PredNav: A Predictive Cognitive Map for Zero-Shot Object Goal Navigation

Anonymous ACL submission

Abstract

Zero-shot Object Goal Navigation requires an embodied agent to locate a target object in an unseen environment, where the choice of which frontier to explore next is critical to both success and efficiency. Existing methods score frontiers from already-observed cues such as nearby objects or scene graphs, leaving the structure beyond each frontier implicit and offering no principled prior for choosing among rooms the agent has yet to see. In contrast, the hippocampal entorhinal system supports navigation through a predictive cognitive map that anticipates unseen space from partial observations. Inspired by this, we propose **PredNav**, which externalizes a hierarchical cognitive map with metric, instance-semantic, and topological layers, and casts frontier selection as room-level structured prediction over its topological layer. A language model infers the likely room type and target relevance behind each frontier, which is fused with CLIP visual evidence, an offline room category prior, and geodesic cost under phase-adaptive weights. Extensive experiments on MP3D and HM3D demonstrate the effectiveness of our method.

1 Introduction

Object Goal Navigation (ObjectNav) requires an embodied agent to locate a target object category in an unseen indoor environment, and has become a central benchmark for embodied intelligence. Among zero-shot approaches, frontier-based exploration is widely adopted: agents build a map, identify boundaries between explored and unexplored regions, and select frontiers as subgoals. Since this selection directly determines where the agent explores next, frontier scoring is crucial to both episode success and navigation efficiency. Recent methods therefore mainly differ in how they score frontiers, using signals such as CLIP-based value maps (Yokoyama et al., 2024), language-based commonsense reasoning (Yu et al., 2023; Zhou et al., 2023), online scene graphs (Yin et al., 2024), and dense belief maps (Zhou et al., 2026).

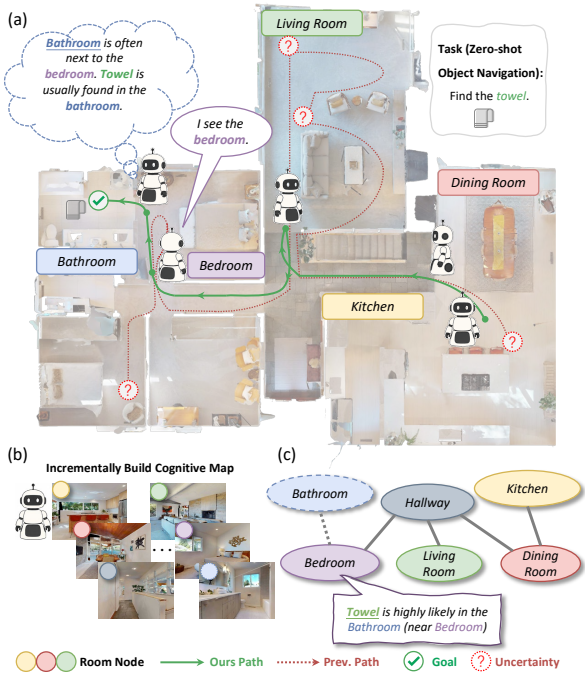

Figure 1: Zero-shot ObjectNav (*"Find the towel"*). (a) Prior methods react only to observations and wander through irrelevant rooms (red), while PredNav predicts unseen areas from commonsense (*bathroom* likely adjoins *bedroom*) and reaches the goal directly (green). (b) A cognitive map is built incrementally during navigation. (c) Its topological layer encodes observed rooms and predicts unobserved ones (dashed).

Most existing scoring functions act as *local mappings of already observed information*, aggregating nearby pixels, neighbouring categories, scene-graph nodes, or voxel features into scalar scores. Methods that predict unobserved regions typically operate at subroom levels, such as pixel semantic completion (Liang et al., 2021), frontier potential fields (Ramakrishnan et al., 2022), voxel belief maps (Zhou et al., 2026), or VLM-driven view imagination (Zhao et al., 2025). These representations leave cross-room layout structure implicit,

rather than predicting it directly. Neither line of work explicitly addresses the question most critical to ObjectNav: *what room lies beyond this frontier, and what objects are likely to appear there?* Even structured local scoring methods such as SG-Nav (Yin et al., 2024) scores observed subgraphs with an LLM and propagates the scores to nearby frontiers, reasoning over what has been seen rather than what lies beyond. Consequently, an agent facing frontiers toward a bedroom, kitchen, and dining room lacks a principled basis for choosing the route most likely to lead to the target bathroom (Figure 1a). The bottleneck is therefore not simply a missing scoring feature, but the absence of an explicit representation of cross-room relational structure.

Such a representation finds a natural analogue in the neuroscience of biological spatial navigation. Prior work suggests that goal-directed navigation relies on a predictive, hierarchical cognitive map in the hippocampal entorhinal system, rather than reactive geometric alignment. Behrens et al. (2018) describe this map as a *factorised structural representation* that supports inference over unobserved states and transitions implied by spatial structure. Computational studies further suggest a hybrid hierarchical map with metric, semantic, and topological layers as a plausible model for embodied agents(Manh et al., 2025; Hao et al., 2026). This perspective reframes frontier scoring as structured prediction over the unobserved part of a cognitive map, rather than local scoring over observed cues.

A natural alternative is to delegate this reformulation to a large language model by textualizing the observations and asking which frontier leads to the target. However, recent evaluations show that LLMs do not reliably construct or traverse cognitive maps from text, often producing edge hallucinations and looping trajectories in tasks requiring latent relational reasoning (Momennejad et al., 2023). Foundation models also exhibit *belief inertia* when maintaining spatial maps internally, resisting updates when new observations contradict prior assumptions (Zhang et al., 2026). These findings motivate *externalizing* the cognitive map by constructing it explicitly from sensor observations, thereby providing the relational structure that LLMs struggle to synthesize and enabling explicit revision when predictions prove incorrect.

Building on this insight, we propose **PredNav**, an ObjectNav framework that uses LLM-based prediction to guide exploration over an externalized cognitive map. Rather than directly reasoning over raw observations or only scoring already observed scene graphs, PredNav explicitly represents both explored regions and candidate frontiers in a structured map (Figure 1b, c), allowing the LLM to infer what semantic space may lie beyond the current observation. These predictions are combined with visual evidence, spatial priors, and navigation costs to select frontiers that are more likely to lead to the target object, and a vision-language verifier adjudicates the candidate target from multiple views before STOP. In this way, PredNav constrains LLM reasoning to a lightweight, single-step prediction problem over unseen space, avoiding open-ended planning while still leveraging commonsense knowledge for efficient object-goal navigation.

Contributions. 1) We propose **PredNav**, a hierarchical navigation framework that combines metric, semantic, and topological map representations with multi-signal frontier evaluation, integrating language-based prediction, visual evidence, spatial priors, and navigation cost in a unified decision process. **2)** We reformulate frontier scoring in zero-shot ObjectNav as room-level structured prediction over an externalized topological cognitive map, enabling the agent to reason about likely unseen spaces rather than only scoring observed regions. **3)** We conduct extensive evaluations on MP3D and HM3D, where PredNav achieves state-of-the-art performance in success rate and SPL and ablation studies further validate the contribution and robustness of its key components.

