When Geometry Matches Language: A Controlled Re-evaluation of InstructNav for ObjectNav

Anonymous authors
Paper under double-blind review

Abstract

We conduct a controlled re-evaluation of the InstructNav pipeline on ObjectNav, using matched semantic perception to isolate the contribution of language-conditioned frontier selection and planning. We introduce two deliberately simple diagnostic variants: FPE, which uses frontier geometry for exploration, and SHF, which augments frontier selection with LLM votes. Under ground-truth semantics, FPE performs comparably to our InstructNav-GT-4.1 reproduction while eliminating API cost and reducing runtime. Under open-vocabulary perception, the methods likewise remain broadly competitive, with no consistent winner across datasets and metrics. Across both perception settings, we do not observe a consistent aggregate advantage from the full LLM/VLM stack over substantially simpler controls. SHF does not consistently improve over FPE overall, although category- and scene-level results reveal selected cases in which semantic voting is beneficial. These findings are limited to ObjectNav and do not question the necessity of language models for tasks involving compositional instructions, route following, or demand-driven navigation.

1 Introduction

Large language models (LLMs) have recently been woven into embodied navigation pipelines, with reports of large zero-shot gains when language priors are injected into classical mapping and planning. Two representative threads are: (i) instruction-following systems such as InstructNav, which unify diverse instructions via a Dynamic Chain-of-Navigation (DCoN) prompt and aggregate multiple value maps into an affordance for A* planning (Long et al., 2024); and (ii) language-as-heuristic approaches such as Language Frontier Guide (LFG), which poll an LLM for votes over candidate frontier regions and use these votes to bias exploration (Shah et al., 2023). Evaluations are typically conducted in Habitat on indoor datasets such as HM3D and MP3D (Ramakrishnan et al., 2021; Chang et al., 2017).

Motivation. First, modern instruction-guided stacks couple language with open-vocabulary perception (e.g., GLEE) and VLM-based "intuition" maps (Wu et al., 2024a). In cluttered indoor scenes, these components can be brittle and expensive, and their errors propagate into the planner. Second, ablations are often performed one component at a time, leaving interdependencies intact; this can obscure whether the purported gains stem from language "intelligence" or from geometry encoded elsewhere in the stack (Long et al., 2024). Meanwhile, independent planning studies suggest that LLMs are weak autonomous planners (single-digit success), but can be useful as heuristic oracles when coupled to structured search (Valmeekam et al., 2023).

What we ask. Within the evaluated InstructNav pipeline on ObjectNav, what aggregate benefit remains attributable to the LLM/VLM components when semantic perception and the downstream mapping/planning stack are matched?

What we find. Under matched ground-truth semantic perception, FPE obtains aggregate performance comparable to InstructNav-GT-4.1 while requiring no API calls and substantially less runtime. Under open-vocabulary perception, the evaluated methods likewise remain broadly competitive, although the best-performing method varies across datasets and metrics. Across both perception settings, the full InstructNav

stack does not demonstrate a consistent aggregate advantage over substantially simpler controls. SHF also does not consistently improve over FPE; its benefits are selective across semantic categories and environments.

Contributions. (1) A controlled re-evaluation of the InstructNav pipeline on ObjectNav that separates frontier-selection and planning effects from semantic perception quality. (2) A deliberately simple, training-free frontier-geometry baseline, implemented within the same value-map and planning framework, that performs comparably to our InstructNav-GT-4.1 reproduction under matched ground-truth perception while requiring no LLM calls and substantially less runtime. (3) A lightweight LFG-inspired frontier-voting variant that enables a direct comparison between geometry-only exploration and localized language guidance, together with analyses of perception noise, action-map failures, computational cost, and the semantic categories and environments in which language guidance helps or fails to help.

Our study is restricted to ObjectNav, where the linguistic input specifies an object category and exploration is a central challenge. We do not evaluate route-following, demand-driven navigation, compositional instructions, constraint satisfaction, or other settings in which interpreting language is itself part of the task. Accordingly, our conclusions apply only to the evaluated InstructNav pipeline on ObjectNav and should not be interpreted as evidence that LLMs are unnecessary for instruction-following navigation more broadly. Altogether, our results motivate matched-perception comparisons, strong geometry-first controls, and explicit cost and runtime reporting when evaluating LLM-augmented ObjectNav systems.

2 Related Work

ObjectGoal Navigation (ObjectNav) tasks an agent to reach an instance of a named object category in an indoor scene under a step cap. Evaluations commonly use the Habitat simulator with datasets such as HM3D and MP3D (Ramakrishnan et al., 2021; Chang et al., 2017). A number of zero-shot or training-free pipelines combine classical exploration with semantic cues.

ZSON proposes open-world ObjectNav via multimodal goal embeddings (Majumdar et al., 2022). ESC transfers commonsense knowledge into exploration via soft logical constraints with VLM/LLM priors (Zhou et al., 2023). L3MVN queries an LLM to select frontiers conditioned on object context, reporting zero-shot gains on Gibson/HM3D (Yu et al., 2023). VLFM builds vision–language frontier maps to prioritize frontiers with language grounding (Yokoyama et al., 2024). VoroNav constructs a Reduced Voronoi Graph and prompts an LLM with path/farsight descriptions to choose waypoints (Wu et al., 2024b). VLMNav studies end-to-end action selection with VLMs (Goetting et al., 2024). Outside HM3D, CLIP-on-Wheels (CoW) explored low-cost, CLIP-based navigation on MP3D-style settings (Gadre et al., 2023).

InstructNav introduces Dynamic Chain-of-Navigation (DCoN) prompting to unify diverse instruction types and aggregates multiple value maps (Trajectory, Semantic, Intuition, Action) into a summed affordance for A* planning; perception uses open-vocabulary segmentation (GLEE) and a VLM-based Intuition map (Long et al., 2024). Rather than issuing low-level plans, LFG (Language-Frontier-Guide) repeatedly polls an LLM with positive/negative prompts over object clusters near frontiers, converts votes to scores, and selects frontiers accordingly (Shah et al., 2023). Frontier-based exploration, where agents move to the boundary between known free space and unexplored regions, dates back to Yamauchi's formulation (Yamauchi, 1997). In ObjectNav, Goal-Oriented Semantic Exploration (SemExp) maintains an episodic semantic map and uses category-specific priors to guide exploration (Chaplot et al., 2020).

Beyond ObjectNav, large foundation models are increasingly being employed in various other embodied tasks. SayCan grounds high-level language in robotic skills and affordances for mobile manipulation (Ahn et al., 2022), while Voyager bootstraps skills in Minecraft by coupling GPT-4 with an evolving code library and curriculum (Wang et al., 2023).

3 Background

3.1 Object-Goal Navigation (ObjectNav)

ObjectNav asks an agent to reach any instance of a named object category (e.g., *Find a <category>*) using egocentric sensing and low-level controls. In this study, we focus on 2D object-goal navigation in unseen environments. Episodes terminate in success when the agent stops within a small metric threshold of the goal (we detail the radius and the step cap in Section 5). Evaluation commonly reports Success Rate (SR) and Success weighted by Path Length (SPL) (Anderson et al., 2018). We use the Habitat stack and HM3D/MP3D datasets for simulation (Ramakrishnan et al., 2021; Chang et al., 2017).

3.2 Mapping and Frontiers

At each time step, RGB-D and pose are fused into a 2D navigability map; free space vs. obstacles are derived from depth ray-casting, following standard Habitat practice. Frontiers are the boundary curves between explored free space and the currently unknown region, a notion originating in classical exploration (Yamauchi, 1997). We detect frontier points by differencing explored and unknown masks, then group contiguous frontier points into clusters via DBSCAN (density-based clustering) (Ester et al., 1996). From now on, we refer to these clusters as "frontiers", which will later serve as anchor sets for scoring or selection.

3.3 InstructNav: DCoN and Multi-sourced Value Maps

Dynamic Chain-of-Navigation (DCoN). InstructNav (Long et al., 2024) turns the instruction and recent observations into a step-wise chain that is re-inferred every decision step with the latest observations. Concretely, the large language model (LLM) prompt is structured into (i) Robot Definition, (ii) Navigation Strategy (task-specific priors), (iii) Prediction Format (keys such as `Reason`, `Action`, `Landmark`, `Flag`), and (iv) Episode Information (recent observations and detected objects). The LLM then outputs the next high-level action (e.g., `Explore`, `MoveForward`, `TurnLeft/Right`, `Approach`) and a landmark name to bias exploration; this process is repeated at each step. InstructNav's prompt templates are included in Appendix C.

Perception and value maps. InstructNav integrates an open-vocabulary perception stack (GLEE) to produce per-frame category masks (Wu et al., 2024a), and a vision language model (VLM) to propose an intuition field of view. These signals, together with the action and trajectory context, are converted into four value maps on the local grid. In all cases, InstructNav forms a binary area/mask $\mathcal{A}_x$ for source $x \in \{\text{sem}, \text{act}, \text{traj}, \text{intu}\}$, computes a distance transform $d_x(p) = \min_{q \in \mathcal{A}_x} \text{dist}(p, q)$ for each point in the navigable area $p \in \mathcal{N}$, normalizes it to $\tilde{d}_x \in [0, 1]$, and then sets

$$V_x(p) = \begin{cases} 1 - \tilde{d}_x(p), & d_x(p) \le r_x, \\ 0, & \text{otherwise}, \end{cases}$$

where r_x is a hyperparameter. Concretely:

- *Semantic Value Map* V_{sem}: $\mathcal{A}_{\text{sem}}$ is the 3D projection of pixels labeled with the current landmark (from GLEE) onto the grid.
- *Action Value Map* V_{act}: $\mathcal{A}_{\text{act}}$ depends on DCoN's action. For `Explore`, it is the set of frontier points; for `MoveForward/TurnLeft/TurnRight`, it is the corresponding local angular sector anchored at the agent.
- *Trajectory Value Map* V_{traj}: $\mathcal{A}_{\text{traj}}$ is the mask of recently visited cells (trajectory footprint).
- *Intuition Value Map* V_{intu}: $\mathcal{A}_{\text{intu}}$ is the VLM-selected field-of-view region (projected to the grid).

