Seeing Together, Acting Apart: Shared Environmental Understanding for Multi-Robot Navigation

Haihong Hao[1†] **Lei Chen**[1†] **Mingfei Han**[1,2,3†‡] **Chenglong Yan**[1] **Dong An**[3]
Yuqiang Yang[4] **Changlin Li**[5] **Minghao Guo**[3] **Salman Khan**[3§] **Xiaojun Chang**[1,2§*]
[1]Future Media Computing Lab, USTC [2]LeapWing AI
[3]MBZUAI [4]Shanghai AI Laboratory [5]Stanford University
[†] Equal contribution. [‡] Project lead. [§] Equal supervision.

Abstract

Multi-robot embodied systems are typically designed to collaborate by coordinating actions, sharing trajectories, or learning joint policies. We ask a complementary question: can collaboration emerge before decision-making, by first building a shared understanding of the environment? We introduce *Shared Environmental Understanding*, a policy-agnostic representation that aggregates partial egocentric observations from multiple robots into a shared latent world model. This shifts collaboration from the policy space to the representation space, enabling decentralized decision-making with shared environmental context. To this end, we propose **S**hared **E**nvironment **E**ncoding and **R**etrieval (SEER), a framework that organizes multi-robot observations into a shared memory, retrieves relevant context via spatio-temporal similarity, and encodes it using a predictive latent world model. The resulting representations are injected into downstream navigation agents as tokens. Experimental results demonstrate that SEER captures action-relevant environmental structure and consistently improves navigation performance across both specialized VLN agents and general vision-language model agents, while preserving single-agent capabilities.

1 Introduction

Multi-robot embodied systems are commonly expected to collaborate by coordinating their behaviors: agents may communicate, share experience, divide tasks, avoid conflicts, or jointly decide what to do next. This question naturally extends classic embodied navigation and vision-and-language navigation settings, where an agent must ground observations and instructions into actions in realistic environments [1, 2]. It has also motivated a large body of work on multi-robot coordination, communication, task allocation, and multi-agent decision making [3–6], as well as recent embodied multi-agent systems that share semantic maps, messages, frontier assignments, action plans, or generated rollouts for cooperative navigation and planning [7–9]. However, most existing approaches frame collaboration at the level of agent-centric traces, such as sharing actions, trajectories, maps, or intermediate decisions. While effective, these methods tightly couple collaboration with policy design and communication protocols. This raises a fundamental question: can collaboration emerge from shared perception before shared decision-making? In other words, instead of exchanging what agents *do*, can they first share what they *understand* about the environment?

In realistic settings, different robots naturally observe complementary parts of the same environment from diverse viewpoints and at different times. Aggregating these partial observations can yield a richer representation of the environment than any single agent can obtain alone. Crucially, such

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

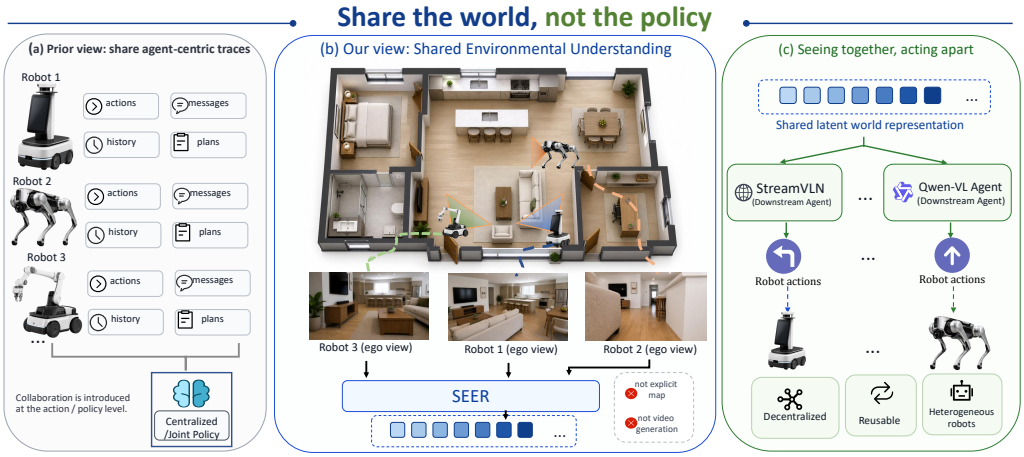


Figure 1: Unlike prior methods, SEER builds a policy-agnostic shared world representation by pooling partial egocentric observations, shift collaboration from policy space to representation space.

a representation can serve as a policy-agnostic interface: it can be shared across agents while allowing each agent to make decisions independently using its own policy. We therefore propose to shift collaboration from policy space to representation space. We introduce *Shared Environmental Understanding*, a policy-agnostic representation that aggregates multi-robot observations into a shared latent description of the environment. This shared representation provides a common environmental context, while allowing each agent to act independently with its own policy.

In this paper, we propose *Shared Environment Encoding and Retrieval* (SEER), a framework that learns shared latent world representations from multi-robot, multi-view observations. SEER organizes distributed observations into a shared memory, retrieves relevant context via spatio-temporal similarity, and encodes it using a predictive latent world model. The resulting representations are exposed as continuous tokens and injected into downstream navigation agents, enabling decentralized decision-making with shared environmental context. We evaluate SEER from both representation and task perspectives. Our results show that the learned representations capture action-relevant environmental structure and consistently improve downstream navigation across both specialized VLN agents and general vision-language model agents.

2 Related Works

Multi-Robot Coordination. Multi-robot coordination covers task allocation, communication, distributed planning, heterogeneous cooperation, distributed optimization, coordinated exploration, mapping, and aerial/multi-UAV swarms [3, 4, 10–13]. Learning-based approaches study inter-agent communication and decision making through differentiable communication and MARL [5, 6, 14–16], while recent embodied systems extend coordination to cooperative visual-semantic navigation, ambiguous instruction resolution, heterogeneous collaboration, LLM/VLM-driven planning, and embodied multi-agent benchmarks [7–9, 17–20]. Related ideas also appear in cooperative driving/perception, city-scale simulation, and multi-view-consistent generation for urban or driving worlds [21–30]. **World Models for Embodied Agents.** Embodied world modeling builds on large-scale datasets, simulators, and 3D language-grounding benchmarks, including Matterport3D, ScanNet, Habitat, HM3D-Sem, EmbodiedScan, EmbodiedQA, ScanRefer, ReferIt3D, ScanQA, SQA3D, OpenEQA, 3D-LLM, and SIG3D [31–42]. World models have evolved from latent dynamics and imagination-based control [43–46] to interactive/generative prediction of future observations, occupancy, and navigation rollouts [47–51]; representation-predictive methods such as V-JEPA and V-JEPA 2 further highlight latent-feature prediction over pixel reconstruction [52, 53]. Recent embodied models also encode explicit 3D/4D structure or navigation geometry and connect perception with language and action [54–61]. **Vision-and-Language Navigation.** VLN grounds natural-language instructions into sequential navigation actions. Early work established standard benchmarks and agents, including speaker-follower training, cross-modal matching, self-monitoring, regretful navigation, and back-translation [1, 62–66]. The field later expanded to multilingual, dialog,

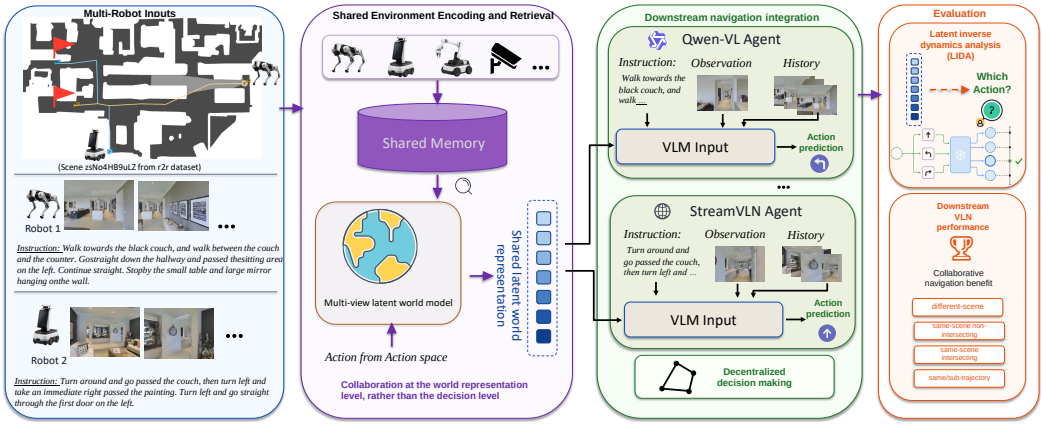


Figure 2: Our SEER organizes distributed observations into a shared memory, retrieves relevant context via spatio-temporal similarity, and encodes it using a predictive latent world model.

object-centric, continuous, and outdoor settings [67–72], and recent agents use large-scale pretraining, transformers, map-centric or history-aware representations, future-view prediction, VLM/LLM reasoning, map-guided planning, causal learning, aerial VLN, long-horizon and streaming settings, physical deployment, semantic-map memory, and world-model-based navigation [2, 73–93]. In contrast to prior coordination works that introduce controllers or communication protocols, world models that primarily focus on single-agent prediction, and VLN methods that mainly improve policies, memory, prompting, or deployment, our work learns a policy-agnostic shared environmental representation from complementary multi-robot observations and uses it as reusable context for downstream navigation agents.

3 Method

We propose a framework that leverages shared environmental understanding to learn policy-agnostic representations. The resulting shared environmental tokens are injected into downstream navigation agents, enabling both representation probing and evaluation on VLN tasks (Figure 2).

