INTENTIONNAV: A BENCHMARK FOR INTENT-DRIVEN OBJECT NAVIGATION FROM IMPLICIT HUMAN INSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Goal-specified navigation matches a known target description against scene objects. An implicit request can suggest several plausible targets, requiring goal interpretation alongside object search. Average success alone obscures consistency across expressions and execution of inferred goals. We introduce IntentionNav, a benchmark that evaluates expression consistency and navigation under shared goal inputs. It pairs four English expressions of each of 500 intents with the same scene, designated target, and starting pose across 176 indoor scenes. A complementary study uses a revised episode manifest to compare two navigation systems under shared category predictions and correct-category controls. Across three reference backends, 36.6–45.4% of tasks exhibit both successful and unsuccessful outcomes across their expressions, despite similar average success rates. Repeated executions provide a reference for within-expression variability. With identical category predictions, the reference system and MTU3D reach the designated target neighborhood in 60.8% and 34.6% of episodes, and finish there in 18.4% and 15.0%, respectively. Task-level consistency and separate measures of search coverage and completion reveal differences that average success obscures. Code and data are available at https://anonymous.4open.science/r/IntentionNav-D7D1 and https://huggingface.co/datasets/Anonymous260726/IntentionNav, respectively.

1 INTRODUCTION

People express everyday needs without necessarily naming the objects that can satisfy them. A request such as "I need to warm up my food" can suggest several plausible targets, including a microwave or an oven. Acting on it requires interpreting the need and locating a suitable object in the scene (Figure 1). IntentionNav pairs each task's requests with a designated scene object to provide a fixed reference for evaluation.

Object-goal navigation benchmarks evaluate search for a specified category (Anderson et al., 2018a; Batra et al., 2020); demand-driven navigation extends this setting to human needs (Wang et al., 2023a; 2024a). Foundation-model agents further support flexible instructions through reasoning and exploration (Long et al., 2025; Kuang et al., 2024). Evaluating reliable assistance requires distinguishing two failure modes. First, different expressions of the same need may produce different outcomes on an unchanged physical task. Second, an agent may predict the correct category yet fail to find the target or to stop after reaching its vicinity. Aggregate success rates obscure both effects by pooling expressions and combining goal inference with downstream navigation.

IntentionNav studies these failure modes through two paired evaluations. The first asks whether different expressions of the same need lead to consistent completion at the designated target. Four English expressions of each of 500 intents share a target, scene, and starting pose across 176 indoor scenes. The second compares two navigators under shared category predictions and correct-category controls, measuring search coverage, endpoint success, and stopping. It uses a revised episode manifest with updated starts and selected target and language annotations; conditions are matched within each study.

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

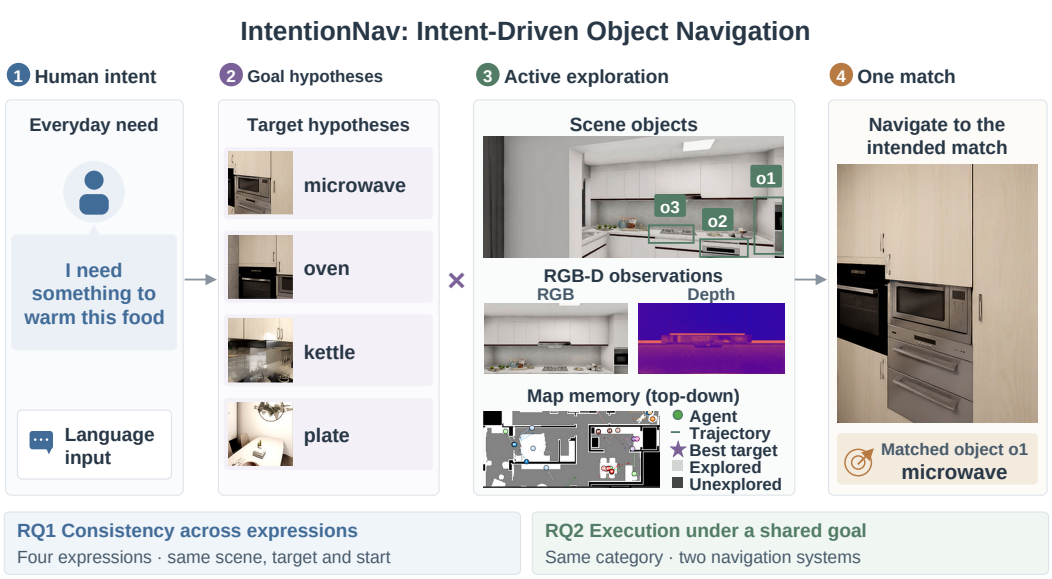


Figure 1: Implicit requests, plausible targets, and navigation to a designated object. The conceptual overview connects implicit requests to scene search and navigation completion. The paired studies test consistency across expressions (RQ1) and execution under shared category inputs (RQ2).

Across 6,000 episodes with three hosted vision-language model (VLM) backends, 36.6–45.4% of tasks have mixed outcomes across expressions, despite similar average success rates. Repeating each expression three times on 40 tasks with Gemini provides an execution reference: outcomes disagree more across expressions than within an expression. In the 2,000-episode system comparison, identical category predictions yield 60.8% and 34.6% target-neighborhood coverage for the reference system and MTU3D, with endpoint success of 18.4% and 15.0%. Correct-category controls and stopping diagnostics relate search coverage to completion.

Our contributions are:

- **Paired evaluation of implicit instructions.** Four expressions per physical task make consistency measurable with the target, scene, and starting pose fixed; semantic cue annotations support analysis across request types.
- **Matched comparisons of goal execution.** Shared-prediction and correct-category controls connect category recovery to search coverage, final success, and explicit stopping across two navigation systems.
- **Findings beyond average success.** Mixed outcomes across expressions and different search and stopping patterns reveal variation hidden by similar aggregate performance.

The fixed benchmark release includes annotations, episode specifications, reference logs, evaluation code, and simulator/scene metadata (Section 3.4).

2 Related Work

Object-goal and language-conditioned navigation. ObjectNav evaluates an agent's ability to find an object from a specified category (Anderson et al., 2018a; Batra et al., 2020). Approaches combine semantic exploration, learned spatial priors, and demonstrations to guide the search (Chaplot et al., 2020; Ramakrishnan et al., 2022; Ramrakhya et al., 2022). Scene datasets and simulators such as Matterport3D, Gibson, Habitat, and HM3D support these evaluations (Chang et al., 2017; Xia et al., 2018; Savva et al., 2019; Ramakrishnan et al., 2021; Szot et al., 2021). Language can specify routes (Anderson et al., 2018b; Chen et al., 2022; Wang et al., 2023b) or refer to objects and regions, as in REVERIE and SOON (Qi et al., 2020; Zhu et al., 2021). Recent benchmarks expand the goals to open-vocabulary categories, multimodal references, personalized objects, and agent capabilities

(Yokoyama et al., 2024b; Khanna et al., 2024; Barsellotti et al., 2024; Ziliotto et al., 2025; Wang et al., 2026; Su et al., 2026). IntentionNav evaluates indirect requests by pairing different expressions with the same physical navigation task.

Implicit needs and free-form goals. DDN formulates navigation goals as user demands, and MO-DDN extends this setting to multiple objects and personal preferences (Wang et al., 2023a; 2024a). IntentionVG examines how human intentions ground to objects in visual observations (Wang et al., 2024b). CogDDN, InstructNav, and OpenFMNav combine reasoning with exploration to handle demands and flexible language goals (Huang et al., 2025; Long et al., 2025; Kuang et al., 2024). Our benchmark builds on this task family to evaluate expression consistency and navigation execution under shared inferred goals. Fixing the target, scene, and starting pose supports comparisons of language robustness; recording both inferred categories and navigation outcomes reveals how each expression is acted upon. Throughout this paper, intent refers to the practical goal communicated by a request.

