BEYOND THE ENDPOINT: RETHINKING TERMINATION IN VISION-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language navigation requires agents to follow language instructions, reach the intended destination, and stop at the appropriate termination. However, standard benchmarks define success by proximity to a single demonstrated endpoint, even though an instruction may permit multiple valid stopping positions. This criterion therefore credits instruction-inconsistent stops and rejects valid alternatives, conflating endpoint proximity with instruction satisfaction. We address this mismatch by representing each instruction with an instruction-conditioned *goal set*, comprising the navigable positions that satisfy its terminal spatial requirements. This formulation induces two evaluation metrics: Goal-Set Success Rate (GSR), which measures whether the agent stops within the goal set, and Goal-Set Oracle Success Rate (GOSR), which measures whether its trajectory reaches the set at any point. To operationalize this, we construct goal sets by extracting these requirements from language, grounding the referenced rooms and objects in the scene, and compiling the resulting constraints into navigable regions. Applying this framework to R2R and RxR produces goal-set annotations for 65,253 training instructions and manually verified evaluation assets for 12,845 validation cases. Re-evaluating 17 open-source VLN methods substantially changes measured success and method rankings, revealing both endpoint successes that never reach an instruction-consistent destination and trajectories that reach a valid destination but fail to stop there. We further introduce NavJev, a lightweight plug-in for termination control that learns from goal-set supervision while keeping the navigation policy frozen. Across six R2R policies, NavJev improves mean GSR from 41.05% to 42.97%. These results establish instruction-conditioned goal sets as a common foundation for evaluating destination arrival and improving termination decisions.

1 INTRODUCTION

Vision-language navigation (VLN) agents follow a language instruction through a visual environment and eventually decide when to issue the STOP action. In Room-to-Room (R2R) and Room-Across-Room (RxR), conventional Success Rate (SR) deems an episode successful if the agent stops within $3m$ of a demonstrated endpoint (Anderson et al., 2018b; Ku et al., 2020; Krantz et al., 2020). This endpoint is the final node of one predefined path shown to an annotator when the instruction was written. It therefore records where one demonstration ends, but does not necessarily characterize all stopping positions that satisfy the instruction's terminal spatial requirements. Consider "Walk into the bedroom and stop near the bed" (Figure 1): any position that is both inside the bedroom and near the bed constitutes a valid stopping location. Under the conventional $3m$ criterion, however, an agent stopping just outside the bedroom door, $2m$ from the demonstrated endpoint (position C), is counted as successful despite never entering the room. Conversely, an agent stopping on the far side of the bed, $3.1m$ from the endpoint (position B), is counted as a failure even though its final position satisfies the instruction. As the invalid stop is closer to the endpoint than the valid stop, adjusting the distance threshold cannot resolve this mismatch. The underlying issue is that success is defined by proximity to a single demonstrated endpoint rather than by whether the stopping position satisfies the instruction. This mismatch affects not only evaluation, but also imitation learning, where endpoint-based supervision is commonly used to train the STOP decision.

This observation motivates a more faithful terminal target: the set of positions at which an agent can validly stop. We formalize this target as an instruction-conditioned *goal set*, $\mathcal{G}(I, E) \subseteq \mathcal{N}(E)$,

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.

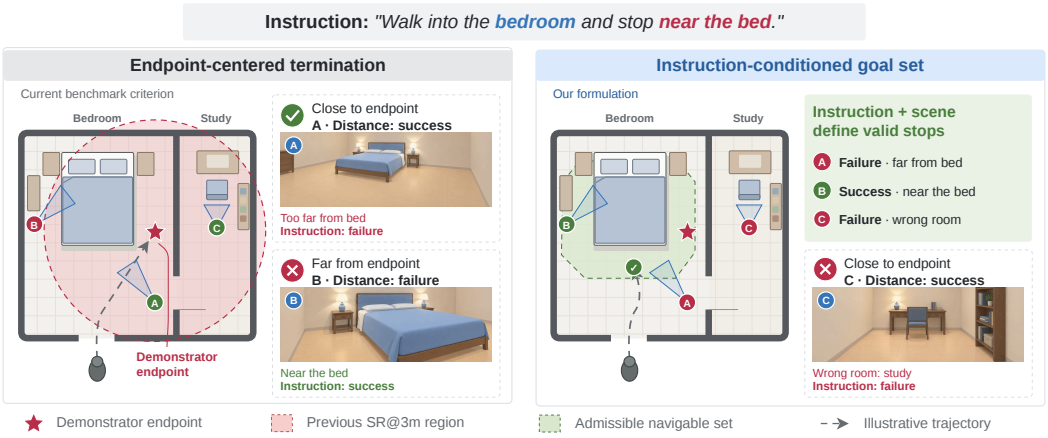

Figure 1: **Why endpoint proximity misjudges termination.** The endpoint-centered 3 m region accepts stops A and C despite their violation of the instruction, while rejecting the valid stop B. The instruction-conditioned goal set instead captures the admissible stopping region and accepts B.

containing all navigable positions in environment E that satisfy the stopping requirements of instruction I. In the bedroom example, the goal set includes positions that are both inside the room and near the bed, with the demonstrated endpoint being one member of this set. This formulation naturally induces two complementary metrics. Goal-Set Success Rate (GSR) credits an episode when the agent executes STOP inside the goal set, whereas Goal-Set Oracle Success Rate (GOSR) measures whether the trajectory enters the goal set at any recorded step. Together, these metrics replace endpoint proximity with a more direct criterion: **Did the agent reach a position that satisfies the instruction, and did it stop there?**

Evaluating this criterion requires converting the terminal meaning expressed in language into explicit regions of the environment. We therefore develop a pipeline that extracts the terminal spatial requirements from each instruction, grounds the referenced rooms and objects in the scene, and compiles the resulting constraints into navigable regions. The pipeline also retains the associated language, grounding, and geometric evidence so that each annotation can be inspected and audited. Applied to R2R and RxR, the pipeline yields goal-set annotations for 65,253 training instructions and evaluation assets for 12,845 validation cases. As these regions directly determine goal-set success, we manually verify every `val_unseen` evaluation case. Reviewers accept 91.2% of the proposed goal sets without modification and revise the remainder, yielding a confirmed goal set for every validation episode. A paired-instruction analysis further shows that even a shared reference route and endpoint can correspond to different valid stopping regions when the instructions name different referents.

With these annotations in place, we re-evaluate 17 open-source methods by re-scoring their frozen trajectories under both terminal criteria. First, replacing endpoint-based success with goal-set-based success substantially changes measured performance, including the relative ordering of navigation methods. Second, decomposing success into arrival and termination shows that many trajectories considered successful under conventional SR never enter an instruction-consistent goal region, while a separate group reaches a valid region but fails to stop there. Together, these results show that endpoint proximity obscures whether an agent has actually reached an instruction-consistent goal region, while goal-set visitation further reveals cases in which the remaining failure lies in the stopping decision.

Motivated by the role of terminal signals, we further introduce NavJev, a lightweight plug-in for improving stopping decisions in existing navigation policies. NavJev estimates task completion from the instruction and visual history and is trained with goal-set supervision derived from our annotations. It attaches to frozen navigation policies through a shared interface without retraining their backbones. Across six navigation policies on R2R, NavJev consistently improves performance, increasing mean GSR from 41.05% to 42.97%. These results show that instruction-defined goal regions provide a useful basis for both navigation evaluation and termination control.

Our contributions are fourfold:

- We formulate VLN termination as an *instruction-conditioned goal set* and introduce GSR and GOSR to measure stopping and goal-region arrival.
- We construct auditable goal-set annotations for R2R and RxR, covering 65,253 training instructions and 12,845 manually verified validation cases.
- We re-evaluate 17 VLN methods and show goal-set evaluation substantially changes success labels and method rankings while separating navigation failures from stopping failures.
- We introduce NavJev, a lightweight goal-set-supervised stopping module that improves six frozen R2R policies, raising mean GSR from 41.05% to 42.97%.

2 Related Work

2.1 Vision-Language Navigation and Its Benchmarks

Vision-language navigation (VLN) asks an agent to follow a natural-language instruction through a scanned environment and stop when the instruction is completed. R2R and RxR instantiate the task in Matterport3D scenes (Anderson et al., 2018b; Ku et al., 2020; Chang et al., 2017), where success is measured by stopping within a fixed radius of a single demonstrated endpoint. Early modeling work strengthened language grounding and generalization through speaker-based augmentation and environmental dropout (Fried et al., 2018; Tan et al., 2019), then scaled to long-horizon instructions with history-aware transformers and graph-based reasoning (Chen et al., 2021; 2022); the move to continuous environments (Krantz et al., 2020) in turn motivated waypoint prediction, topological planning, map pre-training, and dense step-level supervision from imperfect trajectories (Krantz et al., 2021; Hong et al., 2022; An et al., 2025; 2023; Li et al., 2026a). Recent systems generate actions directly from visual history using video, vision-language-action, and large vision-language models (Zhang et al., 2024; Cheng et al., 2025; Wei et al., 2026; Zhou et al., 2024; Chen et al., 2024), with semantic-map memories replacing stored visual history (Zhang et al., 2025b). Evaluation has expanded in parallel: success rate and SPL (Anderson et al., 2018a) were joined by route-coverage and path-fidelity metrics (CLS, nDTW, sDTW) (Jain et al., 2019; Ilharco et al., 2019), fine-grained analyses across instruction nuances (Wang et al., 2024), and platform-transfer studies of robot embodiment (Wang et al., 2025). Throughout these advances, task completion remains a proximity test against one demonstrated endpoint, although an instruction often admits multiple valid stopping positions. We instead formulate termination as an instruction-conditioned goal set while preserving the original trajectories and route-fidelity metrics.

