MULTI-TIMESCALE VERIFICATION FOR TEST-TIME ADAPTATION OF SOCIAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-evolving agents have demonstrated the potential to improve through accumulated experience, particularly in software engineering, where executable feedback can evaluate proposed changes. For social agents in unknown environments, however, both perceived events and their behavioral explanations are uncertain. Repeated reflection can therefore reinforce errors rather than correct them. This raises a central question: how can social agents build on past interactions without compounding unreliable beliefs? We introduce the *Verified Social Evolution Loop* (VSEL), a framework for social agents to evolve through verification at multiple timescales in unseen continuous environments. Within episodes, a *Physical Consistency Verifier* checks perceived states against spatio-temporal constraints. The retained evidence is consolidated into a *decision map* and provided to the visual-language model (VLM) for event-triggered reasoning and evidence reassessment in a zero-shot manner. Across episodes, a *Hypothesis Verifier* evaluates behavioral hypotheses against accumulated support and counterevidence. Using only egocentric RGB-D observations and poses, VSEL sets new state-of-the-art (SOTA) results across benchmarks on the Habitat 3.0 platform for social agents. Particularly, in social navigation, VSEL achieves an episode success rate of 0.73, surpassing the strongest compared reinforcement-learning baseline by 70% in relative terms. Overall, our framework offers a practical route to social agents that accumulate and revise knowledge through verified experience.

1 INTRODUCTION

Social agents (Zhou et al., 2024; Kozlowski & Evans, 2025; Mou et al., 2026) must account for others' behavior when selecting and adapting their own actions (Dafoe et al., 2021). For robots sharing spaces and tasks with people, this capability is essential to move beyond isolated task execution toward effective assistance and cooperation (Puig et al., 2024; Chang et al., 2025). In an unfamiliar environment, such interaction requires the robot to reason under uncertainty at multiple scales, from interpreting the physical context and other agents' states to inferring higher-level behavioral regularities (Scofano et al., 2025; Jha et al., 2026), using only partial observations gathered during interaction (Gavryushin et al., 2026).

Existing approaches develop embodied social intelligence (Zhou et al., 2026b) along several complementary directions. Reinforcement-learning methods improve generalization by exposing agents to diverse partners (Strouse et al., 2021), behaviors (Szot et al., 2023), environments (Jha et al., 2025), or interaction histories (Ma et al., 2024a; Wang et al., 2026a) during training. Another line explicitly models other agents through beliefs (Zhu et al., 2021), latent strategies (Li et al., 2025), or executable behavioral hypotheses (Jha et al., 2026). Meanwhile, advances in vision–language foundation models (Qwen Team, 2026; Gemma Team et al., 2026) provide another route to social intelligence: semantic reasoning (Zhang et al., 2024), memory (Zhang et al., 2025c), communication (Zhou et al., 2026a). In addition, high-level social priors (Chen et al., 2026b) can be integrated with embodied perception and control to support long-horizon cooperation and socially aware behavior. However, accumulating interaction history does not establish its reliability. Video-language models (Rawal et al., 2025) can misdescribe events and temporal relations (Upadhyay et al., 2026). If retained without verification and reused across interactions, such errors may become embedded in persistent beliefs, biasing subsequent reasoning and action selection during test-time evolution. Recent work has begun to address related error-propagation problems (Wang et al., 2026d; Chen

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

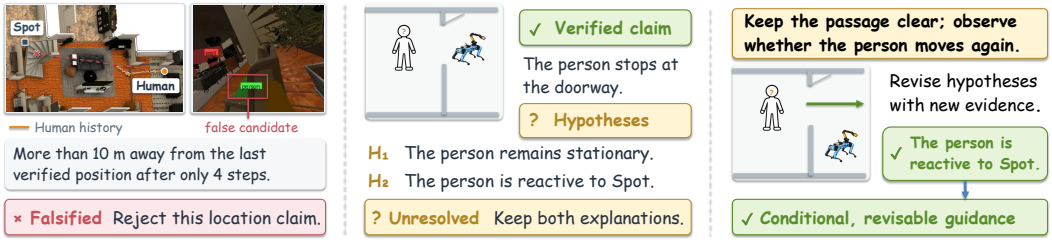


Figure 1: **Multi-timescale verification. Left:** A candidate claim is rejected as inconsistent with recent evidence. **Middle:** Temporally supported evidence that the person has stopped is compatible with competing behavioral predictions. **Right:** Subsequent observations after the passage clears can challenge these predictions, allowing revisable guidance for future decisions.

et al., 2026a). Unlike software-engineering settings, where explicit feedback such as returned values, execution errors, or test outcomes can directly support verification (Pan et al., 2025; Jin et al., 2026; Lee et al., 2026), embodied social agents at test time receive no comparably direct feedback about whether their perceptions are accurate, let alone whether their behavioral models of collaborators are supported by observed interactions. As a result, perceptual ambiguity, incomplete spatial knowledge, and uncertain behavioral interpretations can compound across repeated interactions. This motivates our central question: *how can embodied social agents continually refine their knowledge and strategies through test-time interaction without reinforcing errors arising from uncertain perception and inference?*

Evidence from cognitive science and neuroscience (Hasson et al., 2015; Kumaran et al., 2016; Baldassano et al., 2017) indicates that information is integrated across multiple temporal scales, from ongoing perceptual processing and event-based memory to the gradual acquisition of structured knowledge. Human physical judgments can be modeled through approximate internal simulation of scene dynamics (Battaglia et al., 2013), while social learning updates expectations through accumulated experience and prediction errors (Behrens et al., 2008). These findings motivate *verification at multiple timescales*: short-timescale physical checks constrain which perceived events become evidence, whereas longer-timescale hypothesis testing assesses whether accumulated evidence supports reusable behavioral regularities. As illustrated in Figure 1, multi-timescale verification helps prevent local perceptual errors from becoming persistent beliefs while allowing behavioral knowledge and strategies to be revised as new evidence emerges.

We therefore introduce the *Verified Social Evolution Loop* (VSEL), a framework for test-time evolution of embodied social agents through multi-timescale verification. Its central principle is that *physical consistency constrains admissible evidence, while accumulated evidence constrains the behavioral hypotheses that guide adaptation*. Within episodes, a *Physical Consistency Verifier* checks perceived states against spatial geometry, feasible motion, and visibility conditions derived from egocentric RGB-D and poses. These checks reject physical inconsistencies without certifying semantic correctness, and the agent can further reassess the evidence during event-triggered reasoning of the vision-language model (VLM). Across episodes, a *Hypothesis Verifier* evaluates hypotheses against accumulated episode evidence. Unexplained evidence is retained in an evidence pool and periodically revisited by the VLM to propose new hypotheses. Previously supported hypotheses are considered together with their counterevidence to guide subsequent strategies under task rules, closing a loop of test-time evolution.

We evaluate VSEL on two Habitat 3.0 (Puig et al., 2024) benchmarks, Social Navigation and Social Rearrangement. Using only egocentric RGB-D observations and poses, VSEL establishes new state-of-the-art results on both tasks. Specifically, VSEL improves episode success from 0.43 to 0.73 and further surpasses the oracle baseline in following rate on the Social Navigation benchmark. By leveraging VLM-based reasoning together with verified evidence, VSEL substantially reduces collisions compared with reinforcement-learning baselines. Ablation studies confirm that these gains are driven by multi-timescale verification and VSEL can adapt to collaborators in unseen environments through test-time evolution without parameter updates. Overall, VSEL demonstrates that verification can prevent uncertain interpretations from being accumulated as persistent beliefs, enabling embodied social agents to improve through experience.

2 Related Work

Embodied social agents. Recently, research (Puig et al., 2024; Chang et al., 2025; Chen et al., 2026b) on embodied social agents has gained momentum, with growing emphasis on interaction and cooperation in shared physical environments (Firoozi et al., 2025; Zhou et al., 2026b). Policy-learning approaches acquire socially aware navigation and coordination skills through task-oriented training (Szot et al., 2023; Chen et al., 2026b). To address the challenges of social navigation (Francis et al., 2025), where robots must account for human motion and maintain safe distances under partial observability, Proximity-aware navigation (Cancelli et al., 2023) uses simulator-derived supervision for risk estimation and proximity-compass auxiliary tasks to shape a reinforcement-learning policy. SDA (Scofano et al., 2025) learns a social-dynamics representation from privileged human trajectories during training and estimates it from the robot's state–action history at deployment. Other lines of research address explicit behavioral modeling (Jha et al., 2026), and cooperative planning (Zhang et al., 2024; Liu et al., 2025b; Zu et al., 2025; Zhang et al., 2025a) or history-conditioned adaptation (Ma et al., 2024a; Wang et al., 2026a; Zhou et al., 2026a). In this paper, instead, our method addresses a distinct bottleneck: whether uncertain egocentric experience justifies persistent knowledge revision during test-time evolution. To our knowledge, VSEL is the first multi-scale verification framework for embodied social agents to jointly refine perceptual state estimates and revise behavioral knowledge during test-time evolution in unseen environments.

Spatio-temporal memory for embodied agents. Spatio-temporal memory (Starzyk & He, 2009) supports embodied decision-making by retaining spatial context and tracking changes across observations. Existing approaches organize experience into scene graphs (Huang et al., 2023; Gu et al., 2024), retrievable textual or visual memories (Yang et al., 2025; Anwar et al., 2025), and dynamic scene representations (Kurenkov et al., 2023; Schmid et al., 2024; Liu et al., 2025c; Fan et al., 2025; Gorlo et al., 2026). Learned memory mechanisms further support history compression and task-conditioned recall (Hu et al., 2025; Gupta et al., 2025; Wang et al., 2026c). For VLM-based navigation, MapNav (Zhang et al., 2025b) replaces historical frames with annotated semantic maps, while HSGM (Li et al., 2026) combines geometric, semantic, and decision layers to support map-guided waypoint selection. Beyond storage and retrieval, embodied memory must account for perceptual errors and environmental changes that can introduce inconsistent or outdated information into persistent memory (Ko et al., 2025; Liu et al., 2025c). Unlike previous works, VSEL distinguishes the display of uncertain candidates from their admission as verified evidence. The Physical Consistency Verifier checks perceptual claims against spatiotemporal constraints, retaining unverified candidates while excluding falsified claims from the decision context.

Self-evolving agents. Self-evolving agents seek persistent improvement (Shao et al., 2026; Gao et al., 2026) by updating model parameters (Tian et al., 2025), contexts (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2025; Cheng et al., 2026; Zhang et al., 2026c), tools (Wang et al., 2024; Yang et al., 2026a), reusable skills (Zhang et al., 2026a), agent configurations and workflows (Zhang et al., 2026b; He et al., 2026; Wang et al., 2026b). In embodied settings, self-evolution has similarly been explored through the accumulation of executable skills (Wang et al., 2024), reinforcement fine-tuning from rewards (Tian et al., 2025), verification-guided refinement of reusable skill contracts (Yang et al., 2026b), and evolution of model-external skills and agent harnesses (Wang et al., 2026b). Recent work also explores experience-driven adaptation in embodied environments (Dai et al., 2026). A central challenge, however, is obtaining feedback that can verify the interpretations underlying such adaptation. While verification is easy to access in software-engineering settings (Pan et al., 2025; Jin et al., 2026; Lee et al., 2026), embodied social agents must assess perceived events and behavioral regularities from partial and potentially erroneous observations. Task outcomes alone do not establish whether these intermediate interpretations are correct (Puig et al., 2024; Munje et al., 2025; Rawal et al., 2025; Jha et al., 2026), making it difficult to determine which experiences justify persistent changes in beliefs and strategies. VSEL addresses this challenge through multi-timescale verification in dynamic continuous environments. The emphasis is therefore not merely on accumulating experience, but on assessing what it supports and revising the knowledge retained for future interactions.

