Hypothesis Graph Refinement: Hypothesis-Driven Exploration with Cascade Error Correction for Embodied Navigation

Peixin Chen[1,2⋆], Guoxi Zhang[2], Jianwei Ma[1,3], and Qing Li[2]

[1] Harbin Institute of Technology, Harbin, China
[2] Beijing Institute for General Artificial Intelligence (BIGAI), Beijing, China
[3] Peking University, Beijing, China
jwm@pku.edu.cn, dylan.liqing@gmail.com

Abstract. Embodied agents must explore partially observed environments while maintaining reliable long-horizon memory. Existing graph-based navigation systems improve scalability, but they often treat unexplored regions as semantically unknown, leading to inefficient frontier search. Although vision-language models (VLMs) can predict frontier semantics, erroneous predictions may be embedded into memory and propagate through downstream inferences, causing structural error accumulation that confidence attenuation alone cannot resolve. These observations call for a framework that can leverage semantic predictions for directed exploration while systematically retracting errors once new evidence contradicts them. We propose *Hypothesis Graph Refinement* (HGR), a framework that represents frontier predictions as revisable hypothesis nodes in a dependency-aware graph memory. HGR introduces (1) *semantic hypothesis module*, which estimates context-conditioned semantic distributions over frontiers and ranks exploration targets by goal relevance, travel cost, and uncertainty, and (2) *verification-driven cascade correction*, which compares on-site observations against predicted semantics and, upon mismatch, retracts the refuted node together with all its downstream dependents. Unlike additive map-building, this allows the graph to contract by pruning erroneous subgraphs, keeping memory reliable throughout long episodes. We evaluate HGR on multimodal lifelong navigation (GOAT-Bench) and embodied question answering (A-EQA, EM-EQA). HGR achieves 72.41% success rate and 56.22% SPL on GOAT-Bench, and shows consistent improvements on both QA benchmarks. Diagnostic analysis reveals that cascade correction eliminates approximately 20% of structurally redundant hypothesis nodes and reduces revisits to erroneous regions by 4.5×, with specular and transparent surfaces accounting for 67% of corrected prediction errors.

Keywords: Embodied Navigation · Long-Horizon Exploration · Hypothesis Graph Refinement · Verification-Driven Cascade Correction

⋆ P. Chen and G. Zhang contributed equally. J. Ma and Q. Li are the corresponding authors.

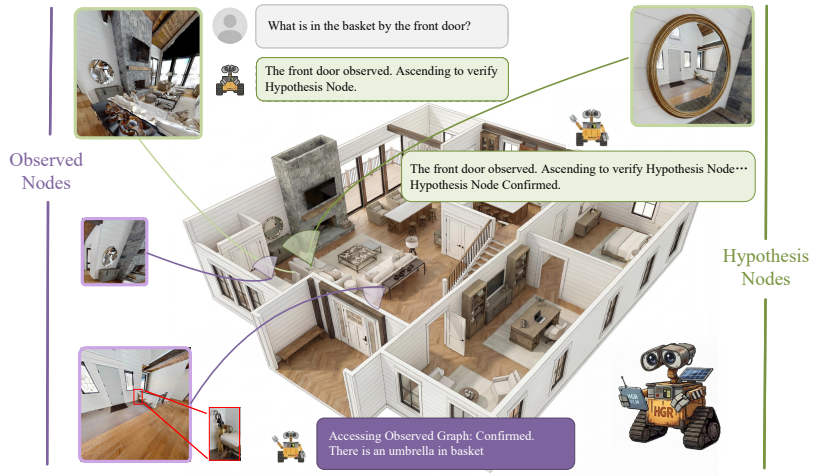

Fig. 1: Overview of Hypothesis Graph Refinement (HGR). The hypothesis graph separates **observed nodes** (purple, verified regions) from **hypothesis nodes** (green, probabilistic frontier predictions), enabling a hypothesis-verification-correction cycle. **(Left)** Given the query "*What is inside the basket?*", observed nodes provide confirmed scene context, while hypothesis nodes project semantic distributions onto unexplored frontiers, guiding the agent toward the most likely location of the target object. **(Right)** Upon arrival, the agent verifies each hypothesis against actual observations. If the prediction is confirmed, the hypothesis node transitions to an observed node; if refuted, cascade correction retracts the erroneous node and all its downstream dependents, preventing error propagation through the graph.

1 Introduction

Embodied agents operating in partially observed environments must decide where to explore next based on incomplete information. Classical frontier-based methods [1, 31] select boundaries between explored and unexplored space using geometric criteria, treating all frontiers as equally unknown. Recent work has begun leveraging vision-language models (VLMs) to predict what may lie beyond observed boundaries, enabling semantically informed frontier selection that prioritizes promising directions over exhaustive search [18, 25, 27]. This shift from geometry-driven to semantics-driven exploration has demonstrated clear efficiency gains in short-horizon navigation tasks.

However, VLM-based frontier predictions are not always reliable—visually ambiguous scenes such as mirror reflections or glass partitions can produce erroneous predictions that, once embedded in graph memory, propagate through chains of dependent hypotheses that inherit and amplify the original mistake. Existing approaches mitigate this through confidence attenuation, but the erroneous subgraph persists and continues to influence downstream decisions, causing cumulative degradation over long-horizon episodes.

We present **Hypothesis Graph Refinement** (HGR), a framework that jointly enables semantics-driven exploration and systematic error correction (Figure 1). HGR maintains a *hypothesis graph* that separates *observed nodes* (verified regions) from *hypothesis nodes* (probabilistic frontier predictions) and records their derivation relationships in a dependency DAG. A *semantic hypothesis module* estimates semantic distributions over frontiers and ranks exploration targets by goal relevance, travel cost, and uncertainty. Upon visitation, a *verification-driven cascade correction* mechanism compares actual observations against predictions; upon mismatch, it retracts the refuted node together with all downstream dependents, allowing the graph to contract by pruning erroneous subgraphs. Our contributions are:

- We introduce a hypothesis graph representation that distinguishes verified observations from revisable frontier predictions and tracks their dependencies, enabling structured hypothesis-driven exploration.
- We propose a cascade correction mechanism that, upon detecting prediction failures, retracts the entire erroneous subgraph along the dependency DAG, preventing error accumulation over long episodes.
- We demonstrate consistent improvements on GOAT-Bench [13] (+3.31% SR, +7.32% SPL over 3D-Mem [31]), A-EQA [19], and EM-EQA [19]. Ablation and diagnostic analyses confirm that both mechanisms contribute meaningfully and that cascade correction substantially reduces error accumulation.