Foundation-model agents and diagnostics. Foundation models guide zero-shot navigation through multimodal embeddings, commonsense reasoning, frontier values, scene graphs, and explicit planning (Majumdar et al., 2022; Gadre et al., 2023; Zhou et al., 2023; Yokoyama et al., 2024a; Yin et al., 2024; Zhou et al., 2024b;a; Chen et al., 2024; Zheng et al., 2024). MTU3D jointly learns object grounding and frontier selection from accumulated 3D observations (Zhu et al., 2025). These systems provide different mechanisms for acting on a category goal. Diagnostic studies examine underspecified instructions, behavioral failures, adversarial language, and alternative descriptions (Feng et al., 2022; Zhu et al., 2022; Yang et al., 2023; Lin et al., 2021; Zhang et al., 2025), while broader embodied evaluations compare multimodal models across tasks (Cheng et al., 2025). LIBERO-Para studies paraphrase robustness in manipulation by varying action expressions and object references (Kim et al., 2026). IntentionNav uses paired expressions to assess the complete agent's consistency and shared category predictions to compare navigation systems with the inference output fixed.

3 Benchmark Design

3.1 Task and Episode Specification

We define an episode as $e = (s, u, q_0, o^\star, c^\star, p^\star)$, where s is the scene, u the request, q_0 the starting pose, and $o^\star$, $c^\star$, and $p^\star$ the designated target's identifier, category, and position. In the implicit condition, the policy receives the request, RGB-D observations, and pose; the target category and designated target metadata are withheld. The agent must search under partial observability and should issue a stop action near the intended object. The reference system selects and executes waypoints using a scene-derived navigability map, with target hypotheses and visual evidence predicted during the episode.

The request may suggest plausible targets; $o^\star$ identifies the designated object used for evaluation. This target anchors comparisons across expressions and agents. SR and OSR evaluate this instance; the stopping analysis also scores the annotated set of same-category instances (Section 5.5).

3.2 Paired Styles and Semantic Cue Groups

Each base intent has four English realizations: formal, natural, casual, and emotional. All four share the same target, scene, and starting pose. They differ in register, length, affect, and descriptive directness, so paired outcomes measure robustness across these combined language changes. Table 1 shows an unmodified example from the release.

Each released item also has one primary cue-mode label: event-script (202 items), inner-state (167), physical-state (72), or affordance (59). These labels identify the main cue linking a request to its target: an activity or routine, the speaker's state, an environmental condition, or the target's function and properties, respectively. Styles are compared within each item. Cue-mode comparisons describe separate item subsets, each with its own category and scene distribution. Representative cue examples appear in Appendix A. Figure 3 summarizes the cue-mode and room composition of the reference split.

Table 1: Four expressions of the same need (microwave item `SEL_087`). All variants share the target, scene, and starting pose. The emotional expression combines affect with a descriptive clue, illustrating the range of language variation.

Style	Instruction
Formal	I should warm something up to satisfy my late night hunger.
Natural	Gonna heat up a quick snack.
Casual	Need to zap a snack.
Emotional	Help, I'm literally starving! I just need to get this snack into the quick-heat box right now!

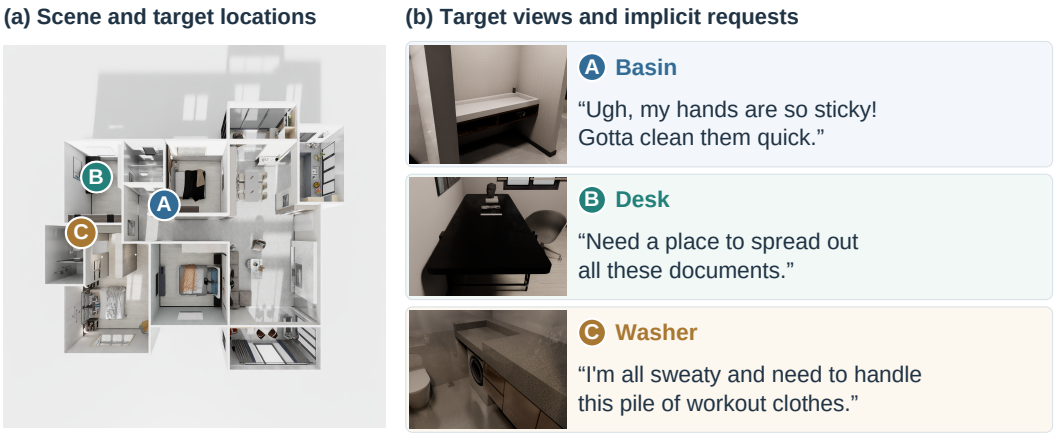


Figure 2: Scene-grounded construction examples. Labels A–C connect target locations and simulator views to illustrative implicit requests. A uses the released basin request `SEL_171`; B and C use this scene's construction records. The views support annotation and quality control.

3.3 Construction and Quality Control

We construct the benchmark by identifying targets in scene graphs, rendering object-centered views of VLNVerse scenes in Isaac Sim (Lin et al., 2025), and generating candidate intent sets with language-model assistance. Figure 2 links three scene targets to their views and illustrative requests. The construction records contain 3,871 surface records and 4,406 view-target observations, producing a pool of 17,624 candidate intent sets. Selection and refinement yield the released 500-item diagnostic split, spanning 176 scenes and 64 annotated categories. Evaluation uses the fixed selected records from this construction pool.

All 500 items undergo automatic lexical filtering, model-assisted checks of target recoverability and style distinguishability, and scene/target consistency checks. The lexical rule excludes the target category name and a fixed set of synonyms. Language refinement requires at least two of the four variants to convey contextual or affordance-based cues; descriptions such as "quick-heat box" are admissible under this rule.

We manually reviewed 250 tasks and all 1,000 expressions for target suitability, expression consistency, implicitness, and ambiguity. Before refinement, 96.4% of tasks had suitable targets, 91.6% had four consistent expressions, and 91.8% of expressions met the implicitness criterion. We made minor edits to 29 tasks and substantially revised or replaced seven. Double review of 100 tasks (400 expressions) yielded 93.0% task-level agreement and Cohen's $\kappa = 0.85$. Appendix F reports the full audit results and cue-mode annotation agreement.

3.4 Benchmark Resources

The anonymous Hugging Face release provides JSONL instructions, 500 target and starting-pose specifications, scene manifests and renders, 6,000 reference episode records, versioned evaluation code, and checksums. The anonymous GitHub mirror hosts collection, construction, and evaluation

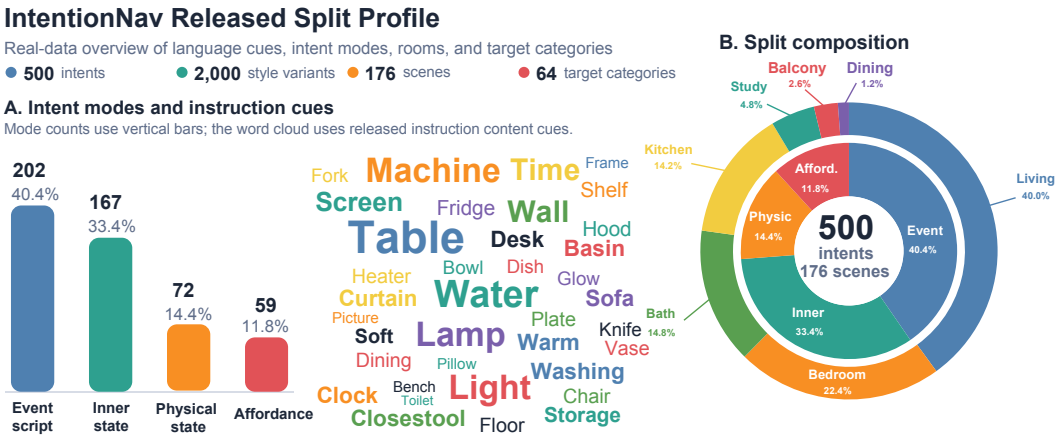


Figure 3: Composition of the released 500-intent reference split. Bars show semantic cue groups alongside illustrative instruction cues. The inner ring shows cue-mode proportions and the outer ring room proportions, each over all 500 intents.

code. Scene IDs, source revisions, and environment metadata identify third-party USD assets obtained under their original licenses.

The hosted-backend study uses this fixed reference release. The shared-category system comparison uses a revised episode manifest, with its own matched conditions and annotated category-goal sets. Section 4.2 specifies the comparison design; Appendix G records the inputs and configuration for reproducing its analysis.

4 Evaluation Framework

The paired-language study tests consistency across requests on fixed physical tasks using the original four-style release. The shared-category study examines how identical category inputs lead to search and completion using a revised task manifest. Each study matches its own scene, target, and starting-pose specifications across conditions.