2 Related Work

Zero-shot Object Goal Navigation and frontier scoring. Zero-shot ObjectNav methods commonly adopt frontier-based exploration, while differing primarily in how frontiers are evaluated. One line of work assigns frontier scores from visual evidence observed near candidate regions. For example, CoW (Gadre et al., 2023) projects CLIP-based relevance onto a top-down map, ZSON (Majumdar et al., 2022) aligns image and object-goal embeddings within an end-to-end RL framework, and VLFM (Yokoyama et al., 2024) constructs a BEV value map using target-conditioned vision-language queries. Another line of work incorporates language-based commonsense reasoning over observed semantic structures. L3MVN (Yu et al., 2023) verbalizes semantic maps for LLM-based

frontier ranking, ESC (Zhou et al., 2023) models object co-occurrence with probabilistic soft logic, while SG-Nav (Yin et al., 2024), UniGoal (Yin et al., 2025), CogNav (Cao et al., 2025), and VoroNav (Wu et al., 2024) reason over online scene graphs or topological representations and transfer the resulting scores to nearby frontiers. A further group delegates high-level planning to language models through textual ReAct-style interaction (Zhou et al., 2024), scene-graph-grounded plans (Rajvanshi et al., 2024), multi-source value-map fusion (Long et al., 2025), or floor-aware coarse-to-fine search (Gong et al., 2026). Despite these differences, existing approaches mainly condition language reasoning on already observed content, and frontier values are often obtained by propagating semantic scores according to geometric proximity. In contrast, our work explicitly models frontiers as interfaces to unseen rooms and predicts their likely semantic outcomes from a structured representation of cross-room layout.

Predictive representations beyond the observed region. A complementary line of work explicitly predicts unobserved regions, but typically through lower-level representations. SSCNav (Liang et al., 2021) learns semantic completion of top-down maps with an auxiliary confidence head, while PONI (Ramakrishnan et al., 2022) predicts frontier-level potential fields for exploration coverage and object presence from full-floor layouts without interaction. L2M (Georgakis et al., 2021) further incorporates ensemble uncertainty and UCB-based long-term goal selection. Recent zero-shot methods extend this direction to 3D beliefs or imagined observations. BeliefMapNav (Zhou et al., 2026) combines LLM-derived target-adaptive priors with a visibility-weighted voxel belief map for global routing, whereas ImagineNav (Zhao et al., 2025) replaces explicit mapping with VLM-based selection over synthesized future views. While these methods provide explicit predictions over unseen space, they operate on pixels, local frontier potentials, voxels, or synthesized views, which are not directly aligned with the room-level decisions required for long-range indoor navigation. PredNav instead formulates prediction at this decision level by performing room-level structured inference over an explicit topological cognitive map.

LLM spatial-reasoning limits and externalised cognitive maps. The need to ground room-level prediction in an explicit map, rather than relying on the language model as an implicit spatial memory, has been increasingly recognized. CogEval (Momennejad et al., 2023) shows that frontier LLMs often hallucinate connectivity and fail in cognitive-map planning. VSI-Bench (Yang et al., 2025a) further attributes many MLLM errors in video-based spatial question answering to spatial reasoning rather than perception, and shows that prompting models to construct a coarse cognitive map can substantially improve relative spatial reasoning. MindCube (Wang et al., 2025) reports similar findings for limited-view spatial reasoning, where map-then-reason training significantly improves VLM performance. Cambrian-S (Yang et al., 2025b) also suggests that increasing token budget, context length, or model scale alone is insufficient for long-horizon spatial reasoning, while foundation models may exhibit belief inertia when observations contradict their internal spatial priors (Zhang et al., 2026). These findings motivate an embodied alternative in which spatial memory is externalized and explicitly maintained. Recent surveys of agentic spatial intelligence (Manh et al., 2025; Hao et al., 2026) advocate hybrid representations that combine metric, instance-level, and topological structures, consistent with factorized spatial representations studied in neuroscience (Behrens et al., 2018; Stachenfeld et al., 2017). BSC-Nav (Ruan et al., 2025) follows this direction by introducing landmark, route, and survey memory for MLLM-based navigation. PredNav adopts the same principle of externalized spatial memory, but constrains the role of the LLM more tightly. Instead of asking the model to perform general spatial planning, PredNav uses the topological layer of the maintained map as the substrate for single-step room-level prediction at each frontier. The resulting prediction is then integrated with visual observation, spatial prior, and navigation cost, and is updated through prediction-observation consistency as exploration proceeds.

3 Method

3.1 Overview

PredNav is a zero-shot ObjectGoal Navigation agent in the standard Habitat ObjectNav setting (Savva et al., 2019). Given a target category g, the agent receives RGB-D-pose observations and selects actions from the default Habitat action space until it reaches the step budget or issues STOP. As shown in Figure 2, PredNav follows a pipeline of

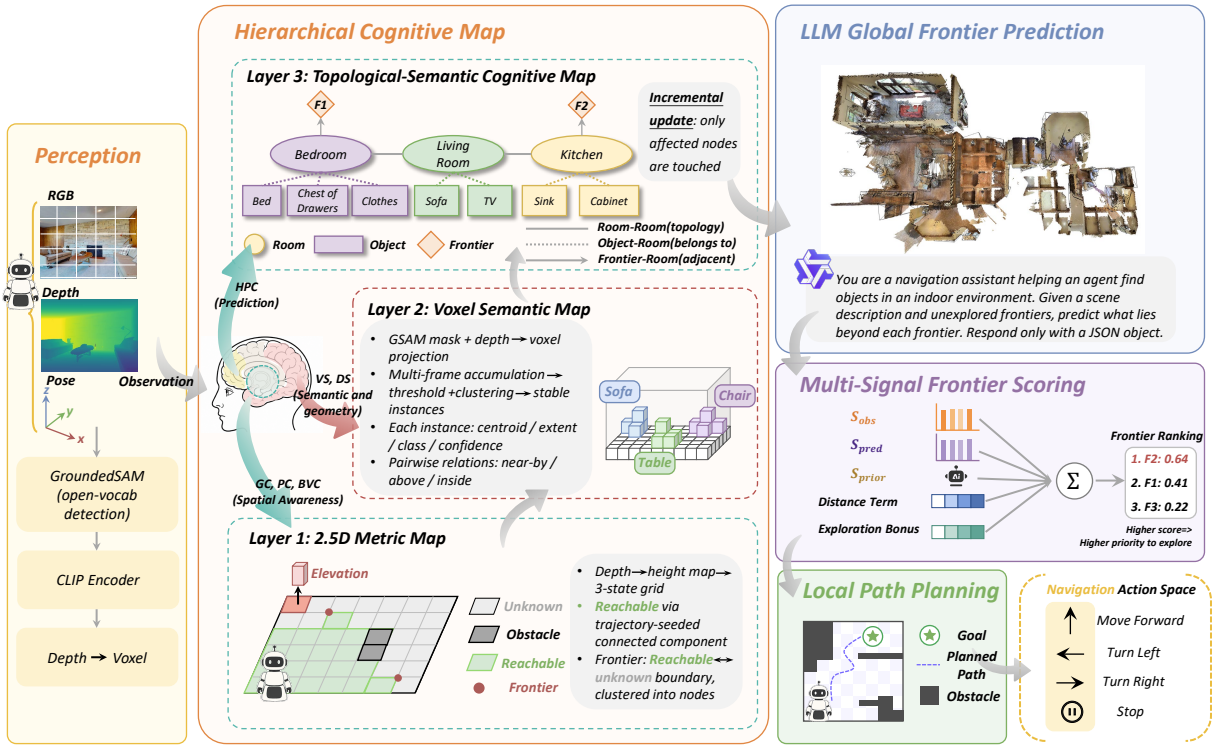

Figure 2: The PredNav pipeline. Per-frame observations update a three-layer cognitive map (metric, instance-semantic, topological). The LLM takes the serialized topological layer to predict each frontier's room type and target relevance; this prediction is fused with a CLIP observation score, an offline room–category prior, and a distance term. The selected frontier is passed to a path planner for action selection, while a vision-language verifier gates STOP. Abbreviations: grid cell (GC), place cell (PC), boundary vector cell (BVC), ventral visual pathway (VS), dorsal visual pathway (DS), hippocampus (HPC).

perception, map maintenance, frontier prediction, frontier scoring, and verification-gated termination. Its core representation is a brain-inspired external cognitive map with metric, instance-semantic, and topological-semantic layers. The topological layer is serialized for LLM-based frontier prediction, whose output is fused with visual evidence, spatial priors, and geodesic cost to select a frontier subgoal. The selected subgoal is executed by an FM^2 planner (Valero-Gomez et al., 2013), and a multi-view vision-language verifier confirms the target before STOP.