3.1 Problem Definition

In conventional multi-robot navigation, consider N robots navigating in the same environment $\mathcal{E}$. At time step t, robot i receives a partial egocentric observation $o_t^i \in \mathcal{O}$, executes an action $a_t^i \in \mathcal{A}$, and robot also be conditioned on a language instruction $x^i \in \mathcal{X}$. The trajectory of robot i is denoted as $\tau^i = \{(o_t^i, a_t^i)\}_{t=1}^{T_i}$, then the multi-robot observation data is collected as $\mathcal{D}_\mathcal{E} = \{\tau^1, \tau^2, \ldots, \tau^N\}$. Since each o_t^i only captures a local egocentric view, no single robot can fully observe the environment. However, the set of observations from different robots,

$$\mathcal{O}_{\leq t}^{1:N} = \{o_s^i \mid i = 1, \ldots, N,\ s \leq t\}, \tag{1}$$

provides complementary multi-view observations about the same underlying world. We define *Shared Environmental Understanding* (SEU) as a policy-agnostic environmental representation that aggregates partial egocentric observations from multiple robots into a common representation, which can be reused by different downstream navigation agents without enforcing a centralized controller. Concretely, we aim to learn a function

$$\mathbf{z}_t^{\text{SEU}} = f_\theta\left(\mathcal{O}_{\leq t}^{1:N}\right), \tag{2}$$

where $\mathbf{z}_t^{\text{SEU}}$ summarizes multi-robot partial observations into a policy-agnostic environmental representation. A downstream navigation agent can then make decisions using its own local observation, instruction, and the shared environmental context: $a_t^i \sim \pi_i\left(a \mid o_t^i, x^i, \mathbf{z}_t^{\text{SEU}}\right)$. Importantly, π_i can be different across robots or downstream models, meaning that *Shared Environmental Understanding* provides shared environmental information without requiring all robots to share the same policy.

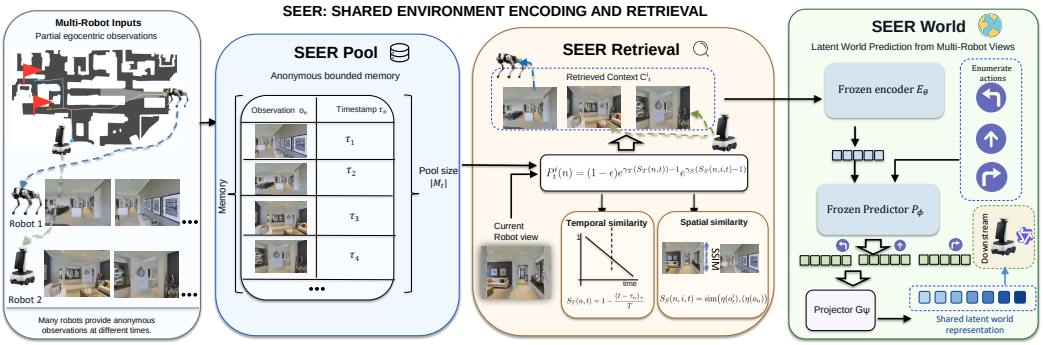


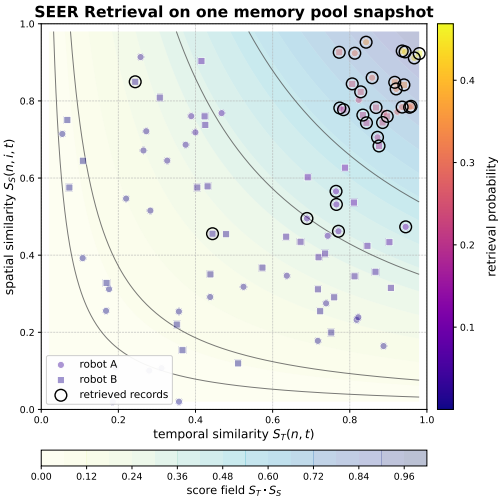
Figure 3: Overview of our shared Environment Encoding and Retrieval with SEER-World.

3.2 SEER: Shared Environment Encoding and Retrieval

We implement *Shared Environmental Understanding* through *Shared Environment Encoding and Retrieval* (SEER). SEER consists of three key components. First, *SEER Pool* is an identity-agnostic memory that aggregates observations or latent features across robots. Second, *SEER Retrieval* selects relevant context from the pool via spatio-temporal similarity-based sampling, enabling effective information sharing. Third, *SEER-World* is a multi-view latent world model that encodes fragmented egocentric observations into a shared environmental representation.

3.2.1 SEER Pool and SEER Retrieval

Different robots observe the same environment from different egocentric viewpoints and at different time steps. Naively concatenating all observations would be inefficient and would introduce many irrelevant views. To address this issue, SEER introduces an anonymous shared memory mechanism, consisting of the *SEER Pool* and *SEER Retrieval*. The goal is to store multi-robot observations as shared environmental context, and retrieve only the observations that are likely to be useful for the current robot under a fixed context budget.



At time t, the shared memory pool *SEER Pool* is denoted as $\mathcal{M}_t = \{m_n\}_{n=1}^{|\mathcal{M}_t|}$, where n indexes memory entries. Each record m_n contains an observation o_n and its timestamp τ_n. The bounded pool size prevents unstructured growth as more robots contribute observations.

To retrieve useful context from $\mathcal{M}_t$ under a fixed budget, we introduce a time-space-aware retrieval probability, inspired by budget-aware temporal sampling in NavFoM [94]. This mechanism prioritizes observations that are both temporally recent and spatially relevant to the querying robot.

Given the current observation o_t^i of robot i, we assign each memory record m_n a spatio-temporal weight and normalize it into a sampling probability:

$$
\begin{aligned}
w_t^i(n) &= (1-\epsilon)\exp\big(\gamma_T(S_T(n,t)-1)\big)\exp\big(\gamma_S(S_S(n,i,t)-1)\big)+\epsilon, \\
P_t^i(n) &= \frac{w_t^i(n)}{\sum_{m=1}^{|\mathcal{M}_t|} w_t^i(m)}, \qquad 0<\epsilon<1.
\end{aligned}
\tag{3}
$$

We use weighted sampling without replacement, renormalizing over the remaining records after each draw. All main experiments use a FIFO pool of 1,024 records, a total context budget of $K = 8$ observations, $\gamma_T = \gamma_S = 1$, and $\epsilon = 0.1$. Missing cross-robot context is supplemented with recent self-history. Only observations with $\tau_n \le t$ are eligible; same-timestep observations from other robots become available once uploaded, while future observations are excluded.

The temporal similarity $S_T(n,t)$ is defined as

$$S_T(n,t) = 1 - \frac{(t-\tau_n)_+}{T}, \tag{4}$$

where τ_n is the timestamp of memory record m_n, $(\cdot)_+$ denotes the positive part, and T is the temporal horizon controlling the decay. The spatial similarity $S_S(n,i,t)$ measures the relevance between the current observation o_t^i and the stored observation o_n:

$$S_S(n,i,t) = \mathrm{sim}\Big(\eta(o_t^i), \eta(o_n)\Big), \tag{5}$$

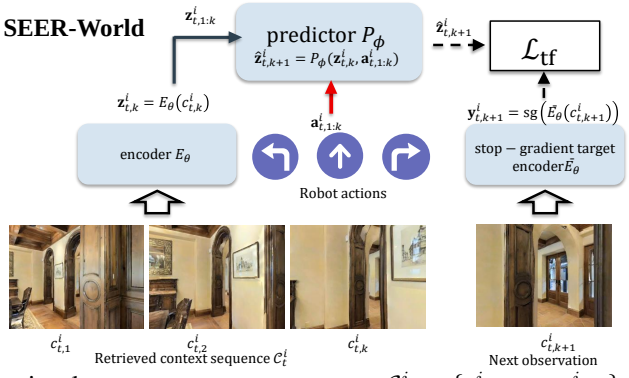
where $\eta(\cdot)$ denotes the visual encoder and $\mathrm{sim}(\cdot,\cdot)$ is implemented as channel-wise SSIM. SSIM used as a lightweight structural correlation for coarse spatial layout agreement under a fixed retrieval budget, inspired by previous work [95–98]. Both $S_T(n,t)$ and $S_S(n,i,t)$ are normalized to $[0,1]$, with larger values indicating higher temporal recency and spatial relevance, respectively. The coefficients γ_T and γ_S control the relative contributions of temporal and spatial similarity in the retrieval process.

Under a fixed context budget, SEER Retrieval samples a compact set of observations from $\mathcal{M}_t$ according to $P_t^i(n)$, without replacement. The sampled observations are then ordered by timestamp and combined with the current observation to form the context sequence $\mathcal{C}_t^i$. This context is then passed to our SEER-World $F_{\theta,\phi}$.

This retrieval formulation exhibits several desirable properties. First, the probability remains strictly positive due to the lower-bound term ϵ, ensuring that older or less similar observations are not entirely discarded. Second, the temporal factor $\exp\big(\gamma_T(S_T(n,t)-1)\big)$ biases sampling toward recent observations, preserving short-term context critical for navigation. Third, the spatial factor $\exp\big(\gamma_S(S_S(n,i,t)-1)\big)$ suppresses temporally recent but irrelevant views, while promoting observations that are spatially or semantically aligned with the current input. Fourth, the fixed context budget ensures that the number of retrieved tokens does not scale with the number of robots, viewpoints, or stored observations. Finally, by operating over an anonymous memory pool, SEER naturally generalizes to varying numbers of robots and viewpoints: additional robots enrich $\mathcal{M}_t$, while each agent still retrieves a bounded, relevance-weighted context.

3.2.2 SEER-World: Latent World Prediction from Multi-Robot Views

SEER-World converts retrieved multi-robot observations into a shared environmental representation by predicting future states in latent space. This design follows the central motivation of V-JEPA-style models [99, 100]: a world model does not need to reconstruct future pixels in order to be useful; it can instead predict future states in a learned representation space. SEER-World uses the same latent prediction principle to organize multi-robot, multi-view observations into a reusable shared environmental context. For robot i at time t, SEER Retrieval returns a context sequence $\mathcal{C}_t^i = \{c_{t,1}^i, \ldots, c_{t,K}^i\}$, where each element $c_{t,k}^i$ is an egocentric observation selected from the shared SEER Pool. The sequence may contain observations from robot i itself and from other robots that observed related regions of the same environment. We also provide the corresponding action condition of the current robot, $\mathbf{a}_{t,1:K-1}^i = \{a_{t,1}^i, \ldots, a_{t,K-1}^i\}$, which tells the predictor how the current robot moves through the retrieved context.



