EGOEXO-VLN: EGOCENTRIC-EXOCENTRIC VISION-AND-LANGUAGE NAVIGATION IN DYNAMIC ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing Vision-Language Navigation (VLN) methods primarily rely on egocentric observations, limiting their ability to perceive important environmental changes beyond the robot's current view. To address this limitation, we propose a novel VLN task, named Egocentric-Exocentric Vision-Language Navigation (EgoExo-VLN), which enables agents to follow language instructions in dynamic indoor environments by combining egocentric observations with information from fixed exocentric cameras. EgoExo-VLN goes beyond simply providing additional views by requiring agents to identify task-relevant information across viewpoints, determine whether observed evidence remains valid over time, and decide whether it should influence the current VLN decision. To systematically evaluate these capabilities in the EgoExo-VLN setting, we construct the first VLN benchmark that explicitly incorporates exocentric cameras, consisting of 8,096 episodes across 90 indoor scenes with scene-disjoint training, validation, and test splits. The benchmark focuses on dynamic target relocation and introduces controlled absent, conflict, and non-conflict pedestrian conditions, while also including fully static tasks to cover common VLN scenarios. Building on this benchmark, we propose a VLN framework that selects task-relevant camera information, updates target representations over time to revise VLN goals, and reasons about pedestrian conflicts to determine when the agent should wait or continue moving. On the test split, adding fixed-camera observations increases SR from 17.15% to 41.93% and collision-free success (SafeSR) on pedestrian-present episodes from 18.33% to 47.33%. EgoExo-VLN establishes a new benchmark for studying VLN with egocentric observations and exocentric camera information in dynamic environments.

1 INTRODUCTION

Vision-and-language navigation (VLN) studies how embodied agents follow natural-language instructions to navigate in unfamiliar environments. Solving this task requires an agent to jointly reason over language, visual observations, spatial relationships, and sequential actions. As a fundamental problem at the intersection of vision, language, and embodied intelligence, VLN provides an important foundation for agents that can understand human instructions and autonomously navigate in indoor environments, with potential applications in service robotics, assistive systems, and intelligent spaces.

A series of benchmarks has progressively expanded the scope of VLN. Room-to-Room (R2R) established the basic paradigm of language-guided navigation in photorealistic indoor environments; Room-Across-Room (RxR) extended it to a larger-scale, multilingual setting with longer routes; and VLN-CE removed the assumption of navigation over a predefined graph and formulated VLN in continuous environments with low-level actions (Anderson et al., 2018b; Ku et al., 2020; Krantz et al., 2020). Subsequent benchmarks have further broadened the task along dimensions such as target specification, instruction diversity, interaction, and navigation complexity. Together, these benchmarks have driven VLN from instruction-guided route following toward increasingly challenging embodied navigation settings.

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: Comparison under the same instruction and robot start state but different dynamic events. Ego-only VLN follows an outdated goal after off-view target relocation (left) or fails to anticipate a pedestrian about to cross its planned path (middle). EgoExo-VLN uses fixed cameras to update the goal and anticipate the crossing risk (right).

Despite these advances, conventional fixed-goal VLN benchmarks generally share two assumptions. First, the instruction-specified target remains at a fixed location during an episode. Second, task-relevant visual information is primarily acquired through sensors mounted on the navigating agent itself. Under these assumptions, the main challenge is to interpret an instruction and reach a predefined goal using egocentric observations. In dynamic indoor environments, however, both assumptions can break. As shown in Figure 1, the target may be relocated after the agent has observed its previous position, leaving the agent with a stale belief about where to go. A pedestrian may approach the planned route from around a blind corner and remain outside the current field of view until a collision is imminent. Human-aware VLN benchmarks have begun to introduce dynamic human activity and safety considerations, but navigation is still primarily formulated around observations acquired by the agent itself (Li et al., 2024; Dong et al., 2025). Existing benchmarks therefore do not systematically study an important setting: how should an agent update its navigation decisions when task-relevant world states change outside its current egocentric view?

Fixed environmental cameras provide a natural source of complementary information for this setting. Such cameras are increasingly available in indoor environments and observe the scene from viewpoints external to the navigating agent (Shim & Cho, 2016; Ravankar et al., 2017). A related principle appears in cooperative perception, where observations from multiple vehicles and infrastructure sensors help recover information hidden by occlusion or distance (Wang et al., 2020; Xu et al., 2022). Motivated by this complementary sensing structure, we introduce **Egocentric-Exocentric Vision-and-Language Navigation (EgoExo-VLN)**, a continuous VLN task that combines egocentric observations with fixed exocentric observations to reason about dynamic events beyond the robot's current field of view. To systematically study this problem, we construct the EgoExo-VLN benchmark with 8,096 episodes across 90 HM3D scenes (Ramakrishnan et al., 2021) and scene-disjoint training, validation, and test splits. The benchmark centers on dynamic target relocation and introduces controlled pedestrian-absent, conflict, and non-conflict conditions, while also including fully static tasks. This setting introduces three challenges that are not explicitly captured by conventional VLN benchmarks. First, *cross-view grounding* requires determining which exocentric observations are relevant to the current instruction, robot state, and navigation route. Second, *temporal reasoning* requires determining whether previously observed evidence remains valid as the environment changes. Third, *action-relevance reasoning* requires deciding whether valid external evidence should actually alter the current action. For example, observing a pedestrian does not necessarily require the agent to wait; intervention is needed only when the pedestrian conflicts with the anticipated navigation path. Thus, under the same language instruction, different off-view world states may require different navigation behaviors.

To address these challenges, we develop EgoExo-VLN to fuse task-relevant external evidence with egocentric observations, update target representations, and predict path conflicts. These capabilities allow the agent to revise its goal, ignore irrelevant or stale evidence, and decide when to move, `WAIT`, or `STOP`. Adding external observations raises Test SR from 17.15% to 41.93% and Reloc. SR from 16.94% to 43.12%; camera relevance further improves them to 54.60% and 55.60%, respectively. Path-conflict reasoning also reduces unnecessary waiting under non-conflicting pedes-

trian motion. These results support using fixed-camera evidence with relevance selection, temporal updates, and action-aware control for dynamic VLN.

Our contributions can be summarized as follows:

1. We introduce **EgoExo-VLN**, a continuous vision-and-language navigation task that combines egocentric observations with fixed exocentric observations. Unlike conventional ego-only VLN, the task requires agents to use off-view dynamic evidence to update navigation decisions, introducing new challenges in cross-view grounding, temporal reasoning, and action-relevance reasoning.
2. We construct the first EgoExo-VLN benchmark, containing 8,096 episodes across 90 HM3D scenes. The benchmark centers on dynamic target relocation, includes pedestrian-absent, conflict, and non-conflict conditions, and supplements these settings with fully static tasks.
3. We develop an EgoExo-VLN model that selectively integrates task-relevant exocentric evidence, updates target representations over time, and reasons about route-conditioned pedestrian conflicts. Experiments demonstrate consistent improvements in navigation performance and safe task completion, validating the usefulness of exocentric information for dynamic VLN.

2 Related Work

Vision-and-language navigation. R2R established language-guided navigation in photorealistic indoor scenes (Anderson et al., 2018b), while VLN-CE extended it from predefined navigation graphs to continuous space (Krantz et al., 2020). Later benchmarks introduced remote object grounding, scenario-oriented object navigation, multilingual instructions, and dialogue (Qi et al., 2020; Zhu et al., 2021; Ku et al., 2020; Thomason et al., 2020), supporting advances in cross-modal learning and long-horizon planning. History-aware and map-based policies use recurrent memory, graph reasoning, and learned topological maps for state tracking and exploration (Hong et al., 2021; Chen et al., 2021; 2022; An et al., 2025; 2023). Recent work explores scene adaptation, imagination-based reasoning, streaming context, and navigation foundation models (Hong et al., 2025; Zhao et al., 2025; Wei et al., 2025; Zhang et al., 2026). These fixed-goal evaluations primarily use robot-mounted observations; EgoExo-VLN uses fixed cameras to observe target changes and pedestrian events beyond the robot's view.