2 Related Work

Embodied Navigation and Exploration. Embodied navigation evolves from reactive policies [2, 14] to memory-augmented architectures [5, 10]. ETPNav [1] introduces topological graphs for long-horizon navigation with Monte Carlo Tree Search for frontier evaluation. FILM [21] maintains episodic memories via transformer-based aggregation. Recent work explores VLM integration for open-vocabulary navigation [18, 27], but treats unexplored regions as lacking semantic information. Further advances include reinforcement fine-tuning for navigation [24], multi-turn active exploration [34], and generative visual imagination [11]. Concurrent work on commonsense-guided frontier exploration includes ESC [35], SCOPE [30], and ReVoLT [16], which also leverage semantic priors to rank and select frontiers. However, these methods treat predictions as one-time scoring signals without recording derivation dependencies among them, so erroneous predictions persist in memory and may propagate to downstream inferences. HGR instead embeds each frontier prediction as a revisable hypothesis node in a dependency DAG, enabling systematic verification upon visitation and cascade correction that retracts an erroneous node together with all its dependents (see the supplementary material for a detailed comparison).

3D Scene Representations. Object-centric scene graphs [3, 8, 29] compress environments into nodes (objects) and edges (relationships). ConceptGraph [8] pioneers open-vocabulary scene graphs via CLIP grounding. 3D-Mem [31] employs

memory snapshots capturing co-visible objects with spatial context. Recent extensions address contrastive pretraining [9], retrieval-augmented reasoning [32], metric-semantic graphs for embodied Q&A [26], and LLM-based semantic integration [33]. These representations handle spatial reasoning well but rely on soft probabilistic updates for error correction. HGR introduces dependency tracking and cascade correction to remove erroneous subgraphs rather than attenuating confidence scores.

Error Detection and Correction in Embodied Systems. VLM prediction errors, where generated content is inconsistent with visual inputs, are studied extensively [15, 28]. Mitigation strategies include retrieval-augmented generation [22], uncertainty quantification [12], and self-consistency checks [20]. In recent robotics, COWS [7] uses contrastive learning and PIVOT [23] employs verification modules. ReLEP [17], HEAL [4], and InterleaveVLA [6] focus on output-level refinement, whereas HGR tracks dependency chains and removes erroneous subgraphs via cascade correction.

3 Method

Frontier-based exploration faces two intertwined challenges: frontiers lack semantic cues to guide efficient exploration, yet VLM-based predictions risk embedding errors that propagate through dependent hypotheses over long horizons. HGR addresses both within a unified graph-based framework: the hypothesis graph separates verified observations from revisable frontier predictions and tracks their dependencies (Section 3.1), evolving through an incremental construction pipeline (Section 3.2). The semantic hypothesis module (Section 3.3) tackles the first challenge by projecting goal-relevant distributions onto frontiers; verification-driven cascade correction (Section 3.4) tackles the second by retracting erroneous subgraphs along the dependency DAG. Figure 2 illustrates the overall architecture.

3.1 Hypothesis Graph Representation

HGR maintains a hypothesis graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{D})$ that serves as the agent's persistent memory of the environment.

Node Types. The node set $\mathcal{V} = \mathcal{V}^{\text{obs}} \cup \mathcal{V}^{\text{hyp}}$ separates two categories. *Observed nodes* $v \in \mathcal{V}^{\text{obs}}$ represent physically visited regions, storing verified semantic labels, visual features, and detected object lists from direct perception. *Hypothesis nodes* $v \in \mathcal{V}^{\text{hyp}}$ represent semantic predictions projected onto unexplored frontiers; each carries a probability distribution $P(\cdot \mid \mathcal{H}_t, f)$ over semantic categories and is marked as unverified.

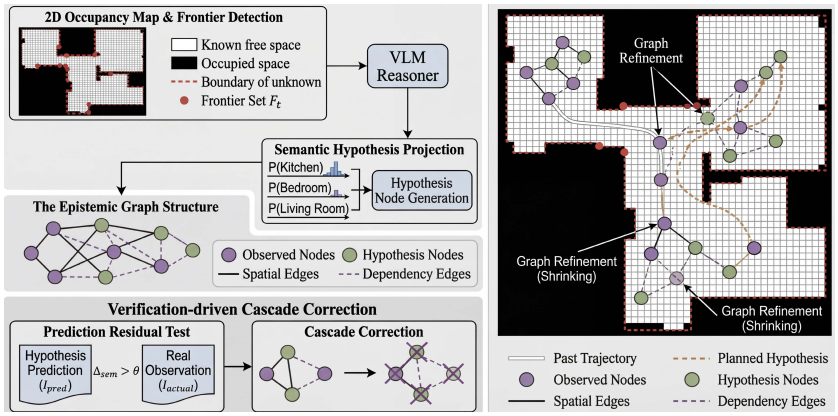

Fig. 2: Architecture of HGR. (Left) Frontiers $\mathcal{F}_t$ detected from the occupancy map are fed to a VLM reasoner for *semantic hypothesis module*, which estimates categorical distributions and generates **hypothesis nodes** linked to **observed nodes** via spatial and dependency edges. Upon visitation, *cascade correction* compares predicted (I_{pred}) and actual (I_{actual}) semantics; if $\Delta_{\text{sem}} > \theta$, the refuted node and all its dependents are removed. **(Right)** Running example on a floor plan. *Graph Refinement* marks confirmed hypotheses promoted to observed nodes; *Graph Refinement (Shrinking)* shows where cascade correction prunes erroneous subgraphs, contracting the graph.