SEER-World consists of an encoder E_θ, a stop-gradient target encoder $\bar{E}_\theta$, and an action-conditioned predictor P_ϕ. Given the retrieved context sequence $\mathcal{C}_t^i$, SEER-World encodes each observation, computes its stop-gradient target, and predicts the next latent state conditioned on previous latents and robot actions:

$$\mathbf{z}_{t,k}^i = E_\theta(c_{t,k}^i), \quad \mathbf{y}_{t,k+1}^i = \mathrm{sg}\Big(\bar{E}_\theta(c_{t,k+1}^i)\Big), \quad \hat{\mathbf{z}}_{t,k+1}^i = P_\phi\left(\mathbf{z}_{t,1:k}^i, \mathbf{a}_{t,1:k}^i\right), \tag{6}$$

$\mathrm{sg}(\cdot)$ denotes stop-gradient. SEER-World is trained in latent space with a teacher-forcing objective, the loss $\mathcal{L}_{\mathrm{tf}}$ supervises one-step latent transitions conditioned on robot actions. The overall objective combines teacher-forcing supervision with a short-horizon rollout regularization:

$$\mathcal{L}_{\mathrm{world}} = \mathcal{L}_{\mathrm{tf}} + \lambda_{\mathrm{roll}}\mathcal{L}_{\mathrm{roll}}, \quad \mathcal{L}_{\mathrm{tf}} = \frac{1}{K-1}\sum_{k=1}^{K-1}\left\|\hat{\mathbf{z}}^i_{t,k+1} - \mathbf{y}^i_{t,k+1}\right\|_1. \tag{7}$$

where the rollout loss $\mathcal{L}_{\mathrm{roll}}$ is detailed in Appendix A.1. After training, SEER-World outputs a shared environmental context $\mathbf{z}_t^{\mathrm{SEU},i}$.

3.3 Integration with Downstream Navigation

SEER is designed as an environmental understanding layer. After SEER-World learns a shared latent world representation $\mathbf{z}_t^{\mathrm{SEU},i}$ from multi-robot observations, we use it to condition downstream navigation agents without replacing their original decision process. This design allows each robot to retain its own policy while receiving additional shared environmental context.

Let h_t^i denote the local navigation state of robot i, including its current observation, visual-language history, and model-specific memory. A downstream navigation agent predicts actions from its local state and instruction $a_t^i \sim \pi_i(a \mid h_t^i, x^i)$, while our SEER-augmented agent additionally conditions on SEER tokens as $a_t^i \sim \pi_i(a \mid h_t^i, x^i, \mathbf{U}_t^i)$. $\mathbf{U}_t^i$ denotes the SEER conditioning tokens provided to robot i. SEER therefore changes what the agent knows about the shared environment, but does not change the agent's action space, controller, or decision interface.

In our implementation, $\mathbf{U}_t^i$ is constructed by querying the frozen SEER-World model with the retrieved multi-robot context and converting the resulting latent predictions into continuous soft tokens. These tokens are inserted immediately before downstream action prediction, acting as world-conditioning prompts. We use the same interface for both Qwen 3.5 4B [101] and StreamVLN [83], demonstrating that SEER can serve as a reusable upstream representation for different types of navigation agents. Given the retrieved context $\mathcal{C}_t^i$, SEER-World predicts one latent world hint for each candidate motion:

$$\hat{\mathbf{z}}_{t+1}^{\mathrm{SEU},i} = F_{\theta,\phi}\left(\mathcal{C}_t^i, \mathbf{a}^i_{t,1:|\mathcal{C}_t^i|-1}, a\right), a \in \mathcal{A}_{\mathrm{move}}, \quad \mathbf{U}_t^i = G_\psi\left(\left[\hat{\mathbf{z}}_{t+1}^{\mathrm{SEU},i}\right]_{a\in\mathcal{A}_{\mathrm{move}}}\right), \tag{8}$$

where $\mathcal{A}_{\mathrm{move}} = \{\textsc{Forward}, \textsc{Left}, \textsc{Right}\}$and G_ψ is a lightweight projector that maps SEER latents into the hidden dimension of the downstream agent. In practice, the projector pools latent tokens, applies a small MLP, and adds action identity embeddings so that the downstream model can distinguish the world hints for FORWARD, LEFT, and RIGHT. These tokens are inserted as continuous embeddings immediately before the downstream model predicts the next action, acting as a world-conditioning prompt. Thus, SEER introduces collaboration at the representation level rather than the decision level: robots share environmental understanding, while each downstream navigation agent independently decides how to act. Detailed construction of SEER tokens and model-specific insertion strategies are provided in Appendix A.2.

3.4 Inverse Dynamics Analysis

To assess whether the representations learned by SEER capture actionable environmental structure, we evaluate them via an inverse dynamics objective. Building on the shared latent representations produced by SEER-World with SEER Pool and Retrieval, this analysis probes whether the latent states encode sufficient information to recover agent actions. As illustrated in Figure 4, given latent representations, an inverse dynamics model predicts the action responsible for the transition between them. We enumerate all candidate motion actions, use the frozen SEER-World predictor to predict the next latent under each action, and select the action whose predicted latent is closest to the true next latent:

$$\hat{a}_t^i = \arg\min_{a\in\mathcal{A}_{\mathrm{move}}} d\left(F_{\theta,\phi}(\mathbf{z}_t^i, a, \mathcal{C}_t^i), \mathbf{z}_{t+1}^i\right), \tag{9}$$

where $F_{\theta,\phi}$ denotes the learned SEER-World forward latent dynamics, $\mathcal{C}_t^i$ is the retrieved context, and $d(\cdot,\cdot)$ is the ℓ_1 distance. We

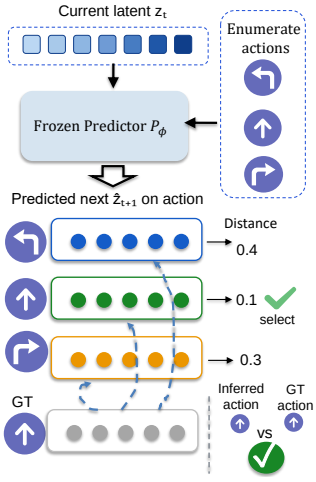


Figure 4: LIDA Illustration

Table 1: Comparison on VLN-CE R2R and RxR **Val-Unseen** splits. Pano, Odo and D respectively represent panoramic view, odometry and depth, SRGB denotes monocular RGB.

Method	Observation				R2R Val-Unseen				RxR Val-Unseen			
	Pano.	Odo.	D.	SRGB	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
CMA [102]	✓	✓	✓		6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0
VLN-BERT [102]	✓	✓	✓		5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7
Sim2Sim [103]	✓	✓	✓		6.07	52.0	43.0	36.0	–	–	–	–
Ego^2-Map [104]	✓	✓	✓		5.54	56.0	47.0	41.0	–	–	–	–
DreamWalker [105]	✓	✓	✓		5.53	59.0	49.0	44.0	–	–	–	–
ETPNav [106]	✓	✓	✓		4.71	65.0	57.0	49.0	5.64	54.7	44.8	61.9
ScaleVLN [107]	✓	✓	✓		4.80	–	55.0	51.0	–	–	–	–
InstructNav [108]	✓	✓	✓	✓	6.89	–	31.0	24.0	–	–	–	–
LAW [109]		✓	✓	✓	6.83	44.0	35.0	31.0	10.90	8.0	8.0	38.0
Sim2Real [110]		✓	✓	✓	5.95	55.8	44.9	30.4	8.79	25.5	18.1	–
Seq2Seq [111]			✓	✓	7.77	37.0	25.0	22.0	12.10	13.9	11.9	30.8
CMA [111]			✓	✓	7.37	40.0	32.0	30.0	–	–	–	–
NaVid [112]				✓	5.47	49.1	37.4	35.9	–	–	–	–
AO-Planner [113]	✓		✓		5.55	59.0	47.0	33.0	7.06	43.3	30.5	50.1
MapNav [86]				✓	4.93	53.0	39.7	37.2	–	–	–	–
NaVid-4D [114]			✓	✓	5.99	55.7	43.8	37.1	–	–	–	–
NavMorph [88]			✓	✓	5.75	56.9	47.9	33.2	8.85	30.8	22.8	44.2
Uni-NaVid [115]				✓	5.58	53.3	47.0	42.7	6.24	48.7	40.9	–
NaVILA [116]				✓	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8
StreamVLN 7B [83]				✓	4.98	64.2	56.9	51.9	6.22	52.9	46.0	61.9
+SEER(ours)				✓	4.89	65.6	58.1	52.5	6.17	54.3	47.2	62.1
Qwen3.5 4B				✓	7.35	49.8	40.3	35.9	8.01	39.7	32.5	45.6
+SEER(ours)				✓	6.97	53.2	44.9	39.8	7.53	42.4	35.6	49.4

report the action recovery accuracy, denoted as Latent Inverse Dynamics Accuracy (LIDA):

$$\text{LIDA} = \frac{\sum_{(\mathbf{z}_t, a_t, \mathbf{z}_{t+1}) \in \mathcal{D}_{\text{eval}}} \mathbf{1}[\hat{a}_t = a_t]}{|\mathcal{D}_{\text{eval}}|}. \tag{10}$$

A higher LIDA indicates that different actions induce distinguishable transitions in latent space, suggesting that the learned shared environmental representations capture navigation-relevant dynamics rather than merely appearance-level semantics. Additional details are provided in Appendix A.3.

4 Experiments

4.1 Experimental Setup

We evaluate SEER in both simulated VLN environments and real-world robot deployments. The simulated evaluation contains three parts: standard single-robot VLN, multi-robot VLN constructed from paired R2R trajectories, and downstream integration with different navigation agents. Detailed construction of multi-robot episodes are provided in Appendix A.4. For standard VLN, we follow the continuous VLN-CE setting and report results on the Val-Unseen splits of R2R-CE [1, 111] and RxR-CE [67] in the Habitat simulator [33]. We use standard VLN metrics, including Navigation Error (NE), Success Rate (SR), Oracle Success Rate (OSR), Success weighted by Path Length (SPL), and normalized Dynamic Time Warping (nDTW) [1, 117]. These single-robot evaluations test whether adding SEER preserves the original navigation ability of downstream agents.

For multi-robot evaluation, we use the paired R2R episodes constructed by ourself. The evaluation set is divided into four subsets according to the spatial relationship between the two robots: *different-scene*, *same-scene non-intersecting*, *same-scene intersecting*, and *same/sub-trajectory*. These subsets allow us to test whether shared environmental understanding helps under different levels of shared context, from no shared scene to highly overlapping trajectories. Following the spirit of collaborative navigation evaluation [118], we report both individual and collaborative success. For a paired episode

Table 2: Multi-agent performance comparison. For dual-agent state tests, CSR and ISR denote complete success rate and individual success rate, respectively. SEER significantly boosts multi-agent navigation performance across all dual-agent settings.