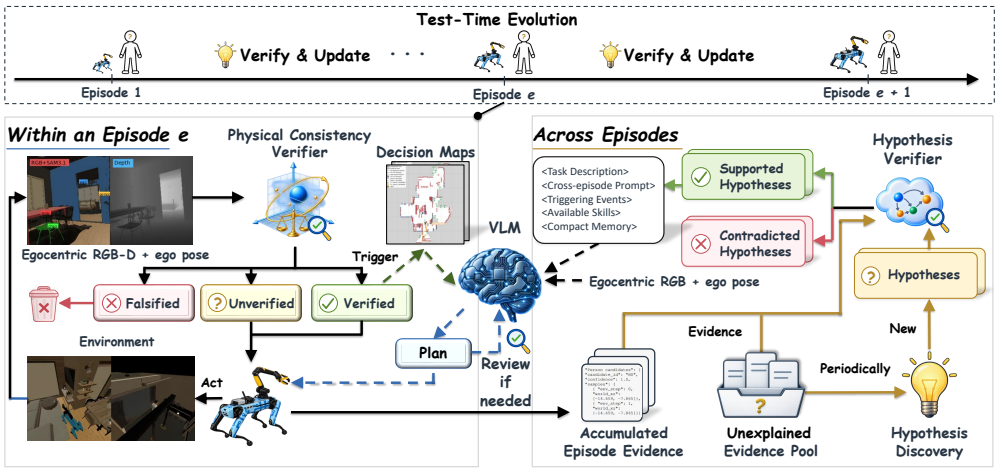


Figure 2: **Overview of the Verified Social Evolution Loop (VSEL).** Within an episode, the *Physical Consistency Verifier* checks RGB-D-derived states against physical constraints and filters inconsistent perceptual evidence. Across episodes, evidence provides the basis for hypothesis discovery and verification, with supporting and contradicting episode evidence retained separately for each behavioral hypothesis. Previously supported hypotheses are considered together with their counterevidence to guide subsequent interactions, forming a closed loop of test-time evolution. Dashed arrows denote VLM invocations triggered by semantic events.

3 Methodology

VSEL separates verification into three interacting timescales: step-wise verification of perceived states, event-triggered vision–language model (VLM) reasoning and reflection, and cross-episode verification of behavioral hypotheses. The distinction is between what the agent observes, what it decides to do, and what it retains as guidance for future interactions.

3.1 Problem Formulation

We consider partially observable interaction in unseen continuous environments for social agents (Puig et al., 2024). At step t of episode e, the robot receives an observation $o_{e,t} = (I_{e,t}, D_{e,t}, \xi_{e,t})$, comprising an egocentric RGB image $I_{e,t}$, aligned depth $D_{e,t}$, and a pose $\xi_{e,t}$. Just as human vision is susceptible to false-positive perception (Palmer & Clifford, 2020; Romagnano et al., 2024), machine perception is fallible, with segmentation models sometimes producing inaccurate masks (Carion et al., 2026). The environment map is unknown, and the states and behavioral regularities of other moving agents must be inferred from incomplete observations.

To study test-time evolution, we organize interaction into a stream of E episodes with the same initially unfamiliar collaborator, while scenes and initial conditions may vary. In episode $e \in \{1, \ldots, E\}$, the robot follows task specification T_e and selects actions according to $a_{e,t} \sim \pi_\theta(\cdot \mid \tau_{e,t}, T_e, \mathcal{K}^{<e})$, where $\tau_{e,t} = (o_{e,0}, a_{e,0}, \ldots, a_{e,t-1}, o_{e,t})$ is its observation–action history, $\mathcal{K}^{<e}$ is knowledge retained from earlier episodes, and θ denotes fixed pretrained model parameters. Episode-local spatial memory resets between episodes, whereas $\mathcal{K}^{<e}$ persists within each partner stream and resets between streams. The objective is to improve subsequent interaction by verifying uncertain perceptual evidence and revising behavioral knowledge, without parameter updates or privileged feedback.

3.2 Spatio-Temporal Evidence Verification per Step

Physical expectations as perceptual checks. A plausible detection is not necessarily a reliable description of the world. Before it is used, its consistency with the unfolding scene must be assessed. Human physical judgments (Battaglia et al., 2013; Bi et al., 2025) offer a useful functional motivation: they combine uncertain observations with expectations about object dynamics, rather

than relying on image appearance alone. Neuroimaging studies (Fischer et al., 2016; Liu et al., 2024) also identify responses associated with violations of physical expectations. Inspired by this principle, we propose the *Physical Consistency Verifier* (PCV), which provides a fast, executable check of perceptual claims before they become evidence for planning or behavioral learning.

Learning an executable verifier. We construct the PCV through feedback-guided program synthesis using a VLM. Let $\mathcal{D}_{\text{train}}$ denote the training data, $\mathcal{P}_{\text{PCV}}$ the prompt to construct the PCV, and $\mathcal{F}_{\text{train}}$ the feedback used to refine the verifier:

$$\text{PCV} \leftarrow \text{VLM}(\mathcal{P}_{\text{PCV}}, \mathcal{D}_{\text{train}}, \mathcal{F}_{\text{train}}). \tag{1}$$

VLM produces the executable consistency rules, not a policy for how another agent should behave. This formulation follows the broader principle of using evaluation feedback to improve LLM-generated programs (Ma et al., 2024b; Romera-Paredes et al., 2024). The PCV is executed at each observation step and frozen during test time, requiring no additional VLM calls.

From physical checks to decision context. Given a candidate claim $\hat{c}^j_{e,t}$ proposed by the perception model and the observation–action history $\tau_{e,t}$, PCV returns

$$\sigma^j_{e,t} = \text{PCV}(\hat{c}^j_{e,t}, \tau_{e,t}) \in \{\textsc{Verified}, \textsc{Unverified}, \textsc{Falsified}\}. \tag{2}$$

A VERIFIED claim has observational support and passes the applicable physical checks, so it is promoted to evidence for decision-making. Each admitted claim retains its verification time and provenance. For time-varying attributes, a newly verified claim may supersede an earlier accepted claim referring to the same grounded referent and attribute. Insufficient support, however, does not imply that a claim is false: an occluded target, for example, cannot be declared absent. Such UNVERIFIED claims remain explicitly uncertain inputs to VLM reasoning and may motivate further observation (Bhatt et al., 2025). Claims that are inconsistent with the accumulated evidence are marked FALSIFIED and discarded from the decision context.

3.3 EVENT-TRIGGERED REASONING AND EVIDENCE-GROUNDED REVIEW

Decision map. An egocentric view alone does not reveal how a newly observed entity relates to previously explored space or earlier observations. We therefore consolidate observation-derived spatial memory and PCV-assessed claims into a single annotated *decision-map image*:

$$\mathcal{I}^{\text{map}}_{e,t} = \text{Render}(\mathcal{M}_{e,t}, \mathcal{C}_{e,t}, \hat{\mathcal{G}}_{e,t}), \tag{3}$$

where $\mathcal{M}_{e,t}$ encodes the current spatial memory derived from observation history, $\mathcal{C}_{e,t}$ contains perceptual claims excluding those marked as FALSIFIED, and $\hat{\mathcal{G}}_{e,t}$ denotes candidate goals. The map provides a unified spatiotemporal reference for exploration, interaction, and decision reassessment, combining observed and unknown regions, task-relevant entities, and retained motion evidence. Unlike HSGM, which lacks an explicit representation of moving targets (Li et al., 2026), our decision map integrates spatial context with candidate historical evidence and predicted future states, enabling the VLM to reason about reliable states and adapt decisions in dynamic environments.

VLM Reasoning triggered by events. Visual perception in embodied navigation can operate at a substantially higher frequency than high-level VLM reasoning (Wei et al., 2026; Ding et al., 2026; Zhong et al., 2026). This raises a practical question: *when should new observations trigger the next decision?* Beyond initialization, the VLM should be invoked when target visibility changes, task-relevant scene semantics change, or the decision made by the VLM is completed. VSEL additionally distinguishes the event that requests a decision from the perceptual feedback that triggers its review. Given the current RGB image and pose, decision map, task rules, strategy context, and compact memory, collectively denoted by $X_{e,t}$, the frozen VLM produces both a proposed decision and task-relevant entity queries: $(\widetilde{g}_{e,t}, \mathcal{Q}_{e,t}) = \text{Propose}_\theta(X_{e,t})$. The queries specify what to check in the scene, while the decision specifies how the agent proposes to respond.

Review triggered by positive grounding. The entity queries are submitted to an open-vocabulary segmentation model, such as SAM 3 (Carion et al., 2026), to re-examine the current image. A positive detection of a queried task-relevant entity triggers a second VLM call. Before this review,

the resulting candidate states are checked by PCV and the decision map is refreshed; the updated context $X^+_{e,t}$ includes the checked grounding results and their evidence status. Let $r_{e,t} = 1$ when segmentation positively matches at least one queried entity, and $r_{e,t} = 0$ otherwise. The resulting decision is

$$\bar{g}_{e,t} = \begin{cases} \mathrm{Review}_\theta(X^+_{e,t}, \widetilde{g}_{e,t}), & r_{e,t} = 1, \\ \widetilde{g}_{e,t}, & r_{e,t} = 0. \end{cases} \tag{4}$$

This review reconsiders that action using the updated context assessed by the PCV. The loop therefore couples a decision proposal to targeted perception and evidence-grounded revision, rather than merely repeating reasoning on an unchanged context.

3.4 Hypothesis Verification across Episodes

From observations to testable explanations. Physically consistent observations do not uniquely determine another agent's behavioral regularities. For example, an agent standing near a doorway may always remain stationary or pause only while the passage is occupied; the same observation is compatible with different predictions about what happens next. Such ambiguity is central to inference about other agents (Baker et al., 2017). We represent competing explanations as hypotheses (Jha et al., 2026), rather than asserting hidden intentions. Verification evaluates these hypotheses against experience instead of accepting plausibility as correctness (Qiu et al., 2024).

Let $\mathcal{C}_e$ denote the record derived from observations of episode e, summarizing estimated trajectories, decision maps, and the robot's actions. Let $\mathcal{H}_e$ contain the current behavioral hypotheses available in the episode. Each hypothesis $h_i \in \mathcal{H}_e$ specifies an observable prediction and the conditions under which it applies. The frozen *Hypothesis Verifier* (HV) evaluates

$$v_{e,i} = \mathrm{HV}(h_i, \mathcal{C}_e) \in \{+, -, ?\}, \tag{5}$$

where $+$, $-$, and $?$ denote support, contradiction, and an unresolved judgment, respectively, and θ denotes the frozen VLM parameters. An occluded outcome or an unobserved precondition leaves the hypothesis unresolved. Each episode receives one verdict per hypothesis, and unresolved or irreconcilably mixed evidence enters neither set. These records track evidential support, not calibrated probabilities.

Hypothesis Discovery from unexplained experience. Verification can assess existing explanations but cannot supply missing ones. We therefore retain evidence $\mathcal{E}_e$ of the episode e in an *unexplained evidence pool* whenever it supports none of the current hypotheses. Every B completed episodes, the VLM revisits the pool alongside existing hypotheses to discover additional conditional regularities (Liu et al., 2025a), rather than forcing unexplained experience into a fixed hypothesis set. Hypothesis discovery is skipped when the pool is empty. Each newly proposed hypothesis is initialized without support and immediately passed to the *Hypothesis Verifier*. Thus, new hypotheses acquire evidential support through explicit verification rather than being accepted solely because they are generated. After these hypotheses are judged, strategy update follows.

Test-time strategy update. Behavioral hypotheses must be translated into guidance rather than simply appended to the decision context. We view this update as *task-conditioned compression*: the VLM combines supported hypotheses with episode evidence to produce a concise strategy for the next interaction, building on experience-based knowledge extraction and context adaptation (Zhao et al., 2024; Zhang et al., 2026c). Let $\mathcal{H}_{e+1}$ denote the hypothesis set available before episode $e+1$, after processing all completed episodes through e. We consider every hypothesis that has received at least one supporting verdict:

$$\mathcal{H}^+_{e+1} = \{h_i \in \mathcal{H}_{e+1} \mid \exists e' \in \{1, \ldots, e\},\ v_{e',i} = +\}\,. \tag{6}$$

Each hypothesis in $\mathcal{H}^+_{e+1}$ is accompanied by its applicability conditions, supporting and contradicting episode verdicts, as well as the corresponding supporting and contradicting episode indices. The VLM synthesizes the strategy p_{e+1} for episode $e+1$ as follows:

$$p_{e+1} = \mathrm{VLM}\big(\mathcal{H}^+_{e+1}; \mathcal{T}\big), \tag{7}$$

where $\mathcal{T}$ specifies the task prompt and its rules. When no applicable guidance is justified, the agent uses its base task strategy. The prompt p_{e+1} remains fixed within the next episode, while current evidence continues to ground individual decisions.

Table 1: Main results for Social Navigation. Env Map indicates whether the environment map is available during testing. hGPS indicates whether ground-truth human coordinates are available during testing. hDet denotes an oracle human detector. GT denotes ground-truth information, and NA indicates that no privileged information is used.