4.1 Reference Navigation System

The three hosted backends share the following navigation architecture. The reference agent first proposes a target category, alternative candidates, and likely rooms. At each step, it acquires RGB-D observations, applies Grounding DINO (Liu et al., 2024), and queries the VLM for target evidence and search directions. Depth and camera pose project accepted observations into spatial memory for later revisits.

Search ranks destinations using target memory, room preferences, unexplored space, directional cues, and a revisit penalty; A* selects the next waypoint. Nearby evidence prompts a completion decision; otherwise, the agent moves and observes again. At the nominal 30-step horizon, an evidence-based rule selects the terminal location. Appendix B specifies observation, evidence admission, movement, and completion.

GPT-5.4, Qwen3.6-Plus, and Gemini-3.1-Flash-Lite supply initial goal interpretations and visual assessments on the same episodes, with a fixed detector and evaluation configuration.

4.2 Shared-Category System Comparison

We compare two complete navigation systems under a common category interface: the reference system with local perception and category-driven control, and MTU3D (Zhu et al., 2025) with learned object grounding and frontier selection. Qwen3.5-4B first predicts a category from each formal request; both navigation systems receive the same cached prediction. The correct-category condition replaces that prediction with the designated category, giving four matched conditions over 500 tasks

and 2,000 episodes. Each navigation policy uses the category, its acquired RGB-D views, pose, and a shared nonsemantic navigability map. Target coordinates and instance identities are used by the evaluator. Both systems have a 30-action budget, with sensing rotations counted as actions.

The shared-category manifest changes starting poses and a subset of target and language annotations relative to the four-style release. All four conditions share these revised specifications. Appendix G details the inputs, system adaptations, and reuse of the reference correct-category trajectories.

4.3 Evaluation measures

We measure category recovery, search coverage, final position, explicit stopping, visual grounding, and cross-expression consistency.

Intent matching (IM). Let $\hat{c}$ be the predicted category and $c^\star$ the designated category. A fixed synonym vocabulary determines category agreement:

$$\mathrm{IM}(e) = \mathbf{1}[\mathrm{syn}(\hat{c}) = \mathrm{syn}(c^\star)]. \tag{1}$$

IM measures target-category recovery under the benchmark vocabulary.

Oracle success (OSR) and success rate (SR). Let q_t be a recorded agent position, $p^\star$ the designated target center, d_t their horizontal distance, and τ the final recorded step. We measure whether the trajectory enters the target neighborhood and whether its final position lies there:

$$\mathrm{OSR}_r(e) = \mathbf{1}\left[\min_{t\le\tau} d_t \le r\right], \qquad \mathrm{SR}_r(e) = \mathbf{1}[d_\tau \le r]. \tag{2}$$

The main comparison uses $r = 2$ m. SR evaluates the final recorded position. Hosted-backend runs include horizon-based terminal selection; shared-category runs end at an explicit STOP or their final budgeted action. The OSR–SR gap counts trajectories that enter the target neighborhood and finish elsewhere.

Explicit stopping (StopSR). StopSR requires an explicit STOP within the 30-action budget and within 2 m horizontal distance of a target's bounding box, with distance zero inside the box. We score the designated instance and other annotated same-category instances separately. This diagnostic uses target-surface geometry; SR/OSR use target-center distance.

Grounded success (GSR). We further combine final proximity with a geometric visibility check at the terminal pose:

$$\mathrm{GSR}_r(e) = \mathrm{SR}_r(e)\,\mathbf{1}[\tilde{v}_\tau = 1], \tag{3}$$

where $\tilde{v}_\tau$ is computed from target-box projection and geometric occlusion checks. This provides a geometric proxy for terminal visual grounding. We report GSR at 1 m as a stricter secondary measure alongside SR and OSR at 2 m. Both studies use the same geometric scorer at their recorded terminal poses (Appendix G).

Cross-expression consistency (All-four). For episode $e_{i,\ell}$ of base intent i under style ℓ, we define

$$\text{All-four} = \frac{1}{N}\sum_{i=1}^{N}\prod_{\ell\in\mathcal{L}}\mathrm{SR}_2(e_{i,\ell}), \tag{4}$$

where $N = 500$ and $\mathcal{L}$ contains the four styles with shared scene, target, and starting pose. All-four is the fraction of intents for which all four executions succeed. One to three successes indicate mixed outcomes.

4.4 Controls and statistical analysis

The explicit-target control supplies the ground-truth category for the same 500 Gemini/formal items while retaining the downstream reference configuration. The paired comparison measures the effect of supplying the target category within that system. We also score each unchanged trajectory with

a first-entry oracle that selects its first position inside the ground-truth neighborhood. This offline oracle uses ground-truth geometry and gives an upper bound on endpoint success along the recorded trajectory.

For the hosted-backend study, we compute confidence intervals from 50,000 bootstrap resamples with seed 3417, keeping all styles and backends for each item together. The shared-category study uses 10,000 resamples with seed 20260919 and pairs the four conditions by task. We also resample by scene and physical target to assess sensitivity to clustering and compare the systems' correct-minus-predicted category effects. The full comparison has one run per condition; two runs per condition on a 40-task development subset assess execution variability. Intervals are unadjusted for multiple comparisons; style and mode comparisons are exploratory. Appendix H reports the additional uncertainty and stopping analyses.

5 Experiments and Findings

5.1 Navigation performance

The two studies in Table 2 examine consistency across expressions and execution under shared category predictions. Across the 6,000 hosted-backend episodes, the agent reaches the target neighborhood in 68.7% of cases and finishes there in 24.93%; GSR@1m is 5.50%.

Table 2: Navigation performance (%, higher is better). SR/OSR use a 2 m target-center radius; GSR uses 1 m. Hosted backends cover four styles (2,000 episodes each); category-input conditions cover 500 matched tasks each. Correct-category rows receive the annotated label and bypass category prediction, so IM is inapplicable (—). Task manifests are specified in Section 4.2.

System / backend	SR ↑	OSR ↑	GSR ↑	IM ↑
Reference / GPT-5.4	24.95	69.1	5.10	50.4
Reference / Qwen3.6-Plus	24.15	68.1	4.80	44.1
Reference / Gemini-3.1-Flash-Lite	25.70	69.0	6.60	50.4
Hosted overall	24.93	68.7	5.50	48.3
Reference / local (predicted category)	18.4	60.8	2.0	49.4
Reference / local (correct category)	24.2	61.2	1.6	—
MTU3D (predicted category)	15.0	34.6	0.8	49.4
MTU3D (correct category)	12.8	30.2	0.8	—

Hosted-backend SR spans 24.15–25.70%, while Gemini and GPT each exceed Qwen by 6.35 percentage points in IM. This narrow SR range motivates comparing which tasks succeed across expressions. Appendix C reports the paired statistical comparisons.

5.2 Consistency across expressions

Task-level outcomes reveal whether success extends across expressions of the same need. Table 3 reports per-style SR alongside the All-four consistency measure defined in Section 4.3.

Table 3: Per-style endpoint success and cross-style consistency in the reference analysis. Each style column contains 500 episodes per model. All-four is the fraction of base intents with endpoint success under every style. Values are percentages.

Backend	Formal	Natural	Casual	Emotional	All-four
GPT-5.4	26.8	24.6	24.4	24.0	5.6
Qwen3.6-Plus	25.4	23.6	22.2	25.4	7.0
Gemini-3.1-Flash-Lite	28.0	24.6	25.0	25.2	9.6

Across the four expressions of each task, 45.4%, 40.0%, and 36.6% of intents have both successful and unsuccessful episodes for GPT, Qwen, and Gemini, respectively. Gemini and GPT have similar average SR (25.70% and 24.95%). Gemini's All-four score is 4.0 percentage points higher, revealing

a difference in task-level expression consistency. GPT and Qwen also reverse order between average SR and All-four. Appendix C reports the paired tests, complete success-count distributions, and cue-mode results.

To compare expression variation with repeated execution, we run each expression three times on 40 uniformly sampled tasks with Gemini, requesting the initial plan anew for every episode. Across 480 executions, outcome disagreement is 9.17% within an expression and 22.78% across expressions (Appendix D). Disagreement across expressions is 13.61 percentage points higher than between repetitions of the same expression.

