RECONAV: RELIABLE CONSTRAINED LONG-HORIZON VISION-AND-LANGUAGE NAVIGATION VIA A DETERMINATION ENGINE

Anonymous authors
Paper under double-blind review

ABSTRACT

Conventional vision-and-language navigation (VLN) asks an agent to follow a described route to a destination. Practical missions can require multiple ordered visits, with hard constraints and preferences governing both target selection and connecting routes. Existing agents use memories, maps, or constraint-guided planning, but reliably propagating scoped requirements across interpretation, grounding, and complete-route planning remains challenging. We introduce ReCoNav, whose instruction contract models route requirements for joint target and route selection. Its Determination Engine uses scene and task rules to propagate route constraints across interpretation, grounding, and planning. Derived evidence determines choices, verifies judgments, and guides repair; vision-language models supply unresolved semantics. We construct a benchmark of 192 scene-grounded tasks across nine indoor scenes to evaluate joint target and route satisfaction. Existing methods show low satisfaction and frequent timeouts. On 44 constraint-resolving tasks without the added time limit, global scene access helps but remains insufficient: ReCoNav achieves 81.82% full instruction satisfaction, versus 29.55% for the best evaluated existing method given global scene information and 38.64% for direct planning with feedback sharing our scene information, backbone, and solver. Across four backbones, ReCoNav improves satisfaction with fewer model tokens than this feedback baseline; ablations support explicit contract-based planning, global determination, and verification with repair.

1 INTRODUCTION

Conventional vision-and-language navigation (VLN) asks an agent to follow a language description of a route to a destination (Anderson et al., 2018; Jain et al., 2019; Krantz et al., 2020; Ku et al., 2020). Recent work studies longer, multi-stage navigation, requiring agents to track progress across successive subtasks and plan between goals (Huang et al., 2023; Song et al., 2025; Zhu et al., 2025; Han et al., 2026). Practical missions additionally impose requirements on the entire journey or particular stages. We study *constrained long-horizon navigation*, where scoped constraints and preferences jointly govern targets and routes.

Figure 1 illustrates the distinction in one scene. Route following specifies movements and landmarks leading to a bed; long-horizon navigation specifies visits to Dining and then a bed, leaving the connecting routes open. The constrained mission limits the second leg to two rooms and prefers avoiding the rug. These requirements apply throughout that leg, including encounters between visits: going through Kitchen violates the preference, while reaching A/B via C exceeds the room limit. Bed C admits a route satisfying both.

Limitations of existing approaches. Observation-driven agents use visual histories and task memories to follow routes and track progress (Zhang et al., 2024; Song et al., 2025; Zhu et al., 2025), but may lack context about distant candidates and connecting routes. Map- and scene-graph-based methods provide broader spatial context, including through pre-exploration (Huang et al., 2023; Bhatt et al., 2026; Zhang et al., 2026). Yet scene access alone does not make scoped requirements govern all relevant targets and route segments. Symbolic planners enforce an interpreted specification, but their guarantees depend on its correctness and scene grounding (Liu et al., 2024;

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.