Settings	Method	Map	hGPS	hDet	ES ↑	F ↑	CR ↓	SPS ↑	S↑
Oracle	Heuristic Expert (Puig et al., 2024)	GT	GT	GT	-	0.51	0.52	**0.97**	**1.00**
RL	DDPPO (Puig et al., 2024)	NA	NA	GT	0.40	0.29	0.48	0.34	0.76
	Proximity (Cancelli et al., 2023)	NA	NA	GT	0.41	0.37	0.58	0.41	0.85
	SDA-S2 (Scofano et al., 2025)	NA	NA	GT	0.43	0.39	0.57	0.45	0.91
Zero-shot	HSGM (Li et al., 2026)	NA	NA	NA	0.00	0.03	0.12	0.23	0.47
	ET-Baseline	NA	NA	NA	0.32	0.26	0.33	0.35	0.71
	VSEL (Ours)	NA	NA	NA	**0.73**	**0.56**	**0.10**	0.47	0.89

4 Experiments

We evaluate the effectiveness, adaptability, and generality of VSEL in this section. The benchmarks, evaluation metrics, and baselines are introduced in Section 4.1. We then evaluate VSEL on Social Navigation and Social Rearrangement in Section 4.2. Finally, our ablation studies in Section 4.3 examine within-episode verification, cross-episode hypothesis verification, and different VLM backbones to analyze the key components and generality of VSEL.

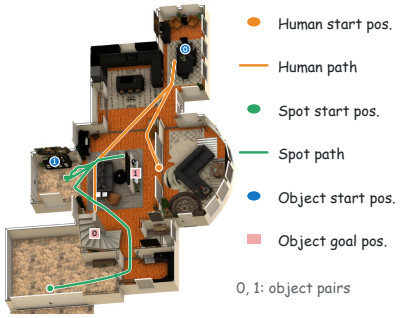
4.1 Experiment Setup

We evaluate VSEL on Habitat 3.0 (Puig et al., 2024), a simulation platform for embodied social interaction in domestic environments, using its Social Navigation and Social Rearrangement benchmarks. The platform provides a diverse library of humanoid avatars spanning different gender representations, body shapes, and appearances, together with planner-based and learned policies that generate various behaviors. Following their evaluation protocols (Puig et al., 2024), we report our results using the simulated Boston Dynamics Spot robot in unseen test scenes.

Social Navigation. The robot must locate a moving humanoid in an unfamiliar indoor environment and follow it at a distance of 1–2 m while facing it. VSEL uses only egocentric RGB-D observations and robot poses, without access to ground-truth humanoid positions or prior geometric or semantic maps. Following Scofano et al. (2025), we report Episode Success (ES), the fraction of episodes in which the robot finds the humanoid and completes the required number of valid following steps. We also report Following Rate (F), and Collision Rate (CR), Finding Success Weighted by Path Steps (SPS), and Finding Success Rate (S) with formal definitions provided in Appendix E.1.

Social Rearrangement. The robot and a humanoid jointly relocate target objects to designated locations (Puig et al., 2024). At test time, the robot is paired with collaborators exhibiting diverse behavioral patterns, challenging it to adjust object selection and action timing while avoiding conflicting or redundant actions. We report *Success Rate* (SR), the fraction of episodes in which all target objects reach their goals, together with *Relative Efficiency* (RE), *Ratio of Completion* (RC), and *Collision Rate* (CR). Appendix E.2 details how these metrics are computed.

Baselines. We choose competent RL and zero-shot baselines for the two benchmarks. For social navigation, the reference baselines comprise a map-privileged *Heuristic Expert* based on shortest-path planning and *DD-PPO* (Puig et al., 2024), a recurrent end-to-end RL policy; *Proximity* (Cancelli et al., 2023), which augments RL with proximity-aware auxiliary tasks; and *SDA-S2* (Scofano et al., 2025), which infers social dynamics from robot histories after training with privileged human trajectories. For social rearrangement, we include *Learn-Single* (Puig et al., 2024), jointly trained with one learned humanoid; *Plan-Pop*$_k$, trained with populations of $k \in \{1, 2, 3, 4\}$ planner-controlled humanoids; and *Learn-Pop* (Puig et al., 2024), trained with a population of learned humanoids (Jaderberg et al., 2019). In addition to the RL baselines, we include *ET-Baseline*, an ablation baseline that retains VSEL's event-triggered VLM invocation mechanism but disables multi-timescale verification and the decision-map input, assessing their combined contribution. For social navigation, we additionally include *HSGM* (Li et al., 2026), which combines hierarchical semantic–



Settings	Method	SR↑	RE↑	RC	CR↓
RL	Learn-Single	56.21	108.77	0.46	0.33
	Plan-Pop$_1$	63.69	116.59	0.44	0.28
	Plan-Pop$_2$	69.63	107.77	**0.50**	0.20
	Plan-Pop$_3$	72.13	107.16	0.51	0.22
	Plan-Pop$_4$	72.02	108.91	0.52	0.21
	Learn-Pop	48.70	100.10	0.45	0.35
Zero-shot	ET-Baseline	60.98	105.95	0.42	0.11
	VSEL (Ours)	**72.67**	**122.33**	0.47	**0.07**

Figure 3: Overview of Social Rearrangement (**left**). Results with 4 unseen planner-based collaborators available in the official Habitat 3.0 release (Puig et al., 2024) (**right**).

(a) Verification within episodes.

Method	ES ↑	F ↑	CR ↓	SPS ↑	S ↑
VSEL (Ours)	**0.73**	**0.56**	**0.10**	0.47	**0.89**
– decision map	0.71	0.55	0.12	**0.48**	**0.89**
– PCV	0.25	0.21	0.33	0.34	0.72
– VLM review	0.69	0.53	0.12	**0.48**	0.88

(b) Different VLM backbones.

VLM backbone	Method	ES ↑	F ↑	CR ↓	SPS ↑	S ↑
`gemma-4-12B-it`	ET-Baseline	0.54	0.42	0.17	0.51	0.84
	VSEL	**0.68**	**0.52**	**0.08**	**0.56**	**0.92**
`Cosmos-Reason2-8B`	ET-Baseline	0.33	0.28	0.29	0.39	0.71
	VSEL	**0.57**	**0.48**	**0.10**	**0.50**	**0.89**

Table 2: Ablation studies on Social Navigation.

geometric mapping with VLM waypoint selection and classical path planning. Implementation details of the zero-shot VLM baselines are provided in Appendix E.3. Despite the differences in inputs summarized in Table 1, all evaluated methods follow the official evaluation protocol (Puig et al., 2024) for each benchmark and are evaluated on the same test split of unseen scenes.

4.2 Main Results

Social Navigation. As shown in Table 1, using only egocentric RGB-D observations and robot poses, VSEL improves the SOTA episode success from 0.43 to 0.73. Compared with ET-Baseline, VSEL achieves a further 128% relative improvement, demonstrating the effectiveness of multi-timescale verification. Notably, VSEL achieves a higher following rate and a lower collision rate than even the heuristic expert. Its remaining gap in S and SPS is mainly due to the difficulty of locating the person without ground-truth human coordinates or an oracle detector, which can require extensive exploration or exceed the time budget. In contrast, HSGM struggles to sustain following since its decisions rely on outdated and unverified target information while its low collision rate reflects limited proximity to people rather than sustained following.

Social Rearrangement. As shown in Figure 3, VSEL substantially improves task success on social rearrangement while consistently reducing collisions, outperforming the existing baselines in success rate, relative efficiency, and collision rate. These gains indicate that VSEL can effectively coordinate with previously unseen collaborators in unfamiliar environments. In particular, VSEL leverages cross-episode interaction evidence to infer recurring collaborator behaviors. We observe that VSEL adapts its object selection toward complementary targets when accumulated evidence reveals a consistent target preference of the human collaborator.

4.3 Ablation Studies

Verification within episodes. Removing the Physical Consistency Verifier causes the largest performance drop among the evaluated components, as illustrated in Table 2. Without physical consistency checks, erroneous perceptual claims can persist in the decision context, potentially destabilizing subsequent VLM decisions. By comparison, removing VLM review has a much smaller effect, suggesting that maintaining reliable perceptual context is more consequential than additional high-level reassessment.

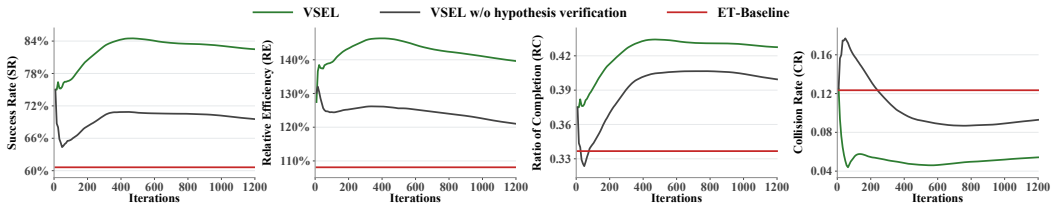


Figure 4: Ablation study of hypothesis verification across episodes on Social Rearrangement.

Different VLM backbones. We evaluate VSEL with `gemma-4-12B-it` (Google, 2026) and `Cosmos-Reason2-8B` (NVIDIA, 2026), while keeping model weights frozen. As shown in Table 2b, VSEL improves episode success over ET-Baseline by 26% and 73% with the two backbones, respectively, while also improving all other reported metrics. These consistent gains support VSEL's applicability across different VLM backbones.

Hypothesis Verification across episodes. Following the official evaluation protocol (Puig et al., 2024), we evaluate 1,200 episodes with a fixed human collaborator in unseen scenes. We set $B = 8$, resulting in 150 hypothesis-discovery iterations, while the collaboration strategy is updated after every episode. As shown in Figure 4, VSEL progressively improves its strategy and consistently outperforms both ET-Baseline and the variant without hypothesis verification. Rather than forcing an interpretation of ambiguous evidence, the Hypothesis Verifier retains unresolved cases and revises behavioral hypotheses as the additional episode evidence accumulates. Repeated interaction with the same collaborator across unseen scenes enables VSEL to identify behavioral tendencies that persist beyond individual environments and progressively adapt its collaboration strategy accordingly.

5 Conclusion and Limitations

VSEL demonstrates that embodied social agents can improve through verified experience without relying on task-specific policy optimization. After offline training of the Physical Consistency Verifier, the agent operates using only egocentric RGB-D observations and robot poses. Rather than adapting model weights, VSEL improves its behavior through physical verification, event-triggered reasoning, and cross-episode hypothesis verification, allowing accumulated experience to inform subsequent decisions. This approach establishes new state-of-the-art results on both Habitat 3.0 benchmarks. Specifically, on Social Navigation, VSEL increases episode success from 0.43 to 0.73 over the strongest compared RL baseline and achieves a higher following rate than the privileged heuristic expert. These results are encouraging: without training a new end-to-end policy or updating model parameters at test time, an embodied agent can become a more effective collaborator by continually determining which aspects of its experience should be retained, revised, or acted upon.

Limitations. Physical consistency is not a guarantee of semantic correctness. Our current Physical Consistency Verifier primarily assesses kinematic regularities that can be inferred from visual observations and pose histories, such as spatial continuity, feasible displacement, and temporal consistency. It does not explicitly reason about dynamics, including acceleration, contact forces, inertia, friction, or the physical consequences of interactions. Extending verification from kinematic plausibility to dynamical consistency is therefore an important challenge, particularly for embodied tasks involving object manipulation, contact-rich interaction, or rapidly changing motion. More broadly, plausible misdetections, occlusion, and uncertainty in associating observations across time can still produce incomplete or misleading evidence. Hypothesis verification and strategy compression also remain fallible, and accumulated support may become stale when other agents change their behavior. In addition, verifier rules synthesized from training feedback and the available execution skills may require revision under new sensing conditions, environments, or embodiments. These limitations suggest several directions for future work, including learning transferable dynamics-aware verification rules, integrating uncertainty over latent physical states, and actively gathering observations that discriminate between competing physical or behavioral hypotheses. Nevertheless, VSEL represents an important step toward embodied social agents that improve through experience while grounding persistent behavioral knowledge and subsequent decisions in verifiable evidence.

AI USE STATEMENT

In this work, we used generative AI tools for general-purpose writing and finding related work. We have not used generative AI tools to draft sections of the paper or prove mathematical claims. Generating synthetic datasets are not applicable to this work. Additionally, we used generative AI tools for research execution. For example, GPT-5.6 Sol was used to build the Physical Consistency Verifier. We have reviewed all AI-assisted work. As detailed in Appendix C, We have checked executable consistency rules automatically generated by AI. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We document the algorithms, interfaces, and inference settings used in VSEL to facilitate reproducibility. Appendix A provides the notation and pseudocode for episode execution, event-triggered proposal and review, and cross-episode hypothesis verification and discovery. Appendix B specifies the shared decision interface and evidence-dependent execution conditions. Appendix C describes the feedback-guided construction of the Physical Consistency Verifier and its executable rule families. Appendix D provides reasoning templates for online decisions, hypothesis verification, and strategy synthesis. Metric definitions and VLM baseline adaptations are described in Appendix E, while Appendix F documents model configurations. Appendix G records VLM-call accounting and inference cost. These descriptions distinguish offline verifier construction from test-time updates to evidence, hypotheses, and strategies with frozen model parameters. We will release the implementation, complete prompts, configuration files, and evaluation scripts to support replication.