3.2 Hierarchical Cognitive Map

The cognitive map is factorized into three layers updated from depth and pose, open-vocabulary detections with CLIP embeddings, and confirmed objects with frontiers, respectively. This metric, instance, and topological decomposition follows biologically inspired accounts of embodied spatial intelligence (Manh et al., 2025; Hao et al., 2026).

Metric layer. The metric layer maintains a 2.5D height map $H_t : \mathbb{Z}^2 \to \mathbb{R} \cup \{\perp\}$ by projecting depth points onto a top-down grid and storing the maximum observed elevation per cell (Miki et al., 2022), where $\perp$ denotes unobserved cells. It is converted into a traversability grid,

$$G_t(c) \in \{\text{UNKNOWN}, \text{OBSTACLE}, \text{REACHABLE}\},$$

where reachability requires both local walkability and connectivity to the agent. Frontiers $\mathcal{F}_t$ are connected boundaries between REACHABLE and UNKNOWN regions, represented by their centroids and information gain $\mathrm{IG}(f)$.

Instance-semantic layer. The instance-semantic layer aggregates open-vocabulary detections into object instances. Each occupied voxel stores category votes, a mask-pooled CLIP embedding, and supporting time indices. Confident voxels are clustered by DBSCAN (Ester et al., 1996) into candidate instances with category g_o, position, and pooled embedding. A candidate is confirmed as an object node o when

$$\hat{n}(o) = \max\big(n_v(o), n_s(o)\big) \qquad (1)$$

exceeds the category-specific threshold k_{g_o}, where $n_v(o)$ is the maximum voxel vote count and $n_s(o)$ is the number of distinct supporting frames. Unconfirmed candidates matching the target category are retained only as provisional evidence.

Topological-semantic layer. The topological-semantic layer is the only layer exposed to the LLM. It represents rooms, confirmed objects, and frontiers as a labelled multigraph with *contains*, *adjacent*, and *borders* edges. Rooms are segmented from the reachable area at doorway-like bottlenecks following (Werby et al., 2024) and labelled from contained object categories using a deterministic affinity table. For frontier prediction, the graph is serialized into a compact text snapshot containing room types, objects, qualitative room adjacencies, and frontier attachments, grounding the LLM in an external spatial representation.

3.3 Predictive Frontier Scoring

PredNav ranks frontiers by jointly considering CLIP-based local observation, LLM-based room-level prediction, offline room-category prior, geodesic-distance cost, and exploration bonus.

LLM frontier prediction. Given the serialized topological-semantic map and target category g, the LLM predicts for each frontier $f \in \mathcal{F}_t$ a tuple $(\hat{r}(f), c(f), \nu(f))$, where $\hat{r}(f)$ denotes the likely room type beyond the frontier, $c(f) \in [0,1]$ is the confidence of this room prediction, and $\nu(f) \in [0,1]$ estimates the probability that the target lies behind the frontier. The prompt also includes frontier-level attributes such as distance, agent-relative direction, size class, and the room currently bordering the frontier. The response is constrained by a strict JSON schema; invalid responses fall back to $(\text{UNKNOWN}, 0.5, 0.5)$. To limit latency and avoid using the LLM as a persistent world model, prediction is invoked only when the map changes substantially, such as when a new object or room is added, or when the current subgoal becomes invalid.

Multi-signal frontier scoring. For each frontier f, PredNav computes

$$s(f) = \sum_{i \in \{\text{obs, pred, prior}\}} w_i S_i(f) + b(f) + \tau(d(f)). \tag{2}$$

Here, $S_{\text{obs}}(f)$ is the maximum CLIP similarity between the target text embedding and confirmed objects near f, capturing direct visual evidence around the frontier. The prediction score is

$$S_{\text{pred}}(f) = c(f)\nu(f),$$

which combines the LLM's confidence in the predicted room type with its estimate of target presence. The prior score is

$$S_{\text{prior}}(f) = P(g \mid \hat{r}(f)),$$

obtained from an offline room-category prior; if $\hat{r}(f) = \text{UNKNOWN}$, the prior falls back to the observed room associated with the frontier, and then to a marginal constant if needed. The information-gain bonus is

$$b(f) = \beta \log(1 + \text{IG}(f)),$$

and the distance term is

$$\tau(d) = \lambda_d \left(1 - \text{clip}\left(\frac{d}{d_{\max}}, 0, 1\right)\right),$$

where $d(f)$ is the FM^2 geodesic distance over reachable cells. Planner-unreachable frontiers are removed from consideration.

Phase-adaptive weighting. Because the reliability of each signal changes during exploration, PredNav uses phase-adaptive weights. Let $\eta = t/T_{\max}$ denote normalized episode progress. The fusion weights are defined as

$$\mathbf{w}(\eta) \triangleq (w_{\text{obs}}, w_{\text{pred}}, w_{\text{prior}}). \tag{3}$$

$$\mathbf{w}(\eta) = \begin{cases} \mathbf{w}^{\text{EARLY}}, & \eta < \eta_1, \\ \mathbf{w}^{\text{MID}}, & \eta_1 \le \eta < \eta_2, \\ \mathbf{w}^{\text{LATE}}, & \eta \ge \eta_2. \end{cases} \tag{4}$$

Early exploration relies more on prediction and priors, while later stages shift toward observation as more objects are confirmed. The highest-scoring reachable frontier is selected as the subgoal and executed by an FM^2 planner.

3.4 Multi-View Target Verification

The STOP action is gated by a separate multi-view verifier. Verification is triggered when the agent approaches a confirmed or provisional target candidate. The agent moves to a standoff pose that frames the candidate, collects several bearing-diverse RGB views with the candidate bounding box annotated, and sends them to a vision-language model under a constrained JSON schema. The

Table 1: Comparison of methods on MP3D and HM3D benchmarks.