Collaborative and external-view perception. Multi-agent perception combines complementary views to reduce sensor blind spots. V2VNet exchanges vehicle features for perception and prediction, while V2X-ViT models vehicle–infrastructure interactions under asynchronous observations and pose errors (Wang et al., 2020; Xu et al., 2022). Where2comm selects informative spatial features to balance perception and communication bandwidth (Hu et al., 2022). These methods motivate infrastructure-assisted perception, primarily for perception and prediction. EgoExo-VLN additionally requires judging whether fixed-camera evidence is task-relevant and current, and whether it should update the goal or trigger movement, `WAIT`, or `STOP`.

Navigation in dynamic and human-aware environments. Object-grounded VLN evaluates navigation to instruction-specified objects (Qi et al., 2020; Zhu et al., 2021), while human-aware benchmarks introduce moving people and safety constraints (Li et al., 2024; Dong et al., 2025). EgoExo-VLN considers off-view target relocation and pedestrians approaching the planned route under an unchanged instruction. It combines fixed-camera observations with matched pedestrian conditions to evaluate when external evidence should update the goal or change the next action.

3 EgoExo-VLN Benchmark

3.1 Task formulation

As illustrated in Fig. 2, EgoExo-VLN requires an agent to follow a natural-language instruction in a continuous indoor environment and stop near the target object's final valid location. Unlike ego-only VLN, it provides observations from M external cameras with different viewpoints and spatial

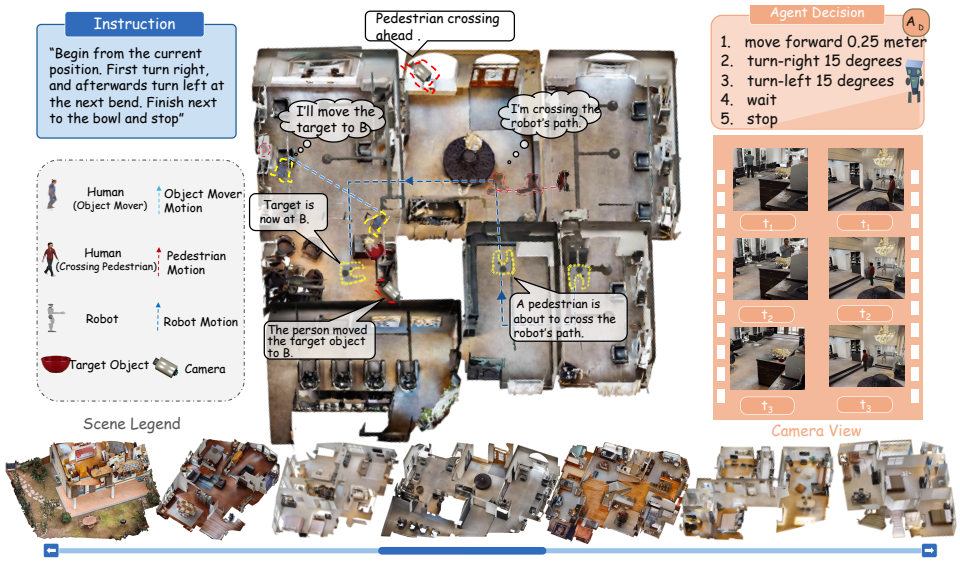


Figure 2: Illustration of an EgoExo-VLN episode, showing the language instruction, target relocation, pedestrian motion, robot trajectory, and corresponding fixed-camera observations. The fixed cameras reveal target movements and pedestrian activity outside the robot's current field of view, providing information for navigation and safety decisions.

coverage, fixed throughout each episode. The instruction specifies the route and target, but not which camera provides relevant information.

At decision step t, the agent receives an egocentric observation o_t^{ego} and external-camera observations:

$$O_t^{\text{cam}} = \left\{ o_{t,1}^{\text{cam}}, \ldots, o_{t,M}^{\text{cam}} \right\}. \tag{1}$$

Given instruction X and observation–action history H_t available before action selection, the policy predicts

$$a_t = \pi_\theta \left(X,\, o_t^{\text{ego}},\, O_t^{\text{cam}},\, H_t \right). \tag{2}$$

External observations extend the agent's view but may be irrelevant or contain outdated evidence about targets or pedestrians. The task therefore requires cross-view grounding to relate external observations to the robot's state, temporal reasoning to assess their continued validity, and action-relevance reasoning to determine whether they should change the next action.

The agent uses a VLN-CE-style continuous navigation interface (Krantz et al., 2020), with standard movement and turning controls, `STOP`, and `WAIT`. `STOP` terminates the episode; `WAIT` preserves the robot's pose while the environment evolves and observations update. The agent may wait when evidence indicates that proceeding is unsafe, while irrelevant evidence need not change its action. The base interface provides egocentric and fixed-camera observations, and the benchmark also supports depth. The experimental setup specifies the observation modalities and additional geometric inputs used by each evaluated configuration.

3.2 Evaluation Metrics

We report success rate (SR), oracle success rate (OSR), success weighted by path length (SPL), navigation error (NE), and normalized dynamic time warping (nDTW) (Anderson et al., 2018a; Ilharco et al., 2019). SR requires `STOP` within 3 m geodesic distance of the final target; OSR measures entry into this region at any time, regardless of stopping. SPL combines success and path efficiency, NE measures final geodesic distance to the target, and nDTW measures executed–reference trajectory similarity. All goal-based metrics use the final location B in relocation episodes.

Let $\mathcal{R}$ denote all target-relocation episodes, with or without pedestrians, and $\mathcal{P}$ all pedestrian-present episodes, including `conflict` and `non-conflict` conditions. For episode i, S_i indicates navigation success, and C_i indicates no robot–pedestrian collision over the complete executed trajectory;

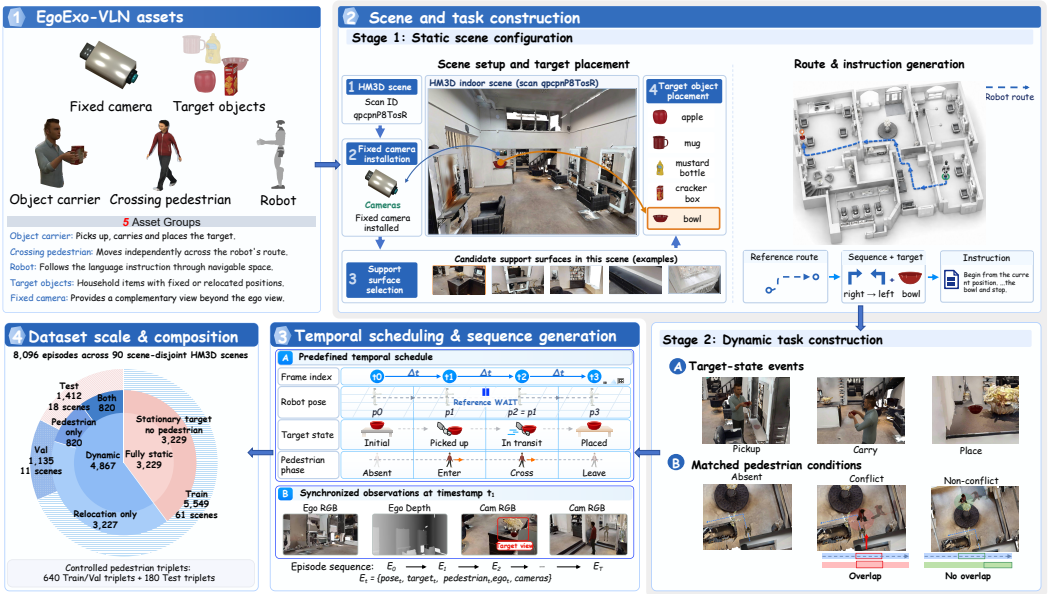


Figure 3: Construction pipeline of EgoExo-VLN. We first prepare the robot, human, camera, and target assets, then configure static scenes with target placements, camera poses, navigation routes, and instructions. Dynamic tasks are created by adding object relocation and matched pedestrian conditions. These events are scheduled on a shared timeline to generate synchronized episode sequences, which form the scene-disjoint training, validation, and test splits.