REFERENCES

Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. ReMEmbR: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

Chris L. Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B. Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1: 0064, 2017. doi: 10.1038/s41562-017-0064.

Christopher Baldassano, Janice Chen, Asieh Zadbood, Jonathan W. Pillow, Uri Hasson, and Kenneth A. Norman. Discovering event structure in continuous narrative perception and memory. *Neuron*, 95(3):709–721.e5, 2017. doi: 10.1016/j.neuron.2017.06.041.

Peter W. Battaglia, Jessica B. Hamrick, and Joshua B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013. doi: 10.1073/pnas.1306572110.

Timothy E. J. Behrens, Laurence T. Hunt, Mark W. Woolrich, and Matthew F. S. Rushworth. Associative learning of social value. *Nature*, 456:245–249, 2008. doi: 10.1038/nature07538.

Neel P Bhatt, Yunhao Yang, Rohan Siva, Daniel Milan, Ufuk Topcu, and Zhangyang Wang. Know where you're uncertain when planning with multimodal foundation models: A formal framework. *Proceedings of Machine Learning and Systems*, 7, 2025.

Wenyan Bi, Aalap D. Shah, Kimberly W. Wong, Brian J. Scholl, and Ilker Yildirim. Computational models reveal that intuitive physics underlies visual processing of soft objects. *Nature Communications*, 16:6303, 2025. doi: 10.1038/s41467-025-61458-x.

Enrico Cancelli, Tommaso Campari, Luciano Serafini, Angel X. Chang, and Lamberto Ballan. Exploiting proximity-aware tasks for embodied social navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10923–10933, 2023.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou,

Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, pp. 138846–138923, 2026.

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.

Peixin Chen, Guoxi Zhang, Jianwei Ma, and Qing Li. Hypothesis graph refinement: Hypothesis-driven exploration with cascade error correction for embodied navigation. In *European Conference on Computer Vision*, 2026a.

Ziyi Chen, Yingnan Guo, Zedong Chu, Minghua Luo, Yanfen Shen, Mingchao Sun, Junjun Hu, Shichao Xie, Yang Kuan, Pei Shi, Zhining Gu, Lu Liu, Honglin Han, Xiaolong Wu, Mu Xu, and Yu Zhang. Socialnav: Training human-inspired foundation model for socially-aware embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28796–28806, 2026b.

Zihao Cheng, Zeming Liu, Yingyu Shan, Xinyi Wang, Xiangrong Zhu, Yunpu Ma, Hongru Wang, Yuhang Guo, Wei Lin, and Yunhong Wang. Mem2evolve: Towards self-evolving agents via co-evolutionary capability expansion and experience distillation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20784–20831, 2026.

Allan Dafoe, Yoram Bachrach, Gillian Hadfield, Eric Horvitz, Kate Larson, and Thore Graepel. Cooperative ai: machines must learn to find common ground. *Nature*, 593(7857):33–36, 2021.

Guangzhao Dai, Shuo Wang, Zihan Wang, Guo-Sen Xie, Yang Yang, Jinshan Pan, Qianru Sun, and Xiangbo Shu. History to future: Evolving agent with experience and thought for zero-shot vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15177–15187, 2026.

Xin Ding, Jianyu Wei, Yifan Yang, Shiqi Jiang, Qianxi Zhang, Hao Wu, Fucheng Jia, Liang Mi, Yuxuan Yan, Weijun Wang, Yunxin Liu, Zhibo Chen, and Ting Cao. AdaNav: Adaptive reasoning with uncertainty for vision-language navigation. In *Proceedings of the Forty-Third International Conference on Machine Learning*, 2026.

Yue Fan, Xiaojian Ma, Rongpeng Su, Jun Guo, Rujie Wu, Xi Chen, and Qing Li. Embodied VideoAgent: Persistent memory from egocentric videos and embodied sensors enables dynamic scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, Brian Ichter, Danny Driess, Jiajun Wu, Cewu Lu, and Mac Schwager. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, 44(5):701–739, 2025. doi: 10.1177/02783649241281508.

Jason Fischer, John G. Mikhael, Joshua B. Tenenbaum, and Nancy Kanwisher. Functional neuroanatomy of intuitive physical inference. *Proceedings of the National Academy of Sciences*, 113 (34):E5072–E5081, 2016. doi: 10.1073/pnas.1610344113.

Anthony Francis, Claudia Pérez-d'Arpino, Chengshu Li, Fei Xia, Alexandre Alahi, Rachid Alami, Aniket Bera, Abhijat Biswas, Joydeep Biswas, Rohan Chandra, et al. Principles and guidelines for evaluating social robot navigation algorithms. *ACM Transactions on Human-Robot Interaction*, 14(2):1–65, 2025.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Qihan Ren, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Transactions on Machine Learning Research*, 2026.

Alexey Gavryushin, Dingxi Zhang, Zhao Huang, Alexandros Delitzas, Jiaqi Chen, Ben Ellis, Cedric Zöllner, Manthan Patel, Manuel Kaufmann, Marc Pollefeys, and Xi Wang. CoMind: Understanding collaborative human activity from multiple minds and views. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2026.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Google. Gemma-4-12B-it model card. Official model card, 2026. URL https://huggingface.co/google/gemma-4-12B-it.

Nicolas Gorlo, Lukas Schmid, and Luca Carlone. Describe anything anywhere at any moment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35002–35013, 2026.

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024.

Gunshi Gupta, Karmesh Yadav, Zsolt Kira, Yarin Gal, and Rahaf Aljundi. Memo: Training memory-efficient embodied agents with reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Uri Hasson, Janice Chen, and Christopher J. Honey. Hierarchical process memory: Memory as an integral component of information processing. *Trends in Cognitive Sciences*, 19(6):304–313, 2015. doi: 10.1016/j.tics.2015.04.006.

Yufei He, Juncheng Liu, Yue Liu, Yibo Li, Tri Cao, Zhiyuan Hu, Xinxing Xu, and Bryan Hooi. Evotest: Evolutionary test-time learning for self-improving agentic systems. In *International Conference on Learning Representations*, volume 2026, pp. 104166–104202, 2026.

Wenbo Hu, Yining Hong, Yanjun Wang, Leison Gao, Zibu Wei, Xingcheng Yao, Nanyun Peng, Yonatan Bitton, Idan Szpektor, and Kai-Wei Chang. 3DLLM-Mem: Long-term spatial-temporal memory for embodied 3D large language model. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2278.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation*, pp. 8277–8283. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10160969.

Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

Kunal Jha, Wilka Carvalho, Yancheng Liang, Simon Shaolei Du, Max Kleiman-Weiner, and Natasha Jaques. Cross-environment cooperation enables zero-shot multi-agent coordination. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27198–27220. PMLR, 2025.

Kunal Jha, Aydan Huang, Eric Ye, Natasha Jaques, and Max Kleiman-Weiner. Modeling others' minds as code. In *International Conference on Learning Representations*, 2026.

Yiyang Jin, Kunzhao Xu, Hang Li, Xueting Han, Yanmin Zhou, Cheng Li, and Jing Bai. ReVeal: Self-evolving code agents via reliable self-verification. In *International Conference on Learning Representations*, 2026.

Po-Chen Ko, Hung-Ting Su, CY Chen, Jia-Fong Yeh, Min Sun, and Winston H. Hsu. Context-aware replanning with pre-explored semantic map for object navigation. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4253–4267. PMLR, 2025.

Austin C Kozlowski and James Evans. Simulating subjects: The promise and peril of artificial intelligence stand-ins for social agents and interactions. *Sociological Methods & Research*, 54 (3):1017–1073, 2025.

Dharshan Kumaran, Demis Hassabis, and James L McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20 (7):512–534, 2016.

Andrey Kurenkov, Michael Lingelbach, Tanmay Agarwal, Emily Jin, Chengshu Li, Ruohan Zhang, Li Fei-Fei, Jiajun Wu, Silvio Savarese, and Roberto Martín-Martín. Modeling dynamic environments with scene graph memory. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17976–17993. PMLR, 2023.

Dongjun Lee, Changho Hwang, and Kimin Lee. Learning to generate unit test via adversarial reinforcement learning. In *International Conference on Learning Representations*, 2026.

Benjamin Li, Shuyang Shi, Lucia Romero, Huao Li, Yaqi Xie, Woojun Kim, Stefanos Nikolaidis, Charles Lewis, Katia Sycara, and Simon Stepputtis. Adaptively coordinating with novel partners via learned latent strategies. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0047.

Kailing Li, Tianwen Qian, Lijin Yang, Yuqian Fu, Jingyu Gong, Xiaoling Wang, and Liang He. Bridging the 2d-3d gap: A hierarchical semantic-geometric map for vision language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15243–15252, June 2026.

Haokun Liu, Yangqiaoyu Zhou, Mingxuan Li, Chenfei Yuan, and Chenhao Tan. Literature meets data: A synergistic approach to hypothesis generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 245–281. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.12.

Jie Liu, Pan Zhou, Yingjun Du, Ah-Hwee Tan, Cees G. M. Snoek, Jan-Jakob Sonke, and Efstratios Gavves. CaPo: Cooperative plan optimization for efficient embodied multi-agent cooperation. In *International Conference on Learning Representations*, 2025b.

Peiqi Liu, Zhanqiu Guo, Mohit Warke, Soumith Chintala, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. DynaMem: Online dynamic spatio-semantic memory for open world mobile manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025c.

Shari Liu, Kirsten Lydic, Lingjie Mei, and Rebecca Saxe. Violations of physical and psychological expectations in the human adult brain. *Imaging Neuroscience*, 2, 2024. doi: 10.1162/imag_a_00068.

Long Ma, Yuanfei Wang, Fangwei Zhong, Song-Chun Zhu, and Yizhou Wang. Fast peer adaptation with context-aware exploration. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33963–33982. PMLR, 2024a.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Jim Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *The Twelfth International Conference on Learning Representations*, 2024b.

Xinyi Mou, Xuanwen Ding, Qi He, Liang Wang, Jingcong Liang, Xinnong Zhang, Libo Sun, Jiayu Lin, Jie Zhou, Huang Xuanjing, et al. From individual to society: A survey on social simulation driven by large language model-based agents. *ACM Computing Surveys*, 58(11):1–41, 2026.

Michael J. Munje, Chen Tang, Shuijing Liu, Zichao Hu, Yifeng Zhu, Jiaxun Cui, Garrett Warnell, Joydeep Biswas, and Peter Stone. SocialNav-SUB: Benchmarking VLMs for scene understanding in social robot navigation. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*. PMLR, 2025.

NVIDIA. Cosmos-Reason2-8B: Physical AI common sense and embodied reasoning model. Official model card, 2026. URL https://huggingface.co/nvidia/Cosmos-Reason2-8B.

Colin J Palmer and Colin WG Clifford. Face pareidolia recruits mechanisms for detecting human social attention. *Psychological Science*, 31(8):1001–1012, 2020.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with SWE-Gym. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47717–47737. PMLR, 2025.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimir Vondrus, Theophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots. In *The Twelfth International Conference on Learning Representations*, 2024.

Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. In *The Twelfth International Conference on Learning Representations*, 2024.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Ruchit Rawal, Reza Shirkavand, Heng Huang, Gowthami Somepalli, and Tom Goldstein. Argus: Hallucination and omission evaluation in video-llms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20280–20290. IEEE, 2025.

Valentina Romagnano, Julian Kubon, Alexander N Sokolov, Andreas J Fallgatter, Christoph Braun, and Marina A Pavlova. Dynamic brain communication underwriting face pareidolia. *Proceedings of the National Academy of Sciences*, 121(16):e2401196121, 2024.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Lukas Schmid, Marcus Abate, Yun Chang, and Luca Carlone. Khronos: A unified approach for spatio-temporal metric-semantic SLAM in dynamic environments. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.081.

Luca Scofano, Alessio Sampieri, Tommaso Campari, Valentino Sacco, Indro Spinelli, Lamberto Ballan, and Fabio Galasso. Following the human thread in social navigation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Shuai Shao, Qihan Ren, Dongrui Liu, Chen Qian, Boyi Wei, Dadi Guo, Jingyi Yang, Xinhao Song, Linfeng Zhang, Weinan Zhang, et al. Your agent may misevolve: Emergent risks in self-evolving llm agents. In *International Conference on Learning Representations*, volume 2026, pp. 99728–99793, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Janusz A Starzyk and Haibo He. Spatio–temporal memories for machine learning: A long-term memory organization. *IEEE Transactions on Neural Networks*, 20(5):768–780, 2009.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14502–14515, 2021.

Andrew Szot, Unnat Jain, Dhruv Batra, Zsolt Kira, Ruta Desai, and Akshara Rai. Adaptive coordination in social embodied rearrangement. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33365–33380. PMLR, 2023.