2.2 Termination in Vision-Language Navigation

Prior work addresses termination in VLN from three complementary directions. History-based methods infer completion from the navigation record: Self-Monitoring predicts instruction progress (Ma et al., 2019), Mind the Gap separates reaching the target from issuing Stop (Zhao et al., 2023), and RATE-Nav and MetaNav extend such decisions to zero-shot navigation, stopping when region-level exploration saturates or metacognitive checks no longer support continued exploration (Li et al., 2025; 2026b). Destination-centric methods make the target explicit: REVERIE and SOON ground referred objects and structured room-attribute-object descriptions (Qi et al., 2020; Zhu et al., 2021), REALM examines proximity and final-view alignment during the final approach (Shi et al., 2026), and zero-shot navigators encode goals as value maps, training-free instruction following, or unified goal graphs (Yokoyama et al., 2024; Long et al., 2024; Yin et al., 2025). A third line supervises the stop decision itself: DAgger relabels learner-visited states with expert actions (Ross et al., 2011), Φ-Nav aligns exploratory trajectories with hindsight instructions (Kim et al., 2026), and re-weighting the stop action alone corrects systematic under-stopping (Xiang et al., 2020). Each line improves how agents recognize, approach, or learn terminal behavior, yet all represent termination through a designated endpoint, target, or terminal state. An instruction defined relative to a room, object, or spatial relation may instead admit several valid stopping locations, and GoalSet-VLN captures this set-valued structure directly: its goal sets define when termination is appropriate and supervise NavJev as a plug-in stopping module for frozen navigators.

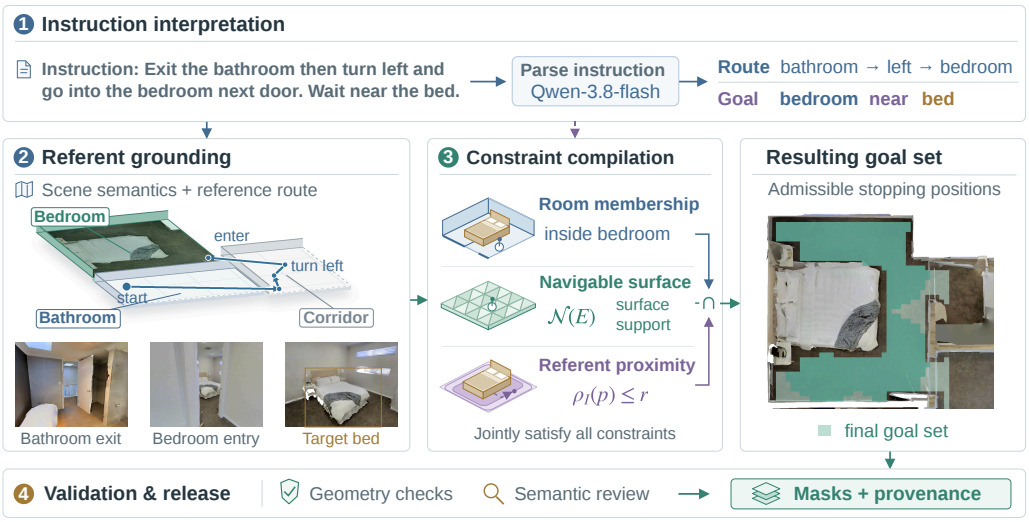

Figure 2: **Goal-set construction.** The pipeline interprets the instruction, grounds its referents in the scene, and combines spatial constraints with navigable geometry to obtain admissible stopping positions. Geometry checks and semantic review validate the resulting goal set.

3 From Endpoints to Goal Sets

A demonstration records one stopping position, while an instruction typically licenses several. This section develops the *goal set*, the set of admissible stopping positions, into the terminal target of evaluation. Section 3.1 defines the goal set and the success criterion it induces. Section 3.2 constructs and validates goal sets for R2R and RxR from language and scene geometry. A paired-instruction comparison then tests whether a shared endpoint specifies a shared stopping region (Section 3.3). Finally, Section 3.4 defines the metrics used to evaluate navigation with these regions.

3.1 Formalizing Termination as Set Membership

An instruction specifies the terminal spatial conditions for completing a navigation task. These conditions may refer to a particular object or constrain the agent only at the region level. For example, "enter the bedroom and stop beside the chair" requires the final position to lie both inside the bedroom and near the referenced chair. An instruction such as "take a few steps into the room" instead constrains the agent to enter the designated room and terminate within a local geodesic neighborhood of the corresponding entry point. We define the *goal set* $\mathcal{G}(I, E)$ as the subset of navigable positions in environment E that jointly satisfy the terminal semantic and spatial constraints induced by instruction I.

Given this set, navigation succeeds when the agent explicitly issues STOP at an admissible position. Conventional VLN evaluation instead treats a stop as successful when it lies within 3 m of the demonstrated endpoint. The two criteria therefore define admissible stopping positions in fundamentally different ways: the conventional criterion accepts a geodesic neighborhood centered at a single demonstrated endpoint, whereas the goal set is determined by the terminal requirements of the instruction and the geometry of the scene.

These acceptance regions need not coincide. In particular, an instruction-inconsistent position may lie closer to the demonstrated endpoint than an instruction-consistent one. Any endpoint-centered radius large enough to include the valid position would then also include the invalid one, showing that a single distance threshold cannot in general reproduce instruction-conditioned admissibility. The resulting disagreement occurs in both directions: positions close to the demonstrated endpoint may violate the instruction, while valid stopping positions may lie beyond the conventional threshold. Section 4 quantifies these cases on frozen trajectories. We next describe how $\mathcal{G}(I, E)$ is constructed at scale for the benchmark.

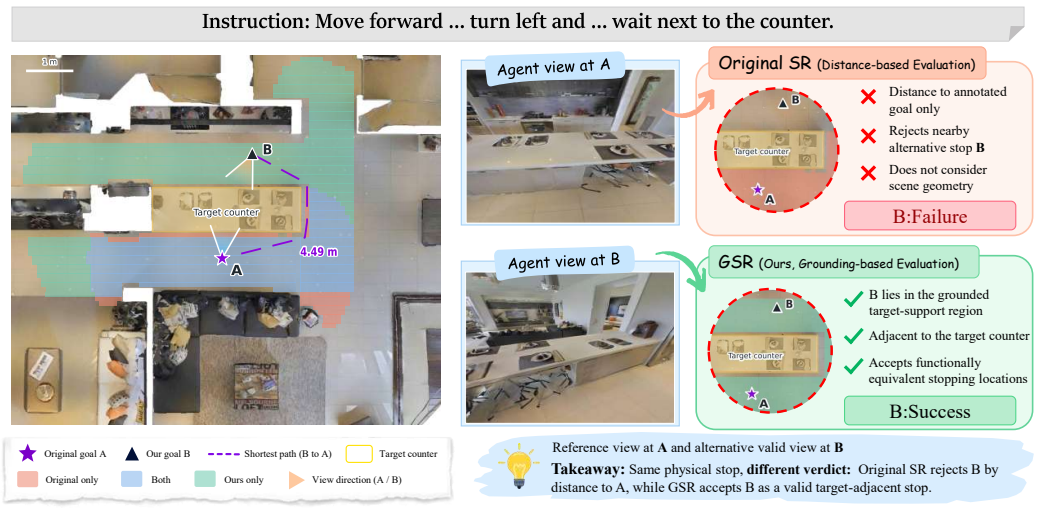

Figure 3: **A goal-set annotation supported by multiple cues.** The instruction, scene geometry, and agent view all support B as a valid stop beside the target counter. GSR accepts B, whereas endpoint-based SR rejects it by distance.

3.2 Constructing Goal Sets from Language and Geometry

The construction pipeline turns instruction-level terminal conditions into navigable regions that can be used directly for evaluation. It combines three sources of information: the instruction, scene context for identifying its referents, and navigable geometry for determining where the agent may validly stop. Figure 2 summarizes the construction process. We separate the problem into two stages. First, we interpret the terminal clause and ground its destination referents in the scene. We then compile the associated stopping constraints into geometric predicates over the navigable space.

Interpreting and grounding the target. We first use Qwen3.8-Flash to parse the terminal clause into a primary target, identifying attributes, and positional constraints. We then use scene context to localize the relevant search area. A top-down view covers the room containing the demonstrated endpoint together with adjacent regions, providing room connectivity, nearby landmarks, and candidate spatial extents. We combine this view with instance-level scene observations and the Matterport3D semantic inventory (Chang et al., 2017) to associate language referents with specific rooms and objects. We verify each grounding with respect to object identity, spatial extent, and floor assignment. Descriptive relations, such as a hat located on a cabinet, are used to disambiguate the referenced entity, while stopping relations determine the spatial constraints applied to the agent. We also use the reference path to recover approach context for directional and entry-related expressions. Appendix A.1 details the grounding procedure, and Appendix A.2 specifies the spatial representation.

From grounded referents to navigable regions. Once the referents are grounded, we convert the instruction into a set $\mathcal{C}_I$ of positional predicates defined over the navigable space $\mathcal{N}(E)$. These predicates encode the spatial conditions required at termination, including navigable proximity to the grounded referent, room membership, entry constraints, and directional or side relations. The operational goal set is then defined as

$$\mathcal{G}(I, E) = \{p \in \mathcal{N}(E) \mid c(p) = 1,\ \forall c \in \mathcal{C}_I\}. \tag{1}$$

We implement these predicates as surface-level geometric checks. Doorway-related conditions identify the traversable opening associated with the target region, while "just inside" conditions restrict admissible positions to the local interior near that entrance. Stair-related conditions additionally use height-resolved surface geometry to identify the requested level.