Navigability Edges and Dependency DAG. The edge set $\mathcal{E}$ encodes traversable paths for navigation. The directed acyclic graph $\mathcal{D}$ records derivation relationships among nodes: an edge $(v_p, v_c, \rho) \in \mathcal{D}$ indicates that hypothesis v_c was derived from parent v_p with confidence $\rho \in [0, 1]$. This structure enables systematic retraction: when a node is invalidated, all dependent descendants can be identified and removed along $\mathcal{D}$.

3.2 Incremental Construction and Update

The hypothesis graph evolves incrementally as the agent explores through a three-phase cycle: *hypothesis generation*, *verification*, and *update*. We outline each phase below; Sections 3.3 and 3.4 then detail the generation and verification mechanisms, respectively.

Phase 1: Frontier Discovery and Hypothesis Generation. At each timestep t, given observation $(I_t, d_t, \mathbf{p}_t)$ (RGB, depth, pose), the system identifies geometric frontiers $\mathcal{F}_t$ from the occupancy map. For each new frontier f_j, a hypothesis node v_j^{hyp} is generated via the semantic hypothesis module (Section 3.3), added to $\mathcal{V}^{\text{hyp}}$, and linked in $\mathcal{D}$ to its parent observed node.

Phase 2: Hypothesis Verification. When the agent reaches a hypothesis node v_j^{hyp}, it obtains actual observations and computes a prediction residual (Section 3.4). Two outcomes follow: (1) *Confirmation*—prediction matches obser-

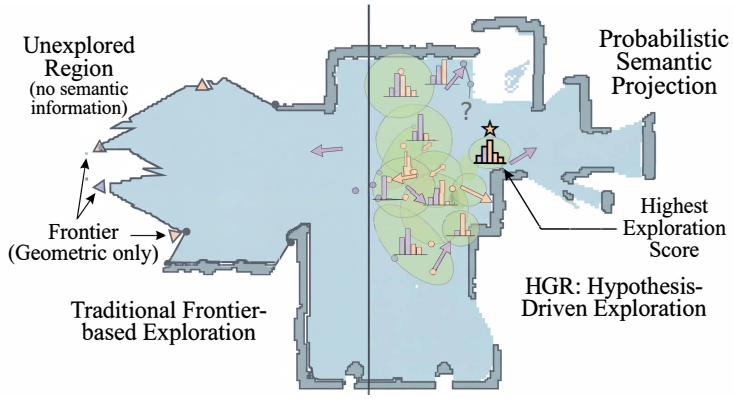

Fig. 3: Semantic Hypothesis Module. Left: Traditional frontier representation treats unexplored regions as undifferentiated boundaries. Right: HGR projects probabilistic semantic distributions onto frontiers as hypothesis nodes, enabling goal-directed exploration.

vation, and the node transitions to $\mathcal{V}^{\text{obs}}$ with updated labels; (2) *Refutation*—significant deviation triggers cascade correction, removing the node and all its dependents from $\mathcal{D}$.

Phase 3: Observed Node Update. When revisiting an observed node, the system updates its object list and visual features from the latest perception. The supplementary material provides full construction rules and worked examples.

3.3 Semantic Hypothesis Module

Rather than treating frontiers as undifferentiated boundaries (Figure 3, left), HGR projects probabilistic semantic distributions onto each frontier, producing hypothesis nodes with estimated categorical distributions that enable goal-directed exploration (Figure 3, right).

Hypothesis Node Generation. Given observation history $\mathcal{H}_t = \{(I_i, d_i, \mathbf{p}_i)\}_{i=1}^{t}$ and the frontier set $\mathcal{F}_t = \{f_j\}$ identified in Phase 1, each frontier f_j is assigned a semantic distribution over categories $\mathcal{C} = \{c_1, \ldots, c_K\}$:

$$P(c_k \mid \mathcal{H}_t, f_j) \propto \phi_{\text{VLM}}(c_k \mid I_j^{\text{frontier}}, \mathcal{O}_{\text{local}}), \tag{1}$$

where I_j^{frontier} is the visual observation toward frontier f_j, $\mathcal{O}_{\text{local}}$ denotes detected objects within spatial radius r_{context}, and ϕ_{VLM} uses the VLM's world knowledge to estimate semantic categories from partial visual cues and spatial context.

Exploration Scoring. To convert frontier exploration into goal-directed hypothesis testing, we define an exploration score for each frontier f_j relative to navi-

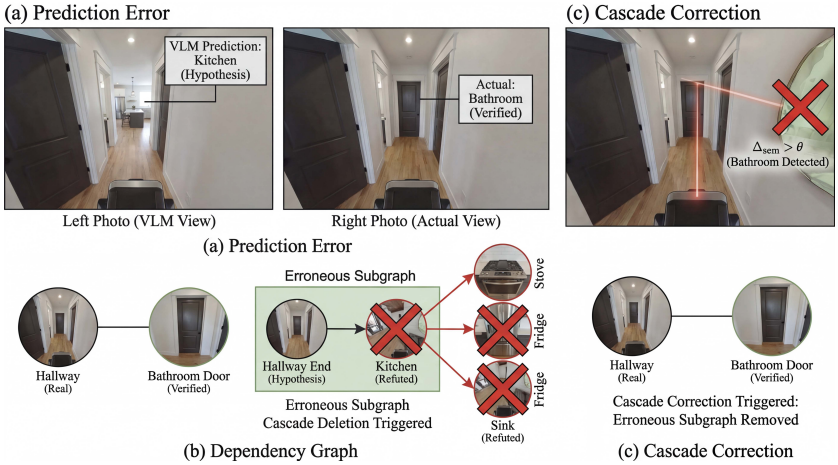

Fig. 4: Cascade Correction Example. (a) Prediction Error: from the hallway, the VLM predicts a *kitchen* beyond the frontier (left, hypothesis), but on arrival the region is actually a *bathroom* (right, verified). **(b) Dependency Graph:** the erroneous kitchen hypothesis and its inferred-object children (stove, fridge, sink) form an erroneous subgraph. **(c) Cascade Correction:** the prediction residual exceeds the threshold ($\Delta_{\text{sem}} > \theta_{\text{refute}}$, bathroom detected), so the system traces the dependency DAG and removes the entire erroneous subgraph, leaving the verified hallway–bathroom structure.