5.3 From shared goals to search and completion

The second comparison holds category predictions fixed and examines how each navigator searches and completes the task. Both conditions have IM of 49.4%, corresponding to the same 247 correct predictions among 500 tasks. The reference system reaches the designated neighborhood in 60.8% of episodes, compared with 34.6% for MTU3D. The 26.2-point coverage difference narrows to 3.4 points in final SR (18.4% versus 15.0%; Table 15), motivating the completion analysis.

These 247 tasks form a common subset for examining execution after successful category recovery. The reference system completes 57 of these tasks (23.1% SR), while MTU3D completes 44 (17.8%). Among the remaining 253 tasks, their success counts are 35 and 31, respectively.

5.4 Correct-category controls

Correct-category controls test whether better goal specification translates into successful completion. The reference system's SR increases from 18.4% to 24.2%, a gain of 5.8 percentage points, while OSR changes by only 0.4 points. The gain is therefore concentrated in the final location. MTU3D changes from 15.0% to 12.8%, with no detectable improvement. The difference in input effects is 8.0 points, and its direction is preserved when resampling by scene or physical target (Appendix H). The benefit of more accurate goal specification depends on how the navigator uses that information.

The hosted-backend control in Table 4 compares implicit and explicit target specification on the same 500 Gemini/formal items. Supplying the category increases SR from 28.0% to 44.8%, a gain of 16.8 percentage points. OSR and GSR also increase.

Table 4: Effect of supplying the correct category on 500 Gemini/formal tasks. Rates are percentages; differences are percentage points.

Metric	Implicit	Explicit	Difference
OSR@2m	70.6	77.6	+7.0
SR@2m	28.0	44.8	+16.8
GSR@1m	6.6	12.4	+5.8

A first-entry oracle scores the unchanged implicit Gemini/formal trajectories at their first visit to the goal neighborhood. This gives 70.6% success, compared with the observed 28.0%, a gap of 42.6 points. The oracle equals OSR by construction: the gap counts episodes that visit the target neighborhood and subsequently finish outside it.

5.5 Stopping behavior and target evidence

The coverage–completion gap motivates examining where agents stop. Table 5 distinguishes reaching a target region without STOP from stopping outside annotated category regions. We use the target-surface StopSR criterion defined in Section 4.3.

With predicted categories, reaching the designated region without STOP dominates reference outcomes (239/500 episodes versus 95/500 for MTU3D). Stopping outside all annotated same-category regions dominates MTU3D outcomes (173/500 versus 69/500 for the reference system). A center-distance variant of StopSR gives 11.0% designated-instance success for both systems; the target-surface measure gives 16.8% for MTU3D and 12.4% for the reference system (Table 16).

Table 5: Completion outcomes under a 2 m horizontal target-surface threshold. Success requires an explicit STOP within 30 actions. Counts are out of 500 per condition; Pred./Corr. denote predicted/correct category inputs. Rows are mutually exclusive, with designated-region matches taking precedence. Each column sums to 500.

Outcome	Ref. Pred.	Ref. Corr.	MTU Pred.	MTU Corr.
Designated-instance success	62	110	84	69
Other same-category success	24	52	30	39
STOP outside category regions	69	56	173	152
Reached designated region; no STOP	239	188	95	97
Reached other category region; no STOP	23	10	20	16
Never reached category region; no STOP	83	84	98	127

Among the 247 tasks with correct predictions, 112 reference episodes reach the designated region without STOP, compared with 49 for MTU3D under the same surface-distance criterion. MTU3D stops outside all annotated category regions in 64 episodes, versus 17 for the reference system. The systems thus exhibit different stopping errors even after correct category recovery.

Supplying the correct category increases the reference system's designated-instance successes to 110, while 188 episodes still reach that region without STOP. Another 52 episodes stop near a different annotated same-category instance. Scoring both instance scopes identifies successful category-level search that ends at a different instance from the benchmark's designated target.

A complementary analysis of the hosted-backend explicit-target runs relates completion to the location of retained target evidence. Of 500 episodes, 439 contain a target-memory cluster derived from DINO detections that passes the agent's label, room, and geometric filters. In 259 episodes, at least one such cluster lies within 2 m of the designated target center. Endpoint success is 206/259 (79.5%) with such nearby evidence versus 3/180 (1.7%) when all admitted clusters are farther away. Among the 276 endpoint failures, 46 have no admitted evidence, 177 have only farther evidence, and 53 have nearby evidence but still finish outside the goal neighborhood.

Distance and grounding criteria further distinguish neighborhood completion from a close final viewpoint. Endpoint success increases from 12.0% at 1 m to 24.93% at 2 m and 34.6% at 3 m in the reference evaluation. In the shared-category comparison, reference GSR is 2.0% with predicted categories and 1.6% with correct categories; MTU3D scores 0.8% under both inputs. The low GSR values show that neighborhood completion rarely extends to the close-range grounding criterion.

6 Discussion and Conclusions

Implications for evaluation. The paired design connects evaluation to two model-development questions: whether a language frontend yields consistent task outcomes, and whether a navigator turns a fixed goal into completion. Per-expression averages, task-level consistency, and repeated executions inform the first. Shared and correct-category inputs, trajectory coverage, endpoint success, and explicit stopping inform the second.

Limitations. Experiments cover English requests in static simulated apartments, three hosted VLM backends, and two category-driven systems. Full comparisons use one run per condition; repeats cover 40 Gemini tasks and 40 shared-category development tasks. Human review covers 250 tasks and 1,000 expressions. Each task specifies a designated target; stopping diagnostics also score annotated same-category instances. Language variants jointly change register, length, affect, and directness. Attributing spatial outcomes to specific perceptual causes requires independent target-visibility labels and component interventions.

Conclusions. IntentionNav evaluates how consistently agents act on implicit requests and how navigators execute inferred goals. Paired expressions show that similar average success can conceal mixed outcomes on the same tasks. Shared-category comparisons reveal distinct search and stopping behavior under identical inferred goals. Its paired evaluations characterize expression consistency alongside the search and completion of an inferred goal.

AI USE STATEMENT

Language and vision-language models were used to generate, refine, and assess benchmark instructions and as components of the evaluated agents. AI tools also assisted with literature retrieval, feedback on methodology and experimental design, code development, analysis and interpretation of results, figure preparation, and manuscript drafting and editing. The authors take responsibility for the final manuscript and research artifacts.

REPRODUCIBILITY STATEMENT

The fixed annotations and episode tuples, 6,000 reference logs, evaluation-code snapshots, configuration files, checksums, and environment/scene manifests are distributed through the anonymous Hugging Face release. Collection, construction, and reference evaluation code is linked through the anonymous GitHub mirror. The shared-category release contains task inputs, cached predictions, 2,000 full-set and 320 development-repeat trajectories, frozen system code, and a standalone metric-recomputation script. Sections 4.3 and 5 define the metrics and report the analyses; Appendix E describes the release structure and data versions. Appendix G specifies the task inputs, fixed model configurations, reused trajectories, and uncertainty procedure for the shared-category comparison. Table 5 reports completion outcomes, with further diagnostics in Appendix H. Large third-party scene assets must be acquired from their original providers. The fixed records support metric recomputation, while hosted-model reruns remain subject to service and model updates.

ETHICS STATEMENT

The benchmark uses simulated indoor scenes and model-generated English requests for diagnostic research. Intent inference can encode cultural assumptions or select an inappropriate object when a request is ambiguous. Real-world assistance requires validation with users and safeguards appropriate to the deployment context. Downstream use should respect scene-asset and model licenses, preserve privacy, and include appropriate human oversight.

REFERENCES

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

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018b.

Luca Barsellotti, Roberto Bigazzi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Personalized instance-based navigation toward user-specific objects in realistic environments. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024.

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020.

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. In *International Conference on 3D Vision*, 2017.

Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Advances in Neural Information Processing Systems*, 2020.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K. Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2024.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Zhili Cheng, Yuge Tu, Ran Li, Shiqi Dai, Jinyi Hu, Shengding Hu, Jiahao Li, Yang Shi, Tianyu Yu, Weize Chen, Lei Shi, and Maosong Sun. Embodiedeval: Evaluate multimodal llms as embodied agents. *arXiv preprint arXiv:2501.11858*, 2025.

Weixi Feng, Tsu-Jui Fu, Yujie Lu, and William Yang Wang. Uln: Towards underspecified vision-and-language navigation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

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*, 2023.