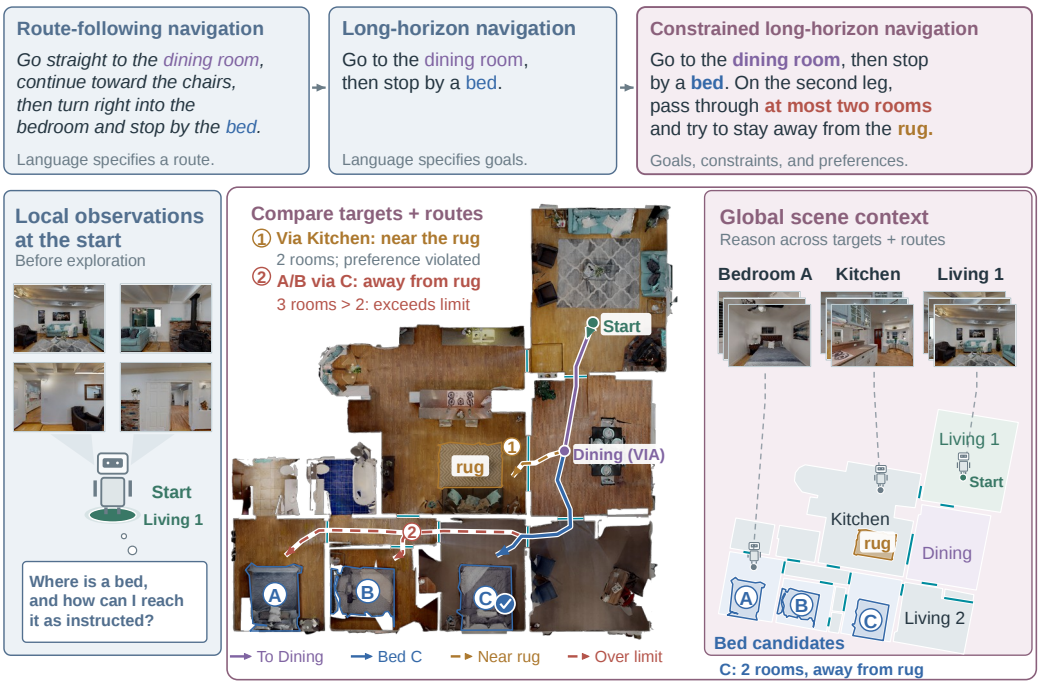


Figure 1: **From route following to constrained long-horizon navigation.** All three instructions start in Living 1 and visit Dining before a bed; the room limit and rug preference apply only after Dining. Global scene context exposes the alternative beds and connecting routes. A–C label candidate beds; VIA denotes the intermediate dining-room visit.

Quartey et al., 2025). For example, semantic and spatial matching in Figure 1 can leave several bed candidates; constraints must govern their selection as well as the routes used to reach them. *Supporting constrained long-horizon navigation requires modeling these tasks and propagating their scoped requirements across instruction interpretation, grounding, and complete-route planning*.

We introduce ReCoNav, modeling visits, order, scoped hard constraints, and preferences in an *instruction contract* for joint target and route selection. An *Affordance Scene Graph* (ASG) grounds this contract in scene entities, spatial relations, and traversability. Our insight is that *uncertainty in individual judgments does not preclude deterministic relations between them*. The *Determination Engine* uses local rules and global route queries to derive commitments, verify judgments, and supply evidence for repair. Vision-language models (VLMs) predict unresolved semantics; revised premises trigger recomputation of dependent conclusions. The same solver supports these queries and constructs final routes. Together, these mechanisms propagate route constraints globally across interpretation, grounding, and complete-route planning while preserving their specified scopes.

We construct a **192-task benchmark across nine indoor scenes**, with scene-grounded scoped requirements and target-resolution annotations. Main results reveal low satisfaction and frequent timeouts among existing methods. Removing the added wall-time limit on Hard44 shows that global context helps but cannot replace scoped route modeling and reliable reasoning: the best evaluated existing method with an ASG reaches 29.55% full instruction satisfaction; Direct-Feedback, sharing our ASG, backbone, and solver, reaches 38.64%, versus **81.82%** for ReCoNav. Across four backbones, comparisons with Direct-ID-Sequence support explicit contract-based planning; Forced-Early and No Verify/Repair ablations show the benefits of global determination and verification with repair. For GPT, global target queries raise resolving-task successes from **28/89 to 89/89**, with 103 controls unchanged. ReCoNav also uses fewer model tokens than Direct-Feedback across backbones.

Our contributions are (i) an instruction contract coupling target selection with whole-route satisfaction under scoped hard constraints and preferences; (ii) ReCoNav, which derives, verifies, and revises interdependent commitments using scene and route evidence; and (iii) a scene-grounded benchmark evaluating these requirements and their role in target selection.

2 Related Work

Navigation and scene representations. VLN agents combine observations with navigation history, topological planning, or hierarchical execution (Chen et al., 2022; Zhou et al., 2024; Chen et al., 2024; Wei et al., 2026). Long-horizon methods use memory or hierarchical subtask planning (Song et al., 2025; Zhu et al., 2025; Han et al., 2026). Semantic maps and scene graphs support spatial grounding (Zhang et al., 2025; Huang et al., 2023; Gu et al., 2024; Werby et al., 2024); pre-exploration provides global context before execution (Zhang et al., 2026; Bhatt et al., 2026).