Method	Single-Agent		Dual-Agent State Test							
	R2R Unseen	RxR Unseen	Different Scenes		Same Scene w/o Intersection		Same Scene w/ Intersection		Overlapped Trajectories	
	SR↑	SR↑	CSR↑	ISR↑	CSR↑	ISR↑	CSR↑	ISR↑	CSR↑	ISR↑
Random	2.50	5.89	0.0	3.05	0.0	2.15	0.0	1.60	0.0	1.40
StreamVLN	56.9	52.9	28.8	52.2	28.2	51.2	31.2	55.7	30.3	54.9
+SEER	58.1	54.3	32.9 (+4.1)	56.1 (+3.9)	32.6 (+4.4)	55.8 (+4.6)	35.9 (+4.7)	60.7 (+5.0)	35.4 (+5.1)	60.1 (+5.2)
Qwen3.5-4B	40.3	39.7	17.0	39.7	17.7	40.6	18.1	41.4	17.2	40.1
+SEER	44.9	42.4	21.6 (+4.6)	43.8 (+4.1)	22.1 (+4.4)	45.3 (+4.7)	23.9 (+5.8)	47.5 (+6.1)	22.4 (+5.2)	45.6 (+5.5)

Table 3: Inverse dynamics analysis.SEER-World dramatically improves inverse dynamics accuracy by capturing how actions transform the environment in latent space.

Method	Single-Agent		Dual-Agent State Test							
	R2R Unseen	RxR Unseen	Different Scenes		Same Scene w/o Intersection		Same Scene w/ Intersection		Overlapped Trajectories	
	SR↑	SR↑	CSR↑	ISR↑	CSR↑	ISR↑	CSR↑	ISR↑	CSR↑	ISR↑
Random	33.8	33.4	11.3	33.6	11.0	33.2	11.4	33.9	11.2	33.4
V-JEPA2	78.2	77.2	66.8	77.4	68.3	78.7	68.0	78.4	70.7	79.5
Ours w/o SEER Retrieval	92.1	91.9	89.1	91.5	89.6	91.8	88.8	91.3	90.1	91.9
Ours	**95.4**	**94.7**	**94.9**	**95.4**	**94.8**	**95.3**	**95.9**	**95.9**	**96.0**	**96.0**

e, let $s_e^A, s_e^B \in \{0, 1\}$ denote whether robot A and robot B successfully reach their respective goals. We define Individual Success Rate (ISR) and Collaborative Success Rate (CSR) as

$$\mathrm{ISR} = \frac{1}{2|\mathcal{D}|} \sum_{e \in \mathcal{D}} (s_e^A + s_e^B), \qquad \mathrm{CSR} = \frac{1}{|\mathcal{D}|} \sum_{e \in \mathcal{D}} s_e^A s_e^B. \tag{11}$$

ISR measures whether each robot can still complete its own task, while CSR measures whether both robots succeed in the paired episode. We emphasize that our paired-R2R evaluation is not intended to model full physical multi-robot coordination; instead, it provides a controlled abstraction for isolating SEER's core question: whether independently collected egocentric observations can be transformed into reusable shared environmental context for decentralized navigation.

We instantiate downstream navigation with two agents: a specialized StreamVLN agent [83] and a general Qwen3.5-based agent [101], testing whether SEER is reusable rather than model-specific. We further conduct real-world indoor experiments on two embodiments, AgileX LiMO Pro and Unitree Go2, which performs inference and returns actions to robot-specific low-level controllers via ROS 2. Details on splits, evaluation, deployment, and implementation are in Appendix A.5.

4.2 Task-Level Validation: Shared Environmental Understanding for Navigation

Single-agent preservation. A key requirement of SEER is that it should not degrade ordinary single-agent navigation. In this sense, single-agent navigation can be viewed as a special case of our setting, where the robot mainly relies on its own observation context and receives no useful cross-robot information. Table 1 evaluates this property, when SEER is added to StreamVLN, the resulting agent preserves the strong performance of the original navigation model and improves the main success-oriented metrics. These results show that SEER tokens provide useful environmental context while preserving the downstream action space and output format. SEER-World is trained independently of downstream navigation policies and frozen during downstream adaptation, while the projector and downstream navigation agent are optimized. Here, policy-agnosticism refers to reusing the same shared world model across downstream navigation policies. **Multi-robot navigation improvement.** Table 2 evaluates SEER in paired multi-robot episodes under varying shared environmental context. Across all subsets, SEER improves both ISR and CSR. ISR gains show that each robot benefits from shared environmental context while solving its own task, whereas CSR gains indicate that both robots are more likely to succeed in the same paired episode. SEER achieves stronger gains in same-scene and overlapped-trajectory settings, where cross-robot observations are more relevant.

Compatibility with different downstream agents. We evaluate SEER with both a specialized VLN agent (StreamVLN) and a general VLM-based navigation agent (Qwen 3.5), using the same integration strategy. Results show that SEER improves StreamVLN and brings even larger gains to the

Table 4: Comparison of different sharing strategies on Qwen3.5-4B. *Share Spatial Structure uses an additional 3D expert InternVL3 8B [119] to extract and share spatial descriptions from observations.

Method	Dual-Agent State Test							
	Different Scenes		Same Scene w/o Intersection		Same Scene w/ Intersection		Overlapped Trajectories	
	CSR↑	ISR↑	CSR↑	ISR↑	CSR↑	ISR↑	CSR↑	ISR↑
No Sharing	17.0	39.7	17.7	40.6	18.1	41.4	17.2	40.1
Share History	16.2	38.9	17.3	39.5	17.8	40.9	17.4	40.3
Share Action	17.6	40.0	18.1	40.4	19.4	42.2	19.7	42.4
Share Observation	18.4	40.5	19.2	41.1	19.8	43.4	20.3	43.4
*Share Spatial	19.6	41.4	20.7	42.8	20.6	45.1	20.9	44.7
SEER	21.6	43.8	22.1	45.3	23.9	47.5	22.4	45.6



Figure 5: Qualitative Analysis of Success and Failure Cases on different Dual-Agent State.

Qwen 3.5 agent. This supports that SEER is not a model-specific trick, but a reusable environmental understanding layer for different downstream navigation agents. It enables collaborative navigation by sharing latent world understanding, without sharing actions, histories, or a centralized policy.

4.3 Representation-Level Validation: Action-Relevant Latent Dynamics

We introduce a representation-level inverse dynamics probe to test whether SEER encodes action-relevant environmental structure. Since the latent state is not directly interpretable, we freeze the learned forward latent dynamics and, for each transition, enumerate candidate actions. The action whose predicted next latent is closest to the true next latent is selected. High recovery accuracy therefore indicates that the latent space preserves how different navigation actions transform the world state, rather than merely encoding static appearance.

Table 3 shows that the probe is non-trivial: V-JEPA2 achieves reasonable accuracy, indicating that generic predictive video features capture motion cues, but still lags behind SEER. *Ours w/o Retrieval* (replaced by uniform random sampling) already performs well, suggesting that SEER-World learns discriminative latent dynamics via multi-view prediction. SEER Retrieval further improves both single- and dual-agent settings, isolating the benefit of shared retrieval: relevant observations from the anonymous SEER Pool provide cleaner cross-robot context than the world model alone. Importantly, LIDA is a held-out counterfactual action recovery test across diverse spatial configurations, not a direct training-loss measure, making high scores unlikely to stem only from action bias, visual repetition, or retrieval leakage.

4.4 What Should Robots Share: World Understanding vs. Agent-Centric Traces

Table 4 compares SEER with several sharing strategies. We position these baselines as controlled comparisons against representative agent-centric sharing strategies, while leaving stronger learned communication, map-sharing, and centralized multi-agent architectures as complementary directions for future work rather than direct targets of this representation-layer study. Sharing actions only reveals what another robot did, without explaining the environmental structure that made the action useful. Sharing spatial structure* improves over these simple signals by providing explicit geometric context, but it requires an additional 3D expert and mainly captures static layout, without modeling how actions change the world state. SEER showing that collaboration benefits more from shared

world understanding than from sharing raw histories, actions, or hand-crafted maps. Notably, semantic or topological maps are not mutually exclusive with SEER; they can be viewed as complementary external structures, while SEER provides a learned, action-aware representation layer.

5 Conclusion

This paper argues for a shift in how multi-robot navigation should be understood. The essence of collaboration is not merely that multiple robots move at the same time, avoid each other, or exchange action histories; it is that each robot can make the world more legible for the others. By formulating multi-robot VLN around shared environmental understanding, we move the problem from surface-level coordination to a deeper question of collective scene comprehension. Complementary embodied experience can be transformed into shared knowledge that improves navigation beyond isolated single-agent perception. The real-robot study further shows that this idea is not confined to benchmark artifacts. We believe this representation-centric view offers a cleaner, more scalable, and more general foundation for embodied multi-agent systems.

Acknowledgments and Disclosure of Funding

This work was partially supported by New Generation Artificial Intelligence-National Science and Technology Major Projection(2025ZD0123100) and by The National Natural Science Foundation of China (NSFC) under no. 62573399 and U25A20530.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.

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

[3] M. Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proc. IEEE*, 94(7):1257–1270, 2006.

[4] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12):399, 2013.

[5] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems 29*, pages 2244–2252, 2016.

[6] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems 30*, pages 6379–6390, 2017.

[7] Xinzhu Liu, Di Guo, Huaping Liu, and Fuchun Sun. Multi-agent embodied visual semantic navigation with scene prior knowledge. *IEEE Robotics Autom. Lett.*, 7(2):3154–3161, 2022.

[8] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. Co-navgpt: Multi-robot cooperative visual semantic navigation using large language models. *CoRR*, abs/2310.07937, 2023.

[9] Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Behzad Dariush, Kwonjoon Lee, Yilun Du, and Chuang Gan. COMBO: compositional world models for embodied multi-agent cooperation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[10] Yara Rizk, Mariette Awad, and Edward W. Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Comput. Surv.*, 52(2):29:1–29:31, 2019.

[11] Trevor Halsted, Ola Shorinwa, Javier Yu, and Mac Schwager. A survey of distributed optimization methods for multi-robot systems. *CoRR*, abs/2103.12840, 2021.

[12] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E. Schneider. Coordinated multi-robot exploration. *IEEE Trans. Robotics*, 21(3):376–386, 2005.