Yuehao Huang, Liang Liu, Shuangming Lei, Yukai Ma, Hao Su, Jianbiao Mei, Pengxiang Zhao, Yaqing Gu, Yong Liu, and Jiajun Lv. Cogddn: A cognitive demand-driven navigation with decision optimization and dual-process thinking. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 5237–5246, 2025.

Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. Goat-bench: A benchmark for multi-modal lifelong navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Chanyoung Kim, Minwoo Kim, Minseok Kang, Hyunwoo Kim, and Dahuin Jung. LIBERO-Para: A diagnostic benchmark and metrics for paraphrase robustness in VLA models. *arXiv preprint arXiv:2603.28301*, 2026. URL https://arxiv.org/abs/2603.28301.

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

Bingqian Lin, Yi Zhu, Yanxin Long, Xiaodan Liang, Qixiang Ye, and Liang Lin. Adversarial reinforced instruction attacker for robust vision-language navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Sihao Lin, Zerui Li, Xunyi Zhao, Gengze Zhou, Liuyi Wang, Rong Wei, Rui Tang, Juncheng Li, Hanqing Wang, Jiangmiao Pang, et al. Vlnverse: A benchmark for vision-language navigation with versatile, embodied, realistic simulation and evaluation. *arXiv preprint arXiv:2512.19021*, 2025.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision*, pp. 38–55, 2024.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2049–2060. PMLR, 2025.

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

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M. Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. In *Advances in Neural Information Processing Systems*, 2021.

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

Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

Xia Su, Ruiqi Chen, Benlin Liu, Jingwei Ma, Zonglin Di, Ranjay Krishna, and Jon Froehlich. Capnav: Benchmarking vision language models on capability-conditioned indoor navigation. *arXiv preprint arXiv:2602.18424*, 2026.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems*, 2021.

Hongcheng Wang, Andy Guan Hong Chen, Xiaoqi Li, Mingdong Wu, and Hao Dong. Find what you want: Learning demand-conditioned object attribute space for demand-driven navigation. In *Advances in Neural Information Processing Systems*, 2023a.

Hongcheng Wang, Peiqi Liu, Wenzhe Cai, Mingdong Wu, Zhengyu Qian, and Hao Dong. Mo-ddn: A coarse-to-fine attribute-based exploration agent for multi-object demand-driven navigation. In *Advances in Neural Information Processing Systems*, 2024a.

Hongcheng Wang, Jinyu Zhu, and Hao Dong. User-centric object navigation: A benchmark with integrated user habits for personalized embodied object search. *arXiv preprint arXiv:2602.06459*, 2026.

Wenxuan Wang, Yisi Zhang, Xingjian He, Yichen Yan, Zijia Zhao, Xinlong Wang, and Jing Liu. Beyond literal descriptions: Understanding and locating open-world objects aligned with human intentions. In *Findings of the Association for Computational Linguistics*, 2024b.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023b.

Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Zijiao Yang, Arjun Majumdar, and Stefan Lee. Behavioral analysis of vision-and-language navigation agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. In *Advances in Neural Information Processing Systems*, 2024.

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

Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. Hm3d-ovon: A dataset and benchmark for open-vocabulary object goal navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024b.

Yue Zhang, Tianyi Ma, Zun Wang, Yanyuan Qiao, and Parisa Kordjamshidi. Vision-and-language navigation with analogical textual descriptions in llms. In *Proceedings of EMNLP*, 2025.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, 2024a.

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

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 *Proceedings of the International Conference on Machine Learning*, 2023.

Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Wanrong Zhu, Yuankai Qi, Pradyumna Narayana, Kazoo Sone, Sugato Basu, Eric Xin Wang, Qi Wu, Miguel Eckstein, and William Yang Wang. Diagnosing vision-and-language navigation: What really matters. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, 2022.

Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, Siyuan Huang, and Qing Li. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8120–8132, October 2025.

Filippo Ziliotto, Jelin Raphael Akkara, Alessandro Daniele, Lamberto Ballan, Luciano Serafini, and Tommaso Campari. Personal: Towards a comprehensive benchmark for personalized embodied agents. *arXiv preprint arXiv:2509.19843*, 2025.

A Language Examples and Benchmark Composition

Table 6 provides representative released instructions for the four mode labels. Each label identifies the main cue linking a request to its annotated target. When an utterance contains several cues, the primary label records the one judged most informative. Style comparisons hold the scene, target, and starting pose fixed within an item; cue-mode groups contain different task subsets.

Table 6: Representative cue-mode examples from the released records. Targets identify the evaluator's designated objects; the implicit policy receives each request as its language input.

Mode	Target	Example instruction
Event-script	Plate	Friends are coming tonight, so I should set little snacks on a serving piece for the table.
Inner-state	Kettle	I'm freezing; let's boil some water for something warm.
Physical-state	Curtain	It's too bright in here, so I'm gonna block the light.
Affordance	Dining table	I need a big spot to lay everything out.

The reference split has five broad target groups: appliances (133), small objects (115), large furniture (113), lighting/decor (81), and bathroom objects (58). Together, these groups contain the 500 base items in the diagnostic sample. Living rooms, bedrooms, bathrooms, and kitchens account for 200, 112, 74, and 71 items, respectively; the remaining items occur in other room types. The sample spans household needs associated with these objects and room types.

B Reference Navigation Procedure

The reference agent repeatedly observes the scene, accumulates target evidence, and selects where to search. Table 7 summarizes this procedure. A scene-derived walkability grid supplies navigation geometry, and room-type regions support room preferences. Language and visual observations provide the predicted target category, candidate objects, and search directions.

Table 7: Reference navigation loop. Visual evidence persists across viewpoints and guides subsequent search decisions.

Stage	Operation
Interpret	Infer the likely target, candidate categories, and promising room types from the request.
Observe	Acquire RGB-D; detect target-related objects and obtain VLM visibility and search-direction assessments.
Remember	Project accepted evidence into the scene and merge nearby observations into persistent target clusters.
Search	Combine target evidence, room preferences, unexplored space, and visit history in a value map; plan the next waypoint.
Complete	Use proximity and evidence confidence to select a final approach, or continue searching.

Initialization and observations. The language planner produces one main target hypothesis with candidate objects and likely rooms. In the correct-category condition, the supplied category replaces this inference while the downstream navigation configuration is retained. The simulator initializes the recorded episode pose and supplies forward RGB-D images at 768×768 resolution with a 90° horizontal field of view. Initial and conditional panoramic scans gather additional detector evidence when the current target memory is sparse. The 30-step horizon counts iterations of the navigation decision loop; the scan schedule and final approach are part of the reference agent's execution configuration.

Evidence accumulation. Grounding DINO is queried with the predicted target and related candidate phrases, using a detection-score threshold of 0.3. The agent combines accepted detections with depth and camera pose to estimate spatial target locations; VLM assessments guide exploration and stopping. Category matching, room compatibility, and geometric admission rules filter these proposals before nearby observations are merged into memory clusters. Each cluster retains a location, confidence, and observation history. Stored observations guide subsequent approaches to previously seen targets.

Exploration and movement. The value map rewards locations near target evidence, likely target rooms, and the boundary of explored space. The VLM provides directional preferences, and a penalty for revisiting locations discourages repeated coverage. The planner selects a high-value destination on the navigability grid and uses A* to choose an intermediate waypoint toward it. The simulator's waypoint interface updates the agent pose to the selected location, after which the observation cycle repeats. Spatial and room preferences guide exploration when target evidence is sparse; accumulated target evidence guides the approach.

Completion. The main stopping branch uses an admitted target detection or memory cluster with confidence at least 0.5 within 1.5 m of the agent. An additional branch uses recently reinforced memory with confidence at least 0.65 within 0.8 m, and a visual-confirmation branch combines geometric proximity with consecutive positive VLM assessments and detector support. The agent can orient toward the selected evidence and make a final approach of up to 1 m before recording its terminal pose. If the decision horizon is reached, it selects a previously visited position near a confident target cluster, or the highest-valued visited position, as its final search location. SR evaluates this recorded terminal location using the designated target center.