These maps are summed to form the affordance/decision map

$$V_{\text{aff}}(p) \;=\; V_{\text{sem}}(p) + V_{\text{act}}(p) + V_{\text{traj}}(p) + V_{\text{intu}}(p), \tag{1}$$

from which the target $p^\star = \arg\max_p V_{\text{aff}}(p)$ is chosen, and a shortest path to $p^\star$ is computed via A* (obstacles masked) with cost $1-V_{\text{aff}}$ (Long et al., 2024; Hart et al., 1968):

$$p^\star \;=\; \arg\max_{p\in\mathcal{N}} V_{\text{aff}}(p), \qquad \pi \;=\; \text{A}^*\big(\text{start}\to p^\star;\ \text{cost}{=}\,1-V_{\text{aff}}\big). \tag{2}$$

When the named goal object is observed, InstructNav's `Approach` behavior uses V_{sem} to close in and issues `Stop` once the success condition is met (Long et al., 2024).

3.4 Language Frontier Guide (LFG)

LFG (Shah et al., 2023) is a complementary paradigm: instead of composing multiple value maps, it forms candidate frontiers and queries a language model multiple times (k trials) with positive and negative prompts to judge which frontier is most promising given currently observed object names. The votes are aggregated into a scalar frontier score, and the planner greedily selects the frontier with the highest score; standard planning (metric or topological) executes the move (Shah et al., 2023). LFG does not assume open-vocabulary detectors or a VLM "intuition" map; its only learned prior is the frozen LLM used for scoring. LFG incorporates ground-truth semantic information to address the suboptimal performance of open-vocabulary detectors in simulations, which is attributed to rendering artifacts (Shah et al., 2023).

4 Methods

We introduce two training-free variants built on the pipeline in Section 3: (i) **FPE: Frontier Proximity Explorer** and (ii) **SHF: Semantic-Heuristic Frontier**. Both methods select targets by maximizing the affordance map V_{aff} and execute A* with cost $1-V_{\text{aff}}$ as in Eq. 1–2. During exploration, both variants use their proposed Action Value Map together with the Trajectory Value Map. When the named goal becomes visible, the agent activates the Semantic Value Map, switches to the standard InstructNav Approach behavior, and issues Stop once the success condition is satisfied.

4.1 FPE: Frontier Proximity Explorer

FPE removes DCoN (no language planning), disables the Intuition map, and uses frontiers directly to define the action prior. Let $\mathcal{F}_t$ be the set of frontier points at time t (detected as explored–unknown boundaries and clustered as in Section 3). Define the frontier distance transform $d_{\mathcal{F}}(p) = \min_{q\in\mathcal{F}_t} \text{dist}(p,q)$ and its normalized version $\tilde{d}_{\mathcal{F}} \in [0,1]$. FPE sets

$$V_{\text{act}}^{\text{FPE}}(p) = \begin{cases} 1-\tilde{d}_{\mathcal{F}}(p), & d_{\mathcal{F}}(p) \le r_{\text{FPE}}, \\ 0, & \text{otherwise}, \end{cases}$$

keeps the trajectory prior V_{traj} to ensure exploration, and, during exploration, sets $V_{\text{sem}} = V_{\text{intu}} = 0$. The affordance map becomes $V_{\text{aff}} = V_{\text{act}}^{\text{FPE}} + V_{\text{traj}}$, from which the target and A* path are computed as in InstructNav. Intuitively, cells closer to any frontier score higher, biasing the planner to expand into nearby unknown space before costly detours.

Design rationale. FPE removes DCoN and the Intuition map to avoid the prompt–metric mismatch (LLMs receive no coordinates) and the VLM overhead; it keeps only the proximity prior to frontiers and the coverage prior V_{traj}. This tests how far geometry alone can go under the same target-selection and A* pipeline.

4.2 SHF: Semantic-Heuristic Frontier

SHF adds a language prior to FPE by scoring frontiers with an LLM in the LFG style (Shah et al., 2023). For each frontier I_j, we form the set of object names currently observed near that frontier, $\mathbf{s}_{I_j} = \{o_1, \ldots, o_m\}$.

Algorithm 1: FPE: Frontier Proximity Explorer update at step t

Input: navigable area $\mathcal{N}$, frontier points $\mathcal{F}_t$, scoring radius r_{FPE}
Output: $V_{\text{act}}^{\text{FPE}}$ on $\mathcal{N}$
$V_{\text{act}}^{\text{FPE}}(p) \leftarrow 0 \;\; \forall p \in \mathcal{N}$;
compute $d_{\mathcal{F}}(p) = \min_{q \in \mathcal{F}_t} \text{dist}(p, q)$ for all $p \in \mathcal{N}$;
normalize $\tilde{d}_{\mathcal{F}}(p) \leftarrow d_{\mathcal{F}}(p) / \max_{q \in \mathcal{N}} d_{\mathcal{F}}(q)$;
for all $p \in \mathcal{N}$**: if** $d_{\mathcal{F}}(p) \leq r_{\text{FPE}}$ **then** $V_{\text{act}}^{\text{FPE}}(p) \leftarrow 1 - \tilde{d}_{\mathcal{F}}(p)$;
return $V_{\text{act}}^{\text{FPE}}$

Algorithm 2: SHF: Semantic-Heuristic Frontier update at step t

Input: navigable area $\mathcal{N}$, frontiers $\{I_j\}$, goal name g, trials k, scoring radius r_{SHF}
Output: $V_{\text{act}}^{\text{SHF}}$ on $\mathcal{N}$
$V_{\text{act}}^{\text{SHF}}(p) \leftarrow 0 \;\; \forall p \in \mathcal{N}$;
for *each frontier* I_j **do**
  assemble object-name set $\mathbf{s}_{I_j}$ from current observations (names only);
  query LLM with k positive and k negative prompts; compute $H(I_j) = n_j^+ - n_j^-$;
  $\alpha(I_j) \leftarrow \dfrac{H(I_j) + k}{2k}$; `// rescale to` $[0, 1]$
  compute $d_{I_j}(p) = \min_{q \in I_j} \text{dist}(p, q)$ for all $p \in \mathcal{N}$;
  for all p**: if** $d_{I_j}(p) \leq r_{\text{SHF}}$ **then** $V_{\text{act}}^{\text{SHF}}(p) \leftarrow V_{\text{act}}^{\text{SHF}}(p) + \alpha(I_j)$;
$V_{\text{act}}^{\text{SHF}}(p) \leftarrow \min\Big(1,\; V_{\text{act}}^{\text{SHF}}(p)\Big) \;\; \forall p$;
return $V_{\text{act}}^{\text{SHF}}$

The LLM is queried with k positive prompts ("Given $\mathbf{s}_{I_j}$, which frontier would you choose to find a g?"), where g is the goal name, and k negative prompts ("... which frontier would you avoid?"); no metric coordinates are provided. Let n_j^+ and n_j^- be the number of positive and negative votes for frontier I_j. We define a simple integer score

$$H(I_j) \;=\; n_j^+ - n_j^- \quad \in [-k, k],$$

and rescale it to $[0, 1]$ via $\alpha(I_j) = \frac{H(I_j)+k}{2k}$. The SHF Action map is then

$$V_{\text{act}}^{\text{SHF}}(p) = \min\Big(1, \sum_{j \in \mathcal{J}(p)} \alpha(I_j)\Big),$$

where

$$\mathcal{J}(p) = \{j : d_{I_j}(p) \leq r_{\text{SHF}}\}, \qquad d_{I_j}(p) = \min_{q \in I_j} \text{dist}(p, q),$$

with $V_{\text{sem}} = V_{\text{intu}} = 0$ during exploration, and V_{traj} unchanged. Target selection and A* proceed exactly as in FPE/InstructNav. SHF's prompt templates are included in Appendix C.

Design rationale. Unlike InstructNav's Chain-of-Navigation (DCoN), which asks the LLM for the next action/landmark based on recent observations without any explicit geometric inputs such as frontier coordinates or map distances (Long et al., 2024), SHF keeps language as a weak, spatially distributed prior: the LLM only expresses preferences over frontiers, and we diffuse those votes over each frontier's proximity field W_j. This preserves the metric bias in the value-map stage while avoiding prompt–metric mismatch.

Figure 1 summarizes the data flow of FPE and SHF, highlighting the modules that we remove or replace.

5 Experimental Setup

All experiments run in Habitat (release 3) with default navigation mesh and physics (Puig et al., 2023). We evaluate on two standard benchmarks: HM3D (validation split: 20 scenes, 2,000 episodes) and MP3D (validation split: 11 scenes, 2,195 episodes). Agents receive synchronized RGB–D at 640×480. The discrete action set is $\{$`forward`, `turn_left`, `turn_right`, `stop`$\}$, with a 0.20 m forward step and 30° turns. Episodes