both are binary. Relocation success rate (Reloc. SR) and safe success rate (SafeSR) are

$$\text{Reloc. SR} = \frac{100}{|\mathcal{R}|}\sum_{i\in\mathcal{R}} S_i, \qquad \text{SafeSR} = \frac{100}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} S_i C_i. \tag{3}$$

Collision-free success weighted by path length (CF-SPL) also accounts for path efficiency:

$$\text{CF-SPL} = \frac{100}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} S_i C_i \frac{\ell_i}{\max(\ell_i, p_i)}, \tag{4}$$

where ℓ_i is the shortest-path distance used in SPL evaluation and p_i the executed path length. Unsuccessful episodes or those with any robot–pedestrian collision contribute zero to SafeSR and CF-SPL. These three metrics use a 0–100 scale, with higher values better. Waiting is analyzed separately under `absent`, `conflict`, and `non-conflict` conditions; `WAIT` alone does not constitute success.

3.3 Dataset Construction

Figure 3 summarizes the assets, scene and task construction, temporal scheduling, and dataset composition.

EgoExo-VLN assets. We build the benchmark in Habitat (Savva et al., 2019) using HM3D indoor scenes (Ramakrishnan et al., 2021). The assets comprise a robot, human object carriers, crossing pedestrians, fixed cameras, and five target categories: apple, mug, mustard bottle, cracker box, and bowl (Fig. 3(1)). Carriers transfer objects, while crossing pedestrians follow independently scheduled motion.

Scene and task construction. Construction comprises static configuration and dynamic task generation (Fig. 3(2)). We place targets on native support surfaces, such as tables and cabinets, identify nearby navigable stopping positions, and sample reference routes through navigable space. Instructions specify route turns and the target category. Each episode has three textual variants of the same instruction, counted as one episode. Fixed cameras provide complementary coverage and 224×224 RGB observations; their poses are shared across episodes within each scene.

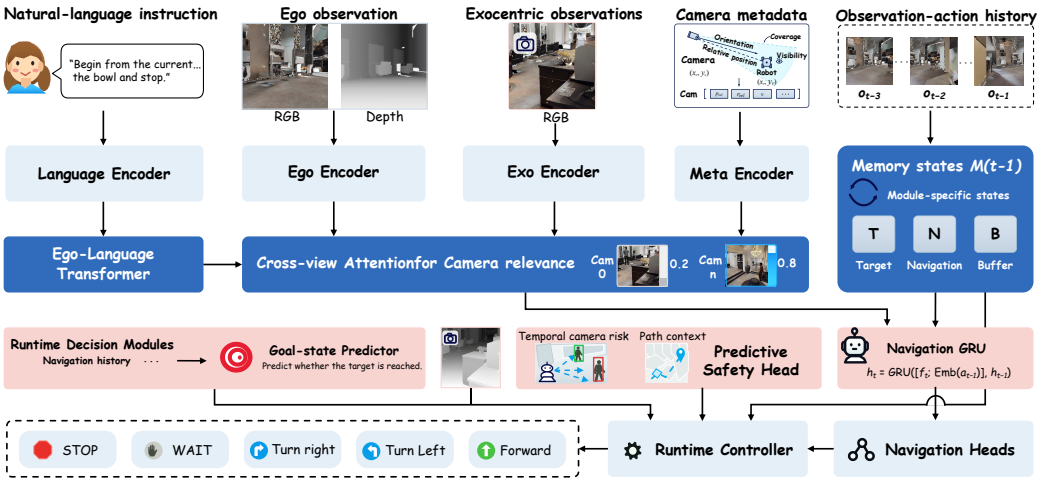


Figure 4: Overview of EgoExo-VLN. Instruction, egocentric RGB-D, fixed-camera RGB, and observation–action history provide the inputs for cross-view grounding, temporal evidence updates, navigation, and predictive safety control.

Dynamic tasks add target-state events and controlled pedestrian motion. For relocation, a human carrier continuously picks up, carries, and places the target from A to B; stationary-target variants preserve its original placement. Pedestrian tasks form matched `absent`, `conflict`, and `non-conflict` triplets with identical scenes, robot initializations, reference routes, target configurations, instructions, and camera poses. Pedestrians are omitted in `absent`; otherwise, their motion is timed to create or avoid spatiotemporal conflict with the reference robot trajectory.

Temporal scheduling and sequence generation. Target-transfer stages and pedestrian motion share a simulation timeline (Fig. 3(3)). Episode records contain scene configurations, instructions, dynamic-event schedules, environment states, and reference actions. During training and evaluation, the simulator instantiates these records and renders observations on demand, synchronizing robot pose, target and pedestrian states, and visual observations at each sampled timestamp. Configurations and event timings can thus be revised without regenerating a fixed video collection.

Dataset scale and composition. EgoExo-VLN contains 8,096 episodes across 90 scenes (Fig. 3(4)). The scene-disjoint training, validation, and test splits contain 5,549, 1,135, and 1,412 episodes from 61, 11, and 18 scenes, respectively. Four mutually exclusive components comprise 3,229 stationary-target and 3,227 relocation episodes without crossing pedestrians, and 820 episodes of each target condition with crossing pedestrians, yielding 3,229 fully static and 4,867 dynamic episodes. The splits respectively include 480, 160, and 180 matched pedestrian triplets, already counted in the episode totals.

4 The EgoExo-VLN Model

EgoExo-VLN (Fig. 4) receives the instruction, egocentric RGB-D, fixed-camera RGB and metadata, and observation–action history at step t. Frozen RoBERTa encodes the instruction (Liu et al., 2019), CLIP ViT-B/16 encodes both RGB views (Radford et al., 2021), and DD-PPO encodes egocentric depth (Wijmans et al., 2020). An ego–language Transformer combines instruction and egocentric features to represent the current navigation state.

Instruction-conditioned spatial attention pools each available camera's RGB features into a token. Cross-view attention identifies evidence relevant to the route and target; a relevance gate controls its contribution to the ego–language representation, and a null token permits ignoring unhelpful external views. Camera-specific temporal memory updates this evidence as observations arrive, replacing stale target locations after relocation.

A navigation GRU combines the fused representation, temporal memory, and previous action to predict action, waypoint, progress, and remaining distance through its output heads. In parallel, a predictive safety head estimates pedestrian conflict from temporal camera evidence and path context,

Table 1: EgoExo-VLN results across validation and test splits.