Figure 3 illustrates the resulting annotation for a representative episode. The instruction requires the agent to stop next to the grounded counter, yielding a region of admissible navigable positions rather

than a single target coordinate. Consequently, an alternative position can satisfy the same terminal condition even when it lies outside the conventional endpoint-centered success radius. Additional examples spanning different target types and spatial relations are provided in Appendix B.6.

Annotation outputs and validation. The pipeline yields 65,253 training annotations and evaluation assets for 12,845 R2R and RxR validation cases. Training annotations provide terminal supervision, while the validation assets undergo episode-level human review of referent identity, stopping relation, and region extent, together with structural checks for floor support, navigability, containment, and mask integrity. Reviewers accepted 91.2% of the initial proposals unchanged and revised the remainder before confirmation, yielding a confirmed goal set for every validation episode. We further validate the resulting membership decisions with an independent blinded position-level audit. A probability sample covers policy stops, while a diagnostic sample covers region interiors, boundaries, and exteriors. Independent annotators judge each candidate position from the instruction and scene views, allowing endpoint proximity and goal-set membership to be evaluated against the same adjudicated positions. Appendices A.3 and A.4 describe annotation coverage and the independent audit, respectively.

3.3 Stopping Region Follows the Instruction

Instructions associated with the same reference route and endpoint offer a direct check of whether the endpoint alone specifies the stopping region. For most such pairs, the annotated goal sets agree, as expected when the instructions name the same destination. In an informative subset, however, the instructions identify different terminal referents and their goal sets differ despite sharing the endpoint. Figure 4 shows one pair: waiting in the hallway and waiting beside the vase of flowers call for different regions, although the demonstrated route ends at the same position. The comparison isolates a representational limit. Any endpoint-distance rule assigns the same accepted region to the two instructions, whereas their terminal requirements distinguish the valid stopping positions. An instruction-conditioned goal set preserves this distinction by grounding each referent separately.



Figure 4: **One endpoint, two stopping regions.** The same reference route and endpoint E accompany instructions targeting the hallway (green) and the vase of flowers (purple). Each instruction selects a different admissible region.

Finding 1. **A shared endpoint need not imply a shared stopping region.** Different terminal referents can license different goal sets for the same reference route, so the admissible region must be specified for each instruction.

3.4 Goal-Set Metrics

To characterize whether an agent reaches and correctly terminates within an admissible region, we introduce two goal-set-based metrics: Goal-Set Success Rate (GSR) and Goal-Set Oracle Success Rate (GOSR). Let $\mathcal{E}$ denote the evaluation population, $N = |\mathcal{E}|$, and b_T^i indicate whether trajectory i explicitly terminates with STOP. Using the fixed 3 m proximity setting, we write each goal set as $\mathcal{G}^i$ and define

$$\mathrm{GSR} = \frac{1}{N}\sum_{i\in\mathcal{E}} b_T^i\,\mathbf{1}[p_T^i \in \mathcal{G}^i], \qquad \mathrm{GOSR} = \frac{1}{N}\sum_{i\in\mathcal{E}} \mathbf{1}[\exists t \in \{0,\dots,T_i\} : p_t^i \in \mathcal{G}^i]. \tag{2}$$

GSR measures whether the agent explicitly stops at an admissible position, whereas GOSR measures whether the trajectory reaches the goal set at any point. We additionally report distance-to-set (GNE) and path-normalized success (GSPL) to characterize residual distance and navigation efficiency, respectively. Their geodesic definitions and edge-case conventions are provided in Appendix B.1.

Table 1: **R2R-CE and RxR-CE validation unseen under goal-set (bold) and endpoint (gray) metrics.** Methods marked with * use the full multilingual RxR-CE set; other RxR-CE results use all English episodes. Method references are enumerated in Appendix B.2.

	R2R-CE								RxR-CE							
Method	SR	**GSR**	OSR	**GOSR**	SPL	**GSPL**	NE	**GNE**	SR	**GSR**	OSR	**GOSR**	SPL	**GSPL**	NE	**GNE**
w/o LLM backbone																
CMA	30.18	**21.70**	37.09	**27.51**	28.20	**18.68**	7.36	**6.67**	–	–	–	–	–	–	–	–
HPN	37.57	**24.14**	41.76	**28.33**	35.33	**21.03**	6.06	**5.41**	–	–	–	–	–	–	–	–
CWP	44.15	**27.30**	53.24	**34.31**	38.89	**21.75**	5.74	**5.05**	–	–	–	–	–	–	–	–
BEVBert-CE	59.22	**36.11**	67.10	**43.67**	49.11	**27.10**	4.63	**3.99**	–	–	–	–	–	–	–	–
ETPNav*	56.93	**38.44**	64.11	**45.24**	49.07	**29.88**	4.78	**4.15**	56.38	**44.49**	65.19	**52.57**	44.92	**28.37**	5.42	**4.33**
g3D-LF	60.14	**37.25**	67.54	**44.64**	50.71	**28.88**	4.59	**3.94**	–	–	–	–	–	–	–	–
ETP-R1*	65.31	**44.05**	71.40	**49.92**	55.72	**34.15**	3.93	**3.34**	60.78	**51.47**	67.01	**57.56**	50.57	**35.06**	4.87	**3.84**
CoMaR (ETPNav)	59.16	**39.04**	66.72	**46.44**	50.85	**30.40**	4.56	**3.93**	–	–	–	–	–	–	–	–
w/ LLM backbone																
NaVid	42.47	**30.51**	50.46	**35.94**	37.17	**24.30**	5.69	**5.08**	42.08	**36.00**	51.81	**43.47**	35.61	**24.59**	7.73	**6.64**
NaVILA	54.70	**39.37**	61.88	**45.51**	50.07	**33.32**	5.21	**4.60**	45.57	**37.91**	55.41	**45.76**	40.53	**28.03**	6.81	**5.75**
Uni-NaVid	51.01	**43.01**	57.26	**47.09**	45.90	**35.63**	5.39	**4.78**	52.30	**47.72**	63.26	**57.15**	41.86	**31.35**	6.50	**5.55**
StreamVLN*	56.33	**46.82**	64.00	**52.47**	50.17	**37.78**	4.97	**4.35**	28.87	**25.58**	41.45	**35.44**	24.71	**18.16**	9.20	**7.94**
ActiveVLN	46.66	**31.70**	51.93	**35.62**	42.56	**26.63**	5.71	**5.07**	16.38	**12.81**	24.39	**18.53**	14.32	**8.55**	10.49	**9.17**
JanusVLN	52.53	**35.13**	58.46	**40.35**	48.78	**30.18**	5.26	**4.65**	–	–	–	–	–	–	–	–
InternVLA-N1*	57.48	**42.74**	62.75	**46.22**	53.20	**36.46**	4.74	**4.14**	32.64	**27.58**	44.66	**37.58**	28.31	**19.80**	8.75	**7.50**
NaVIDA	59.11	**50.35**	65.91	**57.15**	52.85	**41.78**	4.68	**4.13**	55.38	**49.99**	63.23	**57.94**	49.45	**37.53**	5.56	**4.62**
AwareVLN	65.25	**42.52**	75.04	**50.79**	56.73	**33.73**	4.05	**3.46**	56.23	**47.26**	64.90	**55.57**	47.73	**33.19**	5.51	**4.51**

4 Re-Evaluating Navigation Policies

With goal sets and metrics in place, we rescore the same frozen trajectories under the two terminal criteria. We examine changes in success labels and policy comparisons, then separate failures to reach a valid region from failures to stop within one.

Evaluation setup. We evaluate 17 open-source VLN methods on validation goal sets. All R2R results cover the full set of 1,839 val_unseen episodes, while RxR follows the language scope of each method, comprising 11,006 instructions for multilingual settings and 3,669 for English-only settings. Appendix B provides the corresponding dataset coverage and denominator details.

4.1 Termination Criteria Reshape Rankings

Goal-set evaluation changes success labels in both directions, rejecting endpoint-near stops that violate the instruction and recovering valid stops outside the endpoint radius. The changes are strongly asymmetric. On R2R, each policy loses 159 to 430 endpoint successes but gains 3 to 14, with rejections accounting for 93 to 99% of all changed labels. Rejections also dominate on RxR, although recoveries are more frequent than on R2R (Appendix B.3). Consequently, the net change in success rate understates the extent of disagreement between the two criteria. On R2R, 9.30 to 23.65% of labels change, compared with GSR decreases of 7.99 to 23.11 points; across the six English RxR runs, the corresponding ranges are 4.77 to 11.58% and 3.57 to 8.97 points. The lower relabeling rates on RxR for all six policies further show that the discrepancy is benchmark-dependent.

More importantly, the discrepancy is also policy-dependent and changes method rankings. On R2R, NaVIDA trails ETP-R1 by 6.20 SR points but leads by 6.30 GSR points, moving from sixth to first. Only 15.36% of the endpoint successes of NaVIDA are rejected, compared with 32.89% for ETP-R1. StreamVLN and Uni-NaVid similarly rise from ninth to second and from twelfth to fourth. On the matched English RxR population, the 0.85-point SR advantage of AwareVLN over NaVIDA becomes a 2.73-point GSR deficit. These reversals indicate that endpoint-based evaluation does not introduce a uniform offset across policies; it favors methods to different degrees depending on how often endpoint success coincides with instruction-consistent termination. The contrast is particularly informative for training supervision. ETP-R1 combines more than three million pretraining examples with online DAgger and reinforcement fine-tuning, substantially expanding instruction and

state coverage, while its action supervision and optimization signals remain largely tied to reference trajectories and demonstrated endpoints. Its strong SR therefore does not translate proportionally to GSR. This result suggests that increasing data scale and state coverage alone is insufficient to guarantee instruction-consistent completion when the supervision target remains endpoint-oriented.