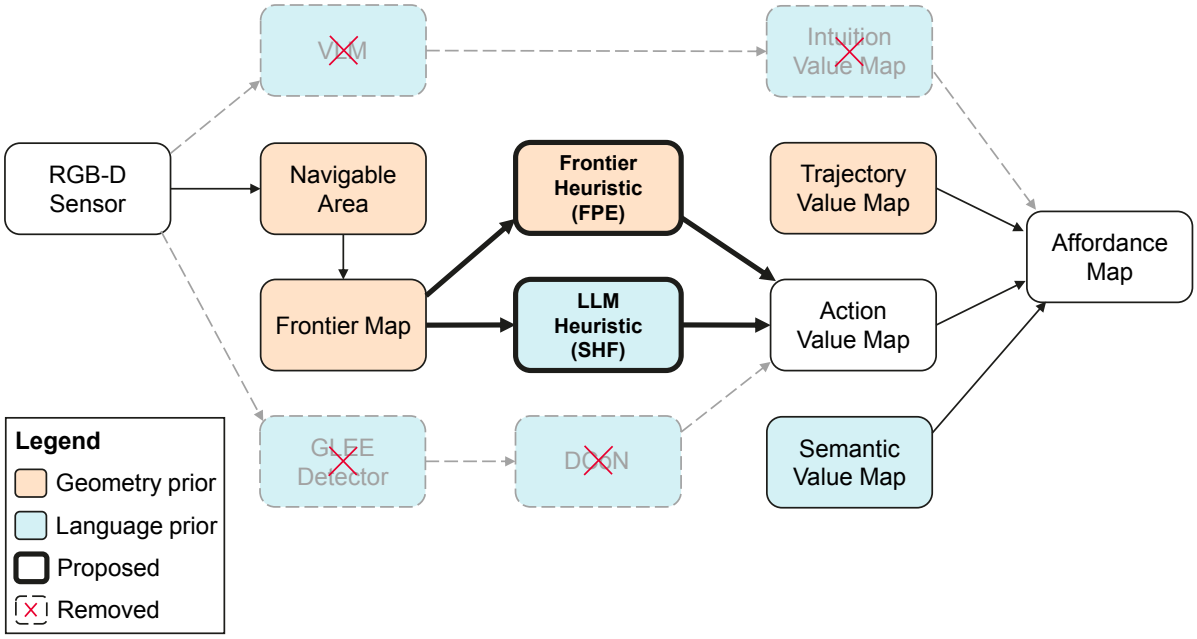


Figure 1: **Planning pipeline with removed and proposed modules.** Peach-color nodes are *geometry priors*; cyan-color nodes are *language priors*; thick black outlines denote our proposed modules. Red × marks indicate components we disable in FPE and SHF. Grey dashed arrows show connections that disappear when a linked module is removed. The RGB-D sensor produces a navigable area map and a frontier map; our FPE injects those frontiers into the Action Value Map. SHF injects LLM heuristics into the Action Value Map, which combines with the Trajectory Value Map to form the Affordance Map during exploration. The Semantic Value Map becomes active after the goal is observed and is used for the standard Approach/Stop behavior.

terminate on `stop` or at a 500-step cap. Success is declared when the goal object is visible and the agent is within 0.25 m. Explored–unknown boundaries define frontier pixels each step; we cluster them with DBSCAN (ε=1.0 m, `min_samples`=1).

Evaluation metrics. We report

- **Success Rate (SR)**: fraction of successful episodes.
- **SPL**: Success weighted by Path Length, $\frac{1}{N}\sum_{i=1}^{N} S_i \frac{\ell_i^\star}{\max(\ell_i,\ell_i^\star)}$. In episode i, $\ell_i^\star$ denotes the length of the optimal path, while ℓ_i represents the length of the trajectory executed by the agent. The variable S_i serves as a binary indicator of success for the episode.
- **ASPL**: Action-based SPL, defined as $\frac{1}{N}\sum_{i=1}^{N} S_i \frac{\min(a_i^\star,a_i)}{a_i}$, where a_i is the number of discrete actions executed (including turns) and $a_i^\star$ is the optimal action count. This mirrors prior action-budgeted evaluation in exploration and navigation, where turns are not "free" (e.g., step-budgeted mapping efficiency (Sun et al., 2025) and success weighted by number of actions (Chen et al., 2020)).
- **API Cost (USD)**: for methods that call an LLM, we log per-experiment dollar cost from the OpenAI dashboard.
- **Runtime (hours)**: wall-clock evaluation time on a single consumer workstation (details in Appendix D), measured end-to-end per dataset split, including simulator stepping and any API calls.

Baselines. We report side-by-side results with representative training-free and LLM-augmented ObjectNav methods: ZSON (Majumdar et al., 2022), CoW (Gadre et al., 2023), ESC (Zhou et al., 2023), L3MVN (Yu

et al., 2023), VoroNav (Wu et al., 2024b), VLMnav (Goetting et al., 2024), VLFM (Yokoyama et al., 2024) and InstructNav (Long et al., 2024). The original LFG paper (Shah et al., 2023) does not provide a full mapping/controller stack or HM3D-wide runs, so we do not include it in the main comparison table. Instead, in Section 6.1 and Appendix B, we instantiate LFG inside our pipeline for ablation experiments.

Local scoring radii. InstructNav applies nonzero Action Value Map scores only near the corresponding masks (Explore: 0.2 m; MoveForward/TurnLeft/TurnRight: 0.1 m). We mirror this policy by fixing two radii for our variants: r_{FPE}=0.2 m and r_{SHF}=0.1 m.

Language and vision model configuration. The released InstructNav implementation uses `gpt-4-0613` for the Dynamic Chain-of-Navigation (DCoN) module and `gpt-4-vision-preview` for the image-conditioned Intuition module (Long et al., 2024). We were unable to access the same pair of model snapshots in our experimental environment; in particular, `gpt-4-vision-preview` was retired from the OpenAI API before our experiments were conducted (OpenAI, 2026). We therefore use `gpt-4.1` for both the text-only DCoN queries and the image-conditioned Intuition queries. SHF also uses `gpt-4.1` for all frontier-voting queries, with identical API settings across datasets.

GPT-4.1 is a newer model for which OpenAI reports improved instruction-following and vision capabilities (OpenAI, 2025). Nevertheless, replacing the original models may change prompt adherence, output distributions, and downstream action-map behavior. We therefore refer to our controlled baseline as *InstructNav-GT-4.1* and treat it as a reproduction of the InstructNav pipeline rather than an exact replication that differs only in the semantic detector of the published system. SHF uses $k = 5$ positive and $k = 5$ negative trials per planning step in the main experiments, following LFG (Shah et al., 2023); Appendix B.2 evaluates smaller values of k. FPE issues no language- or vision-model calls.

Semantic sensor. Open-vocabulary detectors (e.g., GLEE) used in prior pipelines are known to produce high false-positive masks in indoor scenes, which then propagate into the semantic value map and confound the planner (Figure 3). Because our focus is on planning and frontier selection rather than detector accuracy, we use Habitat's ground-truth semantic sensor as a measurement control in evaluation:

- **InstructNav-GT-4.1.** This baseline follows the released InstructNav architecture, prompt templates, value-map construction, and planning pipeline, but differs from the published configuration in two respects. First, semantic masks are obtained from Habitat's ground-truth semantic sensor rather than GLEE. Second, the original language and vision model snapshots are replaced with GPT-4.1, as described above. The baseline therefore provides a matched-perception comparison within our implementation, but it should not be interpreted as an exact detector-only ablation of the published InstructNav result.
- **Usage across methods.** In our runs: (i) FPE does not consume semantics for exploration but, like all methods, follows the standard `Approach/Stop` behavior once the goal is observed using the semantic labels; (ii) SHF forms cluster object-name sets from current observations and uses the same `Approach/Stop` convention; and (iii) InstructNav-GT-4.1 uses ground-truth semantics wherever InstructNav would use detector outputs.

Reproducibility. All API calls used the default decoding settings, and no random seed was specified. The only stochastic component in our experimental pipeline is the LLM output. All methods within each comparison were evaluated on the same ordered episodes. Full evaluations use all episodes in the validation split. Reduced evaluations use episodes 0–9 from each HM3D scene and episodes 0–19 from each MP3D scene. Code, evaluation logs, prompts, configuration files, and implementation details are included in the anonymous supplementary materials available to reviewers. We will provide a public repository link in the camera-ready version.

6 Results

Comparison to prior work. Table 1 summarizes results reported by prior work under their respective original experimental configurations. These results provide context but are not directly controlled comparisons because the methods use different perception and model settings. Our primary comparison is shown

Method	HM3D		MP3D	
	SR(%)↑	SPL(%)↑	SR(%)↑	SPL(%)↑
ZSON (Majumdar et al., 2022)	25.5	12.6	15.3	4.8
CoW (Gadre et al., 2023)	–	–	9.2	4.9
ESC (Zhou et al., 2023)	39.2	22.3	28.7	14.2
L3MVN (Yu et al., 2023)	50.4	23.1	–	–
VoroNav (Wu et al., 2024b)	42.0	26.0	–	–
VLMNav (Goetting et al., 2024)	50.4	21.0	–	–
VLFM (Yokoyama et al., 2024)	52.5	**30.4**	**36.4**	**17.5**
InstructNav (Long et al., 2024)	**58.0**	20.9	–	–

Table 1: **Published zero-shot ObjectNav results on HM3D and MP3D.** Columns report Success Rate (SR) and Success weighted by Path Length (SPL); higher is better for both metrics. Results are taken from the corresponding papers and use their respective original perception, model, and evaluation configurations. They are therefore provided as contextual comparisons rather than as a controlled ranking against our methods. **Bold** denotes the highest reported value for each dataset and metric. "–" indicates that the corresponding result was not reported in the original paper.

Method	HM3D			MP3D		
	SR(%)↑	SPL(%)↑	ASPL(%)↑	SR(%)↑	SPL(%)↑	ASPL(%)↑
InstructNav-GT-4.1	60.6	33.8	21.3	**50.6**	24.0	17.0
FPE (ours)	**61.4**	36.0	**23.5**	48.0	**24.5**	**17.5**
SHF (ours)	61.2	**36.2**	23.0	47.1	23.1	16.4

Table 2: **Controlled ObjectNav comparison under matched ground-truth semantic perception.** All methods are evaluated on the HM3D and MP3D validation sets using the simulator's ground-truth semantic sensor and the same mapping and planning implementation. Columns report Success Rate (SR), Success weighted by Path Length (SPL), and Action-based SPL (ASPL); higher is better for all metrics. **Bold** denotes the highest value within this controlled comparison.

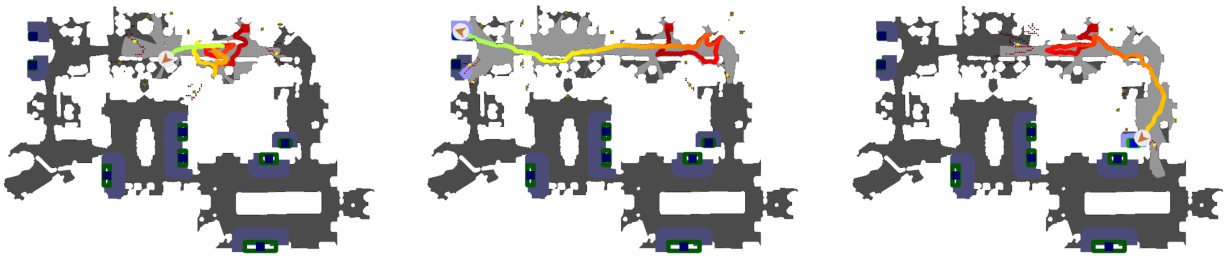
in Table 2, where InstructNav-GT-4.1, FPE, and SHF are evaluated under matched ground-truth semantic perception and the same mapping/planning implementation. Against InstructNav-GT-4.1, FPE improves SR on HM3D from 60.6% to 61.4%, SPL from 33.8% to 36.0%, and ASPL from 21.3% to 23.5%, at \$0 API cost and with $\sim$2.5$\times$ lower runtime (Table 3). On MP3D, InstructNav-GT-4.1 attains the highest SR (50.6%), while FPE achieves the best SPL (24.5%) and ASPL (17.5%) at \$0 cost and $\sim$2.3$\times$ lower runtime. Under matched ground-truth semantic perception, these results show that a simple frontier-geometry exploration policy attains aggregate performance comparable to InstructNav-GT-4.1 while substantially reducing runtime and eliminating API cost.