[13] Soon-Jo Chung, Aditya A. Paranjape, Philip M. Dames, Shaojie Shen, and Vijay Kumar. A survey on aerial swarm robotics. *IEEE Trans. Robotics*, 34(4):837–855, 2018.

[14] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4292–4301, 2018.

[15] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1538–1546, 2019.

[16] Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2961–2970, 2019.

[17] Xinzhu Liu, Xinghang Li, Di Guo, Sinan Tan, Huaping Liu, and Fuchun Sun. Embodied multi-agent task planning from ambiguous instruction. In *Robotics: Science and Systems XVIII*, 2022.

[18] Xinzhu Liu, Di Guo, Xinyu Zhang, and Huaping Liu. Heterogeneous embodied multi-agent collaboration. *IEEE Robotics Autom. Lett.*, 9(6):5377–5384, 2024.

[19] Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. SMART-LLM: smart multi-agent robot task planning using large language models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 12140–12147, 2024.

[20] Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, Siddharth Patki, Ishita Prasad, Xavier Puig, Akshara Rai, Ram Ramrakhya, Daniel Tran, Joanne Truong, John M. Turner, Eric Undersander, and Tsung-Yen Yang. PARTNR: A benchmark for planning and reasoning in embodied multi-agent tasks. In *The Thirteenth International Conference on Learning Representations*, 2025.

[21] 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 *Computer Vision - ECCV*, pages 605–621, 2020.

[22] 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 *Computer Vision - ECCV*, pages 107–124, 2022.

[23] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Conference on Robot Learning*, pages 989–1000, 2022.

[24] Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 2:1–2:22, 2023.

[25] Chen Gao, Baining Zhao, Weichen Zhang, Jinzhu Mao, Jun Zhang, Zhiheng Zheng, Fanhang Man, Jianjie Fang, Zile Zhou, Jinqiang Cui, Xinlei Chen, and Yong Li. Embodiedcity: A benchmark platform for embodied agent in real-world city environment. *CoRR*, abs/2410.09604, 2024.

[26] Nicolas Bougie and Narimawa Watanabe. Citysim: Modeling urban behaviors and city dynamics with large-scale llm-driven agent simulation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - Industry Track*, pages 215–229, 2025.

[27] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *The Twelfth International Conference on Learning Representations*, 2024.

[28] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *The Twelfth International Conference on Learning Representations*, 2024.

[29] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Binyuan Huang, Fan Jia, Yanhui Wang, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea+: Panoramic and controllable video generation for autonomous driving. *IEEE Trans. Circuits Syst. Video Technol.*, 36(2): 2247–2258, 2026.

[30] Junpeng Jiang, Gangyi Hong, Miao Zhang, Hengtong Hu, Kun Zhan, Rui Shao, and Liqiang Nie. Dive: Efficient multi-view driving scenes generation based on video diffusion transformer. *CoRR*, abs/2504.19614, 2025.

[31] Angel X. Chang, Angela Dai, Thomas A. 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*, pages 667–676, 2017.

[32] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2432–2443, 2017.

[33] Manolis Savva, Jitendra Malik, Devi Parikh, Dhruv Batra, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, and Vladlen Koltun. Habitat: A platform for embodied AI research. In *IEEE/CVF International Conference on Computer Vision*, pages 9338–9346, 2019.

[34] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Théophile Gervet, John M. Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4927–4936, 2023.

[35] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, Xihui Liu, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied AI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19757–19767, 2024.

[36] Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. Scanrefer: 3d object localization in RGB-D scans using natural language. In *Computer Vision - ECCV*, pages 202–221, 2020.

[37] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision - ECCV*, pages 422–440, 2020.

[38] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19107–19117, 2022.

[39] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul McVay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent-Pierre Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Alexander Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation

models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16488–16498, 2024.

[40] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. 2023.

[41] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. In *Advances in Neural Information Processing Systems 36*, 2023.

[42] Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. Situational awareness matters in 3d vision language reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13678–13688, 2024.

[43] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pages 2455–2467, 2018.

[44] Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2555–2565, 2019.

[45] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse domains through world models. *CoRR*, abs/2301.04104, 2023.

[46] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, pages 2226–2240, 2022.

[47] Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal M. P. Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott E. Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, pages 4603–4623, 2024.

[48] Xiaofeng Wang, Zheng Zhu, Guan Huang, Boyuan Wang, Xinze Chen, and Jiwen Lu. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *CoRR*, abs/2401.09985, 2024.

[49] Haihong Hao, Shuo Wang, Huixia Ben, Yanbin Hao, Yansong Wang, and Weiwei Wang. Hierarchical space-time attention for micro-expression recognition. *arXiv preprint arXiv:2405.03202*, 2024.

[50] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *Computer Vision - ECCV*, pages 55–72, 2024.

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

[52] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Trans. Mach. Learn. Res.*, 2024, 2024.

[53] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew J. Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *CoRR*, abs/2506.09985, 2025.

[54] Xiaohan Lei, Min Wang, Wengang Zhou, and Houqiang Li. Gaussnav: Gaussian splatting for visual navigation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(5):4108–4121, 2025.

[55] Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: Learning 4d embodied world models. *CoRR*, abs/2504.20995, 2025.

[56] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183, 2023.

[57] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pages 2679–2713, 2024.

[58] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024.

[59] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1702–1713, 2025.

[60] Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied AI. *CoRR*, abs/2405.14093, 2024.

[61] Yufei Guo, Yinan Wu, Haoran Duan, Guiguang Ding, and Jungong Han. Phaselora: Control-regime-conditioned low-rank adaptation for continuous-action vision-language-action policies. *arXiv preprint arXiv:2608.15285*, 2026.

[62] 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 31*, pages 3318–3329, 2018.

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

[64] 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 *7th International Conference on Learning Representations*, 2019.

[65] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6732–6740, 2019.

[66] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2610–2621, 2019.

[67] 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 Conference on Empirical Methods in Natural Language Processing*, pages 4392–4412, 2020.

[68] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9979–9988, 2020.

[69] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. SOON: scenario oriented object navigation with graph-based exploration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12689–12699, 2021.

[70] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision - ECCV*, pages 104–120, 2020.

[71] Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Wang. Aerial vision-and-dialog navigation. In *Findings of the Association for Computational Linguistics: ACL*, pages 3043–3061, 2023.

[72] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12538–12547, 2019.

[73] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13134–13143, 2020.

[74] 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 34*, pages 5834–5847, 2021.

[75] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16516–16526, 2022.

[76] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022.

[77] Jialu Li and Mohit Bansal. Improving vision-and-language navigation by generating future-view image semantics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10803–10812, 2023.

[78] Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *IEEE/CVF International Conference on Computer Vision*, pages 15338–15348, 2023.

[79] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 7641–7649, 2024.

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

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

[82] Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12078–12088, 2025.

[83] 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. *CoRR*, abs/2507.05240, 2025.

[84] 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. *CoRR*, abs/2507.13019, 2025.

[85] Weichen Zhang, Chen Gao, Shiquan Yu, Ruiying Peng, Baining Zhao, Qian Zhang, Jinqiang Cui, Xinlei Chen, and Yong Li. Citynavagent: Aerial vision-and-language navigation with hierarchical semantic planning and global memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31292–31309, 2025.

[86] Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. Mapnav: A novel memory representation via annotated semantic maps for vlm-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13032–13056, 2025.

[87] Yuhang Zhang, Haosheng Yu, Jiaping Xiao, and Mir Feroskhan. Grounded vision-language navigation for uavs with open-vocabulary goal understanding. *CoRR*, abs/2506.10756, 2025.

[88] Xuan Yao, Junyu Gao, and Changsheng Xu. Navmorph: A self-evolving world model for vision-and-language navigation in continuous environments. *CoRR*, abs/2506.23468, 2025.

[89] Haihong Hao, Lei Chen, Mingfei Han, Changlin Li, Dong An, Yuqiang Yang, Zhihui Li, and Xiaojun Chang. Latentpilot: Scene-aware vision-and-language navigation by dreaming ahead with latent visual reasoning. *CoRR*, abs/2603.29165, 2026.

[90] Mingfei Han, Haihong Hao, Liang Ma, Kamila Zhumakhanova, Ekaterina Radionova, Jingyi Zhang, Xiaojun Chang, Xiaodan Liang, and Ivan Laptev. Implicit geometry representations for vision-and-language navigation from web videos. *arXiv preprint arXiv:2603.09259*, 2026.

[91] Haihong Hao, Mingfei Han, Changlin Li, Zhihui Li, and Xiaojun Chang. Conav: Collaborative cross-modal reasoning for embodied navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[92] Pengyang Shao, Naixin Zhai, Lei Chen, Yonghui Yang, Fengbin Zhu, Xun Yang, and Meng Wang. Baldro: A distributionally robust optimization based framework for large language model unlearning. In *Proceedings of the ACM Web Conference*, pages 8874–8884, 2026.

[93] Naixin Zhai, Pengyang Shao, Binbin Zheng, Yonghui Yang, Fei Shen, Long Bai, and Xun Yang. Maximizing local entropy where it matters: Prefix-aware localized llm unlearning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19514–19531, 2026.

[94] 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 He Wang. Embodied navigation foundation model. *CoRR*, abs/2509.12129, 2025.

[95] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4): 600–612, 2004.

[96] Takumi Kobayashi. Structured feature similarity with explicit feature map. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1211–1219, 2016.

[97] Andrew Wentzel, Peter Hanula, Timothy Luciani, Baher Elgohari, Hesham Elhalawani, Guadalupe Canahuate, David M. Vock, Clifton D. Fuller, and G. Elisabeta Marai. Cohort-based T-SSIM visual computing for radiation therapy prediction and exploration. *IEEE Trans. Vis. Comput. Graph.*, 26(1):949–959, 2020.

[98] Keke Zhang, Weiling Chen, Tiesong Zhao, and Zhou Wang. Structural similarity in deep features: Unified image quality assessment robust to geometrically disparate reference. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(3):2581–2595, 2026.

[99] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew J. Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *CoRR*, abs/2506.09985, 2025.

[100] Shangchen Miao, Ningya Feng, Jialong Wu, Ye Lin, Xu He, Dong Li, and Mingsheng Long. JEPA-VLA: video predictive embedding is needed for VLA models. *CoRR*, abs/2602.11832, 2026.

[101] Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

[102] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15418–15428, 2022.