Finding 2. **Endpoint success is not a uniform proxy for instruction-consistent completion.** The mismatch varies across policies and benchmarks, altering rankings and revealing limits of endpoint-oriented supervision.

4.2 Decomposing Goal-Set Success: Arrival and Termination

Goal-set success requires two distinct events: the trajectory must first enter an admissible region, and the policy must then issue STOP while inside it. GOSR measures whether the first event occurs, while GOSR − GSR captures trajectories that reach a valid region but do not terminate there. This decomposition reveals that most endpoint successes overturned by goal-set evaluation originate from arrival errors rather than stopping errors. Among these cases, 74.3 to 95.1% on R2R and 64.4 to 88.8% on RxR never enter the goal set at any point (Appendix B.4). Thus, much of the discrepancy exposed by goal-set evaluation arises before the final STOP decision and cannot be corrected by termination alone.

The decomposition further explains why policies can differ under GSR for qualitatively different reasons. On R2R, the advantage of NaVIDA over ETP-R1 comes almost entirely from arrival: NaVIDA reaches the goal set 7.23 points more often, while the two policies convert arrivals into successful stops at nearly identical rates of 88.1% and 88.2%. AwareVLN exhibits the complementary pattern, reaching the goal set slightly more often than ETP-R1 but converting fewer arrivals into valid stops, which lowers its GSR. Uni-NaVid and InternVLA-N1 compensate for lower arrival rates with conversion rates above 91%. These contrasts separate two sources of performance variation that aggregate success rates conflate: reaching an instruction-consistent region and terminating correctly after reaching it. This distinction motivates the termination intervention in Section 5.

Finding 3. **Most hidden errors arise before termination.** Most endpoint successes overturned by goal-set evaluation never enter a valid region, while arrival-to-stop conversion exposes a distinct termination bottleneck across policies.

5 NavJev: Teaching Frozen Navigators When to Stop

Not every trajectory that reaches the goal set terminates within it. Across R2R policies, the arrival-termination gap identified in Section 4.2 ranges from 3.48 to 8.27 points. This gap indicates that reaching an admissible region does not necessarily lead to a correct stopping decision, motivating a separate mechanism for termination at deployment. NavJev provides such a mechanism as a plug-in module that estimates task completion from the instruction and visual history and determines when a frozen navigator should stop. NavJev is trained once using goal-set labels, and the resulting checkpoint is applied unchanged to different policies without retraining the underlying navigator. We describe the design in §5.1 and evaluate the same checkpoint across six policies in §5.2.

5.1 Plug-In Design

Let h_t contain 16 egocentric frames sampled uniformly from the trajectory prefix through step t, and I denote the episode instruction. NavJev, implemented with the Qwen3-VL-4B-Instruct vision–language model, estimates whether the current state is valid for task completion:

$$c_t = f_\phi(I, h_t) \in [0, 1], \qquad y_t = \mathbf{1}[p_t \in \mathcal{G}(I, E)], \tag{3}$$

where goal-set membership provides the supervision signal. Training states are sampled from the terminal stage of ground-truth demonstration paths over the full R2R training split. In addition to ordinary states, the training pool includes the two disagreement cases identified in Section 4.1: states inside the endpoint neighborhood but outside the goal set are used as hard negatives, whereas states

Table 2: **NavJev on six frozen R2R policies.** Base rows repeat the frozen val_unseen; w/ NavJev re-run the same episodes with each navigator's STOP replaced by the shared stopping controller.

		ETP-R1	AwareVLN	StreamVLN	NaVILA	Uni-NaVid	NaVid	Avg.
GSR	Base	44.05	42.52	46.82	39.37	43.01	30.51	41.05
	w/ NavJev	46.17	45.19	47.42	42.03	44.86	32.14	42.97
GOSR	Base	49.92	50.79	52.47	45.51	47.09	35.94	46.95
	w/ NavJev	50.90	51.33	52.36	46.44	48.61	36.05	47.62

inside the goal set but beyond the endpoint neighborhood are used as hard positives. These examples provide direct supervision on states for which endpoint proximity and instruction-consistent completion disagree.

At deployment, NavJev replaces only the stopping decision. The native STOP output of the frozen navigator is suppressed, while all movement actions remain unchanged. At each eligible step, NavJev evaluates the current instruction-history pair: when its stopping criterion is satisfied, it terminates the episode; otherwise, navigation continues with the next action proposed by the base policy. Thus, NavJev can both prevent premature termination and stop a trajectory after it reaches an instruction-consistent state. Inference requires only the instruction and visual observations, with no access to goal-set annotations or reference endpoints. Appendix C.1 details the input construction and score computation, and Appendix C.2 specifies the inference configuration.

5.2 CROSS-BACKBONE EVALUATION

We evaluate the trained checkpoint and its stopping rule on six frozen policies over the full R2R val_unseen split: ETP-R1 (Ye et al., 2025), AwareVLN (Guo et al., 2026), StreamVLN (Wei et al., 2026), NaVILA (Cheng et al., 2025), Uni-NaVid (Zhang et al., 2025a), and NaVid (Zhang et al., 2024) (Table 2); the six are selected for strong performance on the two benchmarks of Section 4. Base scores come from the frozen runs of Table 1; NavJev-equipped runs re-execute the same episodes under the stopping rule of §5.1. GSR and GOSR are computed on the same 1,839 episodes with the manually confirmed goal sets of Section 3.2.

NavJev improves GSR on all six policies by 0.60–2.67 percentage points and raises the unweighted mean from 41.05 to 42.97, while mean GOSR moves by less than one point, from 46.95 to 47.62. Continuing past a suppressed stop can recover additional arrivals, yet the mean GSR gain exceeds the mean GOSR movement, so most of the improvement comes from converting reached states into stops. The gains also track each policy's termination headroom, measured by the base arrival–termination gap: AwareVLN and NaVILA, which enter with the two largest gaps (8.27 and 6.14 points), gain the most, and the three policies with the smallest gaps show the three smallest gains.

Finding 4. **Termination can be repaired at deployment.** One frozen NavJev checkpoint improves GSR on all six policies, and the gains track the arrival–termination gap of each policy: stopping is a decision that can be layered onto frozen navigators.

6 CONCLUSION

Termination in VLN should reflect the positions at which an instruction permits an agent to stop. We formalize this target as an instruction-conditioned goal set and evaluate both whether a trajectory reaches this set and whether the agent terminates within it. Reassessing 17 methods on R2R and RxR changes both success labels and method rankings, showing that endpoint proximity can mischaracterize task completion. Goal-set evaluation further separates trajectories that fail to reach any valid stopping region from those that reach one but terminate incorrectly. To address the latter failure mode, NavJev learns a stopping policy from goal-set supervision and improves all six frozen R2R policies, increasing the mean Goal Set Success Rate from 41.05% to 42.97%. Since many failures occur before a valid region is reached, improved stopping remains complementary to advances in navigation. Overall, these results establish termination as an explicit, measurable, and improvable component of instruction following.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, including improving grammar, clarity, and readability. They were also used to support routine data statistics and visualization-related coding tasks, such as data aggregation, summary statistics, and figure-generation scripts. All AI-assisted content and code were reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

Section 3 defines the construction, statistics, and annotation evidence; Sections 4 and 5 document the metric analysis, policy diagnosis, and learning interfaces. Appendix A documents annotation construction, Appendix B the evaluation protocol and results, Appendix A.4 the independent position-level audit, and Appendix C the NavJev training and inference protocol. The supplementary material specifies the annotation criteria, evaluation populations, metric definitions, and shared NavJev configuration.

REFERENCES

Dong An, Yuankai Qi, Yangguang 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*, 2023. URL https://arxiv.org/abs/2212.04385.

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. ETPNav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5130–5145, 2025. doi: 10.1109/TPAMI.2024.3386695. URL https://arxiv.org/abs/2304.03047. First published online in 2024.

Peter Anderson, Angel 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. URL https://arxiv.org/abs/1807.06757.

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*, 2018b. URL https://arxiv.org/abs/1711.07280.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *International Conference on 3D Vision*, 2017. URL https://arxiv.org/abs/1709.06158.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K. Wong. MapGPT: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://arxiv.org/abs/2401.07314.

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*, 2021. URL https://arxiv.org/abs/2110.13309.

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. URL https://arxiv.org/abs/2202.11742.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2412.04453.

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*, 2018. URL https://arxiv.org/abs/1806.02724.

Wenxuan Guo, Xiuwei Xu, Yichen Liu, Xiangyu Li, Hang Yin, Huangxing Chen, Wenzhao Zheng, Jianjiang Feng, Jie Zhou, and Jiwen Lu. AwareVLN: Reasoning with self-awareness for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2605.22816.

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. URL https://openaccess.thecvf.com/content/CVPR2022/html/Hong_Bridging_the_Gap_Between_Learning_in_Discrete_and_Continuous_Environments_CVPR_2022_paper.html.

Jiewen Hou, Meina Kan, Lixuan Zhang, Hao Liang, Shiguang Shan, and Xilin Chen. Collaborative map-based and route-based policy learning for continuous vision-and-language navigation. *IEEE Robotics and Automation Letters*, 11(4):4345–4352, 2026. doi: 10.1109/LRA.2026.3662659.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1907.05446.

InternNav Team. InternVLA-N1: An open dual-system navigation foundation model with learned latent plans, 2025. URL https://internrobotics.github.io/internvla-n1.github.io/static/pdfs/InternVLA_N1.pdf. Technical report.

Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1862–1872, 2019. URL https://aclanthology.org/P19-1181/.