Efficiency and cost. FPE has the lowest runtime on both datasets and incurs no API cost. SHF matches or nearly matches InstructNav-GT-4.1 while being faster, because it discards DCoN's step-wise reasoning and Intuition VLM calls. Its higher API cost stems from the LFG-style voting budget (k=5 positive and k=5 negative prompts per step). Given the small gap to InstructNav-GT-4.1, a reduced k (e.g., 1–3) is a promising cost knob; we further explore this in Appendix B.2.

Qualitative trends. Figure 2 visualizes an illustrative MP3D episode: InstructNav-GT-4.1 meanders and times out; FPE explores a few frontiers then reaches the goal; SHF selects a goal-relevant frontier early and follows a shorter route. Figure 4 shows per-step Action Value Maps. A common failure observed in InstructNav-GT-4.1 is an all-zero Action Value Map, for which the DCoN output provides no usable spatial preference to the downstream planner. By contrast, FPE produces dense nonzero values near frontiers (within r_{FPE}), and SHF injects frontier-local boosts (within r_{SHF}) via LLM votes that the shared affordance and A* planner can exploit. We also quantitatively measure this effect in Appendix B.1.

Method	HM3D		MP3D	
	Cost($)↓	Runtime(hrs)↓	Cost($)↓	Runtime(hrs)↓
InstructNav-GT-4.1	242.5	237.0	207.8	204.4
FPE (ours)	**0**	**95.8**	**0**	**87.4**
SHF (ours)	366.3	190.5	424.2	180.0

Table 3: **API cost and wall-clock runtime on HM3D and MP3D.** Costs are cumulative USD from the OpenAI dashboard; runtime is wall-clock hours on a single workstation. Baseline papers generally do not report cost/runtime.



(a) InstructNav-GT-4.1 (b) FPE (c) SHF

Figure 2: **Trajectory comparison on the same MP3D episode.** (a) InstructNav-GT-4.1 wanders, makes many detours, and times out at the 500-step cap without seeing the goal. (b) FPE explores a few frontiers and then reaches the goal. (c) SHF follows a shorter path, guided by the LLM semantic vote, and reaches the goal quickest. Each blue square in the map represents a goal object.

6.1 Ablations

To probe the impact of perception noise, we re-ran InstructNav-4.1, FPE, SHF, and LFG on reduced validation splits (first 10 episodes per HM3D scene, 200 episodes total; first 20 episodes per MP3D scene, 220 episodes total), using either the GLEE detector or the simulator's semantic ground-truth. Since the original LFG implementation only releases prompting code and does not provide a full mapping/controller stack, we instantiate LFG inside the same InstructNav-based pipeline by removing the Trajectory Value Map (which discourages revisiting explored areas) and replacing the action value map with the LFG-style vote-based frontier scores. We report SR, SPL, and Action-based SPL (ASPL).

Using ground-truth semantics, FPE obtains the highest SR and ASPL on the reduced HM3D split, while LFG obtains the highest SPL by 0.4 percentage points (Table 4). On MP3D, FPE obtains the highest SR, SPL, and ASPL (Table 5). These reduced-split results are consistent with the full-split controlled comparison: under matched semantic perception, the simple frontier-geometry policy is competitive with the evaluated instruction-driven and LFG-style controllers.

When ground-truth semantics are replaced with GLEE detections, performance decreases for all methods and the ranking becomes less consistent. On HM3D, InstructNav-4.1 and LFG tie for the highest SR, while LFG obtains the highest SPL and ASPL. On MP3D, InstructNav-4.1 obtains the highest SR, whereas SHF obtains the highest SPL and ties FPE for the highest ASPL (Tables 4 and 5). These results are consistent with perception noise substantially affecting comparisons among planning strategies. Language-conditioned methods remain competitive under GLEE perception, but the full InstructNav stack does not obtain a consistent advantage over the simpler alternatives across datasets and metrics.

Appendix B.3 provides category- and scene-level breakdowns under both ground-truth and GLEE perception. The results show that SHF provides selective gains in particular categories and environments, but does not consistently improve over FPE across either perception setting.

Method	HM3D (GLEE)			HM3D (GT)		
	SR(%)↑	SPL(%)↑	ASPL(%)↑	SR(%)↑	SPL(%)↑	ASPL(%)↑
InstructNav-4.1	**44.0**	22.3	13.8	58.5	34.1	22.1
FPE	41.5	21.4	13.8	**59.0**	34.1	**22.7**
SHF	40.5	21.6	13.5	55.0	34.1	22.0
LFG	**44.0**	**24.6**	**15.1**	55.0	**34.5**	22.0

Table 4: **Detector vs. ground-truth semantics on HM3D.** All methods share the same mapping/planning stack; LFG is implemented following the original specification (no trajectory value map).

Method	MP3D (GLEE)			MP3D (GT)		
	SR(%)↑	SPL(%)↑	ASPL(%)↑	SR(%)↑	SPL(%)↑	ASPL(%)↑
InstructNav-4.1	**23.6**	10.4	6.9	41.8	21.6	15.3
FPE	21.8	11.1	7.4	**47.7**	**26.1**	**18.1**
SHF	20.9	**11.5**	**7.4**	44.6	24.3	16.8
LFG	21.8	10.5	7.0	39.6	21.9	15.1

Table 5: **Detector vs. ground-truth semantics on MP3D.** All methods share the same mapping/planning stack; GLEE is substantially less reliable on MP3D, where target objects are more diverse than in HM3D.

7 Discussion

Geometry under matched perception. Our objective is not to establish that FPE is statistically superior to InstructNav-GT-4.1. Rather, we test whether the added LLM/VLM components yield a consistent aggregate ObjectNav improvement over a matched no-LLM control. Across the full ground-truth evaluations, we do not observe such an improvement: FPE obtains similar aggregate performance while requiring no API calls and substantially less runtime. This finding is limited to our evaluation of InstructNav for ObjectNav and does not establish that FPE is superior to LLM-guided navigation generally or to other published systems under matched perception.

What the SHF results show. SHF does not consistently improve over FPE. On the full validation splits, SHF is slightly stronger on HM3D SPL but weaker on the remaining MP3D metrics, and on the reduced ground-truth splits FPE generally obtains higher aggregate values. The category- and scene-level analysis nevertheless identifies selected cases in which semantic voting is beneficial. We therefore interpret SHF as a diagnostic implementation of localized language guidance rather than evidence that language heuristics provide a systematic aggregate advantage. These conclusions apply to the evaluated LFG-style voting mechanism and do not generalize to all possible language-grounding or metric-aware prompting designs.

Action-map grounding in our reproduction. InstructNav-GT-4.1 frequently produces uniformly zero Action Value Maps, meaning that the generated high-level action does not yield a usable spatial preference for the downstream planner. FPE and SHF avoid this particular failure by constructing frontier-local spatial values directly. Because InstructNav-GT-4.1 uses GPT-4.1 rather than the original language and vision model snapshots, the measured empty-map rate characterizes our reproduction and should not be interpreted as a universal property of DCoN. The result nevertheless suggests that explicitly grounding model outputs in the metric representation used by the planner may improve action-map reliability.

Detection noise and apparent robustness of LLM pipelines. The detector-vs.-GT ablations in Section 6.1 suggest language-conditioned planners may appear more robust under noisy open-vocabulary detection. One mechanism is that FPE, while strong under GT semantics, only uses semantics to trigger goal-approach when the goal is detected; thus a single false negative when the agent is near the goal can become a point of failure. In contrast, InstructNav-4.1, SHF, and LFG consult the detector continuously during exploration (querying surrounding objects and accumulating semantic context over time), which can partially average out intermittent errors and reduce reliance on a single decisive detection. We do not

isolate this recovery mechanism experimentally, so it should be interpreted as a plausible explanation for the observed results rather than as a causal conclusion. That said, even in the noisy GLEE setting, we do not observe a clear advantage for the full InstructNav stack over substantially simpler language-as-heuristic variants (SHF/LFG), suggesting that any robustness benefit is driven more by repeated semantic polling than by step-wise LLM planning or additional VLM components.

SHF vs. LFG and the role of value maps. A second takeaway from Tables 4–5 is that the adapted LFG-style scorer is competitive with, and sometimes stronger than, SHF despite having lower algorithmic complexity. This indicates that several of InstructNav's auxiliary value-map components may contribute little once frontier scoring provides a usable preference signal. In particular, removing the Trajectory Value Map (which is specific to InstructNav and not part of the original LFG specification) does not harm LFG performance and can slightly improve it. Overall, these reduced-split results show that the adapted LFG scorer is competitive with SHF within the shared implementation. Determining the independent contribution of the Trajectory, Intuition, and other value-map components would require a full factorial ablation, which we leave for future work.

Practical deployment considerations. While our experiments are simulation-based, FPE/SHF are designed with deployment constraints in mind. All methods require maintaining a local geometric map from depth for obstacle avoidance and frontier extraction; this memory footprint is dominated by the mapping stack rather than language components. InstructNav-based pipelines additionally incur substantial overhead from step-wise LLM planning (DCoN) and vision–language inference, and may suffer from unpredictable API latency and network dependence. By contrast, FPE eliminates LLM/VLM calls entirely, and SHF restricts language to lightweight frontier voting (with controllable budget k), making runtime and latency more predictable. These properties motivate future evaluation on physical robots under real-time latency and network constraints. However, the present simulation experiments do not establish real-world navigation performance or robustness.