[103] Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *Computer Vision - ECCV*, pages 588–603, 2022.

[104] Yicong Hong, Yang Zhou, Ruiyi Zhang, Franck Dernoncourt, Trung Bui, Stephen Gould, and Hao Tan. Learning navigational visual representations with semantic map supervision. In *IEEE/CVF International Conference on Computer Vision*, pages 3032–3044, 2023.

[105] Hanqing Wang, Wei Liang, Luc Van Gool, and Wenguan Wang. Dreamwalker: Mental planning for continuous vision-language navigation. In *IEEE/CVF International Conference on Computer Vision*, pages 10839–10849, 2023.

[106] 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 Trans. Pattern Anal. Mach. Intell.*, 47(7):5130–5145, 2025.

[107] Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[108] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. In *Conference on Robot Learning*, pages 2049–2060, 2024.

[109] Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X. Chang. Language-aligned waypoint (LAW) supervision for vision-and-language navigation in continuous environments. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4018–4028, 2021.

[110] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Sim-to-real transfer via 3d feature fields for vision-and-language navigation. In *Conference on Robot Learning*, pages 2982–2995, 2024.

[111] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision - ECCV*, pages 104–120, 2020.

[112] 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 XX*, 2024.

[113] Jiaqi Chen, Bingqian Lin, Xinmin Liu, Lin Ma, Xiaodan Liang, and Kwan-Yee K. Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation. In *Thirty-Ninth AAAI Conference on Artificial Intelligence*, pages 23568–23576, 2025.

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

[115] 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. *CoRR*, abs/2412.06224, 2024.

[116] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *CoRR*, abs/2412.04453, 2024.

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

[118] Tianhang Wang, Xinhai Li, Fan Lu, Tianshi Gong, Jiankun Dong, Weiyi Xue, Sanqing Qu, Chenjia Bai, and Guang Chen. Conavbench: Collaborative long-horizon vision-language navigation benchmark. In *The Fourteenth International Conference on Learning Representations*, 2026.

[119] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *CoRR*, abs/2504.10479, 2025.

[120] AgileX Robotics. Limo pro: Ros2 mobile robot platform (product page). URL `https://global.agilex.ai/products/limo-pro`.

A Appendix

This appendix details SEER-World training, downstream integration, inverse dynamics evaluation, paired-episode construction, and physical deployment. The notation follows the main text.

A.1 Rollout Consistency Regularization

Short-horizon prediction. The teacher-forcing objective supervises one-step latent transitions using observed states as predictor inputs. We additionally regularize short autoregressive rollouts, in which predicted latents are fed back to the predictor. Starting from $\mathbf{z}_t^i$ and conditioning on the action sequence $\mathbf{a}_{t:t+R-1}^i$, the R-step prediction and its consistency loss are

$$\widetilde{\mathbf{z}}_{t+R}^i = \mathrm{Roll}_R\Big(P_\phi; \mathbf{z}_t^i, \mathbf{a}_{t:t+R-1}^i\Big), \qquad \mathcal{L}_{\mathrm{roll}} = \Big\|\widetilde{\mathbf{z}}_{t+R}^i - \mathbf{y}_{t+R}^i\Big\|_1, \tag{12}$$

where Roll_R applies the action-conditioned predictor recursively and $\mathbf{y}_{t+R}^i$ is the stop-gradient target latent. This term encourages consistency over successive predicted transitions.

Training objective. SEER-World combines the teacher-forcing loss defined in the main text with rollout consistency:

$$\mathcal{L}_{\mathrm{world}} = \mathcal{L}_{\mathrm{tf}} + \lambda_{\mathrm{roll}}\mathcal{L}_{\mathrm{roll}}. \tag{13}$$

The resulting encoder and predictor form $F_{\theta,\phi}$, which converts retrieved observations and action conditions into latent environmental context for downstream navigation.

A.2 Downstream Navigation Integration

Candidate-action predictions. For robot i, the retrieved context $\mathcal{C}_t^i$ and its action condition are passed to the frozen SEER-World model. We query the candidate motions $\mathcal{A}_{\mathrm{move}} = \{\textsc{Forward}, \textsc{Left}, \textsc{Right}\}$ to obtain action-conditioned world hints:

$$\widehat{\mathbf{z}}_{t+1}^{i,a} = F_{\theta,\phi}\Big(\mathcal{C}_t^i, \mathbf{a}_{t,1:|\mathcal{C}_t^i|-1}^i, a\Big), \qquad a \in \mathcal{A}_{\mathrm{move}}. \tag{14}$$

The downstream policy also predicts STOP as a termination action; latent motion hints are constructed for the three movement actions.

Soft-token construction. The projector pools the predicted latent tokens, applies a small MLP, and adds action identity embeddings. It maps the resulting hints into the hidden dimension of the downstream navigation model:

$$\mathbf{U}_t^i = G_\psi\left(\left[\widehat{\mathbf{z}}_{t+1}^{i,a}\right]_{a\in\mathcal{A}_{\mathrm{move}}}\right). \tag{15}$$

The action embeddings distinguish the predicted effects of moving forward, turning left, and turning right.

Training and insertion. During downstream training, SEER-World is frozen, while the projector and downstream navigation agent are optimized. Thus, the same independently trained environmental model can be reused across downstream policies, with model-specific adaptation at the token interface. For Qwen3.5-4B, SEER tokens are inserted immediately before the assistant's action response. For StreamVLN, they are inserted at each action-prediction round. Both agents retain their original action spaces and output formats, and each robot selects actions from its own local state, instruction, and SEER tokens.

A.3 Inverse Dynamics Evaluation

Action recovery. Latent Inverse Dynamics Accuracy (LIDA) measures whether the learned forward model distinguishes the environmental effects of different navigation actions. For each evaluation transition, we encode the current and next observations as $\mathbf{z}_t^i = E_\theta(o_t^i)$ and $\mathbf{z}_{t+1}^i = E_\theta(o_{t+1}^i)$. We then enumerate the candidate motions using the frozen predictor and recover the action whose predicted latent is closest to the observed next latent:

$$\widetilde{\mathbf{z}}_{t+1}^{i,a} = F_{\theta,\phi}(\mathbf{z}_t^i, a, \mathcal{C}_t^i), \qquad \widehat{a}_t^i = \arg\min_{a\in\mathcal{A}_{\mathrm{move}}} \Big\|\widetilde{\mathbf{z}}_{t+1}^{i,a} - \mathbf{z}_{t+1}^i\Big\|_1. \tag{16}$$

This evaluation uses the ℓ_1 distance, consistent with the main text. Action recovery is computed directly from the frozen forward model by enumerating candidate motions. The next observation

provides the evaluation target; retrieved context is formed from observations available at time t under the protocol in Appendix A.4.

Accuracy. For an evaluation set $\mathcal{D}_{\text{eval}}$ of latent transitions, LIDA is

$$\text{LIDA} = \frac{1}{|\mathcal{D}_{\text{eval}}|} \sum_{(\mathbf{z}_t, a_t, \mathbf{z}_{t+1}) \in \mathcal{D}_{\text{eval}}} \mathbf{1}[\widehat{a}_t = a_t]. \tag{17}$$

Higher accuracy indicates that the predicted latent transitions preserve action-relevant environmental structure. We apply this evaluation to the single-agent data and the four paired-agent configurations described below.

A.4 Paired Multi-Robot Episode Construction

Episode definition. Each paired episode combines two original R2R episodes, assigned to robots A and B. Each robot retains its own instruction, start state, goal, and reference trajectory. The reference paths determine the pairing category, while shared observations are accumulated during execution. The same construction supports inverse dynamics analysis and downstream navigation evaluation.

Path representation. For geometric analysis, we concatenate the start position, reference path, and goal positions, then remove consecutive duplicate points:

$$\rho = \text{Dedup}([p_{\text{start}}; \rho_{\text{ref}}; p_{\text{goal}}]) \,. \tag{18}$$

Including the endpoints preserves intersections and overlaps at the start or goal. The resulting effective paths are used consistently for pair filtering and visualization.

Training and evaluation splits. Training pairs are sampled within scenes from the R2R train split. Duplicate pairs are removed with A/B swaps treated as equivalent, and identical or reversed-identical paths are filtered out. Evaluation pairs are constructed from R2R val-unseen and grouped into four configurations:

Different scenes. The robots navigate in different scenes, $\text{scene}^A \neq \text{scene}^B$. This configuration measures navigation when the shared pool contains observations from unrelated environments.

Same scene without intersection. The robots share a scene, but their effective paths do not intersect. This configuration provides scene-level context with spatially separated routes.

Same scene with intersection. The robots share a scene and their effective paths intersect. This configuration provides opportunities to reuse observations of common local regions.

Overlapped trajectories. The effective paths are identical, reversed-identical, or related by a continuous sub-trajectory. Paired episodes have different instruction texts. This configuration corresponds to the same/sub-trajectory setting and provides highly overlapping environmental context.

Geometric checks and deduplication. Path intersection is computed between polyline segments on the horizontal XZ plane. A projected intersection is accepted only when the corresponding vertical separation satisfies $|y^A - y^B| \leq \epsilon_{\text{floor}}$, which separates crossings on different floors. We remove duplicate episode pairs and repeated geometric path pairs; identical and reversed-identical paths are canonicalized and filtered when they are outside the target category. Sampling uses a fixed random seed, and path coordinates are rounded before canonicalization to reduce floating-point instability. Pair construction preserves the R2R train/val-unseen split.

Causal shared memory. At decision step t, the bounded SEER Pool contains observations already uploaded by either robot, with timestamps no later than t:

$$\mathcal{M}_t \subseteq \{(o_s^j, s) \mid j \in \{A, B\},\ s \leq t\}, \qquad |\mathcal{M}_t| \leq 1{,}024. \tag{19}$$

An observation captured at the current timestep becomes eligible for retrieval once uploaded. Each robot retrieves a bounded context from this pool and selects its own action using its local state, instruction, and SEER tokens. Algorithm 1 summarizes the synchronized evaluation protocol.