Wanxin Tian, Shijie Zhang, Kevin Zhang, Xiaowei Chi, Chun-Kai Fan, Junyu Lu, Yulin Luo, Qiang Zhou, Yiming Zhao, Ning Liu, et al. Seea-r1: Tree-structured reinforcement fine-tuning for self-evolving embodied agents. *Advances in Neural Information Processing Systems*, 38:78458–78499, 2025.

Ujjwal Upadhyay, Mukul Ranjan, Zhiqiang Shen, and Mohamed Elhoseiny. Time blindness: Why video-language models can't see what humans can? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30906–30918, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

Huai-Chih Wang, Hsiang-Chun Chuang, Hsi-Chun Cheng, Dai-Jie Wu, and Shao-Hua Sun. Coot: Learning to coordinate in-context with coordination transformers. In *International Conference on Machine Learning*, 2026a.

Peidong Wang, Zhiming Ma, Ying Chang, Xufang Luo, Xiaocui Yang, Shi Feng, Yuqing Yang, and Dongsheng Li. Self-evolving embodied agents via skill-harness evolution. *arXiv preprint arXiv:2608.11350*, 2026b.

Sen Wang, Bangwei Liu, Zhenkun Gao, Lizhuang Ma, Xuhong Wang, Yuan Xie, and Xin Tan. Explore with long-term memory: A benchmark and multimodal LLM-based reinforcement learning framework for embodied exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026c.

Zheng Wang, Haoran Chen, Haoxuan Qin, Zhipeng Wei, Tianwen Qian, and Cong Bai. Think, then verify: A hypothesis-verification multi-agent framework for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33784–33793, 2026d.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 2025.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Shuo Yang, Caren Han, Xueqi Ma, Yan Li, Mohammad Reza Ghasemi Madani, and Eduard Hovy. Evotool: Self-evolving tool-use policy optimization in llm agents via blame-aware mutation and diversity-aware selection. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 43553–43572, 2026a.

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3D-Mem: 3D scene memory for embodied exploration and reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17294–17303, 2025.

Yunhao Yang, Neel P Bhatt, Kevin Wang, Samuel Tetteh, Zhangyang Wang, and Ufuk Topcu. Vaso: Formally verifiable self-evolving skills for physical ai agents. *arXiv preprint arXiv:2606.05395*, 2026b.

Hanrong Zhang, Shicheng Fan, Henry Peng Zou, Yankai Chen, Zhenting Wang, Jiayu Zhou, Chengze Li, Wei-Chieh Huang, Yifei Yao, Kening Zheng, Xue Liu, Xiaoxiao Li, and Philip S. Yu. Coevoskills: Self-evolving agent skills via co-evolutionary verification. In *Third Conference on Language Modeling*, 2026a.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B. Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. In *International Conference on Learning Representations*, 2024.

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 *International Conference on Learning Representations*, 2025a.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Gödel machine: Open-ended evolution of self-improving agents. In *The Fourteenth International Conference on Learning Representations*, 2026b.

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)*, pp. 13032–13056. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.638.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y. Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *The Fourteenth International Conference on Learning Representations*, 2026c.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47, 2025c.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Yu Zhong, Zihao Zhang, Rui Zhang, Lingdong Huang, Haihan Gao, Shuo Wang, Da Li, Ruijian Han, Jiaming Guo, Shaohui Peng, Di Huang, and Yunji Chen. Run, ruminate, and regulate: A dual-process thinking system for vision-and-language navigation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(22):18845–18854, 2026. doi: 10.1609/aaai.v40i22.38954.

Qinhong Zhou, Chuang Gan, and Anoop Cherian. LLawCo: Learning laws of cooperation for modeling embodied multi-agent behavior. In *Proceedings of the Forty-Third International Conference on Machine Learning*, 2026a.

Qinhong Zhou, Hongxin Zhang, Xiangye Lin, Zheyuan Zhang, Yutian Chen, Wenjun Liu, Zunzhe Zhang, Sunli Chen, Lixing Fang, Qiushi Lyu, et al. Virtual community: An open world for humans, robots, and society. In *International Conference on Learning Representations*, volume 2026, pp. 63971–64010, 2026b.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. Sotopia: Interactive evaluation for social intelligence in language agents. In *International Conference on Learning Representations*, volume 2024, pp. 40975–41019, 2024.

Hao Zhu, Graham Neubig, and Yonatan Bisk. Few-shot language coordination by modeling theory of mind. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12901–12911. PMLR, 2021.

Lizheng Zu, Lin Lin, Song Fu, Na Zhao, and Pan Zhou. Collaborative tree search for enhancing embodied multi-agent collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29513–29522, 2025.

APPENDICES

A Algorithms **19**

B Unified Decision Interface **21**

C Physical Consistency Verifier **23**

D Reasoning Interfaces **25**

E Metrics and VLM Baselines **29**

F Model Configurations **30**

G VLM Invocation Cost **31**

A ALGORITHMS

Algorithms 1–3 expand the three timescales in Section 3: step-wise physical verification, event-triggered proposal and review, and cross-episode hypothesis verification. They use the operators and notation of the main text. The decision interface in Appendix B is shared by social navigation and social rearrangement. PCV construction follows Equation 1 before evaluation. The resulting executable rules and all model parameters remain frozen throughout the evaluation.

A.1 NOTATION

Table 3: Notation used throughout the algorithms.

Symbol	Meaning
e	Episode index.
t	Observation-step index within an episode.
E	Number of episodes in a partner stream.
$o_{e,t}$	Egocentric RGB-D observation and robot pose at step t of episode e.
$\tau_{e,t}$	Observation–action history up to step t of episode e.
$\mathcal{T}$	Task prompt and associated rules.
p_e	Strategy held fixed throughout episode e.
$\hat{c}^j_{e,t}$	Perceptual claim indexed by j at step t of episode e.
$\sigma^j_{e,t}$	PCV verdict for claim $\hat{c}^j_{e,t}$.
$\mathcal{M}_{e,t}$	Spatial Memory, as defined in Equation 3.
$\mathcal{C}_{e,t}$	Context containing non-falsified perceptual claims.
$\hat{\mathcal{G}}_{e,t}$	Set of candidate goals.
$\mathcal{I}^{\text{map}}_{e,t}$	Decision-map image.
$X_{e,t}$	Context provided to the VLM for decision proposal.
$X^+_{e,t}$	Updated context after grounding and physical consistency verification.
$\tilde{g}_{e,t}$	Decision proposed by the VLM.
$\bar{g}_{e,t}$	Final decision after post-grounding review, when triggered.
$\mathcal{Q}_{e,t}$	Entity queries submitted to the perception module.
$r_{e,t}$	Indicator of whether post-grounding VLM review is triggered.
$\mathcal{C}_e$	Claim and action record for completed episode e.
$\mathcal{E}_e$	Verified observations from episode e.
$\mathcal{H}_e$	Hypotheses available during episode e.
h_i	Hypothesis indexed by i.
$v_{e,i}$	Episode-level verification verdict for hypothesis h_i in episode e.
$\mathcal{E}^{\text{pool}}_e$	Unexplained evidence pool after episode e.
B	Hypothesis discovery period, measured in episodes.
$\mathcal{K}^{<e}$	Knowledge retained from episodes completed before episode e.

A.2 TEST-TIME EVOLUTION IN VSEL

$\mathcal{T}$ supplies the corresponding task prompt and rules for strategy synthesis. The initial strategy is empty. An initial hypothesis vocabulary, when supplied, carries no supporting or contradicting episode verdicts. After each interaction, HV processes the completed episode before the next strategy is synthesized. Evaluation scores are computed outside this information flow and are not inputs to adaptation.

Algorithm 1 VSEL

Require: Frozen parameters θ, frozen PCV; task prompt $\mathcal{T}$; initial hypotheses $\mathcal{H}_1$; discovery period $B \geq 1$

Initialize empty completed-episode records and verdict records
$\mathcal{E}_0^{\text{pool}} \leftarrow \varnothing$; $p_1 \leftarrow \varnothing$
for $e = 1, \ldots, E$ **do**
    Reset episode-local state; fix p_e for this episode
    $\mathcal{C}_e \leftarrow \textsc{RunEpisode}(\mathcal{T}, p_e, \theta, \text{PCV})$ ▷ Algorithm 2
    Retain $\mathcal{C}_e$ with its source IDs and evidence status
    Update $\mathcal{H}_{e+1}$, $\{v_{e',i} : e' \leq e\}$, and $\mathcal{E}_e^{\text{pool}}$ using Algorithm 3
    Store the updated records in $\mathcal{K}^{<e+1}$
    if $e < E$ **then**
        Form $\mathcal{H}_{e+1}^{+}$ using Equation 6
        Retain each eligible hypothesis's conditions, support, counterevidence, and limitations
        Synthesize p_{e+1} under $\mathcal{T}$ for episode $e+1$ using Equation 7
        Use the base task strategy when no applicable guidance is justified
    end if
end for

A.3 Step-wise Verification and Event-triggered Decisions

A proposal trigger and a review trigger are different conditions. The former requests a new decision at initialization, a change in target visibility or task-relevant semantics, or decision completion. A change that invalidates the current target or skill preconditions is handled as a decision-relevant change, not as a new source of evidence. The latter is exactly the positive-segmentation condition in Equation 4. In particular, a positive segmentation match still triggers review when PCV rejects the resulting claim. Review then receives the rejection outcome and the remaining non-falsified context, rather than the rejected state as admissible evidence.

Algorithm 2 RunEpisode: PCV at observation steps, VLM reasoning at events

Require: Task prompt $\mathcal{T}$, fixed strategy p_e, frozen θ and PCV
Ensure: Completed-episode record $\mathcal{C}_e$

Initialize episode-local state, empty $\mathcal{C}_e$, and an inactive decision
$t \leftarrow 0$
while episode not terminated **do**
    Acquire $o_{e,t}$; update $\tau_{e,t}$ and observation-derived spatial memory
    Extract currently available perceptual claims $\{\hat{c}_{e,t}^{j}\}_j$
    Assess each claim by Equation 2; update $\mathcal{C}_{e,t}$ and the source-linked episode record
    Update $\mathcal{M}_{e,t}$, $\hat{\mathcal{G}}_{e,t}$, and execution status
    if $t = 0$ or a decision-relevant event occurs **then**
        Render $\mathcal{I}_{e,t}^{\text{map}}$ by Equation 3; assemble $X_{e,t}$
        $(\widetilde{g}_{e,t}, \mathcal{Q}_{e,t}) \leftarrow \text{Propose}_\theta(X_{e,t})$
        Ground $\mathcal{Q}_{e,t}$ in the current image $I_{e,t}$
        $r_{e,t} \leftarrow 1$ if at least one queried entity is positively matched; otherwise $r_{e,t} \leftarrow 0$
        Assess the grounding-derived claims by Equation 2
        Update $\mathcal{C}_{e,t}$ and $\hat{\mathcal{G}}_{e,t}$; refresh the map and checked context $X_{e,t}^{+}$
        Obtain $\bar{g}_{e,t}$ using Equation 4
        Decode $\bar{g}_{e,t}$ using the task-conditioned skill interface in Appendix B
        Store it as the selected decision; suspend execution unless its arguments and evidence-dependent preconditions hold
    end if
    Recheck the active decision against current admissible evidence
    Select and execute $a_{e,t}$ through the local controller; suspend an unsupported or incompletely specified command
    Append $a_{e,t}$ to the history and episode record; $t \leftarrow t + 1$
end while
Process any available terminal observation with PCV and finalize $\mathcal{C}_e$
return $\mathcal{C}_e$

The targeted segmentation pass re-examines the current frame; it does not advance the environment or duplicate that frame in $\tau_{e,t}$. New perceptual claims retain their source IDs, so repeated use of a frame does not create independent evidence. Perception, PCV, and local control continue between proposal events. If $r_{e,t} = 0$, the proposed decision is retained as specified by Equation 4, but it must still satisfy execution preconditions. A local safety stop or an unresolved skill argument is not an additional VLM review trigger. Thus, a yielding request does not bypass the review condition or authorize motion before a valid target has been supplied.

A.4 Cross-episode Verification and Discovery

Existing hypotheses are evaluated on the completed record $\mathcal{C}_e$. New hypotheses are initialized without support and immediately passed to HV, as specified in Section 3.4. Historical verification uses retained completed-episode records; it neither alters earlier actions nor retrospectively improves their evaluation scores.