	Val					Test				
Method	NE↓	SR↑	OSR↑	SPL↑	nDTW↑	NE↓	SR↑	OSR↑	SPL↑	nDTW↑
HA-VLN-CMA	7.277	3.26	12.46	2.74	41.95	7.141	4.22	12.82	3.68	43.89
HPN	8.272	6.85	8.78	6.36	45.41	7.939	8.36	9.14	8.09	46.08
Recurrent	8.161	10.06	27.47	7.35	18.20	7.521	10.06	31.37	8.42	21.69
CMA	7.249	21.32	29.13	20.12	55.50	6.194	28.61	37.39	27.41	63.07
StreamVLN	7.718	22.94	29.32	22.07	49.73	6.771	25.57	32.72	24.41	55.76
ETPNav	5.282	45.18	54.82	38.71	61.96	4.843	42.42	56.66	36.81	66.94
BEVBert	5.636	41.94	54.82	36.20	60.30	4.497	48.51	61.47	41.64	67.53
EgoExo-VLN (Ours)	**3.511**	**52.86**	**62.57**	**52.06**	**76.01**	**3.309**	**55.31**	**61.90**	**54.48**	**76.71**

Table 2: Relocation and safety results across validation and test splits.

	Val			Test		
Method	Reloc. SR↑	SafeSR↑	CF-SPL↑	Reloc. SR↑	SafeSR↑	CF-SPL↑
HA-VLN-CMA	2.82	3.00	2.55	4.82	4.17	3.58
HPN	5.37	6.38	6.17	8.23	8.17	7.88
Recurrent	11.58	9.19	6.62	11.49	10.00	8.36
CMA	5.93	3.13	2.88	10.21	1.78	1.19
StreamVLN	20.34	21.50	20.69	24.82	24.78	23.64
ETPNav	42.37	43.06	36.93	40.85	42.00	36.47
BEVBert	41.24	40.81	35.40	48.65	48.17	41.31
EgoExo-VLN (Ours)	**56.34**	**51.63**	**50.78**	**56.74**	**53.33**	**50.20**

while a goal-state predictor provides task-completion evidence. For geometric execution, the runtime controller back-projects the attended region using the selected camera's depth and calibration, projects the resulting world point onto the Habitat NavMesh, and passes it to a low-level follower. It combines these cues with model outputs to execute `FORWARD`, `TURN LEFT`, or `TURN RIGHT`, issue `WAIT` for predicted conflicts, and issue `STOP` when task-completion criteria are met.

5 Experiments

5.1 Experimental Setup

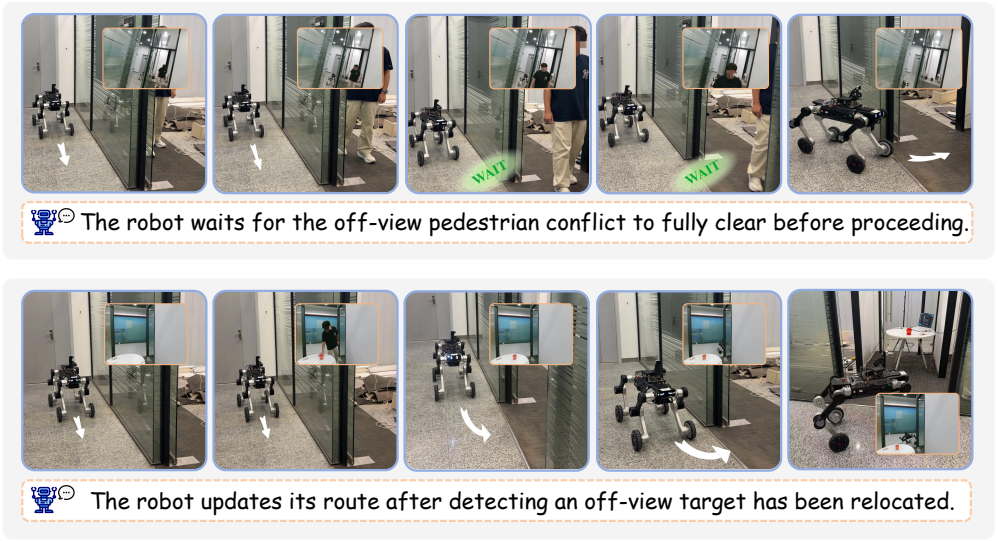
We evaluate EgoExo-VLN against seven VLN baselines on the validation and test splits, following the protocol and metric definitions in Section 3.2. All quantitative results are averaged across multiple random seeds. We also conduct component and depth ablations on the test split, examining exocentric observations, camera relevance, safety control, and egocentric and external-camera depth.

5.2 Comparison with VLN Baselines

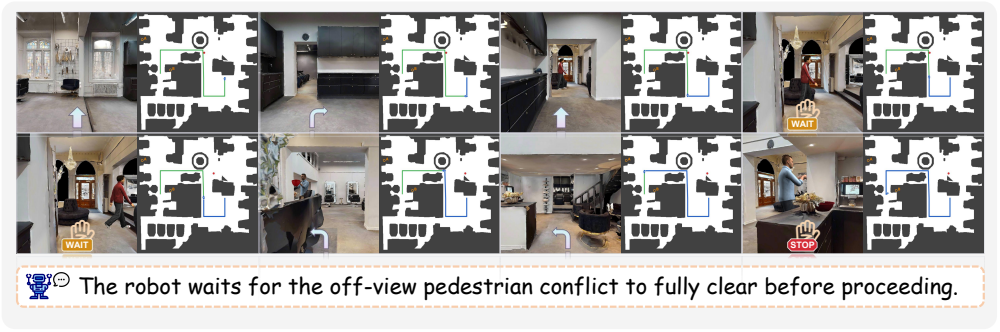
We compare EgoExo-VLN with seven baselines: (1) **HA-VLN-CMA** (Dong et al., 2025) extends CMA with human-aware navigation components for pedestrian interactions. (2) **HPN** (Krantz et al., 2021) predicts intermediate waypoints to guide low-level navigation in continuous environments. (3) **Recurrent** (Krantz et al., 2020) predicts navigation actions from language instructions and egocentric observations using a recurrent policy. (4) **CMA** (Krantz et al., 2020) uses cross-modal attention to integrate language and visual observations. (5) **StreamVLN** (Wei et al., 2025) processes sequential visual observations through a streaming architecture. (6) **ETPNav** (An et al., 2025) combines structured spatial representations with planning for continuous VLN. (7) **BEVBert** (An et al., 2023) integrates spatial observations through a bird's-eye-view representation. We adapt these methods to our benchmark while preserving their core navigation mechanisms.

Table 3: Component analysis on the Test split with temporal updating enabled throughout.

Exo	Relevance	Safety	NE↓	SR↑	SPL↑	Reloc. SR↑	SafeSR↑	CF-SPL↑
–	–	–	6.688	17.15	16.06	16.94	18.33	15.12
✓	–	–	4.328	41.93	41.02	43.12	47.33	46.77
✓	✓	–	3.317	54.60	53.85	55.60	51.28	48.47
✓	✓	✓	**3.309**	**55.31**	**54.48**	**56.74**	**53.33**	**50.20**



(a) Real-world experiments with EgoExo-VLN.



(b) Simulation experiments with EgoExo-VLN.

Figure 5: EgoExo-VLN navigation in real-world and simulated environments.

5.3 Results Analysis

EgoExo-VLN achieves the best reported results across both splits (Tables 1 and 2). On Test, it exceeds BEVBert, the strongest baseline by SR, by 6.80 percentage points in SR and 12.84 points in SPL. The smaller OSR gap of 0.43 points suggests that the gains primarily concern successful completion and path efficiency, rather than merely reaching the goal vicinity. Relocation SR and SafeSR also improve by 8.09 and 5.16 points, respectively, indicating better performance under target relocation and pedestrian interaction.

5.4 Ablation Study

Component analysis. Adding exocentric observations increases SR from 17.15% to 41.93%, and camera relevance further raises it to 54.60% (Table 3). These gains suggest that both external obser-

Table 4: Depth ablation on the Test split.

Ego depth	Exo depth	NE↓	SR↑	SPL↑	Reloc. SR↑	SafeSR↑	CF-SPL↑
–	–	3.890	48.73	47.82	50.21	50.50	47.36
✓	–	3.600	52.27	51.46	53.76	52.28	49.13
✓	✓	3.309	55.31	54.48	56.74	53.33	50.20

Table 5: Condition-specific safety analysis on the Test split.

