted text and suggestions against the method, cited sources, and experimental records, and for the final manuscript.

REPRODUCIBILITY STATEMENT

Section 3 defines the method. Algorithm 1 and Appendices A.2–A.8 provide the inference procedure, intervention construction, mapping feedback, score correction, and training settings. Section 4.1 and Appendix A.9 document the evaluation protocol and final-checkpoint evaluation.

REFERENCES

Dong An, Yuankai Qi, Yanwei Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. BEVBert: Multimodal map pre-training for language-guided navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2737–2748, 2023.

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. ETPNav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5130–5145, 2025. doi: 10.1109/TPAMI.2024.3386695.

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

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, 2018b. doi: 10.1109/CVPR.2018.00387.

Kevin Chen, Junshen K. Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021a.

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*, 2021b.

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*, 2022.

Yucheng Deng, Pingrui Lai, Xinhai Li, Chenjia Bai, Xiaoheng Deng, Chengnuo Sun, Xuelong Li, and Hua Yang. SpaceVLN: A zero-shot vision-and-language navigation agent with online spatial cognitive memory and reasoning, 2026. URL https://arxiv.org/abs/2606.08992.

Tsu-Jui Fu, Xin Eric Wang, Matthew Peterson, Scott Grafton, Miguel Eckstein, and William Yang Wang. Counterfactual vision-and-language navigation via adversarial path sampling. In *Proceedings of the European Conference on Computer Vision*, 2020.

Jianzhe Gao, Rui Liu, Yuxuan Xu, Tongtong Cao, Yingxue Zhang, Zhanguang Zhang, Sida Peng, Yi Yang, and Wenguan Wang. Uncertainty-aware gaussian map for vision-language navigation. In *Proceedings of the International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LPv59noPAy.

Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15460–15470, 2022.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 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.

Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15439–15449, 2022.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *NeurIPS Workshop on Visually Grounded Interaction and Language*, 2019.

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

Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15162–15171, 2021.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 4392–4412. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.356.

Xiwen Liang, Liang Ma, Shanshan Guo, Jianhua Han, Hang Xu, Shikui Ma, and Xiaodan Liang. CorNav: Autonomous agent with self-corrected planning for zero-shot vision-and-language navigation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12538–12559, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.745. URL https://aclanthology.org/2024.findings-acl.745/.

Rui Liu, Wenguan Wang, and Yi Yang. Vision-language navigation with energy-based policy. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 108208–108230. Curran Associates, Inc., 2024. doi: 10.52202/079017-3436. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c3ec89aed795f9deeff3f1390c3bd882-Paper-Conference.pdf.

Kun Luo, Xiangyu Dong, Xiaoguang Ma, Haoran Zhao, and Yaoming Zhou. LightZeroNav: Zero-shot vision language navigation in continuous environments based on lightweight VLMs, 2026. URL https://arxiv.org/abs/2603.16947.

Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Javen Qinfeng Shi, and Anton van den Hengel. Counterfactual vision-and-language navigation: Unravelling the unseen. In *Advances in Neural Information Processing Systems*, 2020.

Jiankun Peng, Jianyuan Guo, Ying Xu, Yue Liu, Jiashuang Yan, Xuanwei Ye, Houhua Li, and Xiaoming Wang. Dynamic topology awareness: Breaking the granularity rigidity in vision-language navigation, 2026. URL https://arxiv.org/abs/2601.21751.

Yanyuan Qiao, Wenqi Lyu, Hui Wang, Zixu Wang, Zerui Li, Yuan Zhang, Mingkui Tan, and Qi Wu. Open-Nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source LLMs, 2025. URL https://arxiv.org/abs/2409.18794.

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

SeungWon Seo, SooBin Lim, SeongRae Noh, Haneul Kim, and HyeongYeop Kang. From assumptions to actions: Turning LLM reasoning into uncertainty-aware planning for embodied agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GODFBZhFcX.

Hanqing Wang, Wei Liang, Jianbing Shen, Luc Van Gool, and Wenguan Wang. Counterfactual cycle-consistent learning for instruction following and generation in vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15471–15481, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Counterfactual_Cycle-Consistent_Learning_for_Instruction_Following_and_Generation_in_Vision-Language_CVPR_2022_paper.html.

Liuyi Wang, Zongtao He, Ronghao Dang, Mengjiao Shen, Chengju Liu, and Qijun Chen. Vision-and-language navigation via causal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13139–13150, 2024.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. GridMM: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15625–15636, 2023a.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12009–12020, 2023b. URL https://openaccess.thecvf.com/content/ICCV2023/html/Wang_Scaling_Data_Generation_in_Vision-and-Language_Navigation_ICCV_2023_paper.html.

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

Jiahao Yang, Zihan Wang, Xiangyang Li, Xing Zhu, Yujun Shen, Yinghao Xu, and Shuqiang Jiang. GA-VLN: Geometry-aware BEV representation for efficient vision-language navigation, 2026. URL https://arxiv.org/abs/2605.22036.

Ting Yu, Yifei Wu, Qiongjie Cui, Qingming Huang, and Jun Yu. MossVLN: Memory-observation synergistic system for continuous vision-language navigation. *IEEE Transactions on Multimedia*, 27:6690–6704, 2025. doi: 10.1109/TMM.2025.3586105.

Lu Yue, Dongliang Zhou, Liang Xie, Feitian Zhang, Ye Yan, and Erwei Yin. Safe-vln: Collision avoidance for vision-and-language navigation of autonomous robots operating in continuous environments. *IEEE Robotics and Automation Letters*, 9(6):4918–4925, June 2024. ISSN 2377-3774. doi: 10.1109/lra.2024.3387171. URL http://dx.doi.org/10.1109/LRA.2024.3387171.

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

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

Wenbin Zhou, Agni Orfanoudaki, and Shixiang Zhu. Conformalized decision risk assessment. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HGz4IGCWg7.

A APPENDIX

A.1 PLANNING ALGORITHM

Algorithm 1 specifies inference for one episode. The map at step t uses feedback retained from the preceding base graph pass. Edge verification is part of every graph forward pass, including the two auxiliary passes, but the mapping feedback is taken from the base pass. During training, waypoint merging uses ρ_{base} and the rollout mixes pseudo-expert actions with policy samples, as described in Appendix A.6.