8 Conclusion

This work presents a controlled re-evaluation of the InstructNav pipeline on ObjectNav. Under matched ground-truth semantic perception, FPE obtains aggregate performance comparable to InstructNav-GT-4.1 while requiring no API calls and substantially less runtime. Under GLEE perception, the evaluated methods likewise remain broadly competitive, with no consistent winner across datasets and metrics. Taken together, these results do not show a consistent aggregate advantage from the full LLM/VLM stack over substantially simpler controls, while also demonstrating that perception quality affects the metric-specific outcomes.

SHF does not consistently improve over FPE at the aggregate level, although the category- and scene-level analyses identify selected cases in which semantic voting is beneficial. We therefore do not conclude that language is generally unnecessary or generally beneficial for navigation. Our narrower finding is that the evaluated LFG-style voting heuristic provides context-dependent gains, whereas a simple geometry-first exploration policy is a strong control for the InstructNav pipeline on ObjectNav.

Methodological implications. Evaluations of LLM-augmented ObjectNav should report runtime and API cost, distinguish published results from matched-perception comparisons, and include strong frontier-geometry controls. A full factorial analysis of DCoN, Intuition, frontier geometry, and the remaining value maps would be a valuable direction for future work.

Limitations and scope. Our experiments are restricted to simulated ObjectNav and do not evaluate route following, compositional instructions, demand-driven navigation, or real-robot deployment. The InstructNav-GT-4.1 reproduction also uses GPT-4.1 in place of the original language and vision model snapshots, and the GLEE comparisons use reduced validation splits. Consequently, the results should be interpreted as a diagnostic study of the evaluated InstructNav pipeline on ObjectNav rather than a general conclusion about the role of language models in embodied navigation.

Broader impact. The results should not be interpreted as evidence that language models are unnecessary for embodied navigation generally. Language remains intrinsic to tasks that require interpreting user instruc-

tions, relations, constraints, dialogue, or multi-step goals. The intended impact of this study is methodological: encouraging matched controls and transparent efficiency reporting when evaluating language-conditioned navigation systems.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

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

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

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

Changan Chen, Sagnik Majumder, Ziad Al-Halah, Ruohan Gao, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Learning to set waypoints for audio-visual navigation. *arXiv preprint arXiv:2008.09622*, 2020.

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

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

Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. End-to-end navigation with vision language models: Transforming spatial reasoning into question-answering. *arXiv preprint arXiv:2411.05755*, 2024.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.

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

OpenAI. Introducing GPT-4.1 in the API. `https://openai.com/index/gpt-4-1/`, April 2025. Accessed: 2026-07-16.

OpenAI. OpenAI API deprecations. `https://developers.openai.com/api/docs/deprecations`, 2026. Accessed: 2026-07-16.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

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

Dhruv Shah, Michael Robert Equi, Błażej Osiński, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. In *Conference on Robot Learning*, pp. 2683–2699. PMLR, 2023.

Boyang Sun, Hanzhi Chen, Stefan Leutenegger, Cesar Cadena, Marc Pollefeys, and Hermann Blum. Frontiernet: Learning visual cues to explore. *IEEE Robotics and Automation Letters*, 2025.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Junfeng Wu, Yi Jiang, Qihao Liu, Zehuan Yuan, Xiang Bai, and Song Bai. General object foundation model for images and videos at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3783–3795, 2024a.

Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. Voronav: Voronoi-based zero-shot object navigation with large language model. *arXiv preprint arXiv:2401.02695*, 2024b.

Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pp. 146–151. IEEE, 1997.

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

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

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

A Open-vocabulary detector failures

(a) GLEE detector on HM3D

(b) GLEE detector on MP3D

Figure 3: **Failure modes of the open-vocabulary segmenter used in InstructNav.** Typical cases include no detections under indoor lighting/texture and near-image-wide false positives on stylized surfaces. These errors corrupt the semantic value map and can mislead downstream planning. Our use of a ground-truth semantic sensor for the fairness control (InstructNav-GT-4.1) prevents such noise in the baselines used for comparison.

B Additional Ablations

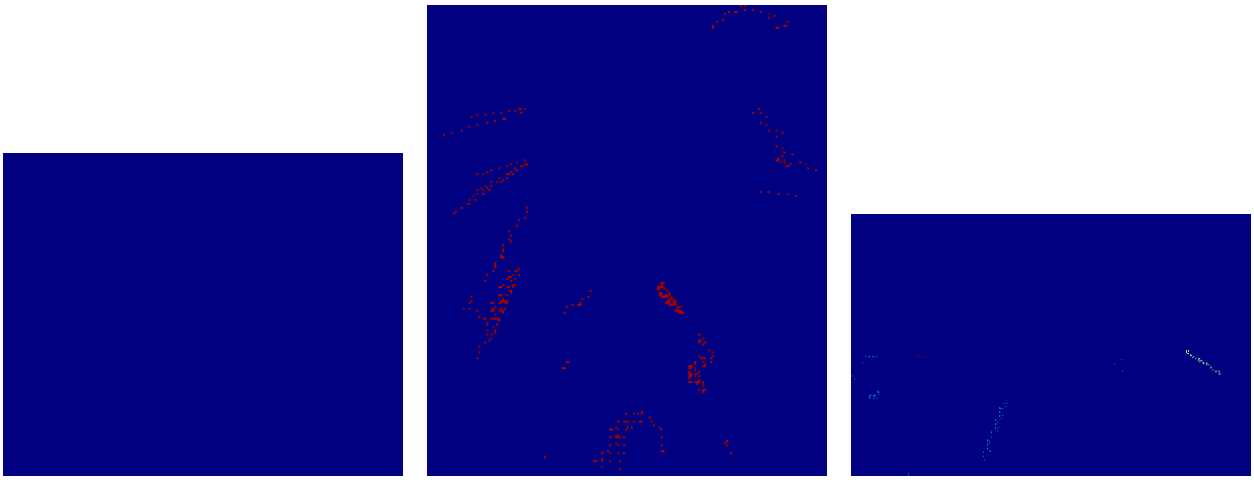
Unless otherwise noted, the experiments in this appendix use the reduced HM3D and MP3D splits introduced in Section 6.1. Appendix B.3 additionally reports category- and scene-level results from the full validation runs under ground-truth semantic perception.

B.1 Empty Action Value Maps

We call an Action Value Map (AVM) empty if all navigable cells receive zero score, i.e., the planner has no preference signal to follow. Such cases can arise when no valid frontier or landmark mask receives positive mass, or when the controller output does not map to a usable spatial preference. Table 6 reports the fraction of empty AVMs generated under ground-truth semantics. The result for InstructNav-GT-4.1 characterizes our GPT-4.1-based reproduction and may differ from the behavior of the original language and vision model configuration used by InstructNav.

Method	HM3D	MP3D
InstructNav-GT-4.1	62.4%	62.5%
FPE	**28.6%**	**36.9%**
SHF	38.6%	41.2%
LFG	38.6%	39.7%

Table 6: **Rate of empty Action Value Maps under ground-truth semantics.** Values denote the percentage of generated AVMs that are uniformly zero. InstructNav-GT-4.1 leaves roughly 62% of planning steps without any action preference. FPE substantially reduces this rate, while SHF and LFG fall in between.



(a) InstructNav-GT-4.1 (b) FPE (c) SHF

Figure 4: **Action Value Maps on the same MP3D episode.** (a) InstructNav-GT-4.1 produces a uniformly zero Action Value Map at this step. This illustrates a failure observed in our GPT-4.1-based reproduction. (b) FPE: dense scores near frontiers within radius r_{FPE}; higher values rendered in red. (c) SHF: frontier-local boosts within r_{SHF}; LLM votes produce per-frontier scores $\alpha(I_j)$ visualized with distinct colors. Differences in the size of local maps arise from the distinct trajectories executed by each method.

B.2 SHF vote budget k

Following LFG (Shah et al., 2023), we set the SHF vote budget to k=5 in the main experiments, issuing k positive and k negative LLM queries per planning step. To probe the cost–accuracy trade-off, we sweep $k \in \{1, 3, 5\}$ under ground-truth semantics.

	HM3D			MP3D		
k	SR(%)↑	SPL(%)↑	ASPL(%)↑	SR(%)↑	SPL(%)↑	ASPL(%)↑
1	**60.0**	**35.5**	**23.0**	**48.2**	**26.0**	**17.9**
3	59.0	34.7	22.3	44.6	24.2	16.7
5	55.0	34.1	22.0	44.6	24.3	16.8

Table 7: **Effect of SHF vote budget k under ground-truth semantics.** Larger k yields more LLM votes per planning step. On these reduced splits, k=1 slightly outperforms k=3 and k=5 in SR/SPL/ASPL on both HM3D and MP3D. For fixed episode lengths, API cost scales approximately linearly with k, so smaller budgets are more cost-effective.

Across both datasets and all three metrics, decreasing k does not hurt performance and in fact slightly improves SR/SPL/ASPL on these subsets. One possible explanation is that averaging over more calls distributes nonzero scores across a more diverse set of frontiers, which can distract the planner from the most useful semantic hints. Since SHF's API cost scales roughly linearly with k, these results further strengthen the case for lightweight, low-budget language priors in this setting and underscore the importance of tuning LLM query budgets.