Method	Unsup.	Zero-Shot	Foundation-Model Backbone		MP3D		HM3D	
			Vision	Language	SR ↑	SPL ↑	SR ↑	SPL ↑
SemEXP (2020)	✗	✗	–	–	36.0	14.4	–	–
PONI (2022)	✗	✗	–	–	31.8	12.1	–	–
SGM (2024)	✗	✗	–	–	37.7	14.7	–	–
ZSON (2022)	✗	✓	CLIP	–	15.3	4.8	–	–
PSL (2024)	✗	✓	–	–	18.9	6.4	–	–
CoW (2023)	✓	✓	CLIP	–	7.4	3.7	–	–
ESC (2023)	✓	✓	GLIP	GPT-3.5	28.7	14.2	–	–
L3MVN (2023)	✓	✓	–	GPT-2	34.9	14.5	36.3	15.7
VLFM (2024)	✓	✓	BLIP-2	–	36.4	17.5	63.6	**32.5**
InstructNav (2025)	✓	✓	BLIP-2	–	–	–	58.0	20.9
SG-Nav (2024)	✓	✓	LLaVA-1.6-7B	GPT-4	40.2	16.0	49.6	25.5
UniGoal (2025)	✓	✓	LLaVA-1.6-7B	LLaMA-2-7B	41.0	16.4	–	–
OpenFMNav (2024)	✓	✓	GPT-4V	GPT-4	37.2	15.7	–	–
ActPept (2024)	✓	✓	GraphSAGE	–	39.8	17.4	–	–
BeliefMapNav (2026)	✓	✓	CLIP	GPT-4	37.3	17.6	–	–
PredNav (Ours)	✓	✓	Qwen3.5-9B	Qwen3.5-9B	**43.4**	**17.8**	**65.3**	28.5

verifier returns the object name, confidence, and visibility. Its output is mapped to PASS, FAIL, or BORDERLINE: PASS issues STOP, FAIL removes the candidate, and BORDERLINE triggers acquisition of an additional view. Thus, termination is based on direct multi-view visual evidence rather than on the frontier prediction that guided exploration.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate PredNav on the two standard Habitat (Savva et al., 2019) ObjectNav benchmarks. Matterport3D (MP3D) (Chang et al., 2017) provides 2195 validation episodes spanning 11 scenes and a 21-category target set; HM3D-Sem-v0.2 (Ramakrishnan et al., 2021) provides 1000 validation episodes spanning 20 scenes and a 6-category set (*chair*, *bed*, *plant*, *toilet*, *tv_monitor*, *sofa*). All episodes use the Habitat default action set, a 500-step budget per episode.

Metrics. We report success rate (SR), success-weighted path length (SPL), and the soft-SPL variant that credits partial progress on failed episodes (SoftSPL) (Anderson et al., 2018). SR captures task completion and SPL captures path efficiency.

Implementation. The main configuration uses GroundedSAM (GroundingDINO + SAM) (Ren et al., 2024) as the open-vocabulary detector and Qwen3.5-9b (Qwen Team, 2026) as both the frontier-prediction LLM and the target-verification VLM. CLIP ViT-B/32 (Radford et al., 2021) computes the observation score S_{obs}. All other hyperparameters are constant across datasets and are listed in Appendix G.

Baselines. We compare against three groups of recent ObjectNav systems. Supervised baselines include SemExp (Chaplot et al., 2020), PONI (Ramakrishnan et al., 2022), and SGM (Zhang et al., 2024). Closed-vocabulary zero-shot baselines include ZSON (Majumdar et al., 2022) and PSL (Sun et al., 2024). Open-vocabulary zero-shot baselines, which are the most relevant comparison for PredNav, include CoW (Gadre et al., 2023), ESC (Zhou et al., 2023), L3MVN (Yu et al., 2023), VLFM (Yokoyama et al., 2024), InstructNav (Long et al., 2025), SG-Nav (Yin et al., 2024), UniGoal (Yin et al., 2025), OpenFMNav (Kuang et al., 2024), ActPept (Guo et al., 2024), and BeliefMapNav (Zhou et al., 2026).

4.2 Main Results

Results on MP3D. Table 1 shows that PredNav achieves 43.4 SR and 17.8 SPL on MP3D val, outperforming all prior open-vocabulary zero-shot methods on this benchmark. Strong recent baselines, including UniGoal (41.0 SR), SG-Nav (40.2),

Table 2: Component ablation on MP3D.

LLM Prediction	VLM Verification	SR ↑	SPL ↑	SoftSPL ↑
✗	✗	38.9	17.3	23.1
✓	✗	39.5	17.3	22.8
✗	✓	42.2	**18.1**	**23.2**
✓	✓	**43.4**	17.8	22.7

Table 3: Detector ablation on MP3D.

Object Detector	SR ↑	SPL ↑	SoftSPL ↑
YOLO-World	36.3	17.5	23.9
OWLv2	44.3	18.3	23.2
G-DINO	43.4	17.8	22.7
Ideal	**61.4**	**29.7**	**33.4**

ActPept (39.8), and BeliefMapNav (37.3), rely on local LLM-based scoring over observed subgraphs or voxel features and propagate these scores to frontiers via geometric proximity. In contrast, PredNav performs language-model reasoning at the room-level granularity, which better aligns with high-level navigation decisions and yields a $+2.4$ SR improvement over UniGoal. PredNav also improves SPL by $+1.4$, indicating that the higher success rate is achieved without sacrificing path efficiency.

Results on HM3D-v0.2. On the full HM3D-Sem-v0.2 validation split with 1000 episodes, PredNav achieves 65.3 SR and 28.5 SPL. This is the highest SR among open-vocabulary zero-shot methods reported in Table 1, exceeding VLFM by $+1.7$ SR and InstructNav by $+7.3$ SR, the two strongest baselines on this benchmark. Although VLFM obtains a higher SPL, PredNav delivers the best success rate, suggesting that room-level prediction is particularly effective for identifying promising exploration targets in the HM3D environments.

4.3 Ablation Studies

We ablate the two language-grounded components (LLM frontier prediction, VLM target verification) and the open-vocabulary detector. All ablations are on the full MP3D val environment (2195 episodes) with the same seed.

Components. Table 2 evaluates the effect of the proposed language-guided prediction and verification design under the same navigation pipeline. Compared with the baseline without these two components, introducing LLM-based frontier prediction improves SR from 38.9 to 39.5, showing that room-level semantic prediction provides useful guidance for frontier selection. When combined with VLM verification, PredNav further improves SR to 43.4, achieving the best overall performance. This demonstrates that the predicted frontier prior and the subsequent visual verification form an effective closed-loop process: LLM prediction first identifies semantically promising exploration directions, and verification then validates candidate object observations before stopping. The slight reduction in SPL and SoftSPL compared with some partial variants reflects the additional verification behavior, which improves decision reliability while introducing a modest path-efficiency trade-off.

Detector. Table 3 evaluates the effect of different object detectors. For fair comparison with prior methods, we adopt G-DINO as the default detector in PredNav. Among real detectors, OWLv2 (Minderer et al., 2023) achieves the best performance with 44.3 SR and 18.3 SPL, slightly outperforming the G-DINO baseline. YOLO-World (Cheng et al., 2024) performs worse, likely because some MP3D categories, such as *cabinet*, *shower*, and *towel*, are less well covered by its training vocabulary. Using an ideal semantic sensor further increases SR to 61.4 and SPL to 29.7, revealing a substantial gap between current open-vocabulary perception and oracle perception. These results show that PredNav can benefit directly from stronger detectors, and that improving object perception is a promising direction for further enhancing navigation performance.

4.4 Failure Analysis

Table 4 provides a breakdown of successful episodes and failure categories on MP3D for different variants of PredNav. Each column sums to 100%, where the first row reports the success rate and the remaining rows correspond to different failure modes.

We categorize failed episodes into five types. A failure is classified as `cross_floor` when the start and goal positions differ in height by more than 1 m, indicating that the target is located on another floor. `detector_fp` denotes cases where the agent emits STOP more than 2 m away from any ground-truth target instance, usually due to false-positive detections. `detector_fn` refers to

Table 4: **Failure case breakdown on MP3D.** Percentages of successful episodes and different failure categories for each model variant. Values in each column sum to 100%.