Algorithm 1 CTPNav inference for one episode

Given Instruction W, graph policy f_θ, waypoint predictor, local controller
Return Executed navigation trajectory

Encode W into token features $\mathbf{X}$ and mask $\mathbf{m}$
Initialize graph memory, $\bar{F}_0^{\text{map}} \leftarrow 0$, and $\rho_1 \leftarrow \rho_{\text{base}}$
for each decision step t within the episode budget **do**
    Observe O_t, encode its panorama, and predict waypoints
    Update $\mathcal{G}_t$ using ρ_t, with visited-node priority over ghost merging
    Construct node inputs $\mathbf{H}_t^{\text{in}}$, geometric bias B_t, and valid actions $\mathcal{A}_t$
    Run the base graph policy, verifying edge responses at each layer
    Retain logits $\mathbf{s}_t^{(0)}$, node representations, and structural feedback F_t^{map}
    Form $\mathcal{K}_t$ by Eq. (15), and set $\mathbf{s}_t^{\text{final}} \leftarrow \mathbf{s}_t^{(0)}$
    if $|\mathcal{K}_t| \geq 2$ **then**
        Neutralize the top-two logit margin by Eq. (16)
        Rerun the graph policy with αB_t to obtain $\mathbf{s}_t^{(e)}$
        Select l^* and rerun with mask $\mathbf{m}^{(-l^*)}$ to obtain $\mathbf{s}_t^{(g)}$
        Compute $\mu_{t,i}^D$, $\sigma_{t,i}^D$, and $r_{t,i}$ on the shared shortlist
        Update shortlisted logits by Eq. (6)
    end if
    Normalize $\mathbf{s}_t^{\text{final}}$ over $\mathcal{A}_t$ to obtain π_t
    Update $\bar{F}_t^{\text{map}}$ and ρ_{t+1} by Eq. (3)
    Select $a_t \leftarrow \arg\max_{i \in \mathcal{A}_t} \pi_t(i)$
    if $a_t = \text{STOP}$ **then**
        Terminate the episode
    else
        Execute the graph route to ghost a_t through the local controller
    end if
end for

A.2 EDGE RESPONSES AND GRAPH PROPAGATION

The graph transformer uses J cross-modal layers. At a layer, let $\mathbf{h}_i$ denote its input node feature and $\mathbf{x}_l$ an encoded instruction token. The geometric, semantic, and instruction-conditioned edge terms are

$$\begin{aligned} E_{\text{geo},ij} &= \phi_g(d_{\mathcal{G}}(i,j)), \\ E_{\text{sem},ij} &= \tanh(\text{MLP}_s([\mathbf{h}_i, \mathbf{h}_j])), \\ \bar{\mathbf{x}} &= \frac{1}{L_{\text{valid}}} \sum_l m_l \mathbf{x}_l, \\ r_i &= \tanh(\text{MLP}_l([\mathbf{h}_i, \bar{\mathbf{x}}])), \\ E_{\text{inst},ij} &= r_i r_j. \end{aligned} \tag{8}$$

Here $d_{\mathcal{G}}(i,j)$ is the pairwise distance supplied by the navigation graph, ϕ_g is a learned affine projection, and m_l marks valid instruction tokens. The semantic MLP receives an ordered node pair. We write $S = w_s E_{\text{sem}}$ and $I = w_l E_{\text{inst}}$, with $w_s = \log(1 + \exp(10\hat{w}_s))/10$ and the same transform for w_l. Both residual terms are signed attention biases, and the geometric term has coefficient one.

To test dependence on each evidence source, we attenuate one term at a time using a shared factor $\alpha \in [0, 1]$

$$\begin{aligned} E^0 &= E_{\text{geo}} + S + I, \qquad E^G = \alpha E_{\text{geo}} + S + I, \\ E^S &= E_{\text{geo}} + \alpha S + I, \quad E^I = E_{\text{geo}} + S + \alpha I. \end{aligned} \tag{9}$$

Each tensor is standardized over valid targets for each source node

$$Z_i(E)_{ij} = \frac{E_{ij} - \bar{E}_i}{\sqrt{\nu_i(E) + \epsilon^2}}, \qquad \bar{E}_i = \frac{1}{|\mathcal{C}_t|} \sum_{j \in \mathcal{C}_t} E_{ij}. \tag{10}$$

The valid context set $\mathcal{C}_t$ contains visited nodes, ghosts, and STOP. The population variance $\nu_i(E)$ is computed over this same set, with padding excluded. Row-wise normalization expresses each response relative to the alternatives available to that attention query.

For either the four normalized edge responses or the four action-support weights, the inference statistics in Eq. (1) are

$$\mu_i = \frac{1}{M} \sum_{m=1}^{M} z_i^{(m)}, \qquad \sigma_i = \sqrt{\frac{1}{M} \sum_{m=1}^{M} (z_i^{(m)} - \mu_i)^2 + \epsilon^2}, \qquad M = 4. \tag{11}$$

For edge verification, $z_{ij}^{(m)} = Z_i(E^{(m)})_{ij}$ yields μ_{ij}^E and σ_{ij}^E. The mean describes edge salience relative to other targets in the same row, while dispersion describes how that relative bias changes across the four evidence conditions. Their combination in Eq. (2) controls the learned attention residual.

Equation (2) converts this support into a residual gate whose multiplier lies in $[1 - \eta_E, 1]$. Setting $\eta_E = 0$ recovers E^0. The verified scalar bias is shared across attention heads and enters each head as

$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\mathsf{T}}{\sqrt{d_h}} + \mathbf{A}_{\text{mask}} + E^{\text{ver}} \right), \tag{12}$$

Here d_h is the head dimension, $\mathbf{Q}$ and $\mathbf{K}$ are query and key projections of the cross-modal node representations, and $\mathbf{A}_{\text{mask}}$ excludes invalid keys. The base pass retains $\sigma_{t,c_t j}^{E,\ell}$ for mapping feedback. Auxiliary graph passes recompute their own edge statistics while preserving the feedback retained from the base pass.