Algorithm 3 Update hypotheses and unexplained evidence after episode e

Require: $\mathcal{H}_e$, records $\mathcal{C}_1, \ldots, \mathcal{C}_e$, stored verdicts, $\mathcal{E}_{e-1}^{\text{pool}}$, period B
Ensure: $\mathcal{H}_{e+1}$, updated verdicts, and $\mathcal{E}_e^{\text{pool}}$

$\mathcal{H}_{e+1} \leftarrow \mathcal{H}_e$
for all $h_i \in \mathcal{H}_e$ **do**
  $v_{e,i} \leftarrow \mathrm{HV}(h_i, \mathcal{C}_e)$ ▷ Equation 5
  store one verdict for (e, i); update the evidence summary based on its observations
end for
Update $\mathcal{E}_e^{\text{pool}}$ with episode evidence supporting none of the current hypotheses
if $e \bmod B = 0$ and $\mathcal{E}_e^{\text{pool}} \neq \varnothing$ **then**
  Propose new conditional hypotheses from $\mathcal{E}_e^{\text{pool}}$ and $\mathcal{H}_{e+1}$
  Remove duplicate proposals and check their originanl references
  for all remaining new hypotheses h_i **do**
    Initialize h_i with no support or counterevidence; add it to $\mathcal{H}_{e+1}$
    for all retained records $\mathcal{C}_{e'}$, $1 \leq e' \leq e$ **do**
      $v_{e',i} \leftarrow \mathrm{HV}(h_i, \mathcal{C}_{e'})$ ▷ Immediate historical verification
      Store one verdict for (e', i); update the evidence summary based on its observations
    end for
  end for
  Remove newly explained evidence from the pool
end if
Retain support, counterevidence, and unresolved verdicts for each hypothesis

Other hypotheses and prior summaries provide interpretive context to HV. Each verdict is accompanied by an explanation of how the observations in $\mathcal{C}_{e'}$ support, contradict, or leave the hypothesis unresolved. The requirement is an explanation based on observations. Discovery annotations do not assign accepted positive verdicts. Newly discovered hypotheses are first checked against the retained episode records that motivated their discovery, ensuring consistency with the accumulated evidence before they can influence the next strategy. Subsequent episodes then provide prospective evidence that can further support, contradict, or leave these hypotheses unresolved.

B Unified Decision Interface

VSEL represents high-level decisions as task-conditioned responses to both the physical environment and other agents, rather than as navigation waypoints alone. Its decisions span information gathering, movement, spatial coordination, and object interaction through a shared proposal–grounding–review interface. This separation between decision semantics, observational support, and physical execution allows the framework to accommodate different skill repertoires without being tied to a single task's action vocabulary.

Table 4: High-level skills in the VSEL decision interface. Target IDs refer to candidate entities or spatial goals; localization and physical execution are handled by the grounding and control components.

Skill	Argument	Decision semantics
`SEEK`	Direction ϕ	Acquire additional visual evidence through viewpoint adjustment or locally feasible exploratory movement.
`WAIT`	None	Remain stationary while continuing to observe the environment.
`NAVIGATION`	Target ID	Move toward a selected exploration, observation, or interaction target.
`APPROACH`	Target ID	Approach and follow a collaborator while maintaining an appropriate distance and person-facing orientation.
`ORIENT`	Target ID	Turn toward a selected entity or goal without translational motion.
`YIELD`	Target ID	Vacate conflicting space by moving to a suitable yielding position.
`PICK`	Target ID	Attempt to acquire the selected object.
`PLACE`	Target ID	Attempt to place the selected held object at its associated destination.

B.1 Shared Decision Representation

At an event-triggered invocation, $(\widetilde{g}_{e,t}, \mathcal{Q}_{e,t}) = \mathrm{Propose}_\theta(X_{e,t})$ produces a proposed decision and a set of entity queries. A decision specifies a skill together with any required target ID or movement argument. It is distinct from a candidate goal in $\hat{\mathcal{G}}_{e,t}$, a low-level environment action $a_{e,t}$, and the strategy p_e held fixed within the episode. A single decision may govern several environment steps before completion or reassessment.

The proposal context $X_{e,t}$ combines the current RGB observation $I_{e,t}$ and robot pose $\xi_{e,t}$, the decision map $\mathcal{I}^{\mathrm{map}}_{e,t}$, candidate goals $\hat{\mathcal{G}}_{e,t}$, non-falsified claims $\mathcal{C}_{e,t}$, compact memory, task instructions, execution feedback, and p_e. Depth $D_{e,t}$ supports geometric grounding and PCV rather than serving as an additional raw VLM image. Proposal and review use the same decision representation, so new evidence can revise either the selected skill or its target without changing the interface.

B.2 Skill Semantics and Target Grounding

Table 4 summarizes the skills used in our evaluations. Task instructions determine the admissible choices, while all skills follow the same distinction between high-level selection and physically grounded execution.

Target-dependent decisions select a supplied *target ID*, rather than generating coordinates or low-level motion commands. Each target has a semantic description, available localization, and admissible uses. Its inclusion among the candidates does not establish that its current position is verified: execution still requires the evidence and physical preconditions appropriate to the selected skill. For example, navigating to a collaborator's last observed location constitutes a search decision, not an assertion that the collaborator remains there. Similarly, reaching a placement destination is distinct from placing an object, and issuing a manipulation command does not establish its success.

B.3 Visual Reports and Partner Predictions

Alongside its proposed decision, the VLM reports visual categories, which are combined with required entity checks to construct $\mathcal{Q}_{e,t}$. Queries are not restricted to entities reported as visible; requesting a check for a collaborator does not assert that the collaborator is present. Visual reports guide perception, whereas localization and observational support are established by grounding and PCV.

When relevant, the decision includes a provisional prediction of the collaborator's next destination, expressed as a target ID or `unknown`. This prediction concerns anticipated behavior rather than the

collaborator's current location and can inform complementary object selection, interaction timing, or yielding. It is neither an observation in $\mathcal{E}_e$ nor a cross-episode conditional hypothesis in $\mathcal{H}_e$, and cannot itself serve as supporting evidence for hypothesis verification.

B.4 Evidence-Grounded Review and Execution

As defined in Section 3.3, review is triggered when segmentation positively matches at least one queried task-relevant entity ($r_{e,t} = 1$). The trigger depends on the perceptual match, not on the subsequent PCV verdict: even a match whose associated claim is falsified can prompt reassessment. Before review, PCV outcomes are incorporated into the updated context $X^{+}_{e,t}$, including the refreshed decision map and candidate records. Falsified states are excluded from admissible targets rather than passed to the VLM as established facts.

Review reconsiders $\widetilde{g}_{e,t}$ using this updated context and produces $\bar{g}_{e,t}$ according to Equation 4. It may revise the skill, its argument, or the provisional partner prediction. The reassessment uses the same observation step and does not initiate another query–review cycle. When review is not triggered, $\bar{g}_{e,t} = \widetilde{g}_{e,t}$; execution nevertheless remains subject to the selected skill's grounding and physical preconditions.

Only decisions with resolved arguments and satisfied execution preconditions can produce the requested motion or interaction. An unresolved request, such as yielding without an eligible target, remains non-executable and is handled by local safety mechanisms; it does not independently trigger an additional review. This boundary prevents a high-level intention from being treated as an executable command before its required physical support is available.

C Physical Consistency Verifier

C.1 Feedback-Guided Construction

We construct the Physical Consistency Verifier (PCV) by using GPT as the offline program generator in Equation 1. The synthesis prompt $\mathcal{P}_{\mathrm{PCV}}$ specifies the observation interface, coordinate and time conventions, and the evidence required to assess a claim. GPT translates these requirements into executable checks over segmentation masks, calibrated RGB-D observations, and observation history. The resulting program determines whether a candidate claim is supported, contradicted, or unresolved under the available evidence; its rules do not specify how the partner should behave.

Training episodes provide RGB-D observations, camera calibration, robot poses, timestamps, and candidate masks with their queried categories. Simulator-side reference information includes the coordinates of the person and task-relevant objects or goals, visibility annotations, and panoptic masks. We distinguish these reference masks from the predicted masks supplied to the verifier: a predicted mask is a grounding proposal, not a ground-truth annotation. Reference information is available when evaluating candidate programs on $\mathcal{D}_{\mathrm{train}}$; the executable physical checks operate on the observation interface rather than querying reference target states.

The training feedback $\mathcal{F}_{\mathrm{train}}$ combines numerical evaluation with diagnostic evidence. Localization feedback reports the distance between an estimated position and its reference in the same coordinate frame. Geometry feedback includes whether an estimated box lies within a reference box and the fraction of its volume contained therein, allowing partial observations to be assessed without requiring full-object reconstruction. Grounding evidence includes the source image, candidate mask, available reference segmentation, and category or association mismatches. Rule diagnostics expose the support available for each estimate and the reason for accepting, rejecting, or withholding it, such as insufficient depth, incompatible geometry, ambiguous association, or excessive displacement. Missing references and missing estimates are recorded explicitly rather than assigned zero error.

We execute candidate implementations on training episodes and return this feedback to GPT to revise the checks and their applicability conditions. For example, feedback can reveal both a spurious detection that passes an overly permissive geometry check and a valid partial observation rejected by an overly restrictive one. This process produces the rule families below. The final program and its thresholds are fixed before test-time execution.

C.2 Executable Rule Families

The rules are organized around the evidence needed for an attribute, rather than around an exhaustive list of object categories. A candidate $\hat{c}^j_{e,t}$ identifies a grounded referent, an asserted attribute, and its observation source. Checks apply only when their required inputs are available. Consequently, the same observation may support an image bearing while leaving a metric location or a temporal identity claim unresolved.

Observation provenance and sensing support. The verifier checks that a mask belongs to the stated observation and has sufficient pixel support. A reused mask does not become a new observation merely because it is processed again. For metric claims, RGB masks and depth measurements are associated through calibrated registration. Let $\widetilde{m}^j_{e,t}$ denote the corresponding support on the depth image grid. The usable samples are

$$\mathcal{V}^j_{e,t} = \left\{ (u,v) \in \widetilde{m}^j_{e,t} : \begin{array}{l} D_{e,t}(u,v) \text{ is finite,} \\ d_{\mathrm{lo}} \le D_{e,t}(u,v) \le d_{\mathrm{hi}} \end{array} \right\}, \tag{8}$$

where $[d_{\mathrm{lo}}, d_{\mathrm{hi}}]$ is the calibrated usable depth interval, excluding sensor saturation. Registration also accounts for visibility and inconsistent depth at boundaries. Insufficient valid support prevents a new metric estimate; it does not establish that the referent is absent.

Geometric plausibility and partial visibility. Valid samples are back-projected using the calibrated camera model and transformed into a common episode coordinate frame. Spatial filtering reduces contamination from isolated points and disconnected background surfaces. Category-dependent checks then assess whether the supported geometry is plausible, including bounds on observed spatial extent. These checks account for partial visibility: a small visible portion of a person need not span the height of a complete body. When a previous surface representation exists, registration residuals and correspondence support provide additional evidence for an associated partial observation. Ambiguous or insufficient correspondence leaves the affected attribute unresolved.

Temporal feasibility and referent association. Temporal checks compare observations only under a supported association with the same referent. For a current position estimate and an earlier verified estimate at $t' < t$, a general feasible-displacement rule is

$$\left\| \hat{\mathbf{x}}^j_{e,t} - \hat{\mathbf{x}}^{j'}_{e,t'} \right\|_2 \le v_{\max} \Delta_{e,t,t'} + \epsilon_{e,t,t'}, \tag{9}$$

where $\Delta_{e,t,t'}$ is elapsed physical time, $v_{\max}$ is an admissible speed bound, and $\epsilon_{e,t,t'}$ accounts for localization uncertainty. In navigation, this check uses ground-plane coordinates and converts environment-step differences to physical time using the control frequency. It requires neither the partner's true velocity nor its future trajectory. Where changing visibility shifts a partial bounding-box center, surface registration provides a more appropriate displacement estimate. If multiple identities remain feasible, the same-referent claim is unresolved. Failure to associate with an old track does not by itself invalidate a newly observed entity, and lack of an earlier location does not invalidate a currently supported estimate. Figure 5 illustrates how the displacement bound becomes an executable check with explicit evidence requirements and diagnostic outputs.

Observability and evidence freshness. A missing detection is informative only when the relevant region is observable under the current camera view, depth coverage, and occlusion conditions. In particular, seeing a remembered location alone is insufficient for negative identity evidence: current depth must support the expected surface still being observable. Farther background can instead indicate that the location has been vacated. Even when negative evidence is available, rejecting a claim about a remembered location does not establish that the entity is absent elsewhere. The verifier also distinguishes current measurements, remembered values, and predictions. Repeated processing of the same evidence does not provide independent confirmation. A forecast retains the time of its source measurement and does not refresh the last verified metric measurement time. Likewise, new evidence of partial visibility can support visibility without establishing a newly measured center position.

```python
def verify_motion(x_t, x_prev, dt, v_max, eps,
                  current_supported, previous_verified,
                  same_referent):
    # Inputs have passed finite-value and calibration checks.
    if not current_supported or not previous_verified:
        return "Unverified", {"reason": "missing metric evidence"}
    if not same_referent:
        return "Unverified", {"reason": "unresolved association"}
    if dt <= 0:
        return "Unverified", {"reason": "invalid time interval"}

    displacement = norm(x_t - x_prev, ord=2)
    bound = v_max * dt + eps

    if displacement <= bound:
        verdict = "Verified"
    else:
        verdict = "Falsified"
    return verdict, {"displacement": displacement,
                     "allowed_displacement": bound}
```

Figure 5: **Executable motion-consistency rule.** Illustrative pseudocode for Equation 9.