Bucket	w/o LLM, w/o VLM	LLM only	VLM only	OWLv2 + both	GT detector
success	38.86	39.45	42.19	44.33	61.37
cross_floor	13.90	13.94	13.94	13.94	13.17
detector_fp	34.58	34.21	30.43	28.47	13.49
same_floor_missed	11.16	11.48	11.66	12.16	11.16
detector_fn	0.36	0.14	0.96	0.59	0.00
timeout_other	1.14	0.77	0.82	0.50	0.82

episodes where a ground-truth target enters the field of view but the detector never produces a valid candidate. `same_floor_missed` describes failures in which the agent explores the correct floor but never acquires a target candidate. Finally, `timeout_other` includes all remaining failures, such as inefficient exploration or navigation timeouts.

Table 4 shows two main trends. First, `cross_floor` failures remain stable at about 13–14% across all configurations, including the oracle detector setting. This indicates that such cases are mainly caused by the limitation of the within-floor topological-semantic representation, rather than by the language-grounded modules or detector quality. Addressing this failure mode therefore requires explicit multi-floor reasoning. Second, among the remaining failures, `detector_fp` is the dominant category, accounting for 28–35% of all episodes with real detectors. VLM verification reduces false-positive stops, while replacing the detector with a ground-truth semantic sensor further lowers this bucket to 13.49% and substantially improves success. In contrast, `detector_fn` stays below 1% in all configurations, suggesting that the system is recall-oriented and rarely misses a visible target once it appears in the field of view. Overall, the analysis identifies multi-floor reasoning and open-vocabulary detector precision as the two main sources of remaining failures.

5 Success rate under different step budgets

Figure 3 reports cumulative SR as a function of the step budget on MP3D. All variants improve as the budget increases, showing that exploration remains useful. Compared with the frontier-only baseline, language-grounded variants achieve higher SR throughout most of the budget range and show clearer gains in the later stages, indicating that semantic prediction and verification help the agent make better use of exploration steps rather than merely exploring more. Among real-detector configurations, the full PredNav pipeline with OWLv2 achieves the best curve, followed by the two partial ablations. This suggests that combining frontier prediction with visual verification improves both early target acquisition and final success. The GT-detector variant provides an upper reference and reaches high SR with fewer steps, confirming that more reliable perception can further amplify the benefit of the proposed navigation strategy.

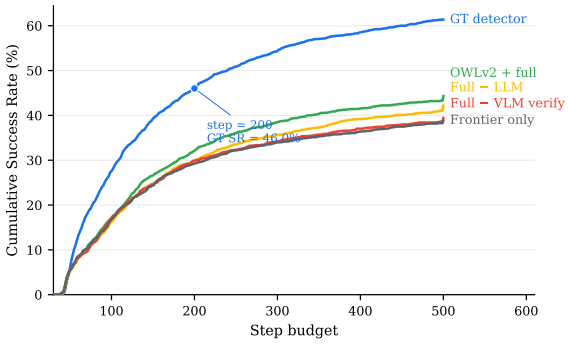

Figure 3: Cumulative SR vs. step budget on MP3D.

6 Conclusion

We propose PredNav, a zero-shot Object Goal Navigation framework inspired by the predictive cognitive map of the hippocampal-entorhinal system. Instead of scoring frontiers only from observed cues, PredNav builds a hierarchical metric, instance-semantic, and topological map, and casts frontier selection as room-level structured prediction over the topological layer. This grounds commonsense reasoning in an explicit spatial map while constraining the language model to schema-guided single-step inference. Experiments on MP3D and HM3D show consistent SR and SPL gains, with ablations validating core component.

Limitations

Within-floor scope. The present formulation of the topological-semantic layer treats each episode as a within-floor navigation task. The metric layer's 2.5 D height map distinguishes elevation differences and can in principle support stair traversal, but the planner does not autonomously decide to ascend or descend between floors in pursuit of a goal on a different level. The cost of this scoping is quantified directly by the failure analysis (Table 4): the `cross_floor` bucket holds at 13.2–14.3% of all MP3D val episodes across every configuration we evaluate — including the GT-detector ceiling — and is the single dominant residual that no within-floor architectural change, language-grounded or otherwise, can recover. Extending the topological layer with explicit inter-floor edges and a floor-aware decision policy is therefore the next architectural step indicated by the empirics, rather than further within-floor tuning.

Goal-specification granularity. We evaluate on the standard ObjectNav category sets shipped with HM3D and MP3D — closed, coarse sets of common indoor object types — which matches both the room-level structured-prediction interface PredNav exposes to the language model (§3.3) and the category-conditioned multi-view verifier (§3.4). Fine-grained instance descriptors ("the brown leather armchair"), open-ended referring expressions, and multi-step instructions ("go to the kitchen, then to the desk") fall outside this interface as currently implemented. The topological-layer serialisation and the verifier prompt could in principle be extended to richer goal specifications — attribute-conditioned scoring on the instance-semantic layer for the former, an open-ended grounded-VQA prompt for the latter — but we do not evaluate this here.

References

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and 1 others. 2018. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*.

Timothy EJ Behrens, Timothy H Muller, James CR Whittington, Shirley Mark, Alon B Baram, Kimberly L Stachenfeld, and Zeb Kurth-Nelson. 2018. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509.

Yihan Cao, Jiazhao Zhang, Zhinan Yu, Shuzhen Liu, Zheng Qin, Qin Zou, Bo Du, and Kai Xu. 2025. Cognav: Cognitive process modeling for object goal navigation with llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9550–9560.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. 2020. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. 2024. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16901–16911.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, and 1 others. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. 2023. 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*, pages 23171–23181.

Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. 2021. Learning to map for active semantic goal navigation. *arXiv preprint arXiv:2106.15648*.

Zeying Gong, Rong Li, Tianshuai Hu, Ronghe Qiu, Lingdong Kong, Lingfeng Zhang, Guoyang Zhao, Yiyi Ding, and Junwei Liang. 2026. Stairway to success: An online floor-aware zero-shot object-goal navigation framework via llm-driven coarse-to-fine exploration. *IEEE Robotics and Automation Letters*.

Yu Guo, Jinsheng Sun, Ruiheng Zhang, Zhiqi Jiang, Zhenqiang Mi, Chao Yao, Xiaojuan Ban, and Mohammad S Obaidat. 2024. An object-driven navigation strategy based on active perception and semantic association. *IEEE Robotics and Automation Letters*, 9(8):7110–7117.

Zhaotie Hao, Bin Guo, Yasan Ding, Linwei Li, Shiqi Liu, Tingting Gao, Sicong Liu, and Zhiwen Yu. 2026. Bio-inspired cognitive navigation for robots. *Nature Reviews Electrical Engineering*.

Yuxuan Kuang, Hai Lin, and Meng Jiang. 2024. Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 338–351.

Yiqing Liang, Boyuan Chen, and Shuran Song. 2021. Sscnav: Confidence-aware semantic scene completion for visual semantic navigation. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 13194–13200. IEEE.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. 2025. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. In *Conference on Robot Learning*, pages 2049–2060. PMLR.

Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. 2022. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352.

Bui Duc Manh, Soumyaratna Debnath, Zetong Zhang, Shriram Damodaran, Arvind Kumar, Yueyi Zhang, Lu Mi, Erik Cambria, and Lin Wang. 2025. Mind meets space: Rethinking agentic spatial intelligence from a neuroscience-inspired perspective. *arXiv preprint arXiv:2509.09154*.