Spatial and constraint-aware navigation. Spatially informed methods use enriched instructions, subinstruction conditions, or semantic–geometric waypoint grounding (Yang et al., 2026; Chen et al., 2025; Li et al., 2026). GC-VLN grounds waypoints through geometric constraints on direction, distance, and passage, retaining alternatives for backtracking (Yin et al., 2025). In ReCoNav, semantic and spatial matching already defines candidate domains; whole-route feasibility under scoped avoidance and counts, followed by explicit preferences, determines target bindings and routes.

Model-assisted reasoning and planning. Model-assisted systems use formal solvers or simulator feedback for reasoning and planning (Ye et al., 2023; Hao et al., 2025; Rana et al., 2023); Lang2LTL-2 and LIMP use grounded LTL specifications to constrain robot plans (Liu et al., 2024; Quartey et al., 2025). ReCoNav couples temporal requirements with distinct-entity counts and preference costs, using scene and route evidence to derive, verify, and revise interdependent commitments across instruction interpretation, grounding, and planning.

3 Method

3.1 Task Definition

Given registered RGB-D observations of a static scene, an instruction u, and a start pose s_0, the task is to produce a traversable, collision-free waypoint route $\pi = (s_0, q_1, \ldots, q_n)$ satisfying u. The instruction contract $\mathcal{U} = (E, O, H, P)$ represents *route constraints*: visits and the final goal (E), their order (O), scoped avoidance and counts ($H = \mathcal{A} \cup \mathcal{C}$), and preferences ($P$). For scene entities N, let $D(v)$ be the visit area of $v \in N$: an object's neighborhood, a region or level interior, or a portal interface.

Visits and order. Each realized event e_i selects a matching target v_i and waypoint index $j_i \in \{0, \ldots, n\}$, with $q_0 = s_0$:

$$\mathrm{VIA}(e_i):\ q_{j_i} \in D(v_i), \qquad \mathrm{GOAL}(e_g):\ j_g = n,\ q_n \in D(v_g). \tag{1}$$

Portal visits require crossing the interface. Only preferred visits may remain unrealized; repeated visits remain distinct events. A realization ω records consistent target choices and visit positions. Precedence imposes $e_i \prec e_j \Rightarrow j_i < j_j$. BEFORE/AFTER use this strict order; NEXT/BETWEEN additionally restrict intervening events (Appendix A).

Scoped constraints. For a scope $\sigma = (e_a, e_b)$ with $j_a < j_b$, let $J_\sigma(\pi, \omega)$ contain distinct entities encountered strictly between its selected boundary occurrences, including along connecting segments. For target sets $N_A, N_C \subseteq N$ and integer $k \geq 0$,

$$\mathrm{AVOID}(N_A, \sigma): \quad J_\sigma(\pi, \omega) \cap N_A = \varnothing, \tag{2}$$

$$\mathrm{COUNT}_{\bowtie k}(N_C, \sigma): \quad |J_\sigma(\pi, \omega) \cap N_C| \bowtie k, \quad \bowtie \in \{\leq, =, \geq\}. \tag{3}$$

Encounters are entries into object neighborhoods, regions, or levels, or portal crossings; repeated encounters count once per identity per scope. Appendix A details boundary conventions.

Preferences and objective. Preferences use the same predicates with positive violation costs, summed by $p(\pi, \omega; P)$. Let $\mathcal{F}(\mathcal{U})$ contain physically valid route–realization pairs satisfying required visits, applicable orders, and hard constraints. Joint target and route selection minimizes

$$(\pi^*, \omega^*) \in \arg\min_{(\pi,\omega)\in\mathcal{F}(\mathcal{U})} p(\pi, \omega; P). \tag{4}$$

3.2 ReCoNav Overview

Figure 2 links scene interpretation, instruction extraction, grounding, and route planning. The Determination Engine propagates route constraints across these stages using scene and task rules. Derived evidence determines choices, verifies commitments, and guides repair; VLMs supply unresolved semantics. The same solver supports route queries and final planning.