A.5 Experimental Setup

Simulation and agents. Single-robot navigation is evaluated on the original R2R-CE and RxR-CE Val-Unseen episodes in Habitat [33]. Multi-robot navigation uses the four paired R2R val-unseen

Algorithm 1 Causal SEER evaluation

Initialize shared memory $\mathcal{M}_0 \leftarrow \emptyset$
for $t = 1, \ldots, T$ **do**
  $\mathcal{M}_t \leftarrow \mathcal{M}_{t-1}$
  for each robot i **do**
    Upload (o_t^i, t) to $\mathcal{M}_t$, retaining the FIFO capacity
  end for
  for each robot i **do**
    Retrieve context $\mathcal{C}_t^i$ from $\mathcal{M}_t$ using o_t^i
    Form $\mathbf{U}_t^i$ from SEER-World predictions and the projector
    Select $a_t^i \sim \pi_i(a \mid h_t^i, x^i, \mathbf{U}_t^i)$
    Execute a_t^i with the robot-specific controller
  end for
end for

configurations in Appendix A.4. We integrate SEER with StreamVLN [83] and Qwen3.5-4B [101], using the soft-token interface in Appendix A.2.

Retrieval settings. All main experiments use a FIFO pool of 1,024 observation records and a total context budget of $K = 8$ observations. Retrieval uses weighted sampling without replacement, with weights renormalized over the remaining records after each draw. The temporal and spatial coefficients are $\gamma_T = \gamma_S = 1$, and the weight-floor parameter is $\epsilon = 0.1$. These settings are shared across both downstream agents and all four paired-agent configurations. When cross-robot observations are unavailable, recent self-history supplies the remaining context.

Navigation metrics. Navigation Error (NE) is the final geodesic distance to the goal. Success Rate (SR) measures the fraction of episodes ending within 3 m of the goal, while Oracle Success Rate (OSR) uses the closest point along the executed trajectory. Success weighted by Path Length (SPL) additionally accounts for path efficiency, and normalized Dynamic Time Warping (nDTW) measures similarity to the reference trajectory. For paired episodes, ISR averages individual goal-reaching outcomes, and CSR counts episodes in which both robots succeed, as defined in the main text.

Sharing baselines. Table 4 compares no sharing, history sharing, action sharing, observation sharing, spatial-description sharing, and SEER using Qwen3.5-4B. The spatial-description baseline uses InternVL3-8B [119] as an additional expert. These comparisons evaluate different shared inputs within the same downstream navigation interface.

A.6 Real-World Deployment

Platforms and server. We deploy SEER on a wheeled AgileX LiMO Pro and a quadruped Unitree Go2 in indoor corridors and rooms. LiMO Pro is equipped with LiDAR and an Intel RealSense D435 camera [120]; Go2 provides an onboard wide-angle camera. SEER consumes egocentric RGB observations from both platforms. A remote workstation with an NVIDIA RTX 4090 runs the shared pool, retrieval, SEER-World, and downstream navigation models. Robot clients handle image capture and physical action execution, as shown in Figure 6.

Observation and action exchange. At each decision step, the client sends a JPEG-compressed RGB image in a multipart HTTP POST request. The server updates the pool, retrieves context, constructs SEER tokens, and evaluates the downstream agent for that robot. Both StreamVLN and Qwen3.5-4B use the discrete action interface $\{\textsc{Forward}, \textsc{Left}, \textsc{Right}, \textsc{Stop}\}$, encoded as $\{1, 2, 3, 0\}$. The server returns a JSON action list, which the client executes sequentially before requesting the next prediction. SSH is used to start or stop clients and reset episodes.

Robot-specific execution. High-level actions are mapped to timed velocity commands. A forward action commands linear speed v for d/v seconds; a turn commands yaw rate ω for θ/ω seconds; a stop action sets commanded velocities to zero. Go2 uses Unitree's SportClient API with (v_x, v_y, ω_z) commands. Its forward settings are $v = 0.3$ m/s and $d = 0.25$ m, and turns use $\omega = 0.5$ rad/s and $\theta = 15^\circ$, followed by a short settling delay. LiMO Pro uses its ROS 2 velocity-control interface with the same action-to-duration mapping and platform-specific speeds. Navigation actions are executed through open-loop velocity control without SLAM or closed-loop waypoint tracking.

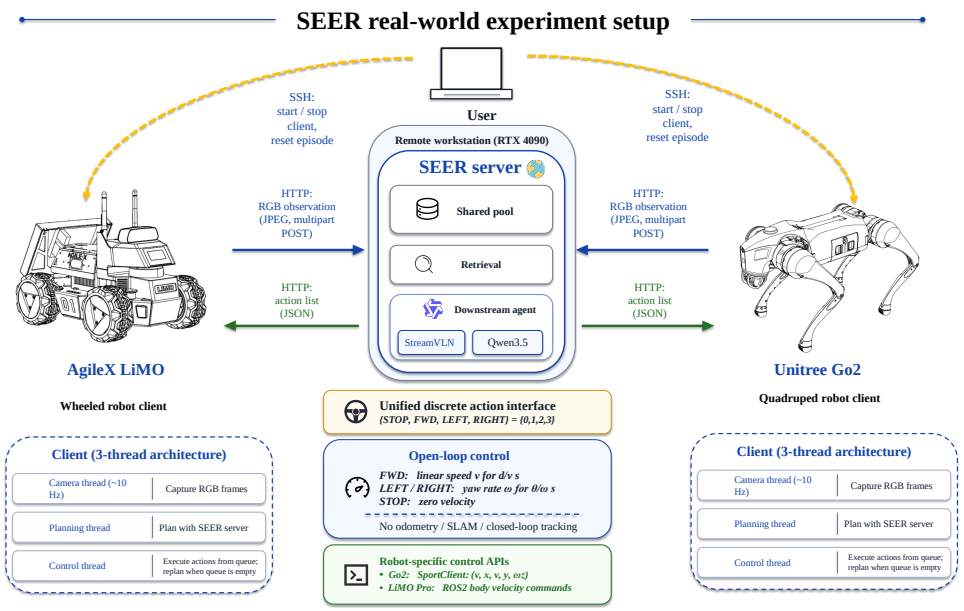


Figure 6: Real-world deployment of SEER. LiMO Pro and Go2 upload RGB observations to a shared server and execute the returned high-level actions through their platform-specific control interfaces.

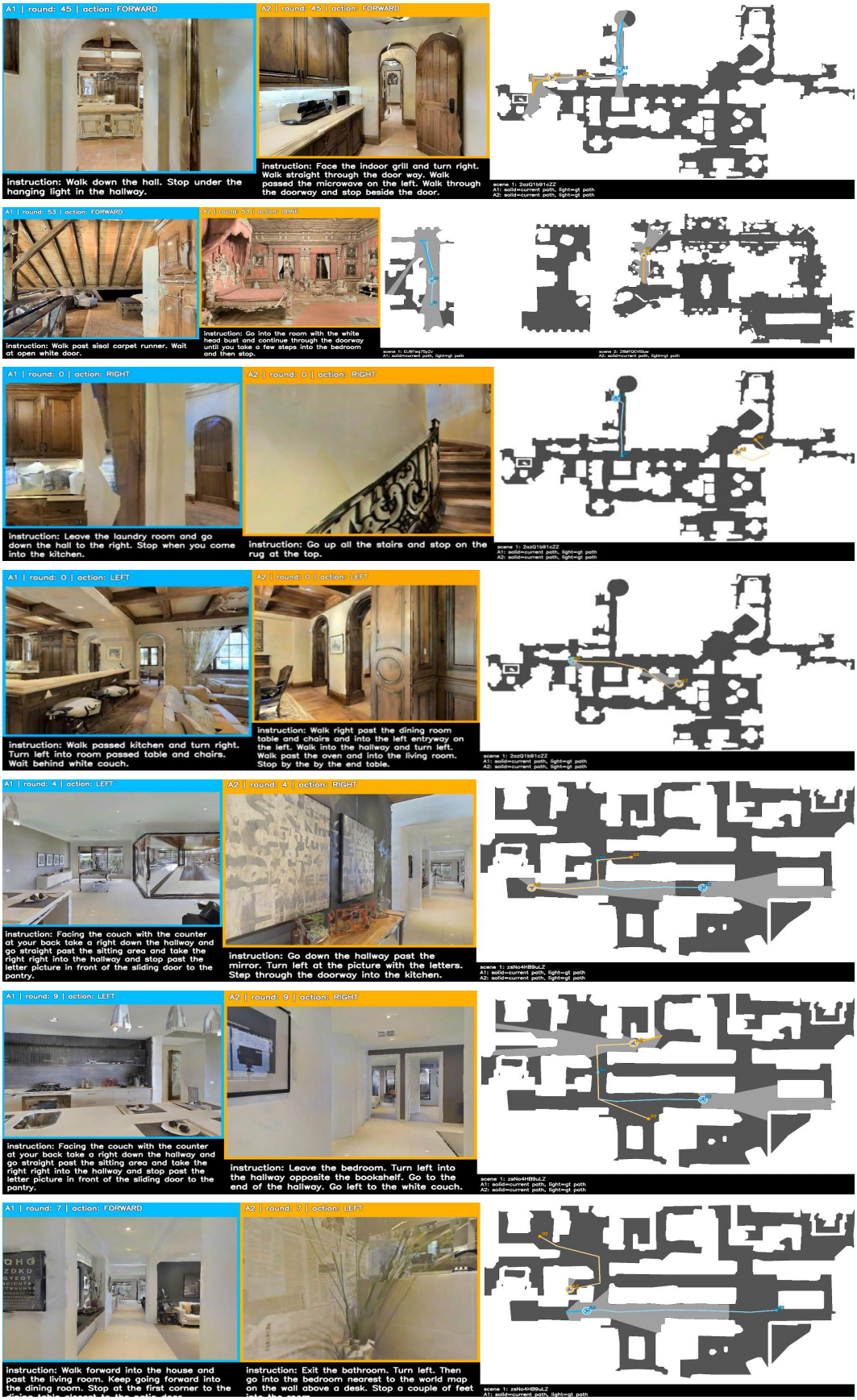
Client scheduling. Each client uses three threads. The camera thread updates the latest RGB frame at approximately 10 Hz; the planning thread sends the latest frame when a new prediction is requested; and the control thread executes the action queue and triggers replanning when the queue becomes empty. This deployment keeps the navigation interface consistent across wheeled and legged platforms while retaining their respective low-level controllers.

A.7 Additional Task-Level Analysis

Single-agent navigation. Table 1 reports both downstream agents with and without SEER on R2R-CE and RxR-CE. On R2R-CE, SR increases from 56.9% to 58.1% for StreamVLN and from 40.3% to 44.9% for Qwen3.5-4B. On RxR-CE, SR increases from 52.9% to 54.3% and from 39.7% to 42.4%, respectively. The corresponding SPL results also improve. These results show that shared-world conditioning remains useful when the context is formed from a single robot's observations.