gation goal g:

$$S(f_j; g) = \underbrace{P(c_g \mid \mathcal{H}_t, f_j)}_{\text{goal alignment}} - \underbrace{\lambda_d \cdot d(f_j, \mathbf{p}_t)}_{\text{travel cost}} + \underbrace{\lambda_h \cdot H[P(\cdot \mid \mathcal{H}_t, f_j)]}_{\text{uncertainty bonus}}, \tag{2}$$

where c_g is the target semantic category, $d(\cdot, \cdot)$ measures geodesic distance, and $H[\cdot]$ is Shannon entropy, with travel-cost weight $\lambda_d = 0.15$ and uncertainty weight $\lambda_h = 0.10$. The three terms respectively prioritize goal-matching frontiers, penalize distant ones, and encourage visiting ambiguous regions with high information gain.

3.4 Verification and Cascade Correction

When a hypothesis node is refuted in Phase 2, HGR must not only remove the erroneous node but also retract all downstream nodes derived from it. This section details the prediction residual test that triggers refutation and the cascade correction that propagates it through $\mathcal{D}$. Figure 4 illustrates a representative case: from a hallway, the VLM predicts a kitchen beyond a frontier and generates descendant hypothesis nodes for inferred objects (stove, fridge, sink); upon arrival the region is actually a bathroom, so the semantic violation triggers cascade correction, which traces $\mathcal{D}$ and removes the entire erroneous subgraph.

Prediction Residual Test. Upon reaching a hypothesis node v_j^{hyp}, the agent obtains actual perception $(I_j^{\text{actual}}, \mathcal{O}_j^{\text{actual}})$. The prediction residual quantifies the discrepancy between predicted and observed semantics:

$$\Delta_{\text{sem}}(v_j^{\text{hyp}}) = \omega_c \cdot \Delta_c + \omega_f \cdot \Delta_f + \omega_o \cdot \Delta_o, \tag{3}$$

where:

$$\Delta_c = 1 - \max_k P(c_k \mid \mathcal{H}_t, f_j) \cdot \mathbb{1}[c_k = c_j^{\text{actual}}], \tag{4}$$

$$\Delta_f = 1 - \text{sim}_{\text{CLIP}}(\text{Enc}(I_j^{\text{frontier}}), \text{Enc}(I_j^{\text{actual}})), \tag{5}$$

$$\Delta_o = 1 - \frac{|\mathcal{O}_j^{\text{pred}} \cap \mathcal{O}_j^{\text{actual}}|}{|\mathcal{O}_j^{\text{pred}} \cup \mathcal{O}_j^{\text{actual}}|}, \tag{6}$$

combining category mismatch Δ_c, visual feature divergence Δ_f via CLIP embeddings, and object-level Jaccard dissimilarity Δ_o. Weights are set to $\omega_c = 0.4, \omega_f = 0.3, \omega_o = 0.3$ and the refutation threshold to $\theta_{\text{refute}} = 0.5$. These hyperparameters, together with λ_d and λ_h in Eq. 2, are held fixed across all three benchmarks; only θ_{refute} was tuned, on a 56-subtask held-out split from GOAT-Bench (see the supplementary material for the per-term ablation and threshold-sensitivity sweep). A hypothesis node is *refuted* when $\Delta_{\text{sem}} > \theta_{\text{refute}}$.

Cascade Correction. Upon refuting a hypothesis node v_p, cascade correction performs a breadth-first traversal from v_p along the dependency edges of $\mathcal{D}$ and deletes all reachable descendants, together with their incident edges in $\mathcal{E}$ and $\mathcal{D}$ (full pseudocode in the supplementary material).

Invalidating a single hypothesis thus removes all downstream nodes inferred from the false premise. Unlike confidence attenuation, cascade correction eliminates erroneous subgraphs entirely, so the hypothesis graph may *shrink* over time—a non-monotonic property distinguishing HGR from conventional additive map construction.

4 Experiments

We evaluate HGR on three complementary benchmarks to validate two hypotheses: (H1) semantic hypothesis module improves exploration efficiency, and (H2) cascade correction prevents cumulative errors from degrading performance.

4.1 Experimental Setup

Benchmarks. **(1) GOAT-Bench** [13]: Multimodal lifelong navigation across 360 scenes requiring sequential navigation to multiple targets specified via category labels, language descriptions, or reference images. Episodes span 100+ steps, emphasizing sustained autonomy. This is our primary benchmark. **(2) A-EQA** [19]: Active Embodied Q&A comprising 557 questions across 63 HM3D scenes. Agents

must explore unknown environments to answer open-ended questions spanning object recognition, spatial understanding, and functional reasoning. **(3) EM-EQA** [19]: Episodic Memory Q&A with 1,600+ questions from 152 ScanNet and HM3D scenes. Given pre-explored trajectories, agents construct scene memory and answer questions without further exploration, isolating memory quality from exploration strategy.

Metrics. **Success Rate (SR)** measures goal achievement within distance thresholds (1.0m for navigation, answer correctness for Q&A). **Success weighted by Path Length (SPL)** penalizes inefficient exploration: SPL $= \frac{1}{N}\sum_{i=1}^{N} S_i \cdot \frac{l_i^*}{\max(l_i^*, l_i)}$. For A-EQA, **LLM-Match** evaluates answer quality via GPT-4 scoring (0–100 scale).