B.3 Category- and Scene-Level Analysis

We analyze performance by goal category and scene under both ground-truth semantic perception and GLEE perception. The ground-truth tables use the full HM3D and MP3D validation runs, whereas the GLEE tables use the reduced splits introduced in Section 6.1. The final two columns report the difference between SHF and FPE in percentage points. Deltas are computed from unrounded values; all displayed values are rounded to one decimal place. These comparisons are descriptive and are intended to identify settings in which localized

HM3D scene	Qualitative description
4ok3usBNeis	House containing a living room, kitchen, bedroom, bathroom, and a basement with narrow corridors and a laundry room.
5cdEh9F2hJL	Living room and kitchen connected by a narrow corridor to bedrooms and a bathroom.
6s7QHgap2fW	Full house with very narrow corridors.
DYehNKdT76V	Modern house with narrow stairs leading to a second floor with narrow corridors.
Dd4bFSTQ8gi	House without a basement.
Nfvxx8J5NCo	Living room and kitchen connected by a narrow corridor to a laundry room, office, bathroom, and bedroom.
QaLdnwvtxbs	Modern house without a basement.
TEEsavR23oF	Full two-floor house.
XB4GS9ShBRE	House with bedrooms and a bathroom on the second floor.
bxsVRursffK	Full two-floor house.
cvZr5TUy5C5	Two-floor house with a large basement containing a miniature collection, office, living room, and bathroom.
mL8ThkuaVTM	Small two-floor house with narrow corridors and a basement.
mv2HUxq3B53	Large house without a basement.
p53SfW6mjZe	Large two-floor house with a basement and a game room on the second floor.
q3zU7Yy5E5s	Second floor of a wooden house with no access to the main floor.
qyAac8rV8Zk	House with a narrow second-floor corridor and a small basement.
svBbv1Pavdk	Full house without a basement.
wcojb4TFT35	Crowded garage containing a car, bicycles, and assorted equipment.
ziup5kvtCCR	Large house without a basement.
zt1RVoi7PcG	Two-floor house.

Table 8: **Qualitative descriptions of the HM3D evaluation scenes.** Descriptions were assigned manually after scene inspection and are included only to support qualitative interpretation of the scene-level results.

MP3D scene	Qualitative description
2azQ1b91cZZ	Large two-floor house with classical architecture.
8194nk5LbLH	Modern house with a gym on the second floor.
EU6Fwq7SyZv	Large three-floor wooden house with classical decoration.
QUCTc6BB5sX	Large garage containing a car, with no access to the main floor.
TbHJrupSAjP	Large two-floor house with a large basement containing a gym.
X7HyMhZNoso	Two-floor house with a grass yard.
Z6MFQCViBuw	Royal-style building containing antique objects and used as a museum.
oLBMNvg9in8	Old three-floor house containing antique objects.
pLe4wQe7qrG	Small old church.
x8F5xyUWy9e	Large modern basement containing a pool, with no access to the main floor.
zsNo4HB9uLZ	Large modern house with narrow corridors and no basement.

Table 9: **Qualitative descriptions of the MP3D evaluation scenes.** Descriptions were assigned manually after scene inspection and are included only to support qualitative interpretation of the scene-level results.

language guidance helps or fails to help. Tables 8 and 9 provide brief scene descriptions assigned manually after inspecting the environments. The descriptions were not used by any navigation method or to define experimental subgroups and serve only to support qualitative interpretation.

B.3.1 Ground-truth semantic perception

Goal categories. On HM3D, the clearest SHF gain relative to FPE occurs for `toilet`, where SHF improves SR by 5.8 percentage points and SPL by 7.7 points. FPE is stronger for `bed`, `chair`, and `sofa`, while the differences for `plant` and `tv_monitor` are small or mixed across metrics. Thus, the HM3D breakdown does not show a systematic category-level benefit from semantic voting.

On MP3D, SHF obtains its largest gains relative to FPE for `clothes` (+16.0 SR and +3.5 SPL), `bathtub` (+11.1 SR and +5.5 SPL), `sofa` (+7.5 SR and +1.6 SPL), and `fireplace` (+6.7 SR and +4.0 SPL). Several of these categories contain relatively few episodes, so these differences should be treated as exploratory. FPE is stronger for a number of other categories, including `plant`, `seating`, `stool`, and `towel`. Overall, semantic voting provides selective rather than consistent category-level gains.

Scenes. On HM3D, SHF obtains its largest gains relative to FPE in `cvZr5TUy5C5` (+8.1 SR and +3.3 SPL), `q3zU7Yy5E5s` (+8.1 SR and +4.3 SPL), and `wcojb4TFT35` (+7.1 SR and +5.6 SPL). These scenes

Goal category	n	SR (%) ↑ InstructNav-GT-4.1	FPE	SHF	SPL (%) ↑ InstructNav-GT-4.1	FPE	SHF	Δ(SHF–FPE), pp SR	SPL
bed	433	**52.7**	49.4	48.7	26.1	**26.3**	25.0	-0.7	-1.4
chair	428	72.4	**82.9**	79.7	40.4	**50.5**	47.9	-3.3	-2.6
plant	84	45.2	**50.0**	48.8	24.1	30.6	**31.3**	-1.2	+0.7
sofa	376	73.1	**76.3**	74.5	43.9	**49.4**	46.6	-1.9	-2.7
toilet	398	**59.6**	48.0	53.8	**32.9**	23.8	31.4	+5.8	+7.7
tv_monitor	281	43.8	**49.1**	48.8	26.5	30.0	**30.1**	-0.4	+0.1

Table 10: **HM3D goal-category-level results under ground-truth semantic perception.** Results use the full validation run. Positive values in the final two columns indicate higher performance for SHF than FPE, while negative values indicate higher performance for FPE. Bold denotes the highest value among the evaluated methods for each row and metric.

Goal category	n	SR (%) ↑ InstructNav-GT-4.1	FPE	SHF	SPL (%) ↑ InstructNav-GT-4.1	FPE	SHF	Δ(SHF–FPE), pp SR	SPL
bathtub	9	**55.6**	33.3	44.4	**36.2**	22.2	27.7	+11.1	+5.5
bed	48	**33.3**	31.2	27.1	**15.9**	14.8	15.3	-4.2	+0.4
cabinet	180	52.2	**53.9**	53.3	29.3	**30.3**	27.6	-0.6	-2.7
chair	434	**54.1**	50.5	50.7	26.3	**26.5**	24.8	+0.2	-1.7
chest_of_drawers	63	**34.9**	27.0	27.0	**16.9**	12.9	13.6	+0.0	+0.7
clothes	25	16.0	20.0	**36.0**	7.7	12.7	**16.2**	+16.0	+3.5
counter	66	**43.9**	39.4	40.9	19.5	**23.2**	22.1	+1.5	-1.2
cushion	239	**55.2**	49.4	49.4	**23.0**	21.5	22.0	+0.0	+0.5
fireplace	30	**73.3**	63.3	70.0	32.9	30.9	**34.9**	+6.7	+4.0
gym_equipment	44	**13.6**	6.8	6.8	**4.8**	2.6	3.4	+0.0	+0.8
picture	161	28.6	**33.5**	32.3	16.3	**18.3**	17.1	-1.2	-1.1
plant	87	29.9	**33.3**	26.4	18.0	**21.8**	18.0	-6.9	-3.8
seating	87	63.2	**77.0**	69.0	36.1	**43.9**	40.2	-8.0	-3.7
shower	70	**55.7**	54.3	**55.7**	25.5	32.0	**32.8**	+1.4	+0.8
sink	82	**47.6**	46.3	43.9	22.6	**22.9**	21.6	-2.4	-1.3
sofa	53	**45.3**	32.1	39.6	**26.5**	21.3	22.9	+7.5	+1.6
stool	68	**38.2**	36.8	29.4	19.5	**20.7**	15.3	-7.4	-5.4
table	340	**75.0**	70.3	69.7	**31.5**	30.7	29.7	-0.6	-1.0
toilet	28	**35.7**	21.4	17.9	**19.5**	9.6	6.7	-3.6	-2.9
towel	73	**32.9**	23.3	16.4	10.3	**12.9**	4.7	-6.9	-8.2
tv_monitor	8	**12.5**	**12.5**	0.0	**6.9**	5.8	0.0	-12.5	-5.8

Table 11: **MP3D goal-category-level results under ground-truth semantic perception.** Results use the full validation run. Positive values in the final two columns indicate higher performance for SHF than FPE, while negative values indicate higher performance for FPE. Bold denotes the highest value among the evaluated methods for each row and metric.

include, respectively, a large multi-room with basement, an isolated upper floor, and a crowded garage with distinctive object context. In contrast, FPE is substantially stronger in `5cdEh9F2hJL` and `DYehNKdT76V`, which contain narrow corridor- or stair-connected layouts. This pattern is consistent with semantic guidance being selectively useful in some semantically distinctive spaces, while frontier proximity can be more effective when geometric connectivity dominates. The pattern is not universal, and the scene descriptions were assigned post hoc; the interpretation is therefore qualitative.

On MP3D, positive SHF–FPE scene differences are generally small. SHF's largest SR gain is 2.5 points in the isolated garage scene `QUCTc6BB5sX`. FPE is substantially stronger in `TbHJrupSAjP` and `x8F5xyUWy9e`, which contain a large basement gym and a pool basement, respectively. These examples do not reveal a single reliable environmental predictor of when language guidance will help.

B.3.2 GLEE perception

The GLEE breakdowns use only 10 episodes per HM3D scene and 20 episodes per MP3D scene. Category counts can therefore be small, and the results should be interpreted as exploratory.