Takahiro Miki, Lorenz Wellhausen, Ruben Grandia, Fabian Jenelten, Timon Homberger, and Marco Hutter. 2022. Elevation mapping for locomotion and navigation using gpu. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2273–2280. IEEE.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. 2023. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007.

Ida Momennejad, Hosein Hasanbeig, Felipe Vieira Frujeri, Hiteshi Sharma, Nebojsa Jojic, Hamid Palangi, Robert Ness, and Jonathan Larson. 2023. Evaluating cognitive maps and planning in large language models with cogeval. *Advances in Neural Information Processing Systems*, 36:69736–69751.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Abhinav Rajvanshi, Karan Sikka, Xiao Lin, Bhoram Lee, Han-Pang Chiu, and Alvaro Velasquez. 2024. Saynav: Grounding large language models for dynamic planning to navigation in new environments. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pages 464–474.

Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, and 1 others. 2021. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*.

Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. 2022. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, and 1 others. 2024. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*.

Shouwei Ruan, Liyuan Wang, Caixin Kang, Qihui Zhu, Songming Liu, Xingxing Wei, and Hang Su. 2025. From reactive to cognitive: brain-inspired spatial intelligence for embodied agents. *arXiv preprint arXiv:2508.17198*.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, and 1 others. 2019. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347.

Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. 2017. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643–1653.

Xinyu Sun, Lizhao Liu, Hongyan Zhi, Ronghe Qiu, and Junwei Liang. 2024. Prioritized semantic learning for zero-shot instance navigation. In *European Conference on Computer Vision*, pages 161–178. Springer.

Alberto Valero-Gomez, Javier V Gomez, Santiago Garrido, and Luis Moreno. 2013. The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories. *IEEE Robotics & Automation Magazine*, 20(4):111–120.

Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, and 1 others. 2025. Mindcube: Spatial mental modeling from limited views. *arXiv e-prints*, pages arXiv–2506.

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. 2024. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. 2024. Voronav: Voronoi-based zero-shot object navigation with large

language model. In *International Conference on Machine Learning*, pages 53757–53775. PMLR.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2025a. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis L Brown II, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, and 1 others. 2025b. Cambrian-s: Towards spatial supersensing in video. In *The Fourteenth International Conference on Learning Representations*.

Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. 2024. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *Advances in neural information processing systems*, 37:5285–5307.

Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. 2025. Unigoal: Towards universal zero-shot goal-oriented navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19057–19066.

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. 2024. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 42–48. IEEE.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. 2023. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3554–3560. IEEE.

Pingyue Zhang, Zihan Huang, Yue Wang, Jieyu Zhang, Letian Xue, Zihan Wang, Qineng Wang, Keshigeyan Chandrasegaran, Ruohan Zhang, Yejin Choi, and 1 others. 2026. Theory of space: Can foundation models construct spatial beliefs through active exploration? In *The Fourteenth International Conference on Learning Representations*.

Sixian Zhang, Xinyao Yu, Xinhang Song, Xiaohan Wang, and Shuqiang Jiang. 2024. Imagine before go: Self-supervised generative map for object goal navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16414–16425.

Xinxin Zhao, Wenzhe Cai, Likun Tang, and Teng Wang. 2025. Imaginenav: Prompting vision-language models as embodied navigator through scene imagination. In *International Conference on Learning Representations*, volume 2025, pages 94387–94401.

Gengze Zhou, Yicong Hong, and Qi Wu. 2024. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7641–7649.

Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. 2023. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning*, pages 42829–42842. PMLR.

Zibo Zhou, Yue Hu, Lingkai Zhang, Zonglin Li, and Siheng Chen. 2026. Beliefmapnav: 3d voxel-based belief map for zero-shot object navigation. *Advances in Neural Information Processing Systems*, 38:83008–83036.

A Episode-Level Scan Phases

Initial panoramic scan. Each episode begins with a fixed 28-step scan,

$$12L \to 2D \to 12L \to 2U,$$

where L, D, and U denote TURN-LEFT, LOOK-DOWN, and LOOK-UP, respectively. This scan covers a full 360° rotation at pitch 0°, lowers the camera to pitch -60°, repeats the rotation, and restores pitch to 0°. The lowered-pitch sweep covers the immediate 0–5 m radial floor and seeds the metric layer's REACHABLE component before normal planning begins: because the height map has no ray casting, cells between the agent and the nearest observed obstacle would otherwise remain UNKNOWN, and the planner's seed cell could land on undefined elevation.

Seal step. After the 28-step scan, residual UNKNOWN cells within a 0.8 m disc around the agent's footprint are forced to REACHABLE. This closes the small NaN holes left by the capsule-shaped self-occlusion at pitch -60°, which would otherwise produce a starburst of UNKNOWN cells radiating from directly beneath the agent.

Arrival mini-scan. When the planner reports AT-GOAL at a frontier, the agent checks the forward $\pm 60^\circ \times 2$ m cone for its UNKNOWN-cell fraction. If the fraction exceeds 0.25, a two-step mini-scan ($1 \times$ LOOK-DOWN, $1 \times$ LOOK-UP at intermediate pitch -30°) is emitted before scoring resumes. The mini-scan covers 0.46–5 m of forward floor and resolves frontiers whose centroid would otherwise remain on momentarily UNKNOWN cells.

B Speed Field, Collision Inflation, and Safe Stripe

Clearance-based speed. Let $\mathrm{clr}(c)$ denote the cell-discrete clearance from cell c to the nearest OBSTACLE cell. The FM^2 per-cell speed is

$$v(c) = \begin{cases} 0 & \mathrm{clr}(c) < r_{\text{in}}, \\ \left(\frac{\mathrm{clr}(c) - r_{\text{in}}}{r_{\text{soft}} - r_{\text{in}}}\right)^{\beta_{\text{FM}}} & r_{\text{in}} \le \mathrm{clr}(c) < r_{\text{soft}}, \\ 1 & \mathrm{clr}(c) \ge r_{\text{soft}}, \end{cases}$$

with $r_{\text{in}} = 2$ cells (inscribed-radius hard zone), $r_{\text{soft}} = 6$ cells (open-corridor saturation), and $\beta_{\text{FM}} = 2$ (soft-zone falloff exponent). The hard zone enforces a strict collision constraint; the soft zone provides a saturating cost-of-narrowness penalty; the open-corridor region carries unit speed.

Inscribed radius below geometric radius. Note that $r_{\text{in}} = 2$ cells at 0.05 m/cell corresponds to 0.10 m, half the geometric agent radius (0.18 m, 4 cells). A strict $r_{\text{in}} = 4$ would close off doorways and furniture gaps whose discretised clearance falls at the boundary. The soft zone penalises but does not block such passages; the collision overlay below re-inflates the obstacle whenever the agent in fact scrapes.

Collision overlay. When the GPS displacement after a FORWARD action falls below a small threshold and the previous action was FORWARD, the cell immediately in front of the agent's footprint is marked persistently as OBSTACLE in a separate collision-overlay channel, which then inflates with the rest of the wall mask on the next speed-field rebuild. This eliminates "re-bumping the same wall" behaviour without storing explicit policy memory.