Baselines. Comparisons include **3D-Mem** [31] and **ConceptGraph** [8] with frontier-based exploration, **Explore-EQA** [25], the commonsense-guided semantic-frontier methods **ESC** [35] and **SCOPE** [30], and ablated variants. For fair comparison, all baselines are re-implemented within our framework using the same GPT-4o VLM, Habitat simulator, low-level controller, and step budget. ConceptGraph and 3D-Mem retain their original graph construction and update logic but use GPT-4o in place of their original semantic modules; ESC and SCOPE apply their semantic-frontier scoring without dependency tracking or cascade correction. See the supplementary material for full re-implementation details.

Protocol. Unless noted otherwise, all GOAT-Bench and A-EQA results are averaged over 3 random seeds, and reported with mean±std where space permits (per-seed variation is small, e.g. Table 2); EM-EQA uses the provided pre-explored trajectories and is therefore deterministic (single run).

4.2 Main Results: GOAT-Bench

Lifelong Navigation Performance. Table 1 presents results on GOAT-Bench. HGR achieves 72.41% SR and 56.22% SPL on the validation subset (278 subtasks), outperforming the strongest baseline 3D-Mem (69.1%/48.9%) by +3.31% SR and +7.32% SPL.

Comparison with Semantic-Frontier Methods. To isolate the contribution of dependency tracking and cascade correction from semantic frontier scoring alone, we re-implement two commonsense-guided exploration baselines, ESC [35] and SCOPE [30], under our common infrastructure (GPT-4o, Habitat 3.0, DD-PPO controller, identical frontier detection, geodesic planner, and 500-step budget; 3 seeds) on the GOAT-Bench subset. Both perform semantic-frontier scoring but lack dependency tracking and cascade correction. As Table 2 shows, semantic scoring alone improves over the non-semantic 3D-Mem baseline (+1.3% SR / +2.8% SPL for SCOPE), but dependency-aware retraction contributes the bulk

Table 1: Performance on GOAT-Bench. HGR achieves the highest success rate and path efficiency on both the full validation set (2,780 subtasks) and the evaluation subset (278 subtasks).

	Full Validation Set		Subset	
Method	SR ↑	SPL ↑	SR ↑	SPL ↑
ConceptGraph [8]	61.2	44.3	67.8	48.1
3D-Mem w/o memory [31]	58.6	38.5	66.2	44.1
3D-Mem [31]	62.9	44.7	69.1	48.9
HGR (Ours)	**64.14**	**50.1**	**72.41**	**56.22**

Table 2: Comparison with semantic-frontier exploration baselines on the GOAT-Bench subset (278 subtasks, mean±std over 3 seeds). ESC and SCOPE use semantic-frontier scoring but lack dependency tracking and cascade correction.

Method	Semantic	Cascade	SR ↑	SPL ↑
3D-Mem [31]			69.1 ±0.6	48.9 ±0.5
ESC [35] (re-impl.)	✓		69.8 ±0.7	50.6 ±0.6
SCOPE [30] (re-impl.)	✓		70.4 ±0.7	51.7 ±0.6
HGR (Ours)	✓	✓	**72.41** ±0.5	**56.22** ±0.4

of HGR's gain, adding a further +4.5% SPL over the strongest semantic-frontier baseline—consistent with the ablation in Table 4 (w/o correction 68.61/51.12; full 72.41/56.22).

Where Do the Gains Come From? As shown in Figure 5, HGR reaches targets earlier and the advantage grows over longer episodes. Semantic hypothesis module directs exploration toward goal-aligned frontiers, improving SPL, while cascade correction eliminates revisits to erroneously predicted regions.

Modality-Specific Analysis. Table 3 decomposes performance by target modality. HGR shows consistent improvements across all three, with the largest gains on language-specified targets (+7.9% SR over 3D-Mem). Language descriptions benefit most from semantic context propagation, as hypothesis nodes encode relational structure that helps disambiguate spatial references.

4.3 Component Ablations

Table 4 systematically isolates the contributions of each component across all three benchmarks.

Component Contributions. Disabling the semantic hypothesis module (uniform frontier selection) degrades performance by −4.70% SR / −6.20% SPL on GOAT-Bench and −7.5 / −7.8 LLM-Match on A-EQA / EM-EQA, confirming H1.

Table 3: Performance by Target Modality on GOAT-Bench Subset.

	Category		Language		Image	
Method	SR	SPL	SR	SPL	SR	SPL
ConceptGraph [8]	65.3	44.7	55.0	38.9	64.0	52.8
3D-Mem [31]	79.2	55.8	61.9	46.0	65.2	44.2
HGR (Ours)	**80.9**	**60.3**	**69.8**	**54.5**	**68.2**	**61.7**

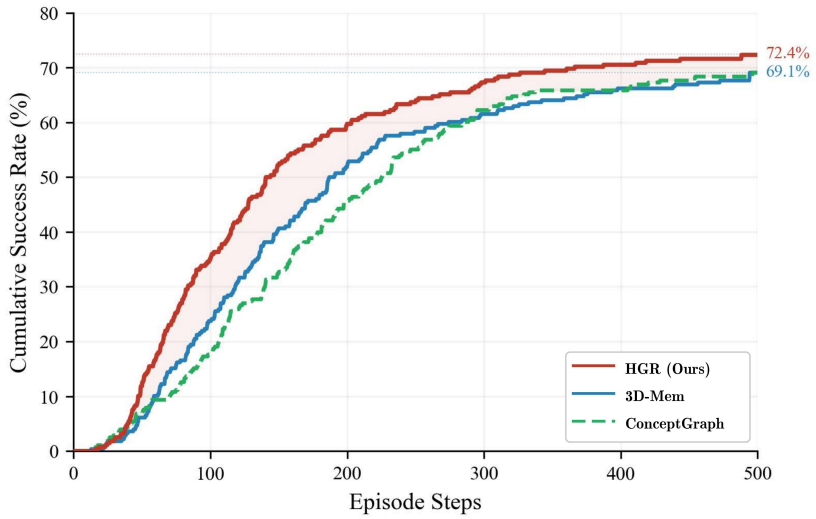

Fig. 5: Cumulative Success Rate vs. Episode Steps. HGR reaches navigation targets earlier than baselines due to hypothesis-driven frontier selection, while 3D-Mem and ConceptGraph require more steps for exhaustive geometric search. The gap widens in later steps as cascade correction prevents error accumulation.