The released run snapshots, identified by prefixes `68c227e0` and `c3c9ee08`, supply the prompts, full parameter settings, and observation schedules for the reference collection. Each record stores the corresponding configuration and action history. To evaluate explicit stopping, we also define

$$\mathrm{SR}_r^{\mathrm{stop}}(e) = \mathbf{1}[a_\tau = \mathrm{STOP},\ \tau \le B,\ d_\tau \le r], \tag{5}$$

where the trajectory ends at its first STOP and B is the declared action budget. This center-distance score complements the surface-distance stopping diagnostic in Equation 6, used in Table 5. Table 2 evaluates the final location as defined in Equation 2.

C Additional Statistical Results

Table 8: SR estimates and 95% bootstrap intervals for Table 2 (%). Hosted estimates resample base intents with their associated styles; shared-category estimates resample matched tasks.

System / backend	SR	95% CI
Reference / GPT-5.4	24.95	[22.30, 27.70]
Reference / Qwen3.6-Plus	24.15	[21.40, 26.95]
Reference / Gemini-3.1-Flash-Lite	25.70	[22.75, 28.70]
Hosted overall	24.93	[22.47, 27.45]
Reference / local (predicted category)	18.4	[15.2, 21.8]
Reference / local (correct category)	24.2	[20.6, 28.0]
MTU3D (predicted category)	15.0	[11.8, 18.2]
MTU3D (correct category)	12.8	[10.0, 15.8]

Table 9: Paired backend differences in percentage points. SR intervals span zero, while IM separates category recovery on the same benchmark items.

Pair	SR difference	95% CI	IM difference	95% CI
Gemini − GPT	+0.75	[−1.45, 3.00]	0.00	[−2.30, 2.30]
Gemini − Qwen	+1.55	[−0.75, 3.90]	+6.35	[4.10, 8.55]
GPT − Qwen	+0.80	[−1.45, 3.05]	+6.35	[4.30, 8.45]

The item-cluster IM intervals are [46.95, 53.85] for Gemini (50.40%), [47.15, 53.70] for GPT (50.40%), and [40.75, 47.35] for Qwen (44.05%). GSR@1m is 6.60% [4.95, 8.35], 5.10% [3.75, 6.55], and 4.80% [3.55, 6.15], respectively. Hosted overall GSR@1m is 5.50% [4.33, 6.77]. Hosted-backend intervals use the 50,000-resample procedure described in Section 4; shared-category estimates use 10,000 resamples.

For the Gemini/formal correct-category control in Table 4, the paired changes and 95% intervals are +7.0 points [2.6, 11.6] for OSR, +16.8 [11.8, 21.8] for SR, and +5.8 [2.8, 9.0] for GSR. Scoring

the implicit trajectories at first entry gives a 42.6-point difference [38.2, 47.0] from their observed endpoint SR. For the paired explicit-target diagnostic, exact McNemar tests give $p = 2.93 \times 10^{-10}$ for SR, $p = 0.00305$ for OSR, and $p = 0.000422$ for GSR. These tests compare outcomes on the same items with and without the supplied category.

Grouping intents by zero through four successful expressions gives counts $(245, 114, 66, 47, 28)$ for GPT, $(265, 98, 61, 41, 35)$ for Qwen, and $(269, 82, 63, 38, 48)$ for Gemini. Each distribution sums to 500 and uses the same endpoint SR@2m labels as Table 3. The middle three groups yield the mixed-success rates in Section 5.2; the final group determines All-four.

An exploratory paired comparison of All-four gives a Gemini–GPT difference of 4.0 percentage points: 29 tasks succeed under all four expressions only for Gemini, and nine only for GPT. The task-bootstrap 95% interval is [1.60, 6.40] points (50,000 resamples, seed 3417); resampling the 176 scenes gives [1.64, 6.39] (10,000 resamples, seed 20260924). The exact McNemar test gives $p = 0.00166$, or $p = 0.00497$ after Bonferroni correction across all three backend pairs. The remaining All-four contrasts are Gemini–Qwen, +2.6 points [0.2, 5.0] ($p = 0.0533$), and GPT–Qwen, -1.4 points $[-3.4, 0.6]$ ($p = 0.248$), with unadjusted exact p-values. These comparisons characterize observed consistency from one execution per expression.

Table 10 reports aggregate style results and outcomes by semantic cue group. The paired formal–casual SR difference is 2.87 points [0.13, 5.53], an exploratory result. Style rows pair the same physical tasks, while cue-mode rows describe distinct subsets.

Table 10: Language variation and semantic cue groups. Style comparisons pair the same 500 tasks; mode rows summarize different task subsets. Mode estimates include 95% item-bootstrap intervals.

Style	SR (%)	Cue mode	Items	SR (%)	95% CI
Formal	26.73	Event-script	202	28.71	[24.63, 32.92]
Natural	24.27	Inner-state	167	25.10	[20.81, 29.54]
Casual	23.87	Physical-state	72	19.21	[14.00, 25.00]
Emotional	24.87	Affordance	59	18.50	[12.85, 24.86]

Event-script items have higher endpoint success than affordance items, with a descriptive difference of 10.21 points [2.69, 17.32]. This association may reflect both the semantic cue and the objects/scenes represented by each group. The 59-item affordance subset has a wider interval, reflecting its smaller sample. Comparisons that match category and scene composition would help separate their contributions from the semantic cue.

The retained-evidence analysis in Section 5 groups episodes by the spatial proximity of admitted memory clusters. For the 500 explicit-target episodes, 61 contain no admitted cluster, 180 contain only clusters farther than 2 m from the designated target center, and 259 contain at least one nearby cluster. Their endpoint failures are 46, 177, and 53, respectively, totaling 276. A complementary trajectory analysis finds that 164 of these failed episodes entered the target neighborhood before finishing outside it; these episodes overlap the three evidence groups.

D Repeated Hosted Executions

Protocol. We uniformly sample 40 of the original 500 tasks with seed 20260924, spanning 36 scenes, and execute each of their four expressions three times with Gemini-3.1-Flash-Lite (480 episodes). Selection uses task identifiers without consulting success labels. All runs use the original language, target, and starting-pose inputs and the frozen reference policy. The 12 expression–repeat combinations are shuffled within each task, with local RNG seeds 202609241–202609243 assigned to the three repetitions. Initial-plan caches are cleared before every episode; all 480 records include a fresh planner-call receipt. The parsed plan content agrees across the three repetitions for each of the 160 task–expression pairs. Each trajectory resets the agent pose, map exploration state, and target memory. The hosted service uses temperature zero; Python, NumPy, and PyTorch RNGs are seeded before each episode. Runs were collected on September 25, 2026, from 02:54 to 14:23 UTC.

Paired outcome disagreement. We apply the same final-position SR@2m criterion as Table 3, including the reference policy's horizon-based terminal selection. For each task, within-expression

disagreement averages the binary outcome differences over three repeat pairs for each of four expressions (12 pairs). Across-expression disagreement averages the six expression pairs over all nine repeat combinations (54 pairs). We average these quantities over tasks and compute their difference within each task. Uncertainty uses 50,000 task-bootstrap resamples with seed 20260924, preserving all associated expressions and repetitions; 10,000 scene-cluster resamples assess sensitivity to scene sharing.

Table 11: Outcome disagreement over 40 tasks with three fresh executions per expression. Differences are percentage points; intervals resample complete tasks.

Comparison	Disagreement (%)	95% CI
Same expression, different repetitions	9.17	[5.42, 12.92]
Different expressions, all repeat combinations	22.78	[15.23, 30.60]
Across-expression minus within-expression	+13.61	[7.64, 20.37]

Across-expression disagreement exceeds within-expression disagreement by 13.61 points; the scene-cluster interval is [7.67, 20.32]. SR across the three repetitions is 25.0%, 25.0%, and 23.75%; All-four is 7.5%, 5.0%, and 10.0%. These repetitions provide an execution-variability reference for the paired language comparison on this backend and task subset. The disagreement difference is descriptive and does not estimate a causal fraction of failures attributable to language.

Execution artifacts. The release retains all 480 trajectories, planning-call receipts, step images/responses, renderer and API-retry logs, and source/input checksums. All episodes are retained, including those with near-uniform views; renderer asset warnings are recorded in the logs. Cached DINO weights and available scene/map files are identified by content hashes. The code clears both planner caches at episode boundaries and verifies the generated plan against the saved record.

E Reproducing the Evaluations

The anonymous repositories linked in the abstract provide the code, dataset, and reference artifacts. The dataset repository stores evaluation-code archives under `reference_results/evaluation_code/`, alongside the released records.