Method	Conflict WAIT↑	Non-conflict WAIT↓	Absent WAIT↓	SafeSR↑	CF-SPL↑
w/o Safety	0.00	0.00	0.00	51.28	48.47
Presence-only	65.00	75.00	2.56	52.00	50.00
EgoExo-VLN	80.00	6.22	1.22	53.33	50.20

vations and the selection of relevant camera evidence aid navigation. Safety control further increases SafeSR from 51.28% to 53.33%, with a smaller navigation gain.

Depth information. Egocentric depth increases SR from 48.73% to 52.27%, while external-camera depth assistance further raises it to 55.31% (Table 4). The latter also improves relocation SR from 53.76% to 56.74%, consistent with geometric assistance helping the robot navigate to relocated targets.

Waiting behavior. Compared with presence-only control, EgoExo-VLN increases conflict-condition waiting from 65.00% to 80.00% and reduces non-conflict waiting from 75.00% to 6.22% (Table 5). This pattern suggests more selective intervention based on path conflict rather than pedestrian presence alone.

5.5 Real-World and Simulation Experiments

We conduct experiments in simulated and real-world indoor environments. For real-world experiments, we use a Daystar quadruped robot with onboard inference on an NVIDIA Jetson Thor GPU. An Intel RealSense D450 RGB-D camera provides egocentric observations, while two additional D450 cameras provide external RGB observations from fixed locations.

The real-world experiments cover target relocation and pedestrian interaction in one indoor scene. For target relocation, an object is moved from location A to location B during navigation. The robot uses fixed-camera observations to update its route toward the relocated target. For pedestrian interaction, the robot waits for a path conflict outside its current field of view to clear before proceeding. When `WAIT` is predicted, the robot commands zero velocity and remains stationary.

Figure 5a shows these real-world examples, while Figure 5b presents a simulated navigation sequence with pedestrian interaction. The simulation example shows the robot moving through the scene, waiting during a path conflict, and resuming navigation before stopping at the destination. Together, these examples illustrate how EgoExo-VLN uses external observations to respond to dynamic events during navigation.

6 Conclusion

We introduce EgoExo-VLN, a continuous navigation task that uses fixed-camera observations to reason about events beyond the robot's view. We construct a benchmark of 8,096 episodes across 90 scenes, covering stationary targets, target relocation, and controlled pedestrian conditions. We further develop a model that combines camera relevance, temporal evidence updates, and predictive safety control, outperforming the evaluated baselines in navigation and safe task completion. Together, the task, benchmark, and model support the study of dynamic navigation with external observations. Future work will address less structured instructions and missing or delayed camera observations.

AI USE STATEMENT

Generative AI tools assisted with literature search, reference formatting, manuscript drafting and editing, source-code inspection, interpretation of experiment records, and figure-generation code. AI-generated suggestions were not treated as experimental evidence. The authors are responsible for verifying the technical claims, citations, implementation descriptions, and reported results. Generative AI is not credited as an author, and the authors take full responsibility for the final manuscript and associated research artifacts.

ETHICS STATEMENT

EgoExo-VLN studies navigation under target relocation and pedestrian interaction. Quantitative benchmark evaluation is conducted in simulation, supplemented by real-world indoor robot experiments. External cameras introduce privacy considerations because they can capture people beyond the robot's immediate surroundings. Real-world use therefore requires appropriate consent and data-handling practices, restricted access to camera observations, and human oversight. Predicted safety decisions do not guarantee collision avoidance. The underlying scene datasets, simulation assets, and pretrained models remain subject to their respective licenses and terms of use.

REPRODUCIBILITY STATEMENT

Section 3.3 describes scene and task construction, event scheduling, and dataset composition. Section 4 presents the model and execution pipeline. Section 5 reports baseline comparisons, ablations, and real-world and simulation experiments. Appendix A provides architecture, training, and controller settings. Section 3.2 and Appendix define the metrics and evaluation populations, while Appendix C specifies the controlled pedestrian conditions and safety-triggered waiting analysis. Quantitative results are averaged across multiple random seeds.

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.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Košecká, 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.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2e5c2cb8d13e8fba78d95211440ba326-Abstract.html.

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.

Yifei Dong, Fengyi Wu, Qi He, Heng Li, Minghan Li, Zebang Cheng, Yuxuan Zhou, Jingdong Sun, Qi Dai, Zhi-Qi Cheng, and Alexander G. Hauptmann. Ha-vln: A benchmark for human-aware navigation in discrete-continuous environments with dynamic multi-human interactions, real-world validation, and an open leaderboard. *arXiv preprint arXiv:2503.14229v1*, 2025. URL https://arxiv.org/abs/2503.14229v1.

Haodong Hong, Yanyuan Qiao, Sen Wang, Jiajun Liu, and Qi Wu. General scene adaptation for vision-and-language navigation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/aebf6284fe85a8f44b4785d41bc8249a-Abstract-Conference.html.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Hong_VLN_BERT_A_Recurrent_Vision-and-Language_BERT_for_Navigation_CVPR_2021_paper.html.

Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2209.12836.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019. URL https://arxiv.org/abs/1907.05446.

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*, pp. 15162–15171, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/papers/Krantz_Waypoint_Models_for_Instruction-Guided_Navigation_in_Continuous_Environments_ICCV_2021_paper.pdf.

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*, 2020. doi: 10.18653/v1/2020.emnlp-main.356. URL https://aclanthology.org/2020.emnlp-main.356/.

Heng Li, Minghan Li, Zhi-Qi Cheng, Yifei Dong, Yuxuan Zhou, Jun-Yan He, Qi Dai, Teruko Mitamura, and Alexander G. Hauptmann. Human-aware vision-and-language navigation: Bridging simulation to reality with dynamic human interactions. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d8087deaf34bb07ddd41c65f8a9fe9b5-Abstract-Datasets_and_Benchmarks_Track.html.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M. Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied AI. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/34173cb38f07f89ddbebc2ac9128303f-Abstract-round2.html.

Abhijeet Ravankar, Ankit Ravankar, Yukinori Kobayashi, and Takanori Emaru. Intelligent robot guidance in fixed external camera network for navigation in crowded and narrow passages. *Proceedings*, 1(2):37, 2017. doi: 10.3390/ecsa-3-D008.

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. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Savva_Habitat_A_Platform_for_Embodied_AI_Research_ICCV_2019_paper.html.

Jae Hong Shim and Young Im Cho. A mobile robot localization via indoor fixed remote surveillance cameras. *Sensors*, 16(2):195, 2016. doi: 10.3390/s16020195.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 394–406, 2020. URL https://proceedings.mlr.press/v100/thomason20a.html.

Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *European Conference on Computer Vision*, pp. 605–621, 2020. doi: 10.1007/978-3-030-58536-5_36. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123470596.pdf.

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

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect PointGoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1gX8C4YPr.

Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European Conference on Computer Vision*, pp. 107–124, 2022. doi: 10.1007/978-3-031-19842-7_7. URL https://arxiv.org/abs/2203.10638.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, Yuxin Fan, Wenjun Li, Zhibo Chen, Fei Gao, Qi Wu, Zhizheng Zhang, and Wang He. Embodied navigation foundation model. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ceb45c48e29e7f49b0b47edb98e43691-Abstract-Conference.html.

Xinxin Zhao, Wenzhe Cai, Likun Tang, and Teng Wang. Imaginenav: Prompting vision-language models as embodied navigator through scene imagination. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/eb261df4322a8bd0a73093c4d8a0d02d-Abstract-Conference.html.

Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. doi: 10.1109/CVPR46437.2021.01250. URL https://openaccess.thecvf.com/content/CVPR2021/html/Zhu_SOON_Scenario_Oriented_Object_Navigation_With_Graph-Based_Exploration_CVPR_2021_paper.html.

A Implementation Details

A.1 Camera Metadata

Each valid external camera is represented by a 19-dimensional vector. Table 6 lists its entries in implementation order. Let $\mathbf{p}_t$ and R_t denote the current robot position and orientation, and $\mathbf{c}_j$ and $\mathbf{f}_j$ the calibrated camera position and unit forward direction in world coordinates. We define

$$\mathbf{u}_{t,j} = \mathrm{clip}\left(R_t^\top(\mathbf{c}_j - \mathbf{p}_t)/d_0, -1, 1\right), \quad \mathbf{v}_{t,j} = R_t^\top \mathbf{f}_j, \quad d_{t,j} = \|\mathbf{c}_j - \mathbf{p}_t\|_2,$$

where $d_0 = 12\,\mathrm{m}$. Positions and directions use the robot's coordinate frame; y is vertical and forward is $-z$. The two bearing entries use $h = \max(10^{-6}, \sqrt{u_x^2 + u_z^2})$. For camera half field of view α_j, the field-of-view score is

$$q_{t,j} = \mathrm{clip}\left(\frac{\mathbf{f}_j^\top(\mathbf{p}_t - \mathbf{c}_j)/d_{t,j} - \cos\alpha_j}{1 - \cos\alpha_j}, 0, 1\right).$$

The view cosine is set to one when $d_{t,j} \le 10^{-6}$.

Table 6: Camera metadata entries, using one-based indices.

Index	Definition and source
1, 2, 3	u_x, u_y, u_z: normalized relative camera position, from calibration and current robot pose.
4, 5, 6	v_x, v_y, v_z: camera viewing direction in the robot frame.
7	$\min(1, d_{t,j}/d_0)$: normalized camera–robot distance.
8, 9	$u_x/h, -u_z/h$: horizontal bearing components.
10, 11	Elapsed and remaining budget fractions: $t/350$ and $(350 - t)/350$.
12	Constant zero, retained for compatibility.
13	Camera-frustum membership of the current robot position, computed from calibration; this is not an occlusion or target-visibility label.
14	Field-of-view score $q_{t,j}$.
15	Distance attenuation $\exp(-d_{t,j}/d_0)$.
16	Constant zero, retained for compatibility.
17, 18	One-hot identity of camera 0 and camera 1.
19	Constant zero; route-dependent camera coverage roles are not supplied.

Calibration and camera identity are fixed before an episode. Relative geometry is recomputed from the current robot pose in the calibrated world frame. The metadata does not include ground-truth target visibility, target coordinates, relocation events, or reference-route coverage labels. Entry 13 tests frustum membership only; it does not test occlusion. An unavailable camera has an all-zero vector and is masked separately.

Clock and action budget. Every episode has a fixed budget of $B = 350$ action decisions. At decision $t \in \{0, \ldots, B-1\}$, entries 10 and 11 are t/B and $(B-t)/B$. Forward motion, turns, WAIT, STOP, and guard-blocked navigation attempts each consume one decision. WAIT and blocked attempts advance the environment clock without moving the robot. STOP terminates the episode without an additional environment update. Neither the action budget nor the clock uses reference-trajectory length. No extra budget is granted for waiting.

The general camera encoder uses the full 19-dimensional vector, including the reserved zero entries. The dedicated camera-risk and target-evidence geometry projections use the 12 entries indexed by 1–9 and 13–15. These projections exclude the clock, camera identity, and reserved coverage-role entry.

A.2 Temporal Evidence and Path Context

For camera j, instruction-conditioned spatial attention produces target evidence, which is combined with language features, the robot state, relative camera geometry, and the normalized predicted waypoint direction to obtain $\mathbf{e}_{t,j}$. Camera-specific temporal updating applies

$$\mathbf{h}_{t,j} = \mathrm{GRU}_{\mathrm{cam}}(\mathbf{e}_{t,j}, \mathbf{h}_{t-1,j}), \qquad \widetilde{\mathbf{e}}_{t,j} = \phi([\mathbf{e}_{t,j}; \mathbf{h}_{t,j}]).$$

The GRU shares parameters across cameras while maintaining a separate state for each camera. Updated evidence feeds camera relevance, target-direction and distance prediction, and the gated waypoint correction. Disabling camera-specific updating bypasses this GRU and temporal fusion, leaving the current-step evidence in their place. This switch is distinct from the explicit instruction-phase memory and from the history-based arrival predictor.

The safety branch obtains path context from the model's predicted local waypoint, rather than from the reference route. Specifically, it normalizes the two-dimensional waypoint available before the infrastructure correction. This direction is combined with the robot–language state, pooled camera risk features, temporal camera risk, camera relevance, and the 12 geometry entries described above. The predictive branch outputs pedestrian presence, path conflict, normalized temporal risk, and release probability. The temporal-risk output is a learned normalized quantity, not a physical time in seconds. In the controller score defined below, a smaller value of this output increases the intervention score when predicted presence and conflict are held fixed.

A.3 Arrival-Predictor Inputs

The arrival predictor receives the 16 features in Table 7. Its history index i refers to the retained controller history. Let s_i, R_i, and p_i denote predicted STOP probability, remaining distance in meters, and progress. Define $r_i = \min(20, R_i)$, $i_k = \max(0, i-k)$, $\mathcal{W}_i = \{\max(0, i-5), \ldots, i\}$, and $\Delta(x) = \mathrm{clip}(x, -10, 10)/10$. Let d_i be the camera-localized target's horizontal distance from the robot, capped at 20 m; failed localization is encoded as 20 m.

Table 7: Arrival-predictor inputs in implementation order.

Index	Feature
1	Current STOP probability s_i.
2	Predicted remaining distance $r_i/10$.
3	Predicted progress p_i.
4	Three-step distance decrease $\Delta(R_{i_3} - r_i)$.
5	Eight-step distance decrease $\Delta(R_{i_8} - r_i)$.
6	Recent minimum remaining distance $\min_{k \in \mathcal{W}_i} R_k/10$.
7	Progress change $\mathrm{clip}(p_i - p_{i_3}, -1, 1)$.
8	Recent maximum STOP probability $\max_{k \in \mathcal{W}_i} s_k$.
9	Current camera-map localization-validity flag.
10	Camera-map target distance $d_i/10$.
11	Three-step map-distance decrease $\Delta(d_{i_3} - d_i)$.
12	Recent minimum map distance $\min_{k \in \mathcal{W}_i} d_k/10$.
13	Relevance score of the selected camera.
14	Attention confidence of the selected image region.
15	Predicted camera-goal distance, capped at 20 m and divided by 10.
16	Camera-goal gate value.

The predictor uses the online features listed in Table 7 and their history. Features 1–8 and 15–16 are current model predictions or summaries of past predictions. Features 9–12 describe online camera-map localization and its history. Features 13–14 are the selected camera's relevance score and the selected image region's attention confidence, respectively. No ground-truth target coordinate, task-subtype label, or relocation indicator is supplied as an input. Features are standardized using the mean and standard deviation stored with the arrival checkpoint. The predictor is a multilayer perceptron with hidden widths 48 and 24; its temporal information comes from the history features rather than an internal recurrent state.