Disabling cascade correction while retaining hypothesis nodes costs −3.80% SR / −5.10% SPL on GOAT-Bench and −6.3 LLM-Match on A-EQA, as refuted nodes persist with lowered confidence. Disabling both is super-additive (−9.0% SR, −10.9% SPL), indicating the two mechanisms reinforce each other: directed exploration brings the agent to verifiable hypotheses that maximize the correction's benefit.

Local Delete vs. Cascade Correction. Removing only the refuted node without DAG propagation ("Local delete only") reaches 70.85% SR / 53.67% SPL—above no correction but −1.56% SR / −2.55% SPL below the full system, showing that descendants of false premises degrade performance and that dependency-aware cascade provides gains beyond simple node removal. The supplementary material

Table 4: Ablation Study. Component contributions on the GOAT-Bench subset and question-answering benchmarks. "Local delete only" removes the refuted node without cascade propagation along the dependency DAG.

	GOAT (Subset)		A-EQA	EM-EQA
Configuration	SR ↑	SPL ↑	LLM ↑	LLM ↑
HGR (full system)	**72.41**	**56.22**	**55.9**	**58.3**
w/o Semantic hypothesis	67.71	50.02	48.4	50.5
w/o cascade correction	68.61	51.12	49.6	52.9
Local delete only	70.85	53.67	52.3	55.9
w/o both (geometry only)	63.42	45.33	43.3	45.2

further shows structural deletion outperforms a soft confidence-decay baseline with revisitation penalty.

Prediction Residual Terms. All three residual terms in Eq. 3 contribute: dropping the category, feature, or object term lowers SR by 1.53%, 1.19%, and 1.37% respectively—the category term most affects refutation precision and the object term most affects recall (full per-term ablation in the supplementary material).

Prediction Residual Threshold. We analyze sensitivity to $\theta_{\text{refute}} \in [0.3, 0.7]$. At $\theta = 0.3$ (aggressive), the refutation rate reaches 42% but incorrectly removes approximately 180 valid nodes, degrading SR by 6.2%. At $\theta = 0.7$ (conservative), only 11% of hypotheses are refuted, allowing errors to accumulate. The optimal $\theta = 0.5$ balances precision (87%) and recall (74%), and SR remains stable within $\pm 1.0\%$ for $\theta \in [0.45, 0.55]$.

Hypothesis Prediction Methods. An entropy-gated hybrid that queries the VLM only for ambiguous frontiers and falls back to heuristics elsewhere attains 95% of VLM-only SPL at 40% of the computational cost; the full prediction-method comparison and open-source VLM results (LLaVA-1.5-13B, InternVL2-8B) appear in the supplementary material.

4.4 Diagnostics: Hypothesis Verification and Correction

We report statistics on hypothesis lifecycle across the GOAT-Bench full validation set (2,780 subtasks).

Hypothesis Lifecycle. Over the full evaluation, HGR created 4,712 hypothesis nodes, of which 3,465 (73.5%) were confirmed upon visitation and transitioned to observed nodes, while 1,247 (26.5%) were refuted and removed. Of the 1,247 refutations, 342 triggered cascade corrections that removed an average of 2.8 nodes per trigger including the refuted node (maximum cascade depth: 4 hops), totalling approximately 960 node removals.

Cascade Events per Trajectory. Cascade corrections are infrequent but consequential: across trajectories, the proportions triggering 0, 1, 2, 3, and ≥ 4 cascade events are 32.4%, 36.0%, 18.0%, 9.4%, and 4.2% (mean 1.23, max 6 per trajectory), and average SPL decreases monotonically with the event count, from 60.5 at zero events to 44.5 at ≥ 4 (full breakdown in the supplementary material).

Revisit Reduction. Cascade correction reduces wasteful revisits to regions based on erroneous predictions. The revisit rate to refuted regions is 4.2% for HGR, compared to 18.7% for 3D-Mem. This 4.5× reduction directly contributes to HGR's SPL improvement.

Error Source Taxonomy. Among the 1,247 refuted hypotheses: mirror reflections (38%), glass partitions (29%), artwork misidentification (18%), and occlusion artifacts (15%). Specular and transparent surfaces account for 67% of all errors, identifying the primary challenge for VLM-based prediction in indoor environments.

Hypothesis Accuracy vs. Performance. We bin trajectories by their hypothesis accuracy r—the fraction of a trajectory's hypotheses confirmed rather than refuted—into low ($r < 0.6$), mid ($0.6 \leq r < 0.8$), and high ($r \geq 0.8$). SPL increases monotonically with r (Pearson 0.42, Spearman 0.45), confirming that more accurate predictions help. Crucially, cascade correction narrows the SPL gap between low- and high-accuracy trajectories from 22.0 without correction to 10.3 with the full system: the dependency DAG is designed to contain prediction errors rather than to rely on perfect predictions, so the higher cascade-event counts reported in the supplementary material reflect harder trajectories rather than harm from correction.

4.5 Secondary Results: A-EQA and EM-EQA

Active Embodied Question Answering. Table 5 presents A-EQA results. HGR achieves 55.9 LLM-Match and 45.0 LLM-Match SPL, outperforming all baselines through more efficient exploration toward question-relevant regions and error-free scene memory.