Scene	n	SR (%) ↑ InstructNav-GT-4.1	FPE	SHF	SPL (%) ↑ InstructNav-GT-4.1	FPE	SHF	Δ(SHF–FPE), pp SR	SPL
4ok3usBNeis	99	**87.9**	85.9	81.8	50.7	**54.4**	51.6	-4.0	-2.9
5cdEh9F2hJL	99	64.6	**89.9**	73.7	30.4	**44.5**	35.6	-16.2	-8.9
6s7QHgap2fW	99	**39.4**	38.4	36.4	23.8	**26.9**	23.3	-2.0	-3.6
bxsVRursffK	99	33.3	**60.6**	51.5	19.5	29.0	**30.4**	-9.1	+1.4
cvZr5TUy5C5	99	37.4	43.4	**51.5**	19.7	25.1	**28.4**	+8.1	+3.3
Dd4bFSTQ8gi	99	70.7	81.8	**85.9**	34.6	**49.1**	48.2	+4.0	-1.0
DYehNKdT76V	99	**72.7**	61.6	51.5	**43.1**	39.8	33.1	-10.1	-6.7
mL8ThkuaVTM	99	33.3	30.3	**34.3**	18.8	16.9	**19.6**	+4.0	+2.7
mv2HUxq3B53	99	**72.7**	65.7	69.7	**40.3**	38.2	39.0	+4.0	+0.8
Nfvxx8J5NCo	99	**66.7**	62.6	65.7	32.3	34.3	**34.8**	+3.0	+0.4
p53SfW6mjZe	99	**49.5**	44.4	39.4	**27.4**	26.6	27.1	-5.1	+0.5
q3zU7Yy5E5s	99	56.6	52.5	**60.6**	31.5	31.8	**36.1**	+8.1	+4.3
QaLdnwvtxbs	99	**88.9**	86.9	81.8	**57.3**	55.4	53.2	-5.0	-2.1
qyAac8rV8Zk	99	34.3	33.3	**37.4**	16.4	18.5	**18.9**	+4.0	+0.4
svBbv1Pavdk	99	85.9	89.9	**90.9**	53.4	**58.1**	57.6	+1.0	-0.5
TEEsavR23oF	99	**55.6**	47.5	50.5	**31.6**	26.6	31.0	+3.0	+4.4
wcojb4TFT35	99	65.7	64.6	**71.7**	40.8	41.2	**46.7**	+7.1	+5.6
XB4GS9ShBRE	99	**52.5**	42.4	45.5	**27.9**	22.2	24.7	+3.0	+2.5
ziup5kvtCCR	99	88.9	89.9	**90.9**	47.5	48.3	**52.1**	+1.0	+3.8
zt1RVoi7PcG	119	55.5	**56.3**	54.6	30.4	**34.0**	33.8	-1.7	-0.2

Table 12: **HM3D scene-level results under ground-truth semantic perception.** Results use the full validation run. Positive values in the final two columns indicate higher performance for SHF than FPE, while negative values indicate higher performance for FPE. Bold denotes the highest value among the evaluated methods for each row and metric.

Scene	n	SR (%) ↑ InstructNav-GT-4.1	FPE	SHF	SPL (%) ↑ InstructNav-GT-4.1	FPE	SHF	Δ(SHF–FPE), pp SR	SPL
2azQ1b91cZZ	200	**48.0**	42.5	44.0	23.6	**24.4**	21.9	+1.5	-2.6
8194nk5LbLH	201	**27.9**	24.4	22.9	**17.3**	15.4	13.9	-1.5	-1.5
EU6Fwq7SyZv	198	**49.0**	45.0	46.5	**25.2**	24.0	24.5	+1.5	+0.5
oLBMNvg9in8	201	**59.2**	56.2	58.2	27.3	**28.6**	27.4	+2.0	-1.2
pLe4wQe7qrG	200	**91.5**	88.5	90.0	40.9	**43.3**	43.1	+1.5	-0.2
QUCTc6BB5sX	197	**29.9**	20.3	22.8	**15.3**	10.2	10.7	+2.5	+0.5
TbHJrupSAjP	200	39.0	**48.0**	36.5	18.2	**22.8**	18.2	-11.5	-4.6
X7HyMhZNoso	200	**28.0**	25.5	26.5	**13.4**	12.1	12.6	+1.0	+0.5
x8F5xyUWy9e	200	**51.0**	**51.0**	46.0	24.4	**29.6**	23.6	-5.0	-6.0
Z6MFQCViBuw	200	**61.0**	**61.0**	60.0	28.4	28.3	**28.6**	-1.0	+0.3
zsNo4HB9uLZ	198	**71.7**	65.1	64.1	**30.5**	30.4	29.5	-1.0	-1.0

Table 13: **MP3D scene-level results under ground-truth semantic perception.** Results use the full validation run. Positive values in the final two columns indicate higher performance for SHF than FPE, while negative values indicate higher performance for FPE. Bold denotes the highest value among the evaluated methods for each row and metric.

Goal categories. On HM3D, SHF improves over FPE for `bed` and `toilet`, but is weaker for the remaining categories. The positive difference for `toilet` is consistent with the full ground-truth run, although its magnitude is smaller under GLEE. InstructNav-4.1 or LFG obtains the highest value for several categories, illustrating that the relative benefit of the language-conditioned methods depends on both the category and the way semantic information is incorporated.

On MP3D, SHF improves over FPE for `counter`, `picture`, and `plant`. The gains for `counter` and `picture` are large but are based on only six and seven episodes, respectively. SHF is weaker for `bed`, `cabinet`, and `chair`, while several categories have zero success for every method. Thus, the reduced GLEE category results again show selective rather than systematic benefits.

Goal category	n	SR (%) ↑ InstructNav-4.1	FPE	SHF	LFG	SPL (%) ↑ InstructNav-4.1	FPE	SHF	LFG	Δ(SHF–FPE), pp SR	SPL
bed	48	**35.4**	31.2	**35.4**	33.3	16.0	15.4	**17.3**	15.4	+4.2	+1.8
chair	30	63.3	**66.7**	60.0	**66.7**	29.9	36.3	33.2	**38.3**	-6.7	-3.0
plant	14	**35.7**	**35.7**	28.6	28.6	**21.1**	17.9	17.0	17.9	-7.1	-0.9
sofa	31	**54.8**	51.6	48.4	**54.8**	**36.0**	31.3	30.7	31.9	-3.2	-0.6
toilet	53	**43.4**	39.6	41.5	**43.4**	19.0	14.8	17.9	**23.7**	+1.9	+3.2
tv_monitor	24	29.2	25.0	20.8	**33.3**	16.1	19.0	14.9	**22.6**	-4.2	-4.1

Table 14: **HM3D goal-category-level results under GLEE perception.** Results use the reduced validation split. Positive values in the final two columns indicate higher performance for SHF than FPE, while negative values indicate higher performance for FPE. Bold denotes the highest value among the evaluated methods for each row and metric.

Goal category	n	SR (%) ↑ InstructNav-4.1	FPE	SHF	LFG	SPL (%) ↑ InstructNav-4.1	FPE	SHF	LFG	Δ(SHF–FPE), pp SR	SPL
bed	11	**36.4**	27.3	18.2	27.3	15.5	14.6	13.2	**16.6**	-9.1	-1.4
cabinet	66	**37.9**	34.8	28.8	30.3	14.5	**15.6**	13.4	13.3	-6.1	-2.1
chair	29	41.4	**48.3**	44.8	41.4	18.9	**27.2**	25.1	19.2	-3.5	-2.2
chest_of_drawers	24	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	+0.0	+0.0
counter	6	0.0	0.0	**33.3**	**33.3**	0.0	0.0	**23.1**	20.8	+33.3	+23.1
gym_equipment	20	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	+0.0	+0.0
picture	7	14.3	0.0	14.3	**28.6**	4.5	0.0	5.1	**9.0**	+14.3	+5.1
plant	32	**25.0**	18.8	21.9	21.9	12.6	9.2	**13.0**	11.7	+3.1	+3.8
shower	23	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	+0.0	+0.0
table	2	**100.0**	**100.0**	**100.0**	**100.0**	**87.1**	**87.1**	**87.1**	63.6	+0.0	+0.0

Table 15: **MP3D goal-category-level results under GLEE perception.** Results use the reduced validation split. Positive values in the final two columns indicate higher performance for SHF than FPE, while negative values indicate higher performance for FPE. Bold denotes the highest value among the evaluated methods for each row and metric.

Scenes. On HM3D with GLEE, SHF obtains higher SR than FPE in `Dd4bFSTQ8gi`, `mL8ThkuaVTM`, `Nfvxx8J5NCo`, `q3zU7Yy5E5s`, `svBbv1Pavdk`, and `TEEsavR23oF`, while FPE is stronger in several other scenes. The largest SHF SPL gain occurs in `Nfvxx8J5NCo`. However, each scene contains only ten episodes, and some scene-level patterns differ from the full ground-truth runs. For example, SHF is stronger than FPE in `cvZr5TUy5C5` under the full ground-truth evaluation but weaker on the reduced GLEE split.

On MP3D with GLEE, SHF obtains its clearest gain in `TbHJrupSAjP`, improving SR by 15.0 points and SPL by 12.8 points relative to FPE. It also obtains smaller gains in `EU6Fwq7SyZv`, `oLBMNvg9in8`, and `QUCTc6BB5sX`. Conversely, FPE is stronger in `2azQ1b91cZZ`, `Z6MFQCViBuw`, and `zsNo4HB9uLZ`. The changes between the ground-truth and GLEE breakdowns indicate that category- and scene-level advantages are not stable across perception settings. This is consistent with detection noise interacting with the exploration policy and with the repeated semantic inquiries used by the language-conditioned methods.

Summary. Across both perception settings, language guidance helps in selected categories and scenes but does not provide a consistent improvement over FPE. Moreover, several SHF–FPE differences change direction between the full ground-truth evaluation and the reduced GLEE evaluation. These results support the interpretation that the value of localized language guidance is context-dependent and interacts with perception quality.

Scene	n	SR (%) ↑ InstructNav-4.1	FPE	SHF	LFG	SPL (%) ↑ InstructNav-4.1	FPE	SHF	LFG	Δ(SHF–FPE), pp SR	SPL
4ok3usBNeis	10	**90.0**	**90.0**	70.0	80.0	38.7	**53.5**	39.3	42.4	-20.0	-14.3
5cdEh9F2hJL	10	50.0	50.0	40.0	**60.0**	21.0	**30.6**	15.3	26.4	-10.0	-15.3
6s7QHgap2fW	10	**20.0**	**20.0**	**20.0**	10.0	**17.6**	13.8	14.5	6.5	+0.0	+0.8
bxsVRursffK	10	**20.0**	**20.0**	**20.0**	10.0	**11.0**	8.6	6.5	6.6	+0.0	-2.1
cvZr5TUy5C5	10	10.0	10.0	0.0	**20.0**	2.7	1.6	0.0	**10.0**	-10.0	-1.6
Dd4bFSTQ8gi	10	60.0	60.0	**80.0**	70.0	**32.1**	16.1	20.4	23.0	+20.0	+4.3
DYehNKdT76V	10	30.0	**60.0**	50.0	20.0	20.7	**30.3**	24.8	15.2	-10.0	-5.5
mL8ThkuaVTM	10	**40.0**	20.0	**40.0**	**40.0**	**27.8**	16.9	23.5	27.8	+20.0	+6.6
mv2HUxq3B53	10	50.0	**60.0**	40.0	**60.0**	19.5	34.9	28.6	**36.8**	-20.0	-6.3
Nfvxx8J5NCo	10	30.0	30.0	**50.0**	30.0	10.6	13.3	**28.4**	12.8	+20.0	+15.0
p53SfW6mjZe	10	20.0	**40.0**	30.0	0.0	10.7	**16.7**	9.8	0.0	-10.0	-6.9
q3zU7Yy5E5s	10	**50.0**	30.0	40.0	40.0	**22.5**	14.9	20.4	22.5	+10.0	+5.5
QaLdnwvtxbs	10	**80.0**	70.0	70.0	**80.0**	41.1	43.1	51.1	**57.4**	+0.0	+8.0
qyAac8rV8Zk	10	**20.0**	0.0	0.0	**20.0**	5.5	0.0	0.0	**12.0**	+0.0	+0.0
svBbv1Pavdk	10	60.0	40.0	60.0	**80.0**	22.5	23.2	23.3	**35.4**	+20.0	+0.1
TEEsavR23oF	10	**30.0**	20.0	**30.0**	20.0	14.4	8.5	**14.7**	7.8	+10.0	+6.2
wcojb4TFT35	10	**50.0**	**50.0**	**50.0**	**50.0**	**41.2**	30.4	32.6	35.5	+0.0	+2.1
XB4GS9ShBRE	10	**40.0**	30.0	20.0	30.0	**19.1**	12.3	14.6	18.2	-10.0	+2.2
ziup5kvtCCR	10	80.0	60.0	60.0	**90.0**	47.7	25.5	34.7	**56.7**	+0.0	+9.2
zt1RVoi7PcG	10	50.0	**70.0**	40.0	**70.0**	20.5	34.5	30.0	**39.5**	-30.0	-4.5