C.3 Claim-Level Outcomes and Deployment

These checks instantiate the claim-level distinction in Equation 2. A VERIFIED claim has sufficient observational support and passes the applicable checks. A FALSIFIED claim conflicts with available evidence. A UNVERIFIED claim lacks sufficient evidence for the asserted attribute. Missing depth, occlusion, or ambiguous identity alone therefore does not establish falsity. Passing physical checks also does not guarantee semantic correctness.

At each observation step, the frozen PCV evaluates these executable checks and supplies their outcomes and diagnostics to evidence maintenance. Its observation state can change as measurements arrive, while the verification program remains fixed. Executing the physical rules requires no additional VLM call; this statement concerns the PCV itself, independently of VLM calls elsewhere in the agent.

D Reasoning Interfaces

VSEL separates high-level decision making, evidence-grounded review, and cross-episode adaptation through shared reasoning interfaces. Task instructions specify the admissible skills and targets, allowing these interfaces to support information gathering, movement, spatial coordination, and object interaction. The templates below instantiate these roles for the evaluated settings; platform names and serialized fields are implementation-specific rather than requirements of the framework. Angle brackets denote substituted inputs, and output examples illustrate syntax rather than experimental results.

Consistent with Section 3.1, the supplied context is derived from task instructions and the agent's observation–action history. Episode summaries, trajectories, and decision maps do not introduce oracle state or privileged evaluation feedback. Generated predictions and prior hypotheses provide interpretive context, not additional observations.

D.1 Online Proposal and Decision Review

At initialization or a decision-relevant event, the VLM proposes a high-level decision $\widetilde{g}_{e,t}$ and entity queries $\mathcal{Q}_{e,t}$, as described in Section 3.3. Events include changes in target visibility or task-relevant scene semantics and completion of the current decision. The decision specifies a skill and, when required, a target ID or movement argument. Reported visual categories guide subsequent perception; neither the proposal nor a category report establishes a verified target state or successful execution.

Listing 1: Compact online proposal prompt.

```
Choose one high-level decision for Spot using the current observation, task context, candidate
records, and memory. Distinguish current observations from remembered positions and motion
predictions. Copy supplied candidate IDs exactly; never invent IDs or target coordinates.

Inputs:
<Task Rules>
<Fixed Cross-Episode Strategy, if available>
<Current Arm RGB and Decision Map>
<Spot Base Pose, Environment Information, and Trigger Event>
<Decision Map Candidates and Compact Navigation Memory>
<Task Assignments, Holding State, and Execution Feedback>

Report in objects only unique category names visible in the current RGB: wall and person, plus the
episode's task-object vocabulary for social rearrangement. A remembered or predicted person position
 does not establish current visibility.

For SEEK, use an angle in {0, +/-22.5, +/-45, +/-67.5, +/-90, +/-112.5, +/-135, +/-157.5, 180}
degrees. Zero requests forward motion; other values request an in-place turn subject to the task's
safety checks.

Return one compact JSON object only, without explanations, Markdown fences, coordinates, or unused
null fields.
```

Reported categories are supplemented with task-relevant entity queries, including checks for the collaborator even when not reported as visible. A query requests perceptual evidence rather than asserting presence. A positive segmentation match for at least one queried entity sets $r_{e,t} = 1$ and triggers review, as defined in Equation 4. This trigger is distinct from the subsequent PCV verdict: a positive match can still require review when its associated claim is falsified. Before review, grounding-derived claims are assessed by PCV and the decision map is refreshed.

Review reassesses the proposal using the updated context under the same task rules and fixed episode strategy. It uses the same observation step, not a later RGB observation: the map refresh incorporates newly computed grounding and verification results, not intervening physical motion. Review may revise the skill or its argument. When no query is positively matched, the proposal is retained without review. In either case, execution remains subject to the selected skill's grounding and physical preconditions.

Listing 2: Compact decision-review prompt triggered by a positive entity-query match.

```
Review the first proposal using the updated decision map and supplied context. Retain or revise its
action under the same task rules and fixed cross-episode strategy. This call provides no new RGB
evidence; do not invent visual details from the first prediction or treat a same-step map refresh as
 observed motion.

Inputs:
<Task Rules>
<Fixed Cross-Episode Strategy, if available>
<First Proposal>
<Refreshed Decision Map>
<Current Environment, Task State, and Estimated Person Trajectory>
<Person Observation Summary and Configured Position Source>
<First and Updated Person Candidates; Current Task Candidates>
<Target Invalidation, Navigation Feedback, and Yield Context>

Use current eligible candidates only. Retired or filtered person tracks cannot be selected. Return
one compact JSON object only.
```

D.2 Compact Memory and Candidate Inputs

Compact memory and candidate records provide complementary context. Memory summarizes observation-derived spatial structure, connectivity, and currently represented unexplored regions; it does not provide prior knowledge of unseen room layouts. Candidate records identify entities and spatial goals for exploration and interaction. Episode-local spatial memory resets between episodes, whereas the cross-episode knowledge in Section 3.1 persists within the same partner stream. The following excerpt illustrates supplied inputs rather than a VLM response.

Listing 3: Compact navigation memory and a separate candidate-table excerpt.

```
Compact navigation memory JSON:
{
  "current_place": {
    "place_id": "<integer or null>",
    "center_xz": "<world [x,z] or null>",
```

```
    "exploration_status": "<exploration state>",
    "view_scan": "<scan coverage and heading-sector record or null>"
  },
  "adjacent_passages": [{
    "passage_id": "<integer>",
    "anchor_xz": "<world [x,z]>",
    "connected_place_ids": "<two place IDs>",
    "target_place_id": "<adjacent place ID>"
  }],
  "unexplored_rooms": [{
    "place_id": "<integer>",
    "center_xz": "<world [x,z] or null>",
    "passages": "<passage records>"
  }]
}

Decision map candidates JSON:
[
  {
    "candidate_id": "<displayed ID>",
    "kind": "<candidate kind>",
    "world_xz": "<world [x,z]>"
  }
]
```

Each candidate is referenced by a target ID, with its semantic role, available localization, evidence status, observation time, and eligibility conveyed through the candidate and environment context. Consistent with Section 3.2, UNVERIFIED claims may inform further observation but are not promoted to verified evidence; FALSIFIED claims are excluded from the active decision context. Neither an assigned target, a remembered location, nor a predicted position establishes the entity's current presence or a verified metric position. Invalid or range-saturated depth samples are excluded from metric localization.

D.3 HYPOTHESIS VERIFICATION

At the end of episode e, the frozen Hypothesis Verifier (HV) assesses existing behavioral hypotheses using the episode observations and task context summarized in $\mathcal{C}_e$. Episode summaries, estimated collaborator trajectories, action records, and decision maps provide complementary views of the interaction. For each hypothesis, HV returns a verdict and explains how the observed behavior relates to its applicability conditions and predicted outcome. PCV assessments may inform this context, but HV does not require citations to individual PCV-verified observations. Its requirement is to explain the judgment from the available episode observations, distinguishing observed behavior from predictions, robot actions, and unsupported interpretations.

Listing 4: Episode-level hypothesis verification with explanations based on observations. One call evaluates all supplied hypotheses for the completed episode.

```
Assess only the existing person-behavior hypotheses against the completed episode. Return exactly
one verdict for every supplied hypothesis_id: support, contradiction, or unclassified. Explain each
verdict from the episode observations. Do not propose, rename, remove, or rewrite hypotheses.

<Person-Behavior Evidence Rules>

Use the episode summary, task context, estimated person trajectory, final VLM output, and decision
maps together. Distinguish observations from predictions and intended actions from actual execution.
 The agent's actions do not establish the person's intentions, and task assignments alone do not
establish a freely expressed preference. Other hypotheses and previous verdicts provide context, not
 new observations.

Explain how the observed events relate to the hypothesis's conditions and prediction. Use
unclassified when relevant preconditions or outcomes are unobserved, the available observations
cannot distinguish the alternatives, or the evidence is irreconcilably mixed. State the uncertainty
rather than inventing missing events. A particular detector output is not required when the other
supplied observations are sufficient.

Trajectory lines join observations; they do not prove continuous motion through gaps. Reused RGB,
localization jumps, and identity changes cannot establish motion. Compressed step ranges may hide
accumulated small displacements. Distinguish Spot's trajectory from the person's.

For each hypothesis, return one concise cumulative evidence_summary. Explain the current verdict
using the episode observations while preserving relevant earlier support, counterevidence, and
uncertainty. New support must not erase counterevidence, and a contradiction must not erase earlier
support. An unclassified episode may clarify uncertainty but must not replace prior evidence with
its lack of information. The program maintains episode IDs and counts; the summary explains the
judgments rather than listing verified-observation citations.

Inputs:
```

```
<Existing Hypotheses and Cumulative Evidence>
<Completed Episode Summary and Task Context>
<Estimated Person Trajectory and Final VLM Output>
<Matched Decision Map>

Return only:
{"verdicts":[
  {"hypothesis_id":"<existing ID>",
   "verdict":"<support | contradiction | unclassified>",
   "evidence_summary":"<explanation based on observations, retained support and counterevidence, and
 uncertainty>"}
]}
```

The serialized verdicts `support`, `contradiction`, and `unclassified` correspond to $+$, $-$, and $?$ in Equation 5, which denotes the verdict component of HV. The accompanying `evidence_summary` explains the current judgment and preserves the relevant accumulated observations and interpretations. Each episode contributes at most one supporting or contradicting verdict per hypothesis; an unresolved judgment contributes to neither set. Episode identifiers and counts organize these judgments without imposing an observation-level citation requirement. The same explanatory requirement applies when HV evaluates retained historical episodes.

D.4 Strategy Synthesis

Before episode $e+1$, strategy synthesis considers $\mathcal{H}^{+}_{e+1}$ from Equation 6: hypotheses with at least one accepted supporting HV verdict from a completed episode. Their applicability conditions, counterevidence, source-linked evidence, and limitations remain available. Eligibility does not imply acceptance for guidance. Together with the next task's instructions, these records inform a concise partner-conditioned strategy that may adjust action selection, target ranking, exploration order, routes, timing, or observation behavior. Historical evidence may rank permitted choices, but cannot establish current state or satisfy an action's observational and physical preconditions.

Listing 5: Synthesis of a partner-conditioned strategy from hypotheses with accepted supporting verdicts.

```
Derive a concise person-specific policy for Spot by applying the supplied task rules to supported
predictions of the person's behavior. The policy should improve task progress or coordination while
respecting those rules.
Inputs:
<Task Rules: authoritative task_system_prompt>
<Episode Context>
<Hypotheses with Accepted Supporting Verdicts, Counterevidence, and Source-Linked Evidence>

Identify the most specific applicable supported tendency, including its actor, target, conditions,
and uncertainty. Hypotheses describe the person; they are not instructions for Spot. Task mappings
alone do not establish the person's current state or an agent-specific assignment.

Evaluate the joint outcome if the person behaves as predicted and Spot follows the proposed policy.
Preserve actor ownership. Whether their choices should coincide, differ, or be sequenced follows
from the task rules. Reject a primary choice that creates the interference it claims to reduce; a
fallback or safety caveat cannot justify it.

Translate the tendency into a concrete conditional change in Spot's action choice, target ranking,
search order, route, timing, or observation behavior. Ground references through task context and
authoritative online candidates. Do not invent identifiers, coordinates, visibility, intentions,
availability, or successful actions.

A historical prior may rank otherwise permitted choices before live confirmation, but does not
establish current state or satisfy an observation or action precondition. Preserve actual hypothesis
 conditions, current commitments, mandatory event rules, eligibility, and safety. Specify applicable
 counterevidence and a useful task-valid fallback without indefinite waiting or permanent
reservation of an option.

Return 2-4 concise sentences covering the supported tendency and context, Spot's response and
coordination benefit, and conditions for applying the guidance or using a task-valid fallback. The
resulting strategy remains fixed during the episode; current evidence determines which conditional
recommendation applies. Return an empty string if no supported pattern yields a useful task-valid
adaptation beyond the base task. Do not return analysis, raw statistics, historical coordinates,
temporary detection IDs, or controller tuning.

Return only:
{"spot_decision_tendency":"<conditional policy or empty string>"}
```

E METRICS AND VLM BASELINES

E.1 SOCIAL NAVIGATION METRICS

We evaluate E episodes indexed by $e \in \{1, \dots, E\}$. Within each benchmark, all episodes share the same maximum step budget $L^{\max}$. Let y_e^{find} indicate finding success and y_e^{follow} indicate finding followed by at least 400 valid following steps. A valid following step requires the robot to remain within 1–2 m of the person and face toward them under the referenced evaluation protocol (Scofano et al., 2025). We report finding success and episode success as

$$\text{S} = \frac{1}{E}\sum_{e=1}^{E} y_e^{\text{find}}, \qquad \text{ES} = \frac{1}{E}\sum_{e=1}^{E} y_e^{\text{follow}}. \tag{10}$$

Let $L_e^\star$ denote the evaluator's oracle minimum encounter steps, L_e^{find} the steps taken by the robot to first satisfy the finding criterion, and w_e the number of valid following steps. Search efficiency and following rate are defined as

$$\text{SPS} = \frac{1}{E}\sum_{e=1}^{E} y_e^{\text{find}} \frac{L_e^\star}{\max(L_e^\star, L_e^{\text{find}})}, \tag{11}$$

$$\text{F} = \frac{1}{E}\sum_{e=1}^{E} \frac{w_e}{\max(L^{\max} - L_e^\star, w_e)}. \tag{12}$$

Episodes without finding success contribute zero to SPS. These normalizers follow the benchmark implementations (Puig et al., 2024; Scofano et al., 2025); the oracle quantities used for scoring are not provided to VSEL.