Episodic Memory Q&A. Table 6 evaluates memory quality in isolation. Given identical pre-explored trajectories, HGR achieves 58.3 LLM-Match with 3.1 average snapshots. Although the exploration path is fixed, HGR still generates hypothesis nodes for visible frontiers as it processes each trajectory step, and verifies them when later steps pass through the predicted regions—so both semantic hypotheses and cascade correction contribute to memory construction quality rather than exploration strategy. The dependency-aware structure enables removal of contradictory information from memory—a capability absent in frame-based representations.

Table 5: Performance on A-EQA (184-question subset). HGR achieves the highest answer quality and exploration efficiency.

Method	LLM-Match ↑	LLM-Match SPL ↑
Blind LLMs (no visual input)		
GPT-4o	35.9	—
Question Agnostic Exploration		
LLaVA-1.5 Frame Captions	38.1	7.0
Multi-Frame	41.8	7.5
VLM-Guided Exploration		
Explore-EQA	46.9	23.4
ConceptGraph w/ Frontier	47.2	33.3
HGR (Ours)	**55.9**	**45.0**

Table 6: Performance on EM-EQA. HGR achieves the highest answer quality with comparable memory footprint.

Method	Avg. Frames	LLM-Match ↑
Blind LLM (GPT-4)	0	35.5
ConceptGraph Captions	0	34.4
Frame Captions	0	38.1
Multi-Frame	3.0	48.1
HGR (Ours)	**3.1**	**58.3**
Human (Full trajectory)	Full	86.8

4.6 Qualitative Analysis

Figure 6 presents a representative episode searching for a "television in the living room." At step 12, the VLM misidentifies a large mirror as an entryway to an adjacent living room, generating hypothesis nodes for a couch and TV. Upon reaching the mirror (step 15, $\Delta_{\text{sem}} = 0.72 > 0.5$), cascade correction removes the false living room subgraph, redirecting exploration to the actual living room (steps 16–22). In contrast, 3D-Mem retains the erroneous nodes with reduced confidence, causing repeated revisits and step-limit failure.

5 Conclusion

We present Hypothesis Graph Refinement (HGR), which enables semantic prediction for efficient exploration while preventing cumulative error accumulation. On GOAT-Bench, HGR achieves 72.41% SR and 56.22% SPL, outperforming 3D-Mem by +3.31% and +7.32%, with consistent gains on A-EQA and EM-EQA, and ablations confirm both mechanisms contribute. Diagnostic analysis

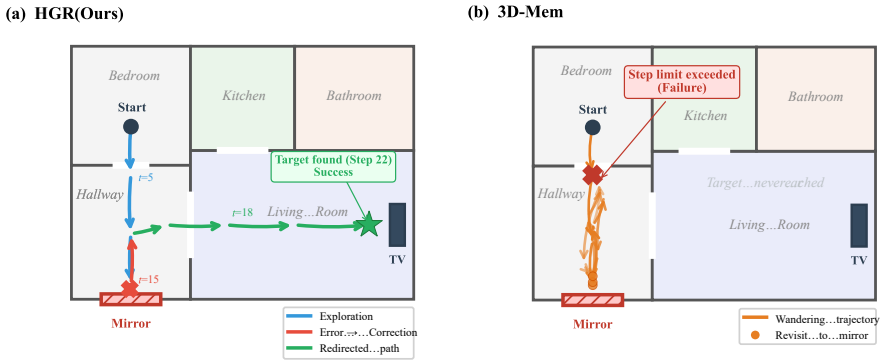

Fig. 6: Qualitative Comparison: Mirror-Induced Prediction Error. Left: HGR detects the prediction violation and removes the erroneous subgraph via cascade correction. Right: 3D-Mem retains erroneous nodes with lowered confidence, leading to repeated misnavigation.

attributes 67% of prediction errors to specular and transparent surfaces, a concrete target for improving VLM robustness indoors. Runtime and memory analysis are in the supplementary material.

The main limitation is that cascade correction inherits the accuracy of the prediction residual test—false negatives let erroneous nodes persist and false positives may prune valid hypotheses—and prediction quality is ultimately bounded by the underlying VLM, which struggles with mirrors, small, proximal objects, and spatial relationships among multiple objects, the dominant error source in our diagnostics (see the supplementary material). Future work includes improving residual calibration and extending the dependency structure to dynamic environments.

References

1. An, D., Wang, H., Wang, W., Wang, Z., Huang, Y., He, K., Wang, L.: Etpnav: Evolving topological planning for vision-language navigation in continuous environments. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
2. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3674–3683 (2018)
3. Armeni, I., He, Z.Y., Gwak, J., Zamir, A.R., Fischer, M., Malik, J., Savarese, S.: 3d scene graph: A structure for unified semantics, 3d space, and camera. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5664–5673 (2019)
4. Chakraborty, T., Ghosh, U., Zhang, X., Niloy, F.F., Dong, Y., Li, J., Roy-Chowdhury, A.K., Song, C.: HEAL: An empirical study on hallucinations in em-