Trajectory safe stripe. The agent's past trajectory, rasterised by Bresenham line drawing and dilated by one cell on each side (a 3-cell tube), is OR'd into the speed field with a floor of 0.1 after the clearance-based formula above is applied. This is required for stair corridors: the roughness filter that defines OBSTACLE cells flags stair risers (whose step height matches the agent's clearance threshold) as obstacles, which would otherwise cut the stair corridor across its own centreline. The safe stripe restores the agent's own track at low but non-zero speed.

C Stuck-Escape Ladder

When the agent fails to make progress — measured as GPS displacement falling below an ϵ over a sliding window — three stages of escape behaviour are layered, with strictly increasing trigger thresholds in steps of no-progress:

- **Stage 1** (> 12 steps): the local planner samples a random reachable cell within a short radius as a temporary escape target and routes to it. This is sufficient to dislodge most micro-collisions and tight-corner oscillations.
- **Stage 2** (> 15 steps): the agent-level controller blacklists the current frontier (or, if currently locked onto a target instance, re-routes to the next nearest candidate). The blacklist entry carries both an identifier and a 0.5 m spatial key, so it survives the next DBSCAN rebuild that may re-issue a different identifier for the same cluster.

- **Stage 3** (> 30 steps): the agent emits a diagnostic `agent_stuck` event for offline analysis; no further action is taken inside the episode.

The thresholds form a strict ladder ($12 \leq 15 \leq 30$) and reflect the upper bound on legitimate zero-displacement intervals: a full 360° rotation takes 12 TURN-LEFT actions, so a window of ≤ 12 steps with zero displacement may still be a legitimate panorama and is not treated as a stuck event.

D Target Selection Hysteresis

When more than one candidate target instance of the requested category g is simultaneously visible, the agent locks onto the geodesically nearest. Let A denote the currently locked candidate at geodesic distance d_A and B a competing candidate at distance d_B. A switch from A to B is admitted only if both of the following hold:

$$d_B < 0.5\, d_A \quad \text{and} \quad d_B < 3\,\text{m}.$$

The first condition demands a substantial relative-distance advantage and rules out jitter between near-tied candidates; the second is an absolute proximity floor that prevents far-away candidates from displacing the current lock just because the agent's geodesic estimate to A has temporarily grown (e.g., a transient un-reachable estimate from a half-built map).

E Room-Segmentation Hyperparameters

Table 5 lists the geometric segmentation hyperparameters used in §3.2; values are unchanged from the procedure of (Werby et al., 2024). The aspect-ratio fallback labels rooms with longest-axis to shortest-axis ratio above 3.0 as *hallway*; the room-id continuity threshold (re-using the previous id when a re-segmented room's centroid lies within 2.0 m of an existing room) avoids id churn under incremental map updates.

F LLM and VLM Prompts

This appendix records the exact prompts and JSON schemas consumed by the LLM frontier predictor (§3.3) and the VLM target verifier (§3.4). Endpoint-specific adaptations are noted at the end.

Frontier prediction — system prompt.

> You are a navigation assistant helping an agent find objects in an indoor environment. Given a scene description and unexplored frontiers, predict what lies beyond each frontier. Respond ONLY with a JSON object — no explanation, no markdown.

Parameter	Default	Role
`footprint_close_m`	0.45	morph. closing radius
`point_dilate_m`	0.10	wall-OBB dilation
`wall_dilate_m`	0.08	final wall-mask dilation
`min_room_area_m2`	1.5	min room area
`min_line_length_m`	0.6	Hough min line length
`max_line_gap_m`	0.25	Hough max line gap
`structural_lateral_m`	0.25	parallel-line lateral tol.
`structural_merge_gap_m`	0.45	along-line merge gap
`branch_prune_m`	0.6	skeleton branch prune
`door_clearance_min_m`	0.35	doorway min radius
`door_clearance_max_m`	0.85	doorway max radius
`doorway_relief_m`	0.35	two-sided radius rise
`cut_margin_m`	0.18	cut-line thickness

Table 5: Room-segmentation hyperparameters used in §3.2, following (Werby et al., 2024).

Frontier prediction — user template. The user message is filled into the template

```
Target object: {target}
Current scene:
{scene_context}
Unexplored frontiers:
{frontier_descriptions}
For each frontier, predict:
- predicted_room: most likely room type
beyond it
- predicted_objects: likely objects
(from the ObjectNav categories)
- confidence: your confidence in this
prediction (0.0-1.0)
- target_relevance: probability the
target "{target}" is beyond this
frontier (0.0-1.0)
Respond with this exact JSON format:
{"predictions": [{"frontier_id": ...,
...}, ...]}
```

where `scene_context` is the serialised topological-semantic snapshot of §3.2 and each `frontier_descriptions` line is formatted as, e.g.,

```
- F001: ahead (12°), 4.2m away, size=187
cells, near R3
```

quantising the agent-relative bearing to {AHEAD, TO THE LEFT, TO THE RIGHT, BEHIND}.

Frontier prediction — JSON schema.

```
{
  "type": "object",
  "properties": {
    "predictions": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "frontier_id": {"type": "string"},
          "predicted_room": {"type": "string"},
          "predicted_objects": {
            "type": "array",
            "items": {"type": "string"}
          },
          "confidence": {"type": "number"},
          "target_relevance": {"type": "number"}
        },
        "required": ["frontier_id", "predicted_room",
          "predicted_objects", "confidence",
```

```
          "target_relevance"],
        "additionalProperties": false
      }
    }
  },
  "required": ["predictions"],
  "additionalProperties": false
}
```

On endpoints with native JSON-schema support, this schema is enforced via `response_format = json_schema`; on endpoints lacking that feature, an equivalent structural example is injected into the system prompt (see "endpoint adaptation" below).

Spatial-prior generation prompt. The offline room–category prior table $P(g \mid r)$ used by S_{prior} is generated once per dataset by a single LLM call:

> You are an expert on indoor scene layouts. For each room type, estimate the probability (0.0–1.0) that each object category is present in a typical instance of that room. Room types: `{rooms}`. Object categories: `{objects}`. Return a JSON object where keys are room types and values are objects mapping category to probability. Include ALL room types and ALL object categories.

The resulting table is loaded from disk and used deterministically at scoring time; it is *not* re-queried during episodes.

Target verification — system prompt.

> You are a strict, neutral visual grader for an object-goal navigation robot. Each image is a full frame with a red rectangle drawn around a candidate object that the robot's detector flagged. All images depict the SAME physical candidate from different angles. Identify the object inside the red rectangle on its own merits — do not let the robot's stated hypothesis bias you. Always return JSON conforming to the schema.

Target verification — user prompt. The user message lists four numbered steps:

> **Step 1 — `looks_like`.** Name the object inside the red rectangle with one natural lowercase common noun (e.g., `chair`, `sofa`, `stool`, `cabinet`, `counter`, `mantelpiece`). If the rectangle is genuinely empty of any discrete object — pure wall texture, an open door, the bare floor, a window pane — name that structural surface. Do not name a structural surface when a real object is in fact inside the rectangle, and do not bias toward `{g}`.
>
> **Step 2 — `confidence`.** Real value in $[0, 1]$ for the `looks_like` label. Use ≥ 0.85 only when the name is unambiguous; use ≤ 0.5 when the rectangle is blurry, occluded, or genuinely ambiguous between two nouns.
>
> **Step 3 — `fully_visible`.** True iff the whole object is in frame in at least one image.
>
> **Step 4 — `is_target`.** True iff `looks_like` is the same kind of thing as `{g}` (counting common synonyms). Be honest about what you actually see; the agent re-derives this deterministically from `looks_like` via a synonym table, so the decision is robust to wording.