Fixed inputs. `selected_500_intents.jsonl` contains the 500 selected records and their four English variants. `episodes.jsonl` specifies target identities/positions and starting poses, joined by `selection_id`. Per-scene directories contain intent records, manifests, and inspection renders. Construction views and target annotations support dataset curation and scoring; navigation observations are generated at the agent's own viewpoints.

Logs, code, and provenance. `reference_results/logs/` contains the reference logs and record index. `reference_results/episode_artifacts/` contains retained images/responses and their indexes. `reference_results/evaluation_code/` records the execution revisions, while `reference_results/reproducibility/` supplies environment, simulator, scene-ID, and upstream-source metadata. The release's manifest and SHA-256 list identify exact files. Large scene assets are obtained from their original providers using the pinned source information.

The shared-category study is distributed separately under `system_comparison/20260924/` in the same dataset repository. Its archive includes `reviewed-v3` annotations and starting poses, the 500 cached predictions, both systems' exact category inputs and frozen code, all 2,000 full-set trajectories, and 320 development-repeat trajectories. It identifies the 500 reused reference correct-category records and retains the original annotation and source provenance. After extraction, `python recompute.py .` verifies the checksums and reproduces the center-distance, STOP/surface, and full-set GSR labels from the distributed records and geometry, without a simulator or model API.

The hosted repeat study is released under `hosted_repeats/20260925/`. It includes the original task inputs, frozen policy and repeat wrapper, all 480 episode records with their retained images/responses, and initial-planner call receipts. Its `recompute.py` verifies the distributed

checksums and reconstructs the disagreement rates, task/scene intervals, and per-repeat SR/All-four directly from the records.

Reanalysis procedure. First verify the selected-record and episode IDs and the release checksums. Next reconstruct the 500-item by four-style by three-backend reference index, checking missing or duplicated records before aggregation. Run the corresponding snapshot's metric computation to reproduce the navigation tables. For uncertainty, resample base intent IDs with replacement, preserve all associated style/model rows, and recompute each aggregate or paired difference for 50,000 replicates with seed 3417. Report the 2.5th and 97.5th percentiles. The evaluation settings specify the action handling, target geometry, starting conditions, and geometric visibility test associated with each result.

Evaluation records. The hosted-backend comparisons use the fixed inputs in the reference release. Episode identifiers, source revisions, and checksums specify the task and agent used for each record. Paired analyses match physical episodes and scoring definitions across the compared conditions. Dataset updates preserve the preceding release and document changes to target annotations, language variants, or starting conditions in their accompanying records. Hosted-model reruns may vary with service updates; released trajectories provide a fixed basis for recomputing the reported metrics.

F Annotation and Matching Details

The synonym-aware matcher normalizes category predictions and checks a finite list of aliases. Examples include `basin`: sink/washbasin; `fridge`: refrigerator; `closestool`: toilet/wc; `couch`: sofa; and `table_lamp`: desk_lamp/lamp. IM measures category agreement under this vocabulary, including its broad aliases; the main navigation scores evaluate localization of the designated physical target.

Validation coverage. Table 12 lists the checks applied to the full dataset and the additional human-reviewed subsets. The human review assessed each task's four expressions and scene context for target suitability, expression consistency, implicitness, and ambiguity.

Table 12: Validation coverage. Human review covers 250 tasks and their 1,000 expressions. Task-quality judgments and semantic cue labels have distinct agreement statistics.

Coverage	Checks
All 500 base items	Automatic category-name/synonym screening, model-assisted recoverability and style assessment, and scene/target consistency checks.
250 tasks / 1,000 expressions	Manual review of target suitability, expression consistency, implicitness, and ambiguity in the task's scene context.
Remaining 250 items	Whole-set automatic and model-assisted checks.
100 tasks / 400 expressions	Double review of task quality; 93 of 100 task-level judgments agree (93.0%, Cohen's $\kappa = 0.85$).
100 mode labels	Double annotation of the primary semantic cue; agreement 91%, Cohen's $\kappa = 0.86$.

Human review outcomes. Table 13 reports the assessments before refinement and the resulting task dispositions. Reviewers judged the target suitable in 241 of 250 tasks and found all four expressions consistent in 229 tasks. Reviewers flagged nine tasks for target suitability and 21 for cross-expression consistency. Of the 1,000 expressions, 918 met the implicitness criterion, while 82 were judged overly direct or strongly suggestive of the target. Ambiguity was assessed over 250 task–scene contexts: 176 were approximately unique, 57 weakly ambiguous, and 17 clearly ambiguous. These counts refer to tasks, which can share a scene. We retained 214 tasks without changes, made minor edits to 29, and substantially revised or replaced seven.

Reviewer agreement. Two reviewers assessed 100 tasks containing 400 expressions, with matching task-level review judgments on 93 tasks (93.0% agreement, Cohen's $\kappa = 0.85$). For semantic cue-mode labels, double annotation of 100 items yields 91% agreement and $\kappa = 0.86$.

Table 13: Human review of 250 tasks and their 1,000 expressions. Quality assessments describe the inputs before refinement. Ambiguity categories partition the 250 tasks; task dispositions form a separate partition. Denominators distinguish task-level and expression-level assessments.

Review outcome	Count	Rate (%)
Suitable designated target	241 / 250	96.4
All four expressions consistent	229 / 250	91.6
Expression meets implicitness criterion	918 / 1,000	91.8
Approximately unique interpretation	176 / 250	70.4
Weak ambiguity	57 / 250	22.8
Clear ambiguity	17 / 250	6.8
Retained without changes	214 / 250	85.6
Minor edits	29 / 250	11.6
Substantial revision or replacement	7 / 250	2.8

Descriptive cues. The lexical filter screens explicit target labels and listed synonyms. Language refinement requires at least two variants per item to convey contextual or affordance-based cues and allows descriptive expressions among the four variants. For example, "quick-heat box" provides a functional description of a microwave while meeting the category-name rule.

Construction and evaluation roles. The construction judge scores target recoverability and style distinguishability from a target-centered photograph, the annotated category, and the four variants. Candidates below the acceptance threshold are revised or replaced. The navigation agent acquires its own observations while searching from the episode start. Final evaluation uses the fixed category matcher and geometric scoring rules in Section 4.3.

G Shared-Category Systems and Experimental Settings

Inputs and pairing. The system comparison uses the same 500 task identifiers across 176 scenes and 461 designated physical objects. Qwen3.5-4B predicts a category once from each formal request using greedy decoding; the prediction cache contains 500 parsed outputs with 247 category matches. Both systems receive the same predicted category, or the designated category in the matched control. During navigation, each system acquires its own observations under that goal.

The hosted-backend study uses the original 64-category reference release, while the system comparison uses the revised 65-category manifest `reviewed-v3` dated September 18, 2026. Relative to the original release, it changes all 500 starting poses, 119 formal requests, 70 category strings, and six designated object identifiers; seven target-center positions change by more than 1 cm. The median start-position displacement is 3.49 m. All four conditions use the same revised inputs for paired comparisons. The earlier hosted-backend analyses retain their original task inputs and matching. The revised annotations include AI-assisted scene/target checks; their semantic review is nonexhaustive, and `SEL_077` and `SEL_314` retain ambiguity annotations.

Category-driven reference system. The reference system uses Grounding DINO detections, depth-projected spatial memory, exploration values, and map-based waypoint planning. The category input supplies its target query; local perception and spatial evidence guide subsequent search and completion. This configuration uses collision-checked movement segments and budgeted actions, with no online language-model calls after category inference. The hosted-backend configuration described in Appendix B uses VLM assessments during navigation.

MTU3D configuration. MTU3D retains the released stage-one pretrained and stage-two OVON-finetuned checkpoints (Zhu et al., 2025). Its learned policy grounds objects and selects exploration frontiers from accumulated observations. An initial scan acquires four views, with rotations charged to the action budget; subsequent decisions reuse views obtained during movement. The Isaac Sim interface supplies calibrated RGB-D projections, persistent frontier execution, and collision-checked movement on the same permitted nonsemantic map. Object approaches use learned evidence and

acquired camera geometry; STOP requires a current learned mask and the corresponding proximity check. The adaptation runs the pretrained perception, memory, and decision model through the common simulator action interface.

Table 14: Shared-category experiment configuration. The two full-scale predicted-category conditions consume the same frozen prediction cache.