Sung June Kim, Sangpil Kim, and Honglak Lee. Path-level hindsight instructions for semantic exploration in vision-language navigation. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2607.01754.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, 2020. URL https://arxiv.org/abs/2004.02857.

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

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

Haoyuan Li, Rui Liu, Hehe Fan, and Yi Yang. Let's reward step-by-step: Step-aware contrastive alignment for vision-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 482–500, 2026a. doi: 10.1007/978-3-032-37211-6_27.

Junjie Li, Nan Zhang, Xiaoyang Qu, Kai Lu, Guokuan Li, Jiguang Wan, and Jianzong Wang. RATE-Nav: Region-aware termination enhancement for zero-shot object navigation with vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6564–6574, 2025. URL https://arxiv.org/abs/2506.02354.

Xueying Li, Feng Lyu, Hao Wu, Mingliu Liu, Jia-Nan Liu, and Guozi Liu. Stop wandering: Efficient vision-language navigation via metacognitive reasoning. *arXiv preprint arXiv:2604.02318*, 2026b. URL https://arxiv.org/abs/2604.02318.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. InstructNav: Zero-shot system for generic instruction navigation in unexplored environment. In *Proceedings of the 8th Conference on Robot Learning*, volume 270, pp. 2049–2060, 2024. URL https://arxiv.org/abs/2406.04882.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r1GAsjC5Fm.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. REVERIE: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. URL https://arxiv.org/abs/1904.10151.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Xiangyu Shi, Ruoxi Yang, Wei Tao, Jiwen Zhang, Yanyuan Qiao, and Qi Wu. From region arrival to instance-level grounding in vision-and-language navigation. *arXiv preprint arXiv:2607.03792*, 2026. URL https://arxiv.org/abs/2607.03792.

Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2610–2621, 2019. URL https://arxiv.org/abs/1904.04195.

Liuyi Wang, Xinyuan Xia, Hui Zhao, Hanqing Wang, Tai Wang, Yilun Chen, Chengju Liu, Qijun Chen, and Jiangmiao Pang. Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2507.13019.

Zehao Wang, Minye Wu, Yixin Cao, Yubo Ma, Meiqi Chen, and Tinne Tuytelaars. Navigating the nuances: A fine-grained evaluation of vision-language navigation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. URL https://arxiv.org/abs/2409.17313.

Zihan Wang and Gim Hee Lee. g3D-LF: Generalizable 3D-language feature fields for embodied tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14191–14202, 2025. URL https://arxiv.org/abs/2411.17030.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. StreamVLN: Streaming vision-and-language navigation via SlowFast context modeling. In *IEEE International Conference on Robotics and Automation*, 2026. URL https://arxiv.org/abs/2507.05240.

Jiannan Xiang, Xin Eric Wang, and William Yang Wang. Learning to stop: A simple yet effective approach to urban vision-language navigation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 699–707, 2020. URL https://aclanthology.org/2020.findings-emnlp.62/.

Shuhao Ye, Sitong Mao, Yuxiang Cui, Xuan Yu, Shichao Zhai, Wen Chen, Shunbo Zhou, Rong Xiong, and Yue Wang. ETP-R1: Evolving topological planning with reinforcement fine-tuning for vision-language navigation in continuous environments. *arXiv preprint arXiv:2512.20940*, 2025. URL https://arxiv.org/abs/2512.20940.

Hang Yin, Xiuwei Xu, Lingqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. UniGoal: Towards universal zero-shot goal-oriented navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2503.10630.

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM: Vision-language frontier maps for zero-shot semantic navigation. In *IEEE International Conference on Robotics and Automation*, 2024. URL https://arxiv.org/abs/2312.03275.

Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, Xing Wei, and Ning Guo. JanusVLN: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.22548.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. In *Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2402.15852.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks. In *Robotics: Science and Systems*, 2025a. URL https://arxiv.org/abs/2412.06224.

Lingfei Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A novel memory representation via annotated semantic maps for vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025b. URL https://arxiv.org/abs/2502.13451.

Zekai Zhang, Weiye Zhu, Hewei Pan, Xiangchen Wang, Rongtao Xu, Xing Sun, and Feng Zheng. ActiveVLN: Towards active exploration via multi-turn RL in vision-and-language navigation. *arXiv preprint arXiv:2509.12618*, 2025c. URL https://arxiv.org/abs/2509.12618.

Chongyang Zhao, Yuankai Qi, and Qi Wu. Mind the gap: Improving success rate of vision-and-language navigation by revisiting oracle success routes. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 4349–4358, 2023. doi: 10.1145/3581783.3611736. URL https://arxiv.org/abs/2308.03244.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. NavGPT-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2407.12366.

Fengda Zhu, Xiwen Liang, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. SOON: Scenario oriented object navigation with graph-based exploration. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2103.17138.

Weiye Zhu, Zekai Zhang, Xiangchen Wang, Hewei Pan, Teng Wang, Tiantian Geng, Rongtao Xu, and Feng Zheng. NaVIDA: Vision-language navigation with inverse dynamics augmentation. *arXiv preprint arXiv:2601.18188*, 2026. URL https://arxiv.org/abs/2601.18188.

Appendix Contents

A Goal-Set Construction and Annotation Quality 14
A.1 From terminal conditions to grounded constraints 14
A.2 Geometric representation 15
A.3 Annotation coverage and human review 15
A.4 Independent audit of stopping positions 16
B Evaluation Protocol and Supporting Results 17
B.1 Metrics and evaluation conventions 17
B.2 Methods and evaluation populations 18
B.3 Score changes and policy comparisons 18
B.4 Separating arrival from termination 20
B.5 Instruction-specific goal sets 20
B.6 Qualitative examples of the two disagreement directions 21
C NavJev Training and Deployment 21
C.1 Visual-history input and completion score 21
C.2 Stepwise completion decisions and stopping rule 21
C.3 Training and paired evaluation 23

A Goal-Set Construction and Annotation Quality

This appendix specifies how terminal conditions are grounded and compiled into navigable regions, then describes annotation review and an independent audit of stopping positions.

A.1 From Terminal Conditions to Grounded Constraints

The construction follows the two stages in Section 3.2: grounding terminal conditions in the scene and compiling them into a navigable region. Each instruction is treated separately, even when several instructions describe the same reference trajectory. Language specifies the terminal requirements; the reference trajectory and scene geometry resolve their spatial referents.

Interpreting the terminal conditions. Qwen3.8-Flash extracts a primary destination, additional referenced entities, and terminal spatial constraints from the complete instruction. Each entity and constraint is linked to its supporting language span. Route actions and identifying attributes are distinguished from stopping conditions: a turn along the route does not automatically impose a terminal side constraint, and "the cabinet with a hat on it" identifies the cabinet without requiring the agent to stand on the hat. The stopping constraints are combined conjunctively. For example, "enter the bedroom and stop beside the chair" yields a room predicate and an object-proximity predicate, each associated with its grounded entity.

Grounding with scene context. The reference endpoint identifies the floor and a local search area covering its source room and rooms connected through adjacent openings. The Matterport3D semantic inventory supplies candidate rooms and objects, including objects crossing room boundaries. Category matches and reference-path proximity organize the search; top-down views establish connectivity, while RGB-D panoramas and object crops resolve appearance and spatial extent. Source room polygons are checked against the functional space named in the instruction.

Travel-relative constraints use the reference approach direction; object-relative constraints require evidence of the relevant side. An entry condition uses the opening consistent with the reference path's final entry into the target room. Competing interpretations remain separate hypotheses for review. The endpoint supplies search context rather than the center of object-proximity expansion, so the mask is an operational realization of $\mathcal{G}(I, E)$ under this reference context and the chosen navigable support.

If the local evidence does not resolve a referent or a required relation, the proposal is held for human review rather than dropping that constraint. Only a confirmed grounding is compiled into the final region.

Table 3: **Spatial constraints used in construction.** Required predicates are intersected on navigable support.

Predicate	Geometric rule
Inside / outside	Retain cells inside / outside the grounded room region, checked against the functional space named in the instruction.
Near / beside	Retain $d_{A,o}(v) \leq 3\,\text{m}$, using surface-gap initialization and constrained graph propagation (Eq. 4).
Entry	Select the opening crossed by the reference path on its final entry into the target room. An unqualified "enter" retains the room; "just inside" and "a few steps into" also require room-restricted graph distance from the opening's interior cells of at most 3 m.
Side	Retain cells in the requested half-plane through the grounded referent center. Travel-relative left/right uses the normal to the final nonzero reference-approach direction; object-relative sides use the identified face orientation from scene views. Ambiguous orientations are resolved in human review.
Stair / level	Retain navigable cells on the grounded destination landing or level, identified from the reference path's final level transition and scene views. Stored surface height separates vertically overlapping cells.

A.2 Geometric Representation

Grounded constraints are intersected with navigability to form $\mathcal{G}$. Object proximity uses a single 3 m threshold; room, entry, and directional predicates remain hard constraints.

Coordinates and navigable support. Positions use Habitat coordinates (x, y, z), with y vertical; Matterport3D house coordinates are transformed by $[x, y, z]_{\text{house}} \mapsto [x, z, -y]$. A grid with spacing $h = 0.1\,\text{m}$ samples navigation-mesh surfaces on the selected floor that are reachable from the episode start. Each cell retains its surface height to distinguish overlapping floors. Adjacent cells are connected only when movement is traversable in both directions, and diagonal connections cannot cut occupied corners.