Target verification — JSON schema.

```
{
  "type": "object",
  "properties": {
    "is_target":     {"type": "boolean"},
    "confidence":    {"type": "number"},
    "fully_visible": {"type": "boolean"},
    "looks_like":    {"type": "string"}
  },
  "required": ["is_target","confidence",
               "fully_visible","looks_like"],
  "additionalProperties": false
}
```

Synonym table. The agent-side synonym table that maps the verifier's `looks_like` label to a canonical target category is a fixed dictionary of approximately a dozen entries, each listing all surface forms that refer to the same physical object class. Representative entries are

> `sofa` $\mapsto$ {sofa, couch, loveseat, settee, divan, sectional};
> `counter` $\mapsto$ {counter, countertop, kitchen counter, bar, kitchen island};
> `chest_of_drawers` $\mapsto$ {chest of drawers, dresser, drawers, bureau};
> `plant` $\mapsto$ {plant, potted plant, houseplant}.

The table is hand-specified once per benchmark, decouples target recognition from the verifier's own internal synonym handling, and is unchanged across all reported experiments.

Endpoint adaptation. The frontier predictor calls a chat-completion endpoint with the system prompt, user prompt, and JSON schema. On OpenAI-compatible endpoints with JSON-schema support, `response_format=json_schema` enforces strict output. On endpoints that lack JSON-schema support (e.g., some DeepSeek-compatible deployments), the system prompt is augmented with an explicit `json` marker and a structural example derived from the schema. Where the underlying model defaults to "thinking" output (wrapping responses in `<think>...</think>`) we disable thinking via the standard `extra_body={"thinking": {"type": "disabled"}}` option. These adaptations are confined to the client; the rest of the pipeline is unaware of them.

G Hyperparameter Summary

Table 6 collects every numerical hyperparameter referenced in the Method and the Appendices above and their semantics and origins are detailed in the relevant sections. All values are constant across scenes, target categories, and datasets.

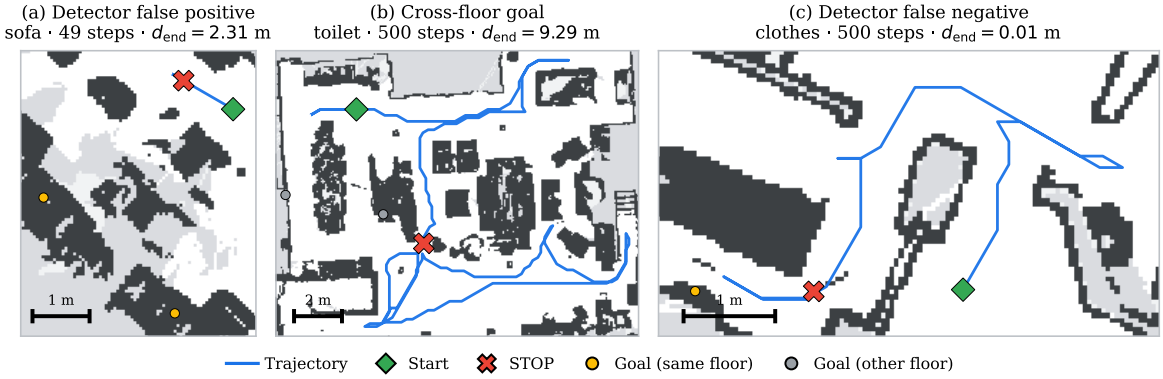

Figure 4: **Representative failure cases on MP3D, one per bucket from Table 4.** Each panel shows the cropped traversability map, where UNKNOWN is light gray, OBSTACLE is dark gray, and FREE/REACHABLE is white. The agent trajectory is shown as a blue line, the start pose as a green diamond, the STOP pose as a red cross, and goal instances by floor membership, with orange for the same floor as the start and gray for a different floor. Panels are the median-DTG episode per bucket under the *LLM-only* configuration in Table 2. Each title strip reports the target category, episode length, and final distance d_{end} to the nearest goal.

H Qualitative Failure Cases

Figure 4 shows three representative failures, one median-DTG episode per bucket. Case (a) is a `detector_fp` on a *sofa* target ($d_{\text{end}} = 2.31$ m): the agent emits STOP only 49 steps into the episode, well before it has explored the scene, because the detector latched onto a surface labelled *sofa* with high confidence and the verifier passes on it under the same visual evidence the detector used; the two same-floor sofa instances (orange markers) sit in regions the agent never reaches. This is the structural reason why the verifier's -3.78 point reduction in `detector_fp` (Table 4) is so much smaller than the GT-detector's -20.72 point reduction: when both the detector and the verifier consult the same image evidence, an error that fools one tends to fool the other. Case (b) is a `cross_floor` episode on a *toilet* target: every toilet instance in the scene is on a different floor from the agent's starting position (all goal markers in the panel are gray), and the within-floor topological-semantic layer is by construction incapable of redirecting the agent across floors; the agent exhausts its 500-step budget exploring within-floor frontiers and terminates 9.29 m from the nearest goal. Case (c) is the most diagnostic. The agent is searching for *clothes* and physically reaches a same-floor clothes instance ($d_{\text{end}} = 0.01$ m), but no frame ever produces a confirmed *clothes* detection on that instance, so the agent never emits STOP; the episode times out at 500 steps despite having geometrically completed the task. `detector_fn` is the rare ($< 1\%$) bucket precisely because the recall-favoured calibration of §3.4 usually catches such instances, but when it does miss, the architecture has no recovery path — the agent has no signal that it is standing on the answer.

Parameter	Default	Description
Cognitive map		
cell size	0.05 m	metric grid resolution
k_{g_o} (default)	2	confirmation threshold
Frontier scoring		
η_1, η_2	$0.1,\ 0.4$	phase boundaries
$\mathbf{w}^{\text{EARLY}}$	$(0.2, 0.4, 0.4)$	obs/pred/prior
$\mathbf{w}^{\text{MID}}$	$(0.4, 0.35, 0.25)$	"
$\mathbf{w}^{\text{LATE}}$	$(0.6, 0.25, 0.15)$	"
λ_d	0.30	distance-term weight
$d_{\max}$	10 m	distance-term cutoff
β	0.015	info-gain bonus coeff.
S_{obs} radius	3 m	CLIP aggregation radius
Verification		
approach trigger	2.5 m	enter verification
ρ	0.45	target image fraction
$d_{\min}^{\text{stand}}, d_{\max}^{\text{stand}}$	$0.8, 4.5$ m	standoff bounds
evidence frames	≤ 4	per verifier call
bearing diversity	30°	evidence separation
accept conf.	0.6	pass-condition floor
partial-accept conf.	0.8	no-visibility pass floor
reject conf.	0.3	fail-condition ceiling
orbit novelty	45°	active-acquisition gate
call budget	40	per-episode VLM cap
Planning		
K	3	detour top-K
γ	$\lambda_d / d_{\max}$	detour distance coeff.
$r_{\text{in}}, r_{\text{soft}}, \beta_{\text{FM}}$	$2, 6, 2$	FM^2 speed params
safe-stripe floor	0.1	trajectory-tube speed
Recovery		
stuck stages	$12, 15, 30$ steps	ladder thresholds
switch ratio	0.5	d_B/d_A admissibility
switch absolute	3 m	d_B proximity floor

Table 6: All numerical hyperparameters used by PredNav, constant across scenes, target categories, and datasets.