A.3 Topological Mapping and Structural Feedback

The feedback signal aggregates the edge-response standard deviation of the current visited node c_t. With J graph layers and valid context set $\mathcal{C}_t$, it is

$$F_t^{\text{map}} = \frac{1}{J} \sum_{\ell=1}^{J} \left[1 - \exp\left(- \frac{\sum_{j \in \mathcal{C}_t} \sigma_{t,c_t j}^{E,\ell}}{T_E |\mathcal{C}_t|} \right) \right]. \tag{13}$$

This statistic lies in $[0, 1]$ and increases with the dispersion of the current node's outgoing relations. The smoothing and radius update follow Eq. (3). Here $0 \le \gamma < 1$ controls smoothing and $\beta > 0$ sets sensitivity. We initialize $\bar{F}_0^{\text{map}} = 0$ and $\rho_1 = \rho_{\text{base}}$. The radius decreases continuously with structural fragility. Its one-step delay makes the current graph's edge responses available before the next waypoint association. Competition and action-grounding responses are used in action verification, while this feedback uses edge dispersion.

Mapping uses visited-node priority. For a predicted waypoint $\mathbf{p}_k$, we first find its nearest visited location. If no visited location is within ρ_t, we search existing ghost means using the same radius. For either candidate set $\mathcal{B}$, association is defined by

$$\begin{aligned} v^*(\mathbf{p}_k, \mathcal{B}) &= \arg\min_{v \in \mathcal{B}} \|\mathbf{p}_k - \mathbf{p}_v\|_2, \\ \text{localize}(\mathbf{p}_k, \mathcal{B}) &\iff \|\mathbf{p}_k - \mathbf{p}_{v^*}\|_2 \le \rho_t. \end{aligned} \tag{14}$$

Here $\mathbf{p}_v$ is the stored global position, or the mean position of an existing ghost. An empty association set yields no match. A visited-node match adds a connection from the current node. A ghost match updates its accumulated position, feature, and frontier connections. A waypoint unmatched in both searches initializes a new ghost. Changes to the radius govern associations for newly predicted waypoints while leaving the stored coordinates of previously visited locations intact. Training uses ρ_{base}, and evaluation enables the structural-feedback rule.

A.4 Action Verification

The verification shortlist contains competitive actions and STOP. Let $s^{(0)}_{t,\max}$ be the largest valid action logit. With score margin δ, we select

$$\mathcal{K}_t = \{\text{STOP}\} \cup \text{Top}_{K_{\max}}\Big(\{i \in \mathcal{A}_t : s^{(0)}_{t,i} \geq s^{(0)}_{t,\max} - \delta\}\Big), \tag{15}$$

The operator $\text{Top}_{K_{\max}}$ retains all entries when the set is smaller than the cap. STOP is added after capping if necessary, giving $|\mathcal{K}_t| \leq K_{\max} + 1$. The same score-based rule is used in training and inference. Pseudo-expert labels are defined over the full action set and leave shortlist membership unchanged.

For $|\mathcal{K}_t| \geq 2$, let $a^\star_t$ be the highest-scoring action in the shortlist and $r^\star_t$ the highest-scoring remaining action. The shared competition response neutralizes their logit margin:

$$\begin{aligned} \bar{s}_t &= \text{LSE}\Big(s^{(0)}_{t,a^\star_t}, s^{(0)}_{t,r^\star_t}\Big) - \log 2, \\ s^{(c)}_{t,i} &= \begin{cases} \bar{s}_t, & i \in \{a^\star_t, r^\star_t\}, \\ s^{(0)}_{t,i}, & i \in \mathcal{A}_t \setminus \{a^\star_t, r^\star_t\}. \end{cases} \end{aligned} \tag{16}$$

Here $\text{LSE}(x, y) = \log(e^x + e^y)$ is computed by stable log-sum-exp. The common logit satisfies $2\exp(\bar{s}_t) = \exp(s^{(0)}_{t,a^\star_t}) + \exp(s^{(0)}_{t,r^\star_t})$. Thus, at unit temperature, the softmax denominator and all probabilities outside the pair are preserved under either full-action or shortlist normalization. Each selected action receives half of the pair's original probability mass. Ties in selecting the pair are resolved by the lower action index. All graph nodes, features, and action masks are retained. For a two-action shortlist, competition yields equal support, while a singleton shortlist retains the base response.

The structure response attenuates the learned geometric attention bias $B_t = \phi_g(D_t)$ obtained from pairwise graph distances D_t

$$\mathbf{s}^{(e)}_t = f_\theta(\mathbf{H}^{\text{in}}_t, \mathbf{X}, \mathbf{m}, \alpha B_t). \tag{17}$$

The node inputs $\mathbf{H}^{\text{in}}_t$, encoded instruction $\mathbf{X}$, and token mask $\mathbf{m}$ are fixed across passes. Dynamic edge terms and their stability gates are recomputed at each layer. Attenuation applies to the projected attention bias, with graph distances and simulator connectivity unchanged.

The grounding response selects the instruction token most aligned with the highest-scoring valid movement action i^*. Let $\widehat{\mathbf{h}}_{i^*}$ be its base graph representation and $\mathcal{T}$ the interior token positions, excluding the first and last valid tokens. We use

$$\begin{aligned} l^* &= \arg\max_{l \in \mathcal{T}} \cos(\widehat{\mathbf{h}}_{i^*}, \mathbf{x}_l), \\ \mathbf{s}^{(g)}_t &= f_\theta(\mathbf{H}^{\text{in}}_t, \mathbf{X}, \mathbf{m}^{(-l^*)}, B_t). \end{aligned} \tag{18}$$

The mask removes direct access to that token during graph reasoning and instruction-edge pooling. The remaining cached token vectors retain their contextual information. One shared token mask produces responses for every action, including STOP. The shared dropout convention used during training is specified in Appendix A.6.

A.5 Full Action Policy and Correction Bounds

Equation (6) modifies shortlisted actions, including STOP, and leaves other valid logits unchanged. The policy normalizes over all valid actions

$$\pi_t(i) = \frac{\exp s^{\text{final}}_{t,i}}{\sum_{j \in \mathcal{A}_t} \exp s^{\text{final}}_{t,j}}, \qquad i \in \mathcal{A}_t. \tag{19}$$

Padding and visited-node actions are masked before normalization. The STOP token remains valid, including when no movement goal is available. If $|\mathcal{K}_t| < 2$, all logits remain unchanged. A movement action outside $\mathcal{K}_t$ can still be selected and supervised.