Constraint composition and proximity. Let V denote the reachable surface cells and $A \subseteq V$ the domain remaining after the required room, entry, directional, and level predicates, with occupied referent footprints excluded. Table 3 summarizes these predicates. For object o, expansion starts from navigable seed cells $S_o \subseteq A$ with an unobstructed connection to its surface. Let $g_o(s)$ be the surface gap at seed s and d_A the shortest-path distance on the graph induced by A. The construction distance and final mask are

$$d_{A,o}(v) = \min_{s \in S_o} \big[g_o(s) + d_A(s, v) \big], \qquad M = \big\{ v \in A : d_{A,o}(v) \leq 3\,\text{m},\ \forall o \in O_I \big\}, \tag{4}$$

where O_I contains objects with terminal proximity requirements. Unreachable cells have infinite distance. If no object-proximity predicate is required, $M = A$. Each grounding hypothesis is compiled separately, and an empty mask is referred for inspection. Propagation respects walls and the other hard constraints; it does not expand an entry band. Distances to the completed set for evaluation are defined separately in Appendix B.1.

Continuous-position membership. The policy position and mask share the coordinate transform and floor convention. A position must lie on the selected reachable navigation surface; it is then assigned to the horizontally nearest surface cell on that surface. A tie across mask labels is rejected, and a position with no matching surface is outside the set. This assignment adds no distance margin to the mask. The mask represents terminal positional admissibility; GSR does not assess final heading or intermediate route events. Figure 5 illustrates the construction.

A.3 Annotation Coverage and Human Review

The construction produces 65,253 training annotations: 9,878 R2R and 55,375 RxR instructions. Validation annotations cover all 1,839 R2R and 11,006 RxR val_unseen episodes. Training annotations supply terminal supervision; validation annotations define policy-evaluation targets.

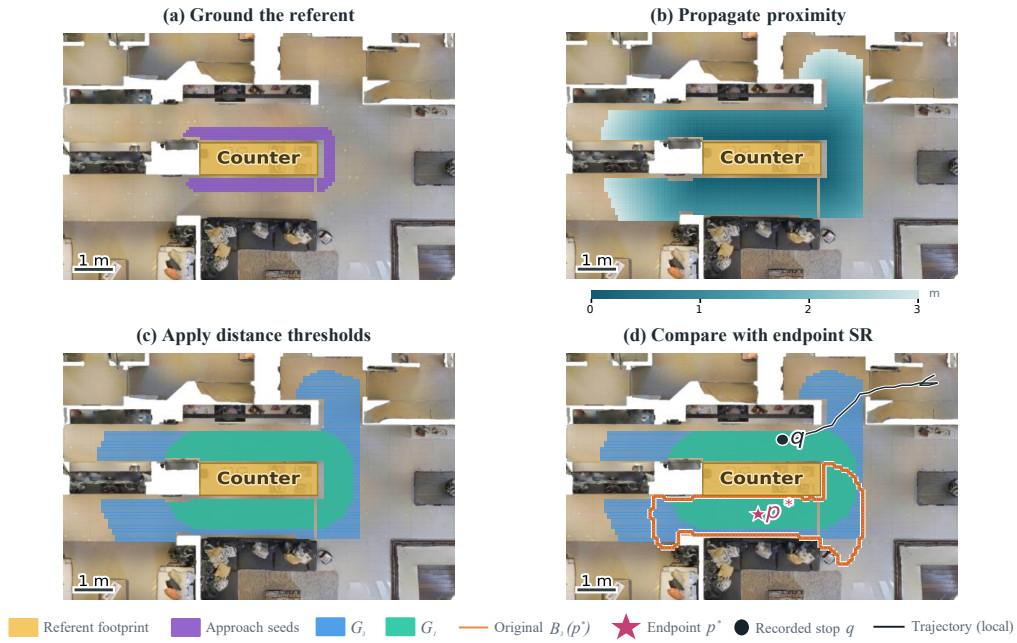

Figure 5: **Construction stages of one goal set.** The grounded referent (a), navigable proximity propagated from it (b), the thresholded goal set (c), and comparison with the endpoint SR ball (d).

Construction checks combine geometry validation with model-assisted semantic review of target identity, stopping constraints, room boundaries, and spatial extent. A complementary check judges sampled interior, boundary, and exterior positions from scene views and measured distances, without showing the candidate mask or reference endpoint. Automatic acceptance requires unambiguous grounding and agreement across region and position checks; other proposals undergo human inspection.

Every validation annotation additionally undergoes human review against the instruction and scene. Reviewers accept 91.2% of initial proposals unchanged and revise and re-check the remainder. Endpoint consistency retrieves cases for inspection but does not determine acceptance. This proposal-acceptance rate describes the correction workload; the independent position-level audit below assesses membership decisions.

A.4 Independent Audit of Stopping Positions

The audit separates policy stops from spatial diagnostics. Two annotators independently judge whether each position satisfies the terminal spatial conditions; a third adjudicates disagreements. They see the instruction and scene views, with route context when needed, while masks, automatic decisions, model names, and highlighted reference endpoints are hidden.

Sampling. All audited positions come from R2R val_unseen episodes with confirmed goal sets. For each of three frozen policies (AwareVLN, StreamVLN, and NaVIDA, whose endpoint-success rejection rates on this split span 15.4–35.4%; Table 5), 100 final stops are drawn uniformly at random from the policy's frozen run, giving 300 policy-stop positions. The diagnostic sample draws 400 further positions from the annotated goal sets of the same split, sampling region interiors, both sides of boundaries, and exteriors. Policy and diagnostic results are reported separately because deliberate boundary sampling changes the position distribution.

Agreement and paired errors. Agreement is computed before adjudication on positions that received two valid initial labels (297 of the 300 policy stops and 395 of the 400 diagnostic positions); the remaining positions go directly to adjudication. Both the endpoint positional test $\mathbf{1}[d_E(p, p^*) < 3\,\mathrm{m}]$ and goal-set membership $\mathbf{1}[p \in \mathcal{G}]$ are compared with the same adjudicated hu-

Table 4: **Independent position-level audit.** Agreement uses initial independent labels; errors, precision, recall, and Cohen's κ use adjudicated labels on identical positions.

Quantity	Policy stops	Spatial diagnostics
Adjudicated positions	300	400
Human-valid / human-invalid	216 / 84	192 / 208
Agreement before adjudication (%)	91.6	84.8
Endpoint FP / FN	36 / 18	57 / 61
Goal-set FP / FN	8 / 13	17 / 19
Goal-set precision / recall (%)	96.2 / 94.0	91.1 / 90.1
Cohen's κ (goal-set vs. human)	0.83	0.82
Error reduction (pp)	+11.0	+20.5

man labels. A false positive accepts a human-invalid position; a false negative rejects a human-valid one. These are positional judgments, since diagnostic positions have no stopping action. Table 4 reports the sample composition, agreement, paired errors, and goal-set quality measures. Paired error reduction is the endpoint error rate minus the goal-set error rate on the same positions; positive values favor goal-set membership.

B Evaluation Protocol and Supporting Results

This appendix connects the evaluation definitions to the findings in Section 4. We specify the metrics and evaluation populations, report the score changes and paired outcomes associated with Table 1, and then use trajectory visits and qualitative cases to explain how arrival differs from successful termination.

B.1 Metrics and Evaluation Conventions

All metrics use the 3 m goal set $\mathcal{G}$ defined in Appendix A.2. For episode i, let b_T^i indicate an explicit STOP, p_T^i be the final position, and $\{p_t^i\}_{t=0}^{T_i}$ be the recorded trajectory, including its final position. Over an evaluation population $\mathcal{E}$ of size N, success and arrival are

$$\mathrm{GSR} = \frac{1}{N} \sum_{i \in \mathcal{E}} b_T^i \, \mathbf{1}[p_T^i \in \mathcal{G}^i], \tag{5}$$

$$\mathrm{GOSR} = \frac{1}{N} \sum_{i \in \mathcal{E}} \mathbf{1}[\exists t \in \{0, \ldots, T_i\} : p_t^i \in \mathcal{G}^i]. \tag{6}$$

Every episode contributes once to each metric. GOSR uses recorded positions without interpolation, so it measures whether the recorded trajectory visits the set. Because successful termination entails such a visit, $\mathrm{GSR} \leq \mathrm{GOSR}$.

Distance and Path Efficiency. Let d_E denote navigable shortest-path distance in the scene. The remaining distance to the set and the shortest start-to-set distance are

$$\mathrm{GNE}_i = \inf_{q \in \mathcal{G}^i} d_E(p_T^i, q), \qquad \ell_i^* = \inf_{q \in \mathcal{G}^i} d_E(p_0^i, q). \tag{7}$$

For executed path length L_i, define the episode efficiency as

$$\mathrm{GSPL}_i = b_T^i \, \mathbf{1}[p_T^i \in \mathcal{G}^i] \frac{\ell_i^*}{\max(\ell_i^*, L_i)}. \tag{8}$$

The ratio is one when both lengths are zero. Thus, a successful stationary trajectory starting in the set receives efficiency one, whereas a positive-length trajectory starting in the set has efficiency zero. Mean GNE and GSPL are computed over their respective sets of episodes with finite, defined values; GSR and GOSR retain the full evaluation population.

Paired Changes in Success Labels. Let n_{ab} count episodes labeled a by endpoint success and b by goal-set success, with both criteria requiring the same explicit STOP. The four counts sum to N, and

$$\text{GSR} - \text{SR} = \frac{n_{01} - n_{10}}{N}, \qquad D = \frac{n_{10} + n_{01}}{N}. \tag{9}$$

Here D is the fraction of labels that change. To distinguish the two directions, we also report

$$\text{FIR} = \frac{n_{10}}{n_{10} + n_{11}}, \qquad \text{FER} = \frac{n_{01}}{n_{00} + n_{01}}. \tag{10}$$

FIR is the fraction of endpoint successes rejected by the goal set, and FER the fraction of endpoint failures accepted by it. An empty conditioning group has an undefined conditional rate. These rates describe disagreement between the criteria; human judgments are assessed separately in Appendix A.4.