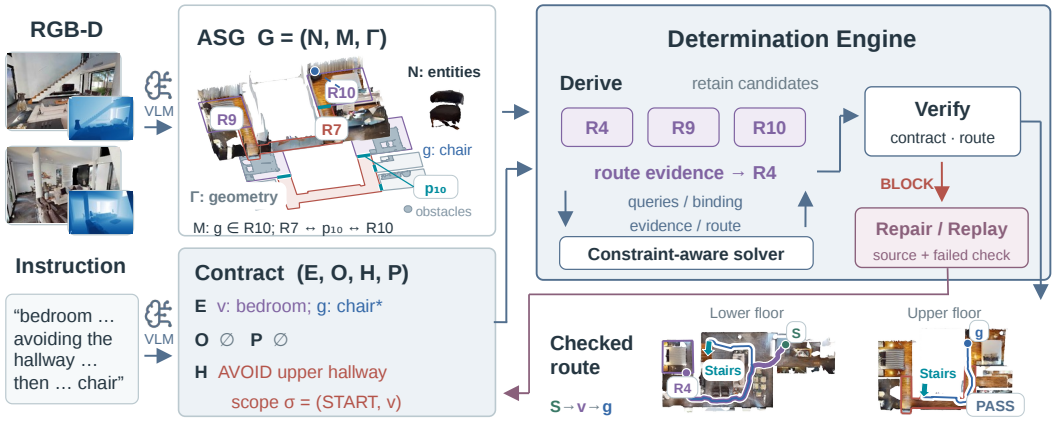


Figure 2: **ReCoNav overview.** ASG: affordance scene graph $G = (N, M, \Gamma)$; N, M, Γ: entities, relations, geometry. R: region (R4: region 4); p_{10}: portal 10. E, O, H, P: events, orders, hard constraints, preferences. S, v, g: start, bedroom visit, chair goal; σ: AVOID scope. $\varnothing$: empty; PASS/BLOCK: accepted/blocked checks. VLM: vision-language model; RGB-D: color and depth. The contract follows repair in Figure 3; the asterisk abbreviates the chair's nightstand qualifier.

Affordance Scene Graph. The ASG $\mathcal{G} = (N, M, \Gamma)$ contains labeled objects, regions, portals, and levels (N), their membership and portal-connectivity relations (M), and geometry (Γ). Geometry defines spatial predicates, visit areas, encounters, and a navigation graph $G_{\text{nav}} = (V_{\text{nav}}, E_{\text{nav}})$ of navigable positions and traversable connections. We instantiate it from pre-exploration RGB-D using MaskClustering instances (Yan et al., 2024), VLM recognition, and region/portal reconstruction (Appendix B.5).

Intent extraction and grounding. The VLM extracts $\mathcal{U}$, retaining source phrases. Rules resolve unambiguous semantic matches; the VLM supplies remaining matches against the ASG vocabulary. Membership and spatial conditions, including *in*, *near*, and *nearest*, restrict each event to $\mathcal{D}_i \subseteq N$. A binding $\beta : E \to N$ selects $v_i = \beta(e_i) \in \mathcal{D}_i$; coreferent events share a target. AVOID and COUNT descriptions resolve to sets N_A, N_C under the same matching conditions.

3.3 Determination Engine

A *commitment* K records a decision, such as a semantic label, event binding, order, or scope boundary, with its source and supporting premises. The engine maintains adopted commitments $\mathcal{K}$ and unresolved alternatives; VLM judgments remain revisable premises. The *commitment frontier* separates rule-determined decisions from unresolved choices. Derivation advances it; verification checks adopted commitments, and repair revises premises and recomputes dependent conclusions.