The bounded update gives an explicit constraint on the amount of reranking at a fixed state. Put $b = \eta_D \Delta_{\max}$ and let $e_i = s_{t,i}^{\text{final}} - s_{t,i}^{(0)}$, taking $e_i = 0$ outside the shortlist. Then $|e_i| \le b$, and

$$\left| \log \frac{\pi_t(i)}{\pi_t(j)} - \log \frac{\pi_t^{(0)}(i)}{\pi_t^{(0)}(j)} \right| = |e_i - e_j| \le 2b, \tag{20}$$

where $\pi_t^{(0)}$ is the baseline softmax over the same action set. This follows directly because the common softmax normalizer cancels in each ratio. Consequently, a baseline margin exceeding $2b$ cannot be reversed by the action correction at that state. With $\eta_D = 0.05$ and $\Delta_{\max} = 0.25$, the maximum pairwise change is 0.025 logit units. The bound applies to the action-correction stage, not to changes in graph representation or future trajectories.

The bound constrains action changes at a fixed state. Appendix A.10.3 compares the base and corrected argmax on the same recorded states: 1.50% of decisions change, all with original margins below $2b$. The corresponding SR point estimate increases by 0.24 points, but the paired difference is not statistically significant ($p = 0.095$). Training with the response-derived target is evaluated separately in Table 3 and is not subject to this inference-time bound.

Edge gating has a related algebraic property. Writing $U_{ij} = S_{ij} + I_{ij}$, Eq. (2) gives

$$|E_{ij}^{\text{ver}} - E_{ij}^{0}| \le \eta_E |U_{ij}|, \qquad E_{ij}^{\text{ver}} - E_{\text{geo},ij} = a_{ij} U_{ij}, \quad a_{ij} \in [1 - \eta_E, 1]. \tag{21}$$

Thus, the gate preserves the sign of the combined residual while contracting its magnitude.

A.6 Imitation Learning and Stability Targets

The model is initialized from the dataset-specific pretrained stack. The RGB and depth visual backbones and the waypoint predictor remain frozen. We fine-tune the language encoder, panorama encoder, and graph policy for 20,000 iterations on each dataset. Training mixes pseudo-expert actions with samples from the uncorrected policy $\pi_t^{(0)}$ at a fixed merge radius. Equation (6) is disabled for both training supervision and rollout sampling. At evaluation, the agent enables structural-feedback mapping and selects the action with the largest final probability.

Within $1.5\,\mathrm{m}$ geodesic distance of the goal, the pseudo-expert emits STOP. Outside this region, ghost selection uses distance to the reference route for RxR-CE and geodesic distance to the goal for R2R-CE. Simulator and reference-route information are used only to obtain the training labels. A step is excluded from supervision if no valid movement label is available outside the stopping region. These labels do not change shortlist membership.

Training normalizes each response vector $\mathbf{s}_t^{(m)}$, $m \in \{0, c, e, g\}$, over the complete valid action set. The competition vector is given by Eq. (16), and the other vectors are graph-policy outputs. The response distributions and detached target are

$$\begin{aligned} \bar{p}_{t,i}^{(m)} &= \mathrm{Softmax}_{i \in \mathcal{A}_t}(\mathbf{s}_t^{(m)}), \\ v_{t,i} &= \left[\mathrm{Mean}_m\, \bar{p}_{t,i}^{(m)} - \lambda_D\, \mathrm{Std}_m\, \bar{p}_{t,i}^{(m)} \right]_+, \\ r_{t,i}^{\text{train}} &= \begin{cases} v_{t,i} / \sum_j v_{t,j}, & \sum_j v_{t,j} > \epsilon, \\ \mathrm{Mean}_m\, \bar{p}_{t,i}^{(m)}, & \text{otherwise.} \end{cases} \end{aligned} \tag{22}$$

Here $[\cdot]_+$ clips negative support to zero, and the fallback retains the mean response when the remaining mass is at most ϵ. Full-action response logits and target statistics are retained in FP32. Training uses unit temperature and the population standard deviation without additive smoothing. The auxiliary cross-entropy in Eq. (7) is accumulated only at supervised steps, and the rollout loss is divided by the number of active environment-decision pairs. Online verification uses the same response logits with shortlist normalization and temperature T_D.

At the start of each graph forward pass, CPU and GPU random-number states are retained. Each auxiliary graph pass restores those states within a forked random context. The base and intervened passes therefore share graph-dropout realizations during training without advancing the main random stream twice. Instruction encoding and panoramic feature extraction are reused across the three graph passes.

A.7 Numerical Conventions

Response statistics are evaluated in FP32 with $\epsilon = 10^{-6}$. For each valid source row, normalization uses the population variance over valid targets and denominator $\sqrt{\nu_i + \epsilon^2}$. The mask contains valid graph-context tokens, including STOP and visited nodes, and it excludes padding but does not restrict attention to physically adjacent pairs. Invalid source–target entries receive zero edge bias. Graph action masks separately exclude visited locations and padded actions.

Action-response weights use a masked softmax over $\mathcal{K}_t$ with temperature T_D. Competition computes log-mean-exp in FP32, preserving the selected pair's total probability at the unit temperature used for training targets and reported action responses. Invalid-action masks remain unchanged. Grounding retains the original mask when a valid movement action or interior instruction token is unavailable. Structure retains the base response when geometric attention bias is absent. Singleton shortlists leave final logits unchanged and set the action diagnostic to zero, while retaining the independently computed structural feedback.

For diagnostics, action fragility combines support-weight dispersion with disagreement about the preferred action. With $w_m = \arg\max_{i \in \mathcal{K}_t} p_{t,i}^{(m)}$, the recorded value is

$$F_t^{\mathrm{dec}} = \frac{1}{2} \min\left(1, \frac{2}{|\mathcal{K}_t|} \sum_i \sigma_{t,i}^D\right) + \frac{1}{6} \sum_{m \in \{c,e,g\}} \mathbf{1}[w_m \neq w_0]. \tag{23}$$

It lies in $[0, 1]$ and is logged as an action diagnostic. The merge-radius rule uses the separate structural signal F_t^{map}. Both signals are reset with the episode state.