B.2 Methods and Evaluation Populations

The evaluation covers 17 open-source VLN methods. Table 1 includes CMA (Krantz et al., 2020) and HPN (Krantz et al., 2021), CWP (VLN-BERT) (Hong et al., 2022), BEVBert-CE (An et al., 2023), ETPNav (An et al., 2025), g3D-LF (Wang & Lee, 2025), ETP-R1 (Ye et al., 2025), CoMaR (ETPNav) (Hou et al., 2026), NaVid (Zhang et al., 2024), NaVILA (Cheng et al., 2025), Uni-NaVid (Zhang et al., 2025a), StreamVLN (Wei et al., 2026), AwareVLN (Guo et al., 2026), ActiveVLN (Zhang et al., 2025c), NaVIDA (Zhu et al., 2026), JanusVLN (Zeng et al., 2026), and InternVLA-N1 (InternNav Team, 2025). CMA and HPN are separate policies; InternVLA-N1 uses the InternNav configuration.

Every R2R run covers all 1,839 val_unseen episodes. RxR results follow the language coverage of the corresponding checkpoint: ETPNav, ETP-R1, StreamVLN, and InternVLA-N1 use all 11,006 multilingual episodes; NaVid, Uni-NaVid, AwareVLN, ActiveVLN, NaVIDA, and NaVILA use all 3,669 English episodes (en-US and en-IN). All included episodes have manually confirmed goal sets. Each trajectory is scored under the endpoint and goal-set criteria with the policy, path, and stopping action held fixed, so within-run score differences isolate the terminal criterion.

B.3 Score Changes and Policy Comparisons

Table 5 reports, on exactly the populations of Table 1, both reclassification directions with their counts n_{10} and n_{01}, the changed-label fraction D of Eq. 9, the conditional disagreement rates, and the visit structure of the rejected successes; the agreement counts n_{00} and n_{11} follow from N, SR, and GSR, and the arrival–termination gap follows from GOSR−GSR in Table 1.

The Relabeling Mostly Removes Unsupported Successes. On R2R, every policy loses between 159 and 430 endpoint successes to the goal set while gaining at most 14 endpoint failures. Rejections therefore account for between 93% and 99% of all changed labels, and the net GSR drop stays within 1.6 points of the changed-label fraction D. On RxR the exchange is more balanced, with the acceptance rate FER reaching 3.24% against at most 1.37% on R2R, yet rejections still dominate every run, n_{10} rising to 1,428 while n_{01} stays at or below 140. Across both benchmarks, the relabeling thus acts chiefly by withdrawing success from endpoint-near stops, most of which never entered an admissible region (Appendix B.4). The acceptance direction is smaller in volume, but it is the component that no distance threshold can reproduce: it credits instruction-consistent stops that lie beyond the endpoint radius.

The R2R Leader Reversal Is Arrival-Driven. NaVIDA trails ETP-R1 by 6.20 points of SR on R2R yet leads by 6.30 points of GSR. Writing GSR as the product of arrival and conversion, where conversion is the fraction of arrivals that end in successful stops (GSR/GOSR), locates the reversal in arrival: NaVIDA reaches a goal set on 57.15% of episodes against 49.92% for ETP-R1, a 7.23-point GOSR lead, while the two policies convert arrivals at nearly identical rates of 88.1% and 88.2%. The survival lens agrees: FIR rejects 15.36% of NaVIDA's endpoint successes against 32.89% for ETP-R1. ETP-R1's endpoint advantage is thus concentrated in stops near the demonstrated endpoint that lie outside the admissible region, and once arrival governs the score, the ranking reverses.

Table 5: **Endpoint–goal-set label changes on the populations of Table 1.** n_{10} (n_{01}): endpoint successes rejected (endpoint failures accepted) by the goal set; D: fraction of changed labels; FIR (FER): share of endpoint successes rejected (failures accepted); Never: % of n_{10} whose trajectory never visited the set.

Method	n_{10}	n_{01}	D (%)	FIR	FER	Never (%)
R2R-CE ($N = 1{,}839$)						
CMA (Krantz et al., 2020)	165	9	9.46	29.73	0.70	87.3
HPN (Krantz et al., 2021)	252	5	13.97	36.47	0.44	92.1
CWP (VLN-BERT) (Hong et al., 2022)	324	14	18.38	39.90	1.36	94.8
ETPNav (An et al., 2025)	348	8	19.36	33.24	1.01	90.8
BEVBert-CE (An et al., 2023)	430	5	23.65	39.49	0.67	92.1
g3D-LF (Wang & Lee, 2025)	427	6	23.55	38.61	0.82	90.9
ETP-R1 (Ye et al., 2025)	395	4	21.70	32.89	0.63	89.4
CoMaR (ETPNav) (Hou et al., 2026)	376	6	20.77	34.56	0.80	90.2
NaVid (Zhang et al., 2024)	228	8	12.83	29.19	0.76	88.2
NaVILA (Cheng et al., 2025)	285	3	15.66	28.33	0.36	91.2
StreamVLN (Wei et al., 2026)	186	11	10.71	17.95	1.37	91.9
Uni-NaVid (Zhang et al., 2025a)	159	12	9.30	16.95	1.33	92.5
AwareVLN (Guo et al., 2026)	425	7	23.49	35.42	1.10	94.8
ActiveVLN (Zhang et al., 2025c)	283	8	15.82	32.98	0.82	95.1
NaVIDA (Zhu et al., 2026)	167	6	9.41	15.36	0.80	74.3
JanusVLN (Zeng et al., 2026)	327	7	18.16	33.85	0.80	92.0
InternVLA-N1 (InternNav Team, 2025)	274	3	15.06	25.92	0.38	94.5
RxR-CE, multilingual ($N = 11{,}006$)						
ETPNav (An et al., 2025)	1,428	120	14.07	23.01	2.50	87.9
ETP-R1 (Ye et al., 2025)	1,164	140	11.85	17.40	3.24	85.5
StreamVLN (Wei et al., 2026)	477	115	5.38	15.01	1.47	66.7
InternVLA-N1 (InternNav Team, 2025)	690	133	7.48	19.21	1.79	74.5
RxR-CE, English ($N = 3{,}669$)						
NaVid (Zhang et al., 2024)	248	25	7.44	16.06	1.18	84.3
Uni-NaVid (Zhang et al., 2025a)	204	36	6.54	10.63	2.06	72.5
AwareVLN (Guo et al., 2026)	377	48	11.58	18.27	2.99	84.1
ActiveVLN (Zhang et al., 2025c)	153	22	4.77	25.46	0.72	85.0
NaVIDA (Zhu et al., 2026)	236	38	7.47	11.61	2.32	64.4
NaVILA (Cheng et al., 2025)	321	40	9.84	19.20	2.00	88.8

Reversals Recur Within Matched RxR Populations. RxR runs must be compared within the same instruction population, since the multilingual and English runs evaluate different episodes. Within the English runs, the R2R pattern repeats: AwareVLN leads NaVIDA by 0.85 points of SR yet trails by 2.73 points of GSR, and NaVIDA again supplies the arrival margin, leading GOSR by 2.37 points. On that same English population, NaVILA records 37.91% GSR versus 45.57% SR, with 321 endpoint successes rejected and 40 endpoint failures accepted by the goal set. Within the multilingual runs, ETP-R1's 6.98-point GSR lead over ETPNav draws on both components at once, combining a 4.99-point GOSR lead with a conversion advantage of 89.4% over 84.6%. Arrival and conversion thus move GSR independently across policy families, which is the separation that Section 4.2 develops into distinct competencies.

B.4 Separating Arrival from Termination

The difference between GOSR and GSR identifies trajectories that reach an admissible region but do not stop successfully there. To decompose all semantic failures, let V indicate a recorded visit to the set, H final-position membership, and S an explicit STOP. Since the recorded trajectory includes the final position, H implies V, and the failure modes partition as

$$1 - \mathrm{GSR} = \Pr(\neg V) + \Pr(V \wedge \neg H) + \Pr(H \wedge \neg S). \tag{11}$$

The first term measures failure to arrive, the second arrival followed by termination outside the set, and the third ending inside without issuing STOP. The last two terms sum to $\mathrm{GOSR} - \mathrm{GSR}$, so the main-table scores already separate never-arrival from the combined termination failure.

Among the endpoint successes that GSR rejects, recorded as the paired counts n_{10} in Table 5, only the first two modes can occur, because these trajectories already issued STOP. Aggregate scores leave the split between the two undetermined, so the separation draws on the underlying episode records; the Never column of Table 5 reports it for every run as the fraction of n_{10} that never visited the set, with the visited-and-lost share as its complement.

Most Lost Successes Never Arrive. Never visiting the set accounts for 74.3–95.1% of n_{10} on R2R and 64.4–88.8% on RxR, and the dominance holds for every one of the 27 runs. Most endpoint successes that the goal set rejects are therefore stops near the demonstrated endpoint whose trajectories never entered the admissible region: the endpoint criterion certifies proximity that, in the large majority of cases, never became arrival.

Visited-Then-Lost Cases Concentrate in a Few Policies. The visited component stays small on R2R, where every policy other than NaVIDA lies at or below 12.7% and NaVIDA alone reaches 25.7%. On RxR it separates the runs into two groups: NaVIDA, StreamVLN, Uni-NaVid, and InternVLA-N1 lose 25.5–35.6% of their rejected successes after a visit, while the remaining six runs lose at most 15.9%. Arrival frequency alone cannot explain this split, because StreamVLN pairs the lowest GOSR among the multilingual runs, 35.44%, with the second-largest visited share, 33.3%. For the high-share policies, the endpoint label is therefore lost late, after the region has been reached and passed through. These visited cases are precisely the episodes that better stopping decisions could recover, and they define the termination target that Section 5 addresses.