Paired-agent navigation. Table 2 reports higher ISR and CSR with SEER in all four configurations. For example, in the same-scene intersecting setting, StreamVLN's CSR increases from 31.2% to 35.9%, while Qwen3.5-4B's CSR increases from 18.1% to 23.9%. The corresponding gains are 4.7 and 5.8 percentage points. Different-scene pairs provide a control with unrelated cross-robot observations, while intersecting and overlapped routes provide shared local evidence. ISR summarizes individual success across robots, and CSR captures the joint outcome of each pair.

Shared information and qualitative examples. In Table 4, SEER achieves higher ISR and CSR than the evaluated history-, action-, observation-, and spatial-description-sharing alternatives. Together with the inverse dynamics results in Table 3, these comparisons connect action-relevant latent structure with downstream navigation performance. The following visualizations show paired observations, language instructions, and executed trajectories across the evaluated spatial configurations.


A1 | round: 45 | action: FORWARD
A2 | round: 45 | action: FORWARD
instruction: Walk down the hall. Stop under the hanging light in the hallway.
instruction: Face the indoor grill and turn right. Walk straight through the door way. Walk passed the microwave on the left. Walk through the doorway and stop beside the door.
A1 | round: 53 | action: FORWARD
A2 | round: 53 | action: DONE
instruction: Walk past sisal carpet runner. Wait at open white door.
instruction: Go into the room with the white head bust and continue through the doorway until you take a few steps into the bedroom and then stop.
A1 | round: 0 | action: RIGHT
A2 | round: 0 | action: RIGHT
instruction: Leave the laundry room and go down the hall to the right. Stop when you come into the kitchen.
instruction: Go up all the stairs and stop on the rug at the top.
A1 | round: 0 | action: LEFT
A2 | round: 0 | action: LEFT
instruction: Walk passed kitchen and turn right. Turn left into room passed table and chairs. Wait behind white couch.
instruction: Walk right past the dining room table and chairs and into the left entryway on the left. Walk into the hallway and turn left. Walk past the oven and into the living room. Stop by the by the end table.
A1 | round: 4 | action: LEFT
A2 | round: 4 | action: RIGHT
instruction: Facing the couch with the counter at your back take a right down the hallway and go straight past the sitting area and take the right right into the hallway and stop past the letter picture in front of the sliding door to the pantry.
instruction: Go down the hallway past the mirror. Turn left at the picture with the letters. Step through the doorway into the kitchen.
A1 | round: 9 | action: LEFT
A2 | round: 9 | action: RIGHT
instruction: Facing the couch with the counter at your back take a right down the hallway and go straight past the sitting area and take the right right into the hallway and stop past the letter picture in front of the sliding door to the pantry.
instruction: Leave the bedroom. Turn left into the hallway opposite the bookshelf. Go to the end of the hallway. Go left to the white couch.
A1 | round: 7 | action: FORWARD
A2 | round: 7 | action: LEFT
instruction: Walk forward into the house and past the living room. Keep going forward into the dining room. Stop at the first corner to the
instruction: Exit the bathroom. Turn left. Then go into the bedroom nearest to the world map on the wall above a desk. Stop a couple of feet

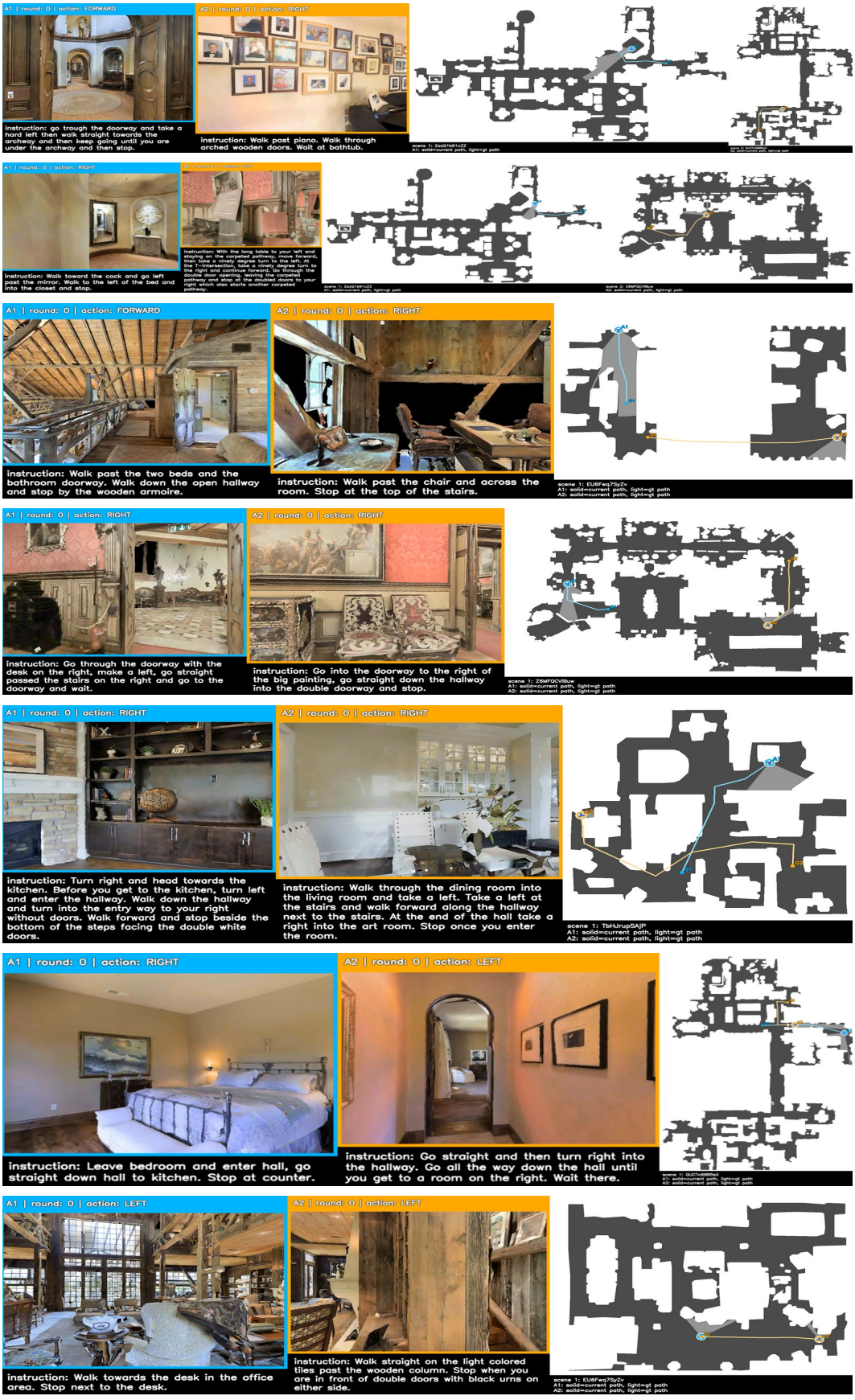


Figure 7: More Qualitative Analysis of our SEER.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main claim that multi-robot collaboration can emerge from *Shared Environmental Understanding* rather than directly sharing agent-centric traces. The claimed contributions match the proposed SEER framework, inverse dynamics analysis, downstream VLN evaluation, and comparisons against action-, history-, and centralized-policy-sharing alternatives.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations discussion covering the scope of evaluation, reliance on paired R2R-derived episodes, scalability to larger robot teams, sensitivity to retrieval quality, and the gap between simulated VLN evaluation and real-world deployment.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical theorems or proofs. The mathematical expressions are used to define the problem setting, SEER training objective, retrieval probability, downstream integration, and evaluation metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper and appendix describe the SEER architecture, training objectives, retrieval mechanism, downstream integration protocol, multi-robot episode construction, evaluation splits, metrics, baselines, and implementation details needed to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The submission includes an anonymized code and data release or will provide it upon publication. The release contains scripts for constructing paired R2R multi-robot episodes, training SEER, integrating SEER with downstream agents, and reproducing the main experiments.

   Justification: We provide the dataset and usage guidelines in the provided links.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the R2R train and val-unseen split usage, the four multi-robot evaluation subsets, model components, retrieval settings, downstream agents, optimization details, and hyperparameters. Additional implementation details are provided in the appendix.

   Justification: We provide the details in the paper and the benchmarking codes.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports variability estimates for the main experimental results, including repeated runs or bootstrap confidence intervals where applicable. The appendix describes how the reported error bars or confidence intervals are computed.

Justification: Our model is bound with LLMs. Due the experiment costs (API costs and GPU consumption), we cannot afford to run the experiments for multiple times.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The implementation appendix reports the compute resources used for SEER training, downstream agent training, inverse dynamics analysis, and evaluation, including GPU type, number of GPUs, approximate training time, and total compute.

   Justification: We have provided this information in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: To the best of our knowledge, the research conforms to the NeurIPS Code of Ethics. The work uses existing navigation datasets and simulated or controlled evaluation protocols, does not involve deception or new human-subject experiments, and discusses responsible deployment considerations.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses potential positive impacts such as more efficient and robust multi-robot navigation, as well as possible negative impacts such as unsafe deployment, privacy concerns from shared observations, and misuse in surveillance or autonomous tracking applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk pretrained language model, image generator, or scraped web dataset. The proposed paired episodes are derived from existing navigation datasets, and the release focuses on research code, derived metadata, and reproducible evaluation protocols.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original works for R2R, Matterport3D/Habitat-based navigation data, StreamVLN, Qwen2.5-VL, V-JEPA-style models, and other reused baselines or assets. The appendix lists the corresponding licenses or terms of use and states that the derived data construction respects these terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces derived multi-robot episode pairs constructed from existing R2R trajectories. The paper and appendix document the pair construction process, effective path definition, de-duplication rules, spatial categorization, split construction, and intended use.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not conduct new crowdsourcing experiments or new research with human subjects. It reuses existing navigation datasets and constructs derived paired episodes from them.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve new human-subject experiments or crowdsourcing, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper describes the use of LLM-based navigation agent as one downstream model for evaluating whether SEER can serve as a general environmental understanding layer. This use is part of the experimental validation and is not hidden as writing or editing assistance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.