A.8 Optimization and Complexity

The RxR-CE implementation uses AdamW with learning rate 5×10^{-5} for dynamic graph parameters and 10^{-5} for other trainable parameters. Four parallel environments and two accumulated rollouts contribute to each optimizer update. Mixed-precision gradient scaling is followed by unscaling and global gradient-norm clipping at 5.0. Instruction length is capped at 200 tokens and training rollouts at 25 high-level decisions. Waypoint augmentation is enabled during training.

The teacher-action sampling probability is $0.75^{\lfloor u/3000 \rfloor + 1}$, updated at logging-interval boundaries. Here u counts iterations from the start of fine-tuning. R2R-CE and RxR-CE each use 20,000 fine-tuning iterations for the trainable navigation modules on two NVIDIA RTX 2080 Ti GPUs. Tables 1 and 2 report results from the respective checkpoints at the end of these runs.

The RxR-CE settings are $\rho_{\mathrm{base}} = 0.50\,\mathrm{m}$, $\rho_{\min} = 0.40\,\mathrm{m}$, $\beta = 2$, $\gamma = 0.8$, $\alpha = 0.5$, $T_E = T_D = 1$, $\lambda_E = \lambda_D = 1$, $\eta_E = 0.25$, $\eta_D = 0.05$, $\Delta_{\max} = 0.25$, $\delta = 0.5$, $K_{\max} = 8$, and $\lambda_{\mathrm{stab}} = 0.05$. Because $\bar{F}^{\mathrm{map}} \in [0, 1]$, the achievable radius is $[0.40 + 0.10e^{-2}, 0.50]\,\mathrm{m}$. $\rho_{\min}$ is the rule's lower asymptote. R2R-CE uses the same values without retuning. Table 7 varies the shortlist over the ranges shown, where the wider settings differ from the reported configuration by at most 0.07 SPL points and the largest cap yields the highest SR in that sweep.

Edge verification adds $\mathcal{O}(JN_t^2)$ arithmetic to a J-layer graph encoder. Structure and grounding verification each rerun that encoder and its action head, with shared visual and language features. Including cross-modal attention and feed-forward projections, their cost is of order $\mathcal{O}(J(N_t^2 d + N_t L d + N_t d^2))$ per pass. Competition sorts shortlisted indices for deterministic tie-breaking and scans their logits, taking $\mathcal{O}(|\mathcal{K}_t| \log |\mathcal{K}_t|)$ work, plus $\mathcal{O}(|\mathcal{A}_t|)$ to copy the full response vector. Token selection adds $\mathcal{O}(Ld)$. The two extra passes contribute this graph-encoder cost at each decision. Under the detached-target objective, auxiliary passes release their activations after constructing the responses.

A.9 Dataset and Evaluation Details

R2R-CE transfers English R2R instructions and routes to continuous Habitat navigation (Anderson et al., 2018b; Krantz et al., 2020). RxR-CE supplies longer routes with English, Hindi, and Telugu instructions (Ku et al., 2020; An et al., 2025). The evaluation distinguishes Val-Seen from Val-Unseen and includes all configured instruction languages. Episode counts follow the split manifests.

For the RxR-CE guide split used here, Val-Seen contains 6,746 episodes and Val-Unseen contains 11,006. Evaluation uses the final checkpoint at iteration 20,000 for each configuration. Table 2 reports both full splits. During RxR-CE training, obstacle sliding and the inherited backtracking procedure are enabled. Evaluation disables sliding and uses the local controller for backtracking.

TL measures traveled distance and NE final geodesic distance to the goal. OSR records entry into the $3\,\mathrm{m}$ goal region, SR success at termination, and SPL path-efficient success (Anderson et al., 2018a). nDTW and SDTW measure route alignment and success-weighted alignment (Ilharco et al., 2019). Episode IDs are retained for paired comparisons.

A.10 Experimental Analysis Protocols

A.10.1 Component and Intervention Controls

Table 3(a) uses the same pretrained stack, training seed, and optimization schedule across configurations. With Edge and Map fixed, Verify toggles only Eq. (6) at evaluation; Distill toggles the auxiliary term in Eq. (7) during training. Table 4(a) instead varies the response channels used for both target construction and inference correction, retraining each configuration. Its rows therefore measure channel contributions across both phases. The all-channel configuration uses the prespecified run in Table 2. Tables 3, 4(a), 5(b), and 6 report single-run results; Tables 5(a) and 4(b) report three-seed analyses.

Table 4(b) replaces one intervention at a time while retaining the other two channels. The competition control replaces the logits of the top-ranked action and a randomly selected rival from the shortlist with their log-mean-exp, using the same neutralization as Eq. (16). Grounding masks a random interior token in place of the most aligned token. The structural comparator perturbs a randomly selected eligible edge, whereas the full method attenuates the projected geometric bias as in Eq. (17). All rows report means and sample standard deviations over three independent training seeds. The selected-perturbation configuration exceeds the tested random-control variants by 0.60–0.68 SR points and 0.78–0.94 SPL points.

Edge-gate variants. Table 6 reports a separate experiment comparing no gate, a matched constant gate, a mean-only gate, and the full $\mu^E - \lambda_E \sigma^E$ criterion. The constant gate improves SR and SPL by 0.13 and 0.09 points over no gate. The mean-only gate increases these gains to 0.26 and 0.18 points. The full mean–dispersion criterion achieves the highest values among the tested gate variants. Relative to the mean-only gate, it improves SR, SPL, nDTW, and SDTW by 0.16, 0.11, 0.16, and 0.27 points, respectively.

Table 6: Edge-gate variants on RxR-CE Val-Unseen.