bodied agents driven by large language models. In: Findings of the Association for Computational Linguistics: EMNLP 2025 (2025)
5. Deng, Z., Narasimhan, K., Russakovsky, O.: Evolving graphical planner: Contextual global planning for vision-and-language navigation. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33 (2020)
6. Fan, C., Jia, X., Sun, Y., Wang, Y., Wei, J., Gong, Z., Zhao, X., Tomizuka, M., Yang, X., Yan, J., Ding, M.: Interleave-VLA: Enhancing robot manipulation with interleaved image-text instructions. arXiv preprint arXiv:2505.02152 (2025)
7. Gadre, S.Y., Wortsman, M., Ilharco, G., Schmidt, L., Song, S.: CoWs on Pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
8. Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K.M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., Gan, C., de Melo, C.M., Tenenbaum, J.B., Torralba, A., Shkurti, F., Paull, L.: ConceptGraphs: Open-vocabulary 3d scene graphs for perception and planning. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2024)
9. Heo, K., Kim, G., Kim, S., Cho, M.: Object-centric representation learning for enhanced 3d semantic scene graph prediction. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)
10. Hong, Y., Wu, Q., Qi, Y., Rodriguez-Opazo, C., Gould, S.: Vln bert: A recurrent vision-and-language bert for navigation. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 1643–1653 (2021)
11. Huang, Y., Wu, M., Li, R., Tu, Z.: VISTA: Generative visual imagination for vision-and-language navigation. arXiv preprint arXiv:2505.07868 (2025)
12. Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., et al.: Language models (mostly) know what they know. arXiv preprint arXiv:2207.05221 (2022)
13. Khanna, M., Ramrakhya, R., Chhablani, G., Yenamandra, S., Gervet, T., Chang, M., Kira, Z., Chaplot, D.S., Batra, D., Mottaghi, R.: GOAT-Bench: A benchmark for multi-modal lifelong navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16373–16383 (2024)
14. Ku, A., Anderson, P., Patel, R., Ie, E., Baldridge, J.: Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4392–4412 (2020)
15. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023)
16. Liu, J., Guo, J., Meng, Z., Xue, J.: Revolt: Relational reasoning and voronoi local graph planning for target-driven navigation. arXiv preprint arXiv:2301.02382 (2023)
17. Liu, S., Du, J., Xiang, S., Wang, Z., Luo, D.: Long-horizon embodied planning with implicit logical inference and hallucination mitigation. arXiv preprint arXiv:2409.15658 (2024)
18. Majumdar, A., Aggarwal, G., Devnani, B., Hoffman, J., Batra, D.: ZSON: Zero-shot object-goal navigation using multimodal goal embeddings. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)

19. Majumdar, A., Ajay, A., Zhang, X., Putta, P., Yenamandra, S., Henaff, M., Silwal, S., Mcvay, P., Maksymets, O., Arnaud, S., Yadav, K., Li, Q., Newman, B., Sharma, M., Berges, V., Zhang, S., Agrawal, P., Bisk, Y., Batra, D., Kalakrishnan, M., Meier, F., Paxton, C., Sax, A., Rajeswaran, A.: OpenEQA: Embodied question answering in the era of foundation models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16488–16498 (2024)
20. Manakul, P., Liusie, A., Gales, M.J.F.: SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 9004–9017 (2023)
21. Min, S.Y., Chaplot, D.S., Ravikumar, P., Bisk, Y., Salakhutdinov, R.: Film: Following instructions in language with modular methods. arXiv preprint arXiv:2110.07342 (2021)
22. Monaci, G., de Rezende, R.S., Deffayet, R., Csurka, G., Bono, G., Déjean, H., Clinchant, S., Wolf, C.: Rana: Retrieval-augmented navigation. Trans. Mach. Learn. Res. **2025** (2025)
23. Nasiriany, S., Xia, F., Yu, W., Xiao, T., Liang, J., Dasgupta, I., Xie, A., Driess, D., Wahid, A., Xu, Z., Vuong, Q., Zhang, T., Lee, T.W.E., Lee, K.H., Xu, P., Kirmani, S., Zhu, Y., Zeng, A., Hausman, K., Heess, N., Finn, C., Levine, S., Ichter, B.: PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs. In: Proceedings of the 41st International Conference on Machine Learning. pp. 37321–37341. Proceedings of Machine Learning Research, PMLR (2024)
24. Qi, Z., Zhang, Z., Yu, Y., Wang, J., Zhao, H.: VLN-R1: Vision-language navigation via reinforcement fine-tuning. arXiv preprint arXiv:2506.17221 (2025)
25. Ren, A.Z., Clark, J., Dixit, A., Itkina, M., Majumdar, A., Sadigh, D.: Explore until confident: Efficient exploration for embodied question answering. In: Robotics: Science and Systems (RSS) (2024)
26. Saxena, S., Buchanan, B., Paxton, C., Liu, P., Chen, B., Vaskevicius, N., Palmieri, L., Francis, J., Kroemer, O.: GraphEQA: Using 3d semantic scene graphs for real-time embodied question answering. In: Proceedings of the 9th Conference on Robot Learning (CoRL) (2025)
27. Shah, D., Equi, M., Osinski, B., Xia, F., Ichter, B., Levine, S.: Navigation with large language models: Semantic guesswork as a heuristic for planning. In: Proceedings of the Conference on Robot Learning (CoRL) (2023)
28. Sun, Z., Shen, S., Cao, S., Liu, H., Li, C., Shen, Y., Gan, C., Gui, L., Wang, Y.X., Yang, Y., Keutzer, K., Darrell, T.: Aligning large multimodal models with factually augmented RLHF. In: Findings of the Association for Computational Linguistics: ACL 2024. pp. 13088–13110. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024)
29. Wald, J., Dhamo, H., Navab, N., Tombari, F.: Learning 3d semantic scene graphs from 3d indoor reconstructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3961–3970 (2020)
30. Wang, N., Chen, W., Chen, L., Ji, H., Guo, Z., Zhang, X., Sun, H.: Expand your scope: Semantic cognition over potential-based exploration for embodied visual navigation. arXiv preprint arXiv:2511.08935 (2025)
31. Yang, Y., Yang, H., Zhou, J., Chen, P., Zhang, H., Du, Y., Gan, C.: 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)

32. Yu, F., Deng, Q., Tang, S., Li, Y., Cheng, L.: Open-world 3d scene graph generation for retrieval-augmented reasoning. arXiv preprint arXiv:2511.05894 (2025)
33. Zemskova, T., Yudin, D.A.: 3DGraphLLM: Combining semantic graphs and large language models for 3d scene understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
34. Zhang, Z., Zhu, W., Pan, H., Wang, X., Xu, R., Sun, X., Zheng, F.: ActiveVLN: Towards active exploration via multi-turn RL in vision-and-language navigation. arXiv preprint arXiv:2509.12618 (2025)
35. Zhou, K., Zheng, K., Pryor, C., Shen, Y., Jin, H., Getoor, L., Wang, X.E.: Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In: International Conference on Machine Learning. pp. 42829–42842. PMLR (2023)