Table 16: **HM3D scene-level results under GLEE perception.** Results use the reduced validation split. Positive values in the final two columns indicate higher performance for SHF than FPE, while negative values indicate higher performance for FPE. Bold denotes the highest value among the evaluated methods for each row and metric.

Scene	n	SR (%) ↑ InstructNav-4.1	FPE	SHF	LFG	SPL (%) ↑ InstructNav-4.1	FPE	SHF	LFG	Δ(SHF–FPE), pp SR	SPL
2azQ1b91cZZ	20	**30.0**	**30.0**	25.0	**30.0**	**14.9**	13.9	11.2	10.8	-5.0	-2.6
8194nk5LbLH	20	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	+0.0	+0.0
EU6Fwq7SyZv	20	10.0	5.0	10.0	**15.0**	3.7	1.1	3.9	**5.3**	+5.0	+2.8
oLBMNvg9in8	20	**25.0**	10.0	15.0	10.0	**7.5**	5.3	6.7	3.4	+5.0	+1.4
pLe4wQe7qrG	20	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	+0.0	+0.0
QUCTc6BB5sX	20	0.0	0.0	**5.0**	0.0	0.0	0.0	**1.9**	0.0	+5.0	+1.9
TbHJrupSAjP	20	30.0	20.0	**35.0**	**35.0**	14.4	9.3	**22.1**	19.4	+15.0	+12.8
X7HyMhZNoso	20	45.0	**50.0**	**50.0**	45.0	17.7	29.6	**30.3**	22.1	+0.0	+0.7
x8F5xyUWy9e	20	**20.0**	**20.0**	**20.0**	**20.0**	**14.4**	14.1	14.2	11.9	+0.0	+0.2
Z6MFQCViBuw	20	20.0	**25.0**	15.0	20.0	10.4	**12.1**	6.4	8.2	-10.0	-5.8
zsNo4HB9uLZ	20	**80.0**	**80.0**	55.0	65.0	30.8	**36.8**	29.2	34.4	-25.0	-7.6

Table 17: **MP3D scene-level results under GLEE perception.** Results use the reduced validation split. Positive values in the final two columns indicate higher performance for SHF than FPE, while negative values indicate higher performance for FPE. Bold denotes the highest value among the evaluated methods for each row and metric.

C Prompt Templates

C.1 InstructNav: DCoN (LLM) prompt

```
"You are a wheeled mobile robot working in an indoor environment.\
And you are required to finish a navigation task indicated by a human instruction in a new house.\
Your task is to make a navigation plan for finishing the task as soon as possible.\
The navigation plan should be formulated as a chain as {<Action_1> - <Landmark_1> - <Action_2> -
↪  <Landmark_2> ...}.\
To make the plan, I will provide you the following elements:\
(1) <Navigation Instruction>: The navigation instruction given by the human.\
(2) <Previous Plan>: The completed steps in the plan recording your history trajectory.\
(3) <Semantic Clue>: The list recording all the observed rooms and objects in this house from your
↪  perspective.\
```

```
The allowed <Action> in the plan contains ['Explore','Approach','Move Forward','Turn Left','Turn
↪  Right','Turn Around'].\
The action 'Explore' will lead you to the exploration frontiers to help unlock new areas.\
The action 'Approach' will lead you close to a specific object or room for more detailed observations.\
The allowed <Landmark> should be one appeared semantic instance in the <Navigation Instruction> or <Semantic
↪  Clue>.\
Do not output an imagined instance as <Landmark> which has not been observerd in <Semantic Clue> or
↪  mentioned in the <Navigation Instruction>.\
To select the landmark, you should consider the common house layout, human's habit of objects placement and
↪  the task navigation instruction.\
For example, the sofa is often close to a television, therefore, sofa is a good landmark for finding the
↪  television and to satisfy the human entertainment demand.\
If the action and landmark is clearly specified in the instruction like 'walk forward to the television',
↪  then you can directly decompose the instruction into 'Move_Forward' - 'Television' without 'Explore'
↪  action.\
You only need to plan one <Action> and one <Landmark> ahead, besides, you should output a flag to indicate
↪  whether you have finished the navigation task.\
Therefore, your output answer should be formatted as Answer={'Reason':<Your Reason>, 'Action':<Chosen
↪  Action>, 'Landmark':<Chosen Instance>, 'Flag':<Task Finished Flag>}.\
If you find a specific instance of the target object or a synonyms object, the output 'Flag' should be
↪  True.\
Try to select the <Landmark> that is closely related to the <Navigation Instruction> according to the human
↪  habit.\
Try not repeatly select the same <Landmark> as the <Previous Plan>."
```

C.2 InstructNav: Intuition (VLM) prompt

```
"You are an indoor navigation agent. I give you a panoramic observation image, complete navigation
↪  instruction and the sub-instruction you should execute now.  \
Direction 1 and 11 are ahead, Direction 5 and 7 are back, Direction 3 is to the right, and Direction 9 is to
↪  the left. Please carefully analyze visual information in each direction \
and judge which direction is most suitable for next movement according to the act and landmark mentioned in
↪  the sub-instruction. \
You answer should follow \"Thinking Process\" and \"Judgement\". In the \"Judgement: \" field, you should
↪  only write down direction ID you choose. \
If you think you have arrived the destination, you can answer \"Stop\" in the \"Judgement: \" field. Note
↪  that the \"Direction 5\" and \"Direction 7\" are the directions you just came from. \
Generally, the direction with more navigation landmarks in the complete navigation instruction is better."
```

C.3 SHF (LFG-style votes): positive prompt

```
"""You are a robot exploring an environment for the first time. You will be given an object to look for and
↪  should provide guidance of where to explore based on a series of observations. Observations will be
↪  given as a list of object clusters numbered 1 to N.

Your job is to provide guidance about where we should explore next. For example if we are in a house and
↪  looking for a tv we should explore areas that typically have tv's such as bedrooms and living rooms.

You should always provide reasoning along with a number identifying where we should explore. If there are
↪  multiple right answers you should separate them with commas. Always include Reasoning: <your reasoning>
↪  and Answer: <your answer(s)>. If there are no suitable answers leave the space afters Answer: blank.

Example

User:
I observe the following clusters of objects while exploring a house:

1. sofa, tv, speaker
2. desk, chair, computer
3. sink, microwave, refrigerator

Where should I search next if I am looking for a knife?

Assistant:
```

```
Reasoning: Cluster 1 contains items that are likely part of an entertainment room. Cluster 2 contains
↪  objects that are likely part of an office room and cluster 3 contains items likely found in a kitchen.
↪  Because we are looking for a knife which is typically located in a kitchen, so we should check cluster
↪  3.
Answer: 3


Other considerations

1. You will only be given a list of common items found in the environment. You will not be given room
↪  labels. Use your best judgment when determining what room a cluster of objects is likely to be in.
2. Provide reasoning for each cluster before giving the final answer.
3. Feel free to think multiple steps in advance; for example if one room is typically located near another
↪  then it is ok to use that information to provide higher scores in that direction.
"""
```

C.4 SHF (LFG-style votes): negative prompt

```
"""You are a robot exploring an environment for the first time. You will be given an object to look for and
↪  should provide guidance of where to explore based on a series of observations. Observations will be
↪  given as a list of object clusters numbered 1 to N.

Your job is to provide guidance about where we should not waste time exploring. For example if we are in a
↪  house and looking for a tv we should not waste time looking in the bathroom. It is your job to point
↪  this out.

You should always provide reasoning along with a number identifying where we should not explore. If there
↪  are multiple right answers you should separate them with commas. Always include Reasoning: <your
↪  reasoning> and Answer: <your answer(s)>. If there are no suitable answers leave the space after Answer:
↪  blank.

Example

User:
I observe the following clusters of objects while exploring a house:

1. sofa, tv, speaker
2. desk, chair, computer
3. sink, microwave, refrigerator

Where should I avoid spending time searching if I am looking for a knife?

Assistant:
Reasoning: Cluster 1 contains items that are likely part of an entertainment room. Cluster 2 contains
↪  objects that are likely part of an office room and cluster 3 contains items likely found in a kitchen. A
↪  knife is not likely to be in an entertainment room or an office room so we should avoid searching those
↪  spaces.
Answer: 1,2


Other considerations

1. You will only be given a list of common items found in the environment. You will not be given room
↪  labels. Use your best judgment when determining what room a cluster of objects is likely to be in.
2. Provide reasoning for each cluster before giving the final answer
"""
```

D Hardware Details

Workstation. 32-core Intel(R) Core(TM) i9-14900KF CPU; one discrete GPU accelerator with 16 GB VRAM (model: NVIDIA GeForce RTX 4080); 64 GB RAM.