Variant	SR ↑	SPL ↑	nDTW ↑	SDTW ↑
No gate	52.63	43.82	60.91	43.20
Constant matched gate	52.76	43.91	61.03	43.38
Mean-only gate	52.89	44.00	61.18	43.58
Full $\mu^E - \lambda_E \sigma^E$	**53.05**	**44.11**	**61.34**	**43.85**

A.10.2 Pseudo-Expert Agreement and Selective Prediction

The binary label records agreement between the selected ghost or STOP action and the pseudo-expert action at the same planning state. Scores are evaluated on these recorded decisions without changing the executed trajectory. Maximum probability, the Top-1–Top-2 probability margin, and negative predictive entropy provide single-pass baselines. Mean response isolates averaging across channels, while $\mu - \lambda\sigma$ also penalizes dispersion. AUROC and AUPRC evaluate the ranking of pseudo-expert agreement. For AURC, risk is the fraction of disagreements among retained decisions, and coverage is the retained fraction as low-scoring decisions are removed.

Table 5(b) reports the dispersion-weight sweep in the main text. In this sweep, only the training-target weight λ_D in Eq. (22) varies; each row is retrained while the inference-time correction keeps its default weight. The mean-only target reaches 53.88 SR, 0.16 points above the no-Distill configuration (53.72) and 0.65 points below the full target (54.53). Table 7 varies shortlist coverage with $\lambda_D = 1$.

Increasing the cap from 8 to 12 changes SR by $+0.05$ points and SPL by -0.07 points. Widening the margin from 0.5 to 1.0 likewise yields similar performance.

Table 7: Shortlist sensitivity on RxR-CE Val-Unseen.

Sweep	λ_D	δ	$K_{\max}$	SR ↑	SPL ↑
Reported	1	0.5	8	54.53	**45.98**
Narrow shortlist	1	0.25	8	54.08	45.45
Wide shortlist	1	1.0	8	54.51	45.92
Small cap	1	0.5	4	54.12	45.42
Large cap	1	0.5	12	**54.58**	45.91

A.10.3 Online Verification Switch

Table 8: Paired success outcomes with Verify toggled at a fixed checkpoint. Counts cover all 11,006 RxR-CE Val-Unseen episodes.

	Verify ON: success	Verify ON: failure
Verify OFF: success	5,877	99
Verify OFF: failure	125	4,905

Table 9 isolates Verify on the checkpoint trained with Distill. The model, shortlist, mapping rule, and episode set are fixed; only Eq. (6) is toggled. These settings correspond to the final two rows of Table 3(a).

Table 9: Inference-time Verify switch on RxR-CE Val-Unseen. The checkpoint and all training settings, including Distill, are fixed. Verify OFF executes the base argmax, corresponding to Distill ON, Verify OFF in Table 3(a); Verify ON applies Eq. (6). The SR difference uses unrounded success counts.

Setting	SR ↑	SPL ↑	nDTW ↑	SDTW ↑
Verify OFF	54.30	45.78	63.02	45.65
Verify ON	**54.53**	**45.98**	**63.19**	**45.86**
Δ	$+0.24$	$+0.20$	$+0.17$	$+0.21$

Verify gains 125 successes and loses 99, with 5,877 episodes successful under both settings and 4,905 unsuccessful under both. The net increase of 26 successes is 0.24 SR points; 224 of $11{,}006$ episodes (2.04%) change final success status. A continuity-corrected McNemar test gives $p = 0.095$, so the SR difference does not reach significance at the 0.05 level.

Across 142,656 recorded high-level decisions, the corrected top-1 action differs from the base argmax at the same state in 2,139 cases (1.50%). Every observed flip has an original top-1–top-2 logit margin below $2b = 0.025$, consistent with Eq. (20). This decision-level result establishes that the observed action changes are confined to small-margin states; the episode-level test separately measures their effect on final success.

Table 10 records the decision-level response that accompanies this switch. Per decision, the intervention channels disagree with the base response on the preferred action in 5.1% of cases, where the denominator is every recorded decision over the split, and 16.8% of episodes contain at least one such decision. The disagreement concentrates on near-ties: it occurs on 12.4% of decisions classified as low-margin and 1.3% of high-margin decisions, a factor of roughly nine. The intervention set therefore responds where the base policy is least committed, which is the behavior the criterion is designed to produce.

Channel disagreement measures the sensitivity of response rankings, whereas the 1.50% action-flip rate measures the effect of the final bounded update. These diagnostics have different meanings: disagreement need not change the executed action, and a changed action need not change final success.

Table 10: Decision-level channel response on the same fixed checkpoint. Disagreement records a case where any of the three intervention channels prefers a different action than the base response, over all recorded decisions of the split. Low and high margin split those decisions by the base Top-1–Top-2 logit margin, and the episode-level rate counts an episode once when it contains at least one disagreement.

Statistic	Rate
Channel disagreement per decision	5.1%
Episodes with any disagreement	16.8%
Disagreement at low margin	12.4%
Disagreement at high margin	1.3%

Table 11: Inference cost on the 33-episode profiling subset. Accuracy uses the full validation split; memory and timing follow the protocol below.

Method	Params (M) ↓	Mem. ↓	ms/dec. ↓	SR ↑	SPL ↑
ETPNav (An et al., 2025)	**467.7**	**1.63**	**72.3**	53.07	44.16
DGNav (Peng et al., 2026)	479.5	1.70	78.8	53.78	44.37
CTPNav	479.5	1.72	119.6	**54.53**	**45.98**

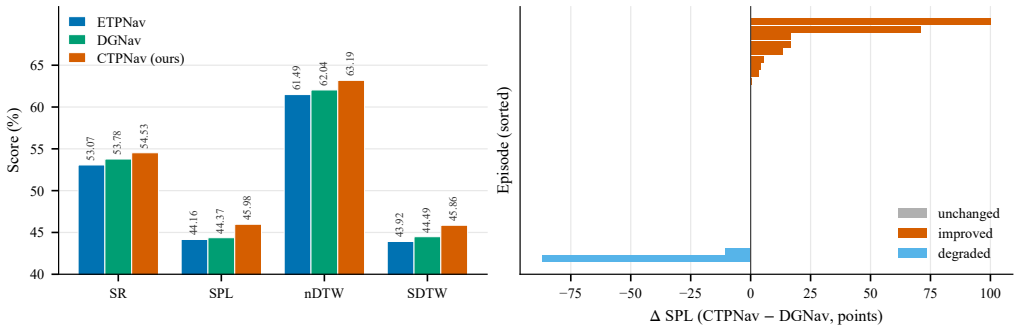

Figure 4: RxR-CE Val-Unseen performance (left) and per-episode SPL change over DGNav on 33 episodes sampled by scene–language group (right).