For social navigation, let $b_e^{\text{follow,coll}}$ indicate whether episode e terminates in a robot–humanoid collision. The collision rate is

$$\text{CR} = \frac{1}{E}\sum_{e=1}^{E} b_e^{\text{follow,coll}}. \tag{13}$$

CR is an episode-level fraction, not the fraction of colliding steps. Under the referenced SDA protocol, reaching 400 valid following steps establishes episode success but does not itself terminate execution; episodes continue until the step budget is exhausted or a collision occurs (Scofano et al., 2025).

E.2 SOCIAL REARRANGEMENT METRICS

Let y_e^{joint} indicate successful placement of all required objects, M_e denote their number, and m_e^{robot} denote the number credited to the robot. Let $b_e^{\text{joint,coll}}$ indicate whether a robot–humanoid collision occurs during episode e. We report

$$\text{SR} = \frac{1}{E}\sum_{e=1}^{E} y_e^{\text{joint}}, \qquad \text{RC} = \frac{1}{E}\sum_{e=1}^{E} \frac{m_e^{\text{robot}}}{M_e}, \qquad \text{CR} = \frac{1}{E}\sum_{e=1}^{E} b_e^{\text{joint,coll}}. \tag{14}$$

SR measures joint task success, RC the fraction of required objects completed by the robot, and CR the fraction of episodes containing a collision (Puig et al., 2024). RC is interpreted alongside SR and RE: a larger robot share alone does not imply better collaboration.

For relative efficiency, let L_e^{human} denote the matched humanoid-only reference completion steps and L_e^{joint} the team's successful completion steps. We correct the RE definition in the original Habitat 3.0 paper (Puig et al., 2024) to match its evaluation implementation:

$$\text{RE}_e = \begin{cases} \dfrac{L_e^{\text{human}}}{L_e^{\text{joint}}}, & y_e^{\text{joint}} = 1, \\ \dfrac{L_e^{\text{human}}}{L^{\max}}, & y_e^{\text{joint}} = 0. \end{cases} \tag{15}$$

Aggregate RE is $\frac{1}{E}\sum_{e=1}^{E} \text{RE}_e$. An unsuccessful episode uses the shared maximum budget $L^{\max}$, even if execution terminates earlier. For successful episodes, 100% matches the humanoid-only completion time, whereas 200% corresponds to completing the task in half that time. The main social rearrangement table reports SR and RE as percentages and RC and CR as fractions.

E.3 VLM BASELINES

We implement the following VLM baselines to assess the contributions of VSEL's multi-timescale verification and decision-map context. All VLM-based methods use the same frozen backbone specified in Section 4.1, with matched permitted observations and geometric execution components. Simulator-derived quantities in the metric definitions above are used only for evaluation, not for online decisions or cross-episode adaptation. Implementation details of each baseline are provided below.

HSGM. HSGM (Li et al., 2026) is transferred to Social Navigation through its hierarchical mapping and geometry-grounded waypoint selection. Geometric, semantic, and decision layers retain occupancy/traversability, detected entity footprints, and candidate waypoints with navigation history. Candidate IDs correspond across the annotated egocentric image and bird's-eye-view map, both supplied to the VLM. The action interface selects a numbered waypoint, a left/right 90° rotation, a 180° rotation, or the completion token. For sustained social navigation, the completion token is adapted to conditional waiting when the humanoid is visible and the required distance and orientation hold. Numbered waypoints remain execution endpoints and use the same geometric planner and local controller as VSEL. This adaptation uses a 0.4–$9.9\,\mathrm{m}$ waypoint range and person-triggered replanning rather than a dedicated continuous-following high-level skill.

ET-Baseline. ET-Baseline is an event-triggered ablation baseline of VSEL for social agents. It shares VSEL's event-triggered VLM invocation mechanism, decision interfaces, low-level controllers, and task prompts, while disabling the PCV, post-grounding VLM review, decision-map input, and cross-episode hypothesis verification. This comparison evaluates the combined contribution of VSEL's multi-timescale verification and decision-map context beyond event-triggered decision making.

F MODEL CONFIGURATIONS

All model parameters remain frozen throughout evaluation. The primary VLM backbone is `Qwen3.8-27B-FP8`, as specified in Section 4.1, and is shared across five roles: decision proposal, post-grounding review, hypothesis verification (HV), hypothesis discovery, and strategy synthesis. The PCV is constructed offline according to Equation 1 and executed as fixed code at test time. Test-time adaptation updates evidence, hypotheses, and strategies rather than model parameters.

Qwen3.8-27B-FP8. All VLM roles use a shared vLLM server through an OpenAI-compatible `/v1/chat/completions` endpoint. Table 5 lists the decoding and serving configuration; explicit request parameters override the checkpoint's generation defaults.

Inputs are tailored to each role. Decision proposals receive the current RGB observation and decision map, whereas post-grounding reviews receive the updated decision map and textual evidence. The Hypothesis Verifier receives the final decision map and structured episode evidence. Hypothesis discovery and strategy synthesis use textual evidence only.

For hypothesis discovery, input tokens are counted using the matching tokenizer or the server's `/tokenize` endpoint. The effective input cap respects both the requested cap and the server's context limit after reserving the full response budget. Under the reference configuration, this leaves $55{,}536$ input tokens within the $65{,}536$-token context window. Cross-episode responses are validated against their output schemas, with validation feedback included in subsequent repair attempts. Requests permit up to three retries with increasing backoff intervals; non-retryable client errors terminate the request earlier.

SAM3.1. We use SAM 3.1 from the SAM 3 family (Carion et al., 2026) for text-conditioned visual grounding, with the `sam3.1_multiplex.pt` checkpoint. Although instantiated through the multiplex predictor builder, the shared service applies its detector independently to individual RGB frames, without point or box prompts. Table 6 summarizes the preprocessing, detection, and mask-processing settings used by this service.

Table 5: Inference and serving configuration for `Qwen3.8-27B-FP8`. The input-cap settings apply to hypothesis discovery and historical hypothesis checks.

Parameter	Value
Serving framework	vLLM
API endpoint	`/v1/chat/completions`
Maximum context length	65,536 tokens
Maximum response length	10,000 tokens per request
Temperature	0.7
Top-p	0.8
Top-k	20
Presence penalty	1.5
Thinking	Disabled
Tensor parallel size	1
Serving dtype	`auto`
GPU memory utilization	0.95
Maximum images per request	2
Reasoning parser	`qwen3`
Request timeout	180 s per attempt
Maximum retries	3

Table 6: SAM3.1 configuration for the shared grounding service. Spatial dimensions are reported as height $\times$ width.

Parameter	Value
Checkpoint	`sam3.1_multiplex.pt`
Inference mode	Single-frame text-conditioned detection
Model weights	Frozen
Point/Box prompts	None
RGB input resolution	240×228 pixels
Processor resolution	1008×1008 pixels
Channel order	RGB
Input scaling	$[0, 1]$
Normalization mean	0.5
Normalization standard deviation	0.5
Detection score threshold	0.2
Binary-mask probability threshold	0.5
Candidate retention	Highest-scoring candidate per text query
Mask-logit interpolation	Bilinear
Interpolation `align_corners`	`false`
Output mask resolution	Original input resolution

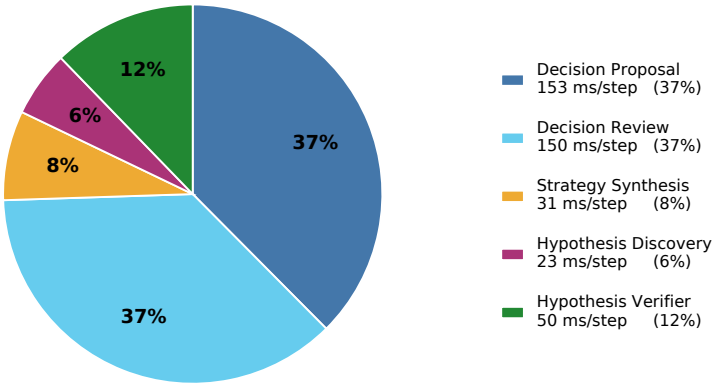
Input frames are resized to the processor resolution and normalized channel-wise. Detection scores are computed from sigmoid-transformed class logits and multiplied by the predicted presence probability when available. For each text query, only the highest-scoring candidate is retained, provided that it meets the detection threshold. Its mask logits are bilinearly resized to the original image resolution with `align_corners=false`; pixels with sigmoid probabilities greater than 0.5 form the binary mask.

G VLM Invocation Cost

We count *logical* VLM requests, grouping internal retries with their originating request. Failed requests are included, and recorded wall times include all executed retries. For episode e, requests

Table 7: Logical VLM requests and average VLM time per executed step across $E = 1{,}200$ episodes. Time shares are rounded independently.

VLM Function	Calls/episode	Time (ms/step)	Time share
Decision Proposal	38.29	153	37%
Decision Review	37.55	150	37%
Strategy Synthesis	1.99	31	8%
Hypothesis Discovery	0.48	23	6%
Hypothesis Verifier	1.00	50	12%
Total	79.31	406	100%



are partitioned into online decision making and cross-episode adaptation:

$$N_{e,\text{online}} = N_{e,\text{proposal}} + N_{e,\text{review}}, \tag{16}$$

$$N_{e,\text{cross}} = N_{e,\text{HV}} + N_{e,\text{discovery}} + N_{e,\text{strategy}},$$

$$N_{e,\text{total}} = N_{e,\text{online}} + N_{e,\text{cross}}. \tag{17}$$

Cross-episode requests are counted once against the episode that triggers them. Historical checks of newly discovered hypotheses count toward $N_{e,\text{HV}}$. A batched request assessing multiple hypotheses counts once; the number of assessed hypothesis–episode pairs is recorded separately. Strategy synthesis in Equation 7 and subsequent strategy review both contribute to $N_{e,\text{strategy}}$. Offline PCV synthesis is reported separately as a preparation cost; executing the frozen PCV requires no VLM calls.

For role r, let $N_{e,r}$ denote the logical request count and $T_{e,r}$ the cumulative VLM wall time in seconds for episode e. Let $L_e^{\text{exec}} \le L^{\max}$ denote its actual executed length. We report mean requests per episode and mean VLM time per executed step, weighting episodes equally:

$$\overline{N}_r = \frac{1}{E}\sum_{e=1}^{E} N_{e,r}, \qquad \tau_r = \frac{1}{E}\sum_{e=1}^{E} \frac{T_{e,r}}{L_e^{\text{exec}}}. \tag{18}$$

Here, τ_r is measured time per executed environment step, and the time share of role r is $p_r = \tau_r / \sum_{r'} \tau_{r'}$. Times and shares are rounded after aggregation, so displayed entries may not sum exactly to their reported totals.

As shown in Table 7, VSEL averages 79.31 logical VLM per episode. Cross-episode adaptation accounts for only 3.47 requests per episode, or 4.4% of all requests. The episode-averaged VLM time is approximately 0.4 s per executed step. VSEL thus performs repeated physical checks without additional VLM calls and requires few extra requests for cross-episode adaptation, keeping the generative-inference workload concentrated in event-triggered decisions and episode-level updates. Although hypothesis discovery is triggered every eight episodes, each round splits accumulated evidence into multiple requests to satisfy the input token budget, yielding 0.48 logical requests per episode versus approximately 0.125 discovery rounds per episode.