B.5 Instruction-Specific Goal Sets

Instructions associated with the same route and endpoint provide a comparison in which the endpoint criterion is fixed. We pair instructions from the same scene and reference route whose endpoints are within 0.01 m, then measure intersection-over-union between their goal sets, giving 885 R2R pairs and 25,123 RxR pairs.

Disagreement Follows the Grounded Referent. Median IoU is 1.0, and only 42 of 885 R2R pairs (4.7%) and 1,226 of 25,123 RxR pairs (4.9%) fall below 0.5. On RxR, same-language pairs disagree at 237 of 5,092 (4.7%), essentially the same rate as the cross-language pairs at 989 of 20,031 (4.9%), so language pairing leaves the disagreement rate unchanged. The agreeing majority is the expected baseline—a shared endpoint trivially predicts it, and instructions describing the same route usually name the same terminal referent—and the informative cases are the pairs that disagree: the two instructions ground different referents within the same room, and their goal sets follow those

referents even though the endpoint, and hence any endpoint-distance rule, is identical for both. Position alone therefore under-determines the stopping region, as Section 3.3 argues from the same counts.

B.6 Qualitative Examples of the Two Disagreement Directions

Policy Stops Exhibit Both Directions. Figure 6 connects the two reclassification directions to the terminal conditions in the original instructions across eight policy stops. In the room-entry cases, stops near the demonstration endpoint remain outside the requested room and are rejected by the goal set. In the object-proximity cases, alternative stops satisfy the relation to the named object despite exceeding the endpoint threshold. These cases illustrate why a change in terminal representation can both reject endpoint-near stops and accept valid alternatives.

C NavJev Training and Deployment

NavJev uses the goal sets as training labels and predicts completion from visual history at deployment. This appendix specifies its input and score, the stopping rule that consumes that score together with its measured serving cost, and the training and paired evaluation protocol. The same model and controller are shared across the six frozen navigation policies.

C.1 Visual-History Input and Completion Score

NavJev implements Eq. 3 with Qwen3-VL-4B-Instruct. At navigation step t, its prompt contains the current episode's instruction and 16 egocentric frames sampled uniformly from that episode's trajectory prefix. The sampling positions are given by $\mathrm{linspace}(0, t, 16)$, so the history spans the initial view through the current observation. The prompt asks for a binary completion decision, expressed as `stop` or `not stop`. These frames and the instruction prompt are the complete inference input.

Inference uses one fixed checkpoint served through vLLM. Let $\ell_t(\texttt{stop})$ and $\ell_t(\texttt{not})$ be the first-token log probabilities for the affirmative and negative labels. The completion score is

$$p_{\mathrm{stop},t} = c_t = \frac{\exp \ell_t(\texttt{stop})}{\exp \ell_t(\texttt{stop}) + \exp \ell_t(\texttt{not})}. \tag{12}$$

Both terms of Eq. 12 therefore read only the first generated token, so the second word of the two-word negative response stays outside the score. The controller consumes this normalized score directly.

C.2 Stepwise Completion Decisions and Stopping Rule

One navigation step is one low-level action—advancing 0.25 m forward or turning 15° to the left or right—and the step index t counts executed low-level actions for every evaluated backbone. NavJev is queried at every step $t \geq 20$ while the episode remains active; before step 20, the navigator follows its base policy alone, with its own STOP output suppressed under the rule below. Let k_t count consecutive queried steps whose completion score is at least 0.5, with $k_{19} = 0$:

$$k_t = \begin{cases} k_{t-1} + 1, & p_{\mathrm{stop},t} \geq 0.5, \\ 0, & p_{\mathrm{stop},t} < 0.5. \end{cases} \qquad (t \geq 20). \tag{13}$$

As soon as $k_t = 2$, the controller replaces the navigator's action with STOP and terminates the episode. Otherwise, the navigator's proposed action is executed, with one exception: a STOP emitted by the navigator itself is never executed. When the navigator proposes STOP, that action is masked from the navigator's action space and the navigator replans at the same step: it is re-queried with STOP excluded until it proposes a movement action, which is executed in place of the suppressed stop. The episode therefore continues on the frozen navigator's movement actions until the controller fires STOP or the step limit is reached. This suppression and replanning apply at every step of the episode, including the first 20, so no episode ends on the navigator's own STOP. A score below the threshold resets the consecutive count, so two separated high scores do not trigger stopping. This decision uses only observations available at the current step. Figure 7 illustrates this interaction



Figure 6: **The two disagreement directions across policies.** (A) Stops accepted by endpoint SR but rejected by the goal set; (B) stops rejected by endpoint SR but accepted by the goal set. Each panel overlays the two success regions on the predicted and reference paths.

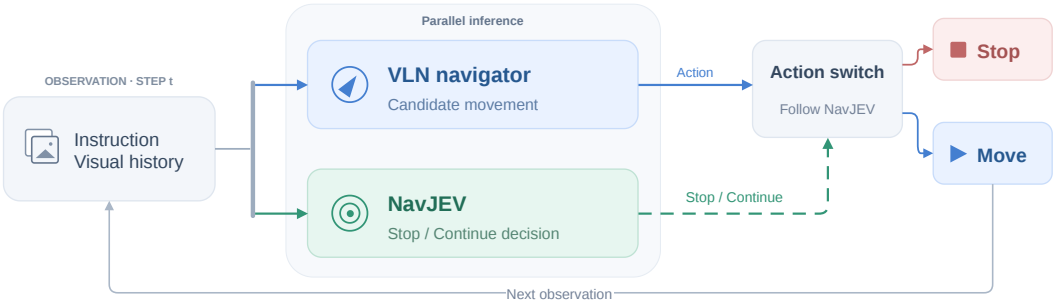

Figure 7: **NavJev deployment.** The navigator proposes actions while NavJev runs in parallel, deciding stop or continue from the instruction and visual history; the controller either forces a STOP or lets the episode continue, and a proposed STOP from the navigator is masked and answered by replanning a movement action.

Table 6: **NavJev serving cost, measured from the deployment server's vLLM logs.** One decision is a single stepwise completion query carrying the instruction and 16 frames; throughput and decision rate are per serving replica.

Item	Measured value
Maximum sequence length	16,384 tokens
GPU-memory share	0.22
KV-cache size	5.05 GiB (36,800 tokens)
Concurrency at full 16,384-token context	2.25 requests
Prompt throughput (per replica)	13.3k tokens/s
Decision rate (per replica)	2.63 decisions/s

between the navigator, NavJev, and the controller. The same checkpoint, query start, threshold, and two-query requirement apply to every evaluated backbone.

Because the controller queries NavJev at every step from $t = 20$ onward, serving cost is the plug-in's main deployment overhead; Table 6 reports the values measured from the deployment server's vLLM logs. A serving replica holds a 0.22 share of GPU memory, with a 5.05 GiB KV cache of 36,800 tokens; at the configured maximum sequence length of 16,384 tokens, this capacity sustains 2.25 concurrent full-length requests. Measured throughput reaches 13.3k prompt tokens per second per replica, corresponding to 2.63 completion decisions per second and roughly 5k prompt tokens per decision. This per-query cost is bounded by design: the uniform 16-frame subsampling of §C.1 holds every prompt to one instruction and 16 frames, however long the trajectory prefix has grown.

C.3 Training and Paired Evaluation

Training states come from the terminal stage of the ground-truth demonstration paths over the full R2R training split, and use membership in the 3 m goal set as the binary target. The sampler combines ordinary examples with the disagreement subsets

$$\mathcal{H}^- = B_3(p^*) \setminus \mathcal{G}, \qquad \mathcal{H}^+ = \mathcal{G} \setminus B_3(p^*), \tag{14}$$

which supply endpoint-near hard negatives and endpoint-far hard positives. Each example pairs the instruction and sampled visual history with a completion label. Training examples require a confirmed goal-set label.

The resulting training set contains 168,726 terminal-stage decisions, all drawn from the R2R training split, including 32,290 `stop` examples and 136,436 `not stop` examples. Optimization runs for one epoch with a global batch size of 128, yielding 1,319 updates when the final partial batch is included. A checkpoint is saved every 500 updates, with an additional checkpoint saved at the end of training. This schedule produces checkpoints at steps 500, 1,000, and 1,319. The checkpoint from step 1,319 is used unchanged for all six policies, so checkpoint selection is determined entirely by the training schedule rather than by downstream evaluation performance.

Table 7: **NavJev training configuration.**

Item	Setting
Completion model	Qwen3-VL-4B-Instruct
Visual encoder	Frozen
Language-model adaptation	LoRA, rank 32 and scale 32
Precision	bfloat16
Maximum context length	8,192 tokens
Optimizer	AdamW
Learning rate	10^{-5}
Schedule	Cosine decay, warm-up ratio 0.05
Global batch size	128
Training steps	1 epoch

Table 7 summarizes the training configuration. All policy experiments combine this fixed checkpoint with the shared 16-frame stopping rule defined in §C.2. Table 2 reports paired evaluations of six frozen policies on the complete R2R val_unseen split comprising 1,839 episodes. GSR and GOSR are computed over the same full evaluation population using the manually confirmed goal sets described in Section 3.2. The base rows correspond to the frozen-policy runs reported in Table 1. For the NavJev-equipped rows, the same episodes are executed again with the STOP output of the navigator suppressed, such that an episode terminates only when the controller issues a forced STOP; all movement actions remain determined by the frozen navigator. The mean column reports the unweighted average across the six policies.