A.10.4 Adaptive Mapping and Runtime

Efficiency is measured on two NVIDIA RTX 2080 Ti GPUs with the released RxR checkpoint for ETPNav and DGNav and CTPNav's checkpoint after 20,000 iterations of navigation-module fine-tuning. All configurations run the same 33 Val-Unseen episodes, sampled by scene–language group from the 11,006-episode split. Parameter counts come from the instantiated navigation policy and exclude the waypoint predictor. Memory is peak allocated GPU memory during evaluation. Latency covers the policy forward pass at each high-level decision, excluding the first five decisions of each process as warm-up. ETPNav and DGNav use their released evaluation configurations. CTPNav uses its graph policy with edge verification, action verification, and structural-feedback mapping. The accuracy columns in Table 11 reproduce the full-split values in Table 2. The resource columns describe this profiling run.

The mapping experiment separates the training rule from the evaluation rule. Fixed mapping uses ρ_{base} throughout, and adaptive mapping uses Eq. (3). With fixed training maps, enabling adaptive mapping at evaluation increases SR by 0.57 points and SPL by 0.79 points (Table 12). Adapting the map during both training and evaluation yields 45.33 SPL, below the 45.98 obtained with fixed training maps. Fixed-training/adaptive-evaluation mapping achieves the highest SPL among these three configurations.

The asymmetry follows from what the association radius controls. Changing the radius changes which frontiers merge into a single node, so it changes the graph that the policy sees and the action the pseudo-expert supervises at that state. Under a fixed radius, the same observation maps to the

same graph throughout training and the supervision target is stable. Under an adaptive radius, the graph depends on the smoothed structural response of the current rollout, so the same observation can map to different graphs at different iterations and the target drifts with it. Fixed-radius training removes that drift, and the radius rule then acts at evaluation only, where it resamples the existing graph rather than retraining against a moving one.

Table 12: Training–evaluation mapping configurations on RxR-CE Val-Unseen.

Train map	Eval. map	SR ↑	SPL ↑	nDTW ↑	SDTW ↑
Fixed	Fixed	53.96	45.19	62.48	45.02
Fixed	Adaptive	**54.53**	**45.98**	**63.19**	**45.86**
Adaptive	Adaptive	54.12	45.33	62.58	45.10

Evaluation-time mapping controls. Table 3(b) compares five evaluation-time mapping rules using the same checkpoint at iteration 20,000, random seed, and all 11,006 RxR-CE Val-Unseen episodes. Only the association-radius rule changes. Edge verification, action verification, and the local controller remain fixed.

The fixed-radius controls use 0.50 and $0.45\,\mathrm{m}$. The matched-mean control uses the decision-weighted mean of the radii recorded during the full adaptive run,

$$\rho_{\mathrm{matched}} = \frac{\sum_e \sum_{t=1}^{T_e} \rho_{e,t}}{\sum_e T_e}, \tag{24}$$

where T_e is the number of high-level decisions in episode e. The 142,656 recorded decisions yield $\rho_{\mathrm{matched}} = 0.4885226647\,\mathrm{m}$, rounded to $0.48852\,\mathrm{m}$ in the table.

For shuffled replay, we first record the radius sequence from the full adaptive rollout of each episode and randomly permute it within that episode. We then restart the episode from its original initial state and apply the permuted radii in order. If the new rollout exceeds the recorded sequence length, playback cycles from the beginning. Earlier termination uses the corresponding prefix. This control changes the temporal assignment of each episode's recorded radii. Full adaptive feedback instead computes each radius from the preceding smoothed structural-response signal.

Reducing the fixed association radius from 0.50 to $0.45\,\mathrm{m}$ improves SPL from 45.19 to 45.43. Full adaptive feedback reaches 45.98 SPL, exceeding the smaller fixed radius by 0.55 points and the matched-mean constant radius by 0.49 points. It also outperforms within-episode shuffled replay by 0.43 SR, 0.52 SPL, 0.47 nDTW, and 0.58 SDTW points. These comparisons support adapting the association radius to structural feedback beyond reducing its fixed value or matching its mean.

A.10.5 Language Breakdown

Table 13 disaggregates RxR-CE Val-Unseen performance by instruction language. Route fidelity improves across all three languages, with nDTW gains of 1.20, 1.07, and 1.18 points in English, Hindi, and Telugu. The overall SR gain is driven by English (+1.14 points) and Hindi (+1.23 points), while Telugu decreases slightly (-0.11 points). Differences are computed before rounding. DGNav succeeds on 2,018, 1,871, and 2,030 episodes, respectively; CTPNav succeeds on 2,060, 1,916, and 2,026. These counts sum to the paired evaluation totals of 5,919 and 6,002 successes.

Table 13: Language-wise comparison with the locally evaluated DGNav checkpoint (Peng et al., 2026) on RxR-CE Val-Unseen. Episode counts come from the split manifests and cover all $11{,}006$ unseen episodes.

Language	Episodes	DGNav SR	CTPNav SR	DGNav nDTW	CTPNav nDTW
English	3,669	55.00	**56.15**	63.45	**64.65**
Hindi	3,669	50.99	**52.22**	60.48	**61.55**
Telugu	3,668	**55.34**	55.23	62.19	**63.37**

A.10.6 Decision-Level Trajectories on Real Episodes

The trajectory export records the RGB panorama, navmesh-aligned top-down view, visited nodes, frontier ghosts, selected goal, and executed pose at each high-level decision. Figures 5 and 6 show