Setting	Value
Task coverage	500 matched tasks per condition; formal requests for category prediction
Observation	768×768 RGB-D, 90° field of view, known camera pose
Simulator	Isaac Sim 4.5; FXAA; eight render ticks per observation
Navigation budget	30 actions, including rotations and STOP
Category predictor	Qwen3.5-4B; greedy text decoding; no per-step predictor calls
Navigation weights	Fixed reference detector; MTU3D stage-one pretraining and OVON stage-two weights
Full comparison	One run per condition; 2,000 trajectories
Repetitions	Two runs on the same 40 development tasks per condition; 320 trajectories
Randomness	Full run and development run A: 20260909; development run B: 20260910
Uncertainty	10,000 bootstrap resamples; seed 20260919

Reference-arm reuse and numerical records. The 500 reference correct-category trajectories were collected with the same frozen policy and task inputs before the other three full conditions. Before reuse, we checked task identities, actual category queries, starting poses, action budgets, and source hashes. All conditions are scored offline with the revised category-goal sets. The other three full conditions contain 500 trajectories each. Episode-level records bind each outcome to the task, input condition, and source snapshot. We pair records by these identifiers when calculating differences and analyze the 320 development trajectories separately.

Grounded-success scoring. We apply the target-box projection and geometric occlusion checks from the reference scoring snapshot `c3c9ee08` to each saved terminal pose. GSR, SR, and OSR are computed from the same recorded episodes and associated scene geometry. In predicted/correct category order, the reference system finishes within 1 m in 16/15 episodes, of which 10/8 also pass the geometric visibility check. The corresponding MTU3D counts are 7/7 within 1 m and 4/4 passing both criteria, with 500 episodes per condition. These counts yield the GSR entries in Table 2; the score uses the final pose, with no additional STOP requirement.

H Search Coverage and Completion Diagnostics

H.1 Paired effects and cluster sensitivity

Table 15 reports task, scene, and physical-target bootstrap intervals for the main shared-category comparisons. The physical-target cluster is the pair of scene ID and designated object ID. The reference correct-minus-predicted OSR effect is 0.4 points [-2.8, 3.4]. The difference in neighborhood coverage and the endpoint category-input interaction retain their signs under all three clustering choices. Removing the two ambiguity-annotated tasks gives an OSR difference of 26.3 points [20.9, 31.9] for reference minus MTU under predicted categories and an input interaction of -8.0 points [-13.1, -3.0] for MTU minus reference.

H.2 Explicit stopping and goal scope

Let $\mathcal{G}_e$ be either the designated target's horizontal bounding box or the union of the annotated same-category boxes. Define $\delta(q, \mathcal{G}_e)$ as the minimum horizontal distance from pose q to these boxes, equal to zero inside a box. The stopping diagnostic is

$$\mathrm{StopSR}(e; \mathcal{G}_e) = \mathbf{1}[a_\tau = \mathrm{STOP},\ \tau \le 30,\ \delta(q_\tau, \mathcal{G}_e) \le 2\,\mathrm{m}]. \tag{6}$$

Its trajectory counterpart measures whether any budgeted position enters that same surface-distance region. These scores use an explicit action and target-surface geometry; the main SR/OSR columns use target-center geometry, and GSR additionally requires its geometric visibility evidence.

Table 15: Paired center-distance effects in percentage points. Input effects are correct minus predicted category; interaction is MTU's input effect minus the reference input effect. All comparisons use 500 matched tasks.

Contrast	Difference	Task CI	Scene CI	Target CI
Predicted OSR: reference − MTU	26.2	[20.8, 31.8]	[20.3, 31.9]	[20.6, 31.8]
Predicted SR: reference − MTU	3.4	[-1.0, 7.8]	[-1.2, 8.1]	[-1.2, 7.9]
Reference input effect: SR	5.8	[2.0, 9.6]	[1.9, 9.8]	[2.0, 9.7]
MTU input effect: SR	-2.2	[-6.0, 1.4]	[-5.8, 1.2]	[-5.8, 1.6]
Input interaction: SR	-8.0	[-13.0, -2.8]	[-12.9, -3.1]	[-13.1, -3.0]
Input interaction: OSR	-4.8	[-10.0, 0.6]	[-10.9, 1.2]	[-10.1, 0.8]

The six outcomes in Table 5 apply this criterion to the designated instance and the annotated same-category set. When goal regions overlap, designated-instance success takes precedence; among episodes without STOP, reaching the designated region takes precedence over reaching another category region. This ordering gives a mutually exclusive partition of all 500 episodes in each condition.

With predicted categories, MTU's designated StopSR is 16.8% versus 12.4% for the reference system: a difference of 4.4 points with task interval [0.4, 8.4] and physical-target interval [0.0, 8.7]. The ordering of these estimates depends on both the explicit STOP requirement and the distance geometry used to score success.

Completion criteria and scoring geometry. Table 16 crosses endpoint and explicit-STOP evaluation with target-center and target-surface distance on the same trajectories. With predicted categories, the two systems both achieve 11.0% under center-distance StopSR. MTU3D has higher surface-distance success with either completion criterion, whereas the reference system has higher center-distance endpoint success. With correct categories, the reference system scores higher under all four criteria. The comparison separates the effects of the completion requirement and scoring geometry on the observed system ordering.

Table 16: Endpoint and explicit-STOP success (%) on the 2,000 shared-category trajectories, with 500 matched tasks per row. Distance is measured horizontally to the designated target center or bounding box, using a 2 m threshold. STOP must occur within 30 actions.

		Endpoint		Explicit STOP	
System	Input	Center	Surface	Center	Surface
Reference	Predicted	18.4	21.0	11.0	12.4
MTU3D	Predicted	15.0	23.6	11.0	16.8
Reference	Correct	24.2	28.2	19.6	22.0
MTU3D	Correct	12.8	20.8	8.4	13.8

Annotated category goals. Table 17 reports the broader annotated-category scores with start-inside cases separated. All 500 tasks begin outside the designated instance's surface region; 35 begin inside a different same-category region, leaving 465 outside-category starts.

Table 17: Success with an explicit STOP and a 2 m surface-distance threshold (%). Category goals use the annotated same-category instance set. Start conditions and denominators are specific to each goal scope.

System	Input	Designated $N = 500$	Category $N = 500$	Outside $N = 465$	Inside $N = 35$
Reference	Predicted	12.4	17.2	17.0	20.0
Reference	Correct	22.0	32.4	30.5	57.1
MTU3D	Predicted	16.8	22.8	21.3	42.9
MTU3D	Correct	13.8	21.6	19.1	54.3

H.3 DISTANCE TOLERANCE AND REPEATED EXECUTION

Table 18 shows how the target-center radius changes endpoint and trajectory outcomes. Reference search has higher neighborhood coverage at each reported radius, while the predicted-category endpoint ordering changes at 3 m. These scores describe distance tolerance alongside the explicit stopping outcomes.

Table 18: Target-center distance sensitivity (%). Each cell gives SR / OSR for the same 500 trajectories in that condition.

System	Input	1 m	2 m	3 m
Reference	Predicted	3.2 / 28.0	18.4 / 60.8	27.8 / 73.6
Reference	Correct	3.0 / 22.2	24.2 / 61.2	34.6 / 74.6
MTU3D	Predicted	1.4 / 6.2	15.0 / 34.6	32.4 / 61.0
MTU3D	Correct	1.4 / 5.2	12.8 / 30.2	28.4 / 58.8

The two development runs in Table 19 reuse the same 40 tasks and cached predictions, with independently specified execution seeds. The category predictor matches 30.0% of this subset. Both development runs give a zero reference SR input effect and a 7.5-point MTU input effect, differing from the full-run interaction. The repeated runs measure execution variability on these 40 tasks, while the full-set intervals measure sampling uncertainty across the 500 observed tasks.

Table 19: Repeated execution on 40 development tasks. A/B are two runs of each condition. SR uses the main 2 m center-distance definition; changed outcomes count tasks whose binary SR differs between runs.

System	Input	SR A	SR B	Changed	B − A 95% CI
Reference	Predicted	22.5	20.0	1	[-7.5, 0.0]
Reference	Correct	22.5	20.0	1	[-7.5, 0.0]
MTU3D	Predicted	5.0	10.0	2	[0.0, 12.5]
MTU3D	Correct	12.5	17.5	8	[-10.0, 20.0]