A.4 Runtime Arbitration

The full model retains the navigation GRU, camera-specific target memory, camera-risk history, infrastructure route memory, geometric execution, and the 16-input arrival predictor. Explicit instruction-phase memory and learned recovery options are disabled. Ordinary collision recovery remains available. The controller uses the following priority:

1. Synchronize dynamic actors and record the robot and pedestrian states. Compute camera evidence, navigation outputs, stopping confidence, and safety eligibility using only information available at this step.
2. If either the base stopping rule or the arrival predictor confirms completion, select `STOP`.
3. Otherwise, select `WAIT` when the safety branch requests waiting and its duration and total-wait limits permit it.
4. Otherwise, use a valid camera-map follower proposal. If none is available, execute an outstanding collision-recovery turn, initiate recovery after a collision, or follow the predicted waypoint, in that order.
5. Apply the egocentric-depth guard to forward proposals. Apply the pedestrian-motion guard to the resulting navigation proposal; a blocked proposal consumes a step while leaving the pose unchanged.
6. Record the chosen branch, proposed and executed actions, intervention reason, and before/after robot and pedestrian states. Perform the endpoint overlap check before testing termination or resetting the environment.

STOP therefore has priority over safety-triggered waiting, and neither WAIT nor geometric execution can override a confirmed termination. Recovery does not override STOP or an eligible safety WAIT.

The base stopping rule uses a STOP score of 0.03, predicted remaining distance at most 3.5 m, and five confirmation steps. The arrival predictor uses a threshold of 0.52 and two confirmation steps, with exit threshold 0.37. Waypoint execution uses a 12° heading threshold and smoothing factor 0.6. A forward proposal is replaced by a turn when the central egocentric-depth statistic is below 0.45 m.

The predictive safety score is $p_{\text{human}} p_{\text{conflict}} (1 - \widehat{u})$, where p_{human}, p_{conflict}, and $\widehat{u}$ are the predicted pedestrian-presence probability, path-conflict probability, and normalized temporal-risk output, respectively. Entry uses threshold 0.22. Waiting requests a minimum of four steps, unless STOP or the episode budget terminates execution; each event is capped at ten steps and the episode at 24 safety-triggered WAIT steps. After the minimum, waiting can continue while risk is at least 0.11 and release is below 0.50, or while the entry condition remains active. Rearming requires release above 0.55 and two absence steps. Camera-risk history continues to update during WAIT; navigation and arrival histories are held fixed. All waiting steps count toward the common 350-step budget.

Geometric execution inputs. The camera-guided controller uses external depth, camera calibration, the current robot pose, and the Habitat NavMesh to convert an attended image region into a navigable point for the low-level follower. External depth is used by this execution branch rather than the learned RGB fusion encoder. These quantities are system inputs; the resulting point is inferred from camera evidence and is not a supplied ground-truth target coordinate. The predictive pedestrian guard and the learned safety-triggered WAIT branch are separate mechanisms: the former can block a proposed motion, while the latter selects a waiting action.

A.5 Training and Statistical Reporting

Episode and record accounting. All training stages use the training split only. A unique episode is identified by its manifest ID. Camera training uses the target-task subset, and safety training uses structured pedestrian records together with ordinary negative records sampled at a 1:1 ratio. Ordinary negatives are sampled without replacement within a stage and may reuse target-task episodes from camera training. Arrival-training trajectories are collected from training episodes with the frozen navigation controller. These are overlapping training uses, not additional dataset episodes. Each stage stores its selected episode IDs and parent manifest hash; counts are derived from these files rather than added across stages.

Camera training. The inherited visual encoders and navigation backbone are frozen while the external-camera modules are trained with AdamW, learning rate 3×10^{-5}, weight decay 10^{-4}, batches of four 32-step sequences, and four-step gradient accumulation. Training uses 24 epochs,

cosine decay, 5% warmup, and gradient clipping at norm one. Epoch 22 is fixed before evaluation for every corresponding configuration.

$$\begin{aligned}\mathcal{L}_{\text{cam}} = 2\mathcal{L}_{\text{rel}} + 1.5\mathcal{L}_{\text{dir}} + 0.25\mathcal{L}_{\text{dist}} + 2\mathcal{L}_{\text{res}} + 2\mathcal{L}_{\text{wp}} \\ + 5\mathcal{L}_{\text{protect}} + \mathcal{L}_{\text{gate}} + \mathcal{L}_{\text{null}} + 0.2\mathcal{L}_{\text{temp}} + 1.5\mathcal{L}_{\text{ROI}}.\end{aligned} \tag{5}$$

The terms supervise relevance, direction, distance, bounded correction, corrected waypoint direction, preservation of the inherited prediction, gating, null preference, temporal consistency, and target-region attention. For stationary targets, supervision uses the fixed target location. When a target is carried by a person, supervision follows its current location throughout transport, allowing the agent to continue following the moving target. After placement, supervision uses the new target location. The target remains valid during transport; transport alone does not trigger a null label or mask direction and region supervision. Camera-selection labels are not inputs at inference.

Safety training. Only the predictive safety head is updated in this stage. Training uses four epochs, batches of four 64-step sequences, AdamW learning rate 2×10^{-4}, and weight decay 10^{-4}:

$$\mathcal{L}_{\text{safety}} = \mathcal{L}_{\text{human}} + 1.5\mathcal{L}_{\text{path}} + 0.5\mathcal{L}_{\text{risk}} + 0.75\mathcal{L}_{\text{release}} + 2\mathcal{L}_{\text{clear}}. \tag{6}$$

Presence, path conflict, and release use binary cross-entropy; normalized temporal risk uses Smooth-L1 regression. The clear-state term penalizes risk predictions on negative examples. Training annotations define the presence interval and conflict interval; release is positive for the first four annotated steps after conflict ends. No pedestrian-presence or conflict label is supplied to the controller at inference.

Arrival training. The arrival predictor is trained using the 16 inputs listed in Table 7. Training trajectories use the same budget, controller, and reset rules as evaluation. States within 2.5 m of g_B are positive, states beyond 3 m are negative, and the intermediate band is omitted. The loss combines weighted binary cross-entropy with a false-positive penalty of weight 0.35. Hard negatives include misleading STOP predictions and camera-localized targets outside the success region. AdamW uses learning rate 10^{-3}, weight decay 10^{-4}, and gradient clipping at norm one. Normalization statistics are fitted on training data only. Ground-truth goal positions generate training labels but are not inputs.

Repetitions and aggregation. Each reported configuration is trained and evaluated with five random seeds. Within each run, episode-level metrics are averaged over the designated evaluation subset. The same frozen episode IDs and metric denominators are used across the five runs of a configuration. For per-run metric values $m_1, \ldots, m_5$, the reported result is

$$\bar{m} = \frac{1}{5}\sum_{k=1}^{5} m_k.$$

Metrics are aggregated before rounding; rounding is applied only to the displayed values. WAIT rates and other episode-level binary rates use episode counts, rather than action-step counts or scene-wise macro averages. Ablation configurations use identical data partitions, training schedules, input access except for the ablated component, and evaluation rules. Matched runs start from the same inherited checkpoint and matching initializations for shared trainable modules. All optimizers are fresh. Test outcomes do not determine checkpoints, thresholds, or configurations.

B Evaluation Protocol

Evaluation populations. The benchmark contains 8,096 episodes across 90 scene-disjoint HM3D scenes. The training, validation, and test splits contain 5,549, 1,135, and 1,412 episodes from 61, 11, and 18 scenes, respectively. The training and validation splits together contain 640 matched pedestrian triplets, and the test split contains 180 triplets. These triplets are included in the split totals rather than added as extra episodes. Let $\mathcal{R}$ denote all target-relocation episodes, with or without crossing pedestrians, and $\mathcal{P}$ all pedestrian-present episodes, including both conflict and non-conflict conditions. On Test, $|\mathcal{P}| = 360$, and each condition-specific WAIT population contains 180 episodes. Relocation and pedestrian subsets may overlap; their sizes are not added to the total split size. Navigation metrics use the full split, Reloc. SR uses $\mathcal{R}$, and SafeSR and CF-SPL use $\mathcal{P}$.