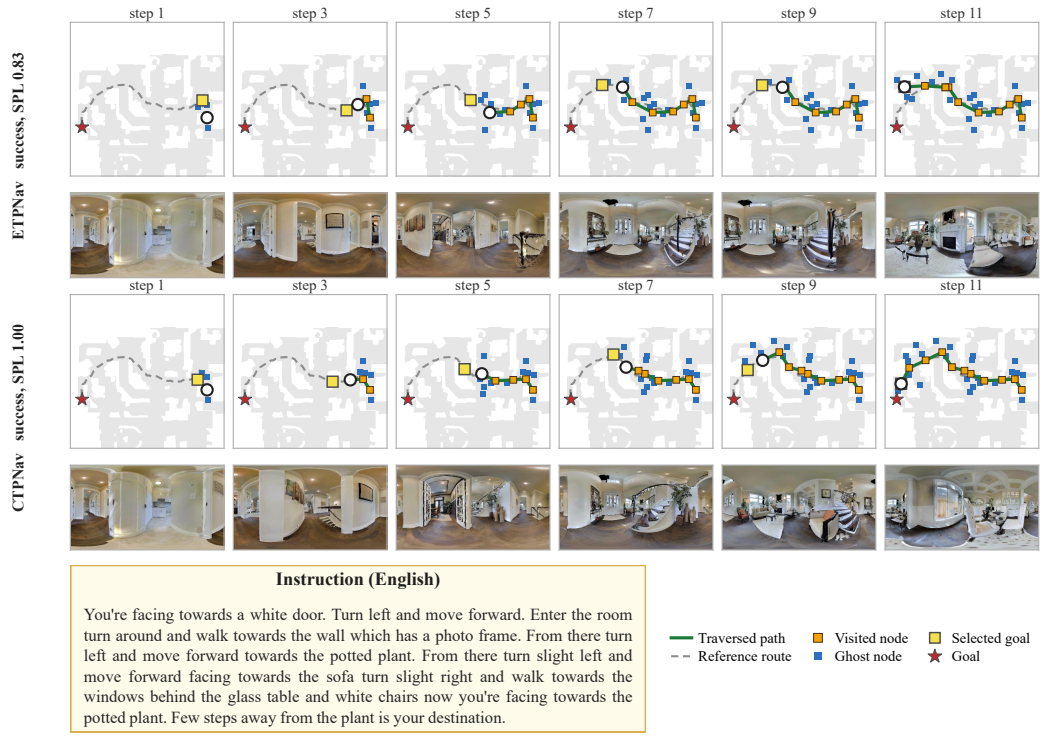

Figure 5: Decision-level traces on an English RxR-CE Val-Unseen episode. Both methods succeed in 11 decisions, with CTPNav following a shorter route.

two Val-Unseen episodes with the same scene and instruction for both methods. The traces expose how their graph states and executed routes develop over the episode.

Figure 5 illustrates route efficiency on an English instruction. Both methods succeed in 11 decisions; CTPNav travels $15.19\,\mathrm{m}$ versus ETPNav's $18.73\,\mathrm{m}$, with SPL 1.00 versus 0.83. In the Hindi episode of Figure 6, ETPNav leaves the reference route and fails after 11 decisions (SPL 0.00). CTPNav reaches the goal in 8 decisions (SPL 1.00), illustrating a difference in goal attainment.

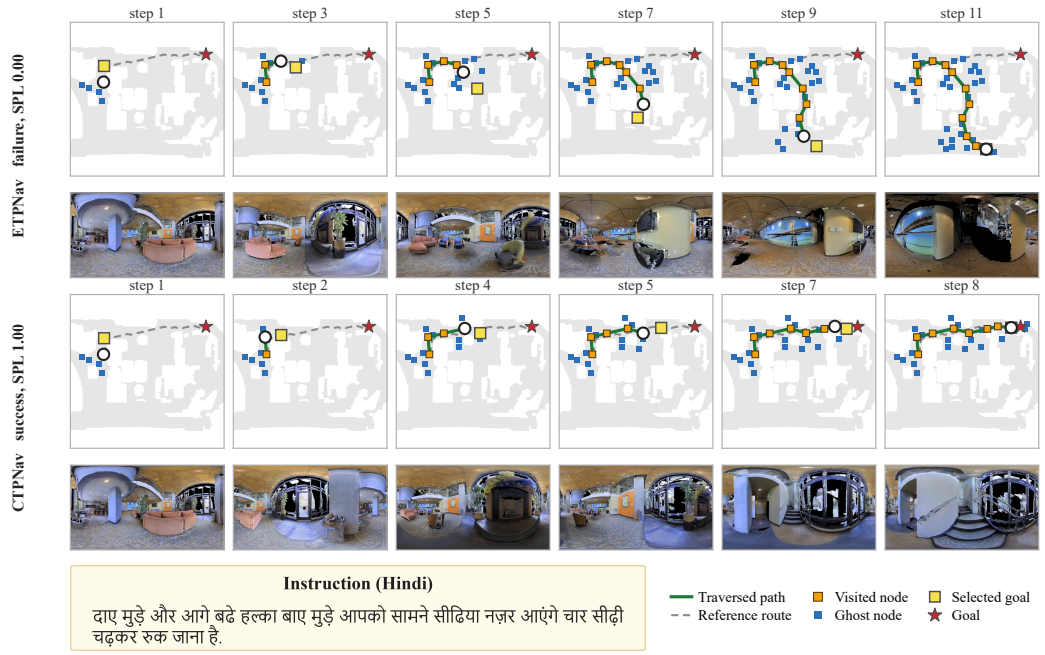

Figure 6: Planning states and observations for a Hindi RxR-CE Val-Unseen episode. ETPNav departs from the reference route, while CTPNav reaches the goal.

A.11 Limitations

The criterion developed here applies to planners that expose candidate goals and the computations supporting them. CTPNav is built on a topological planning stack and inherits its dependence on waypoint prediction and on the local controller, so the results characterize this planning family rather than end-to-end policies that map observations directly to low-level actions. Controller error enters through the graph itself: visited-node positions follow the executed trajectory, so drift changes which node supplies the outgoing edge dispersion that sets the next merge radius.

The comparison is also bounded. The language breakdown in Table 13 uses a single trained model, so the per-language differences describe that model rather than multilingual instruction following in general. The intervention set is fixed by construction: the competition, structure, and grounding responses are specified analytically, and the criterion measures support over that set rather than discovering which internal computations matter.

Finally, the association radius is fixed at ρ_{base} during training, so $\rho_{\min}$ does not come into play until evaluation, where the radius follows the structural-feedback rule of Eq. (3), for the stability reason given in Appendix A.3. Training otherwise proceeds unchanged, and adapting the radius during training while keeping the supervised action fixed to each observation remains open.