Episode independence. Each episode starts with reset recurrent states, controller histories, and geometric caches. Dynamic actors from the preceding episode are removed before new actors are installed; reset checks verify that no previous actor remains. A clean single-episode replay and a replay after an episode prefix must agree on initial observations under deterministic rendering. Data identifiers, selected checkpoints, and configuration hashes are recorded for each run.

Dynamic targets. All goal-based metrics use the stored navigable goal g_B associated with the target's final location B. The goal is fixed before execution and is not the object's elevated centroid. A STOP before or during transport is still evaluated against g_B; no extra transport-completion condition is imposed. Success requires an issued STOP within 3 m geodesic distance. Exhausting the budget without STOP is unsuccessful.

The SPL reference distance is the shortest navigable distance from the initial robot position to g_B. Navigation error uses the final executed endpoint. The nDTW reference is the frozen episode reference trajectory to g_B. Consecutive duplicate positions are removed from both paths using a 10^{-4} m tolerance, and geodesic DTW is normalized by the remaining reference length and the 3 m success threshold. Reference paths and geodesic queries are validated before admission; unavailable distances are flagged as evaluation errors and are not silently replaced by another distance definition. WAIT adds elapsed steps but no geometric path length.

Collision criterion and recording. For the collision-free metrics defined in Section 3.2, a robot–pedestrian collision is operationally defined as geometric overlap at a recorded executed state. The test uses crossing pedestrians and is distinct from predicted interceptions and physics-engine mesh contact. For the horizontal robot position $\mathbf{p}$ and active pedestrian position $\mathbf{h}_j$, clearance is

$$\delta = \min_j \big(\|\mathbf{p}^{xz} - \mathbf{h}_j^{xz}\|_2 - r_R - r_j \big). \tag{7}$$

The robot proxy radius is fixed at $r_R = 0.35$ m. Pedestrian radii r_j come from the frozen actor configuration and are recorded with every trajectory. Missing or invalid radii cause an audit failure; they are not imputed as zero. An executed-pose collision is $\delta \leq 0$. The predictive guard uses a separate 0.05 m margin; this margin is not added to the collision criterion used for SafeSR or CF-SPL.

The trace records the initial state, states after actor synchronization, and the robot endpoint after every motion, turn, WAIT, or blocked attempt. The final endpoint is checked before STOP, budget termination, or reset, without advancing the simulation beyond termination. Intermediate executed states are also checked when exposed by the simulator. Pedestrian keyframes are evaluated according to the simulator's actual update rule; no artificial continuous sweep is inferred between discontinuous actor placements. These metrics describe overlap at recorded executed states and do not claim continuous mesh-contact detection between samples.

Safety aggregation. Let $\mathcal{P}$ contain all frozen pedestrian-present episodes, including both conflict and non-conflict conditions. Define $C_i = 1$ exactly when no executed-pose collision is recorded in episode i, and zero otherwise. Then

$$\text{SafeSR} = \frac{100}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} S_i C_i, \qquad \text{CF-SPL} = \frac{100}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} S_i C_i \frac{\ell_i}{\max(\ell_i, p_i)}. \tag{8}$$

SafeSR is collision-free success rate under this geometric criterion. Failed episodes and successful episodes with any recorded overlap contribute zero.

The environment guard records blocked proposals as interception attempts. Consecutive blocked navigation attempts form one interception event; an unblocked navigation attempt ends the event, while WAIT leaves its state unchanged. Attempts, events, and executed-pose collisions are separate fields. Blocked-attempt counters are not used as collision indicators in SafeSR or CF-SPL.

C Condition-Specific Waiting Analysis

Matched absent, conflict, and non-conflict episodes share the scene, robot initialization, reference route, target configuration, instruction, and camera poses. Their labels describe scheduled conditions

relative to the reference robot trajectory, not the policy's eventual collision outcome. Waiting rates are reported separately for each condition.

This analysis compares safety-control variants using the waiting behavior of the complete controller. The WAIT columns in the main results count episodes containing at least one executed `WAIT`, regardless of the controller branch that produces it:

$$\mathrm{WaitRate}(\mathcal{C}) = \frac{100}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \mathbb{1}[\text{at least one executed } \texttt{WAIT} \text{ in episode } i]. \tag{9}$$

Each episode contributes at most once. A blocked navigation proposal is not counted as `WAIT` solely because the robot remains stationary. All executed actions still count toward the same 350-step budget.

Without Safety disables the predictive safety-control branch. Its WAIT rate is computed from the executed action sequence using the same definition as the other configurations; the reported zero indicates that no `WAIT` was executed in those runs. Presence-only uses the same model's predicted pedestrian presence, rather than ground-truth labels, as its entry signal. It uses the same entry threshold and waiting-duration limits, and releases after the predicted presence falls below the fixed entry threshold, subject to the minimum waiting duration. The full model additionally uses predicted path conflict and temporal risk. Other policy, stopping, execution, and guard settings remain unchanged. False presence detections can produce nonzero waiting under absent conditions. Thresholds are fixed before Test.

D Temporal Evidence Ablation

The temporal comparison keeps the navigation GRU, camera-risk history, infrastructure route memory, arrival-history inputs, safety control, and geometric execution enabled. Explicit phase memory is disabled in every configuration. Each configuration is trained and evaluated with its corresponding observation and memory setting.

Table 8: Settings for the temporal evidence ablation. Target memory means the camera-specific target-evidence GRU and its temporal fusion.

Configuration	External image evidence	Target memory
Frozen Exo	Initial synchronized RGB and depth pair	Enabled
Current-only	Current synchronized RGB and depth pair	Disabled
Full Temporal	Current synchronized RGB and depth pair	Enabled

Frozen Exo holds the initial RGB and its matching depth fixed throughout the episode. Image-derived detections and geometric localization use only this pair; no later external image, depth, or target-location cache is admitted. Fixed camera calibration remains available. Robot-relative camera geometry and the causal budget clock are updated from the current robot state; these are not future external observations. The geometric controller may recompute distance to a point obtained from the frozen pair, but it cannot refresh that point from later frames.

Current-only bypasses the camera-specific target-evidence GRU and its temporal fusion, replacing their output with current-step evidence. It does not remove all system history. The Full Temporal versus Current-only comparison isolates this target-evidence memory under the shared surrounding system. Frozen Exo versus Full Temporal measures the benefit of fresh synchronized external evidence when target memory is retained.

In the component ablations, removing external observations disables all external evidence pathways, including image-based geometric execution. Removing relevance replaces learned camera weights with uniform weights over available cameras while preserving the other camera modules. Removing safety disables only the safety-triggered WAIT branch. Camera-specific target memory remains enabled wherever external evidence is present. This distinguishes component removal from the temporal-memory ablation and makes differences in geometric execution explicit.

Table 9: Temporal ablation of external-camera observations on the test split. NE is reported in meters; other metrics are percentages.

Configuration	NE↓	SR↑	SPL↑	Reloc. SR↑	SafeSR↑	CF-SPL↑
Frozen Exo	4.100	47.00	44.50	43.00	48.00	44.00
Current-only	3.600	52.00	50.50	51.00	51.00	48.00
Full Temporal	**3.309**	**55.31**	**54.48**	**56.74**	**53.33**	**50.20**

Results. As shown in Table 9, Full Temporal improves Reloc. SR from 43.00% to 56.74% over Frozen Exo, supporting the value of refreshing external evidence while retaining target memory. Compared with Current-only, it increases SR by 3.31 percentage points and Reloc. SR by 5.74 points, while raising SafeSR from 51.00% to 53.33%. These results suggest that camera-specific target memory provides useful context beyond the latest external view. This interpretation concerns the target-evidence memory removed by Current-only; the other temporal components listed above remain enabled.