3.3.1 Derivation

Rules propagate premises to consequences or conflicts:

$$K_1 \wedge K_2 \overset{R}{\Longrightarrow} K_3, \qquad K_1 \wedge K_2 \overset{R}{\Longrightarrow} \bot. \tag{5}$$

A conflict excludes candidate hypothesis K_1 only relative to premises K_2 held fixed for that query. A conflict between revisable judgments alone cannot identify which is wrong. Local rules derive candidate domains from types, semantics, membership, and geometry; global route queries resolve ambiguity *after* this semantic and spatial filtering.

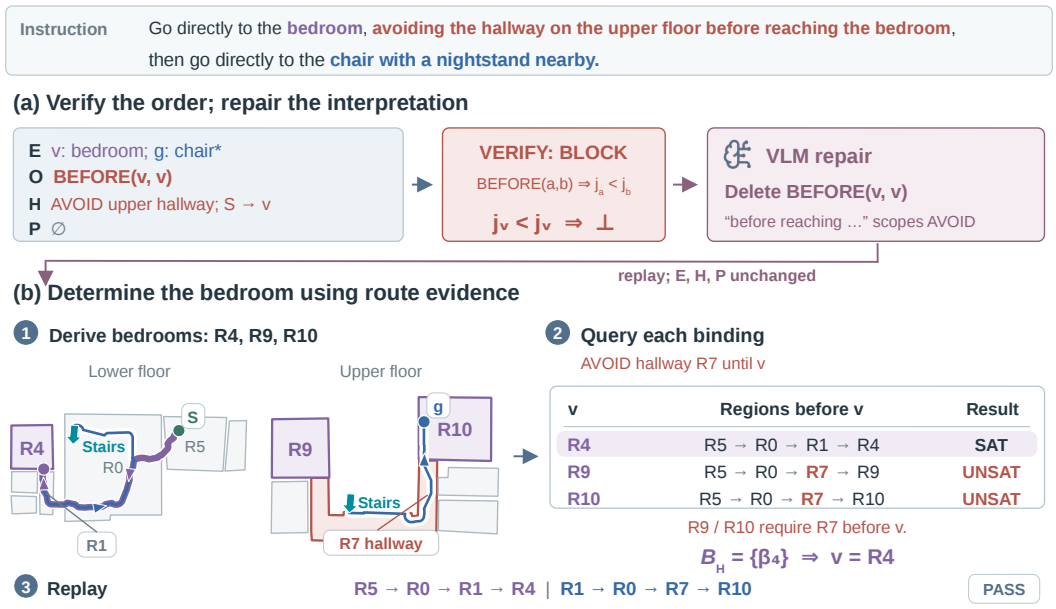


Figure 3: **Determination and repair in one run.** j_a, j_b index the route waypoints of events a, b; $\perp$ denotes contradiction. $\mathcal{B}_H$ contains hard-feasible bindings; β_4 assigns v to R4. SAT/UNSAT denote route feasibility/infeasibility. Table entries show necessary passages for R9/R10 and a feasible prefix for R4; R7 is allowed after v.

Route requirements determine bindings. The solver tests joint bindings, since individually feasible targets may be incompatible together. Let $\mathcal{B}_0$ contain assignments allowed by candidate domains and coreference, and $\mathcal{F}_\beta$ the feasible route–realization pairs for β in the search domain:

$$\mathcal{B}_H = \{\beta \in \mathcal{B}_0 : \mathcal{F}_\beta \neq \varnothing\}, \qquad p^*(\beta) = \min_{(\pi,\omega)\in\mathcal{F}_\beta} p(\pi, \omega; P), \quad \beta \in \mathcal{B}_H. \tag{6}$$

Certified infeasibility excludes an assignment; inconclusive queries leave it unresolved. A certified singleton $\mathcal{B}_H$ determines the binding. Otherwise, preferences select

$$\mathcal{B}_* = \arg\min_{\beta\in\mathcal{B}_H} p^*(\beta). \tag{7}$$

A unique minimizer determines the binding only when the relevant minima are certified; ties remain unresolved.

3.3.2 Route Queries and Planning

For a fixed binding, search augments G_{nav} with event progress, active scopes, encountered identity sets, and preference state. Complete connecting segments update these conditions; hard violations reject transitions. Terminal states must satisfy required visits, applicable orders, and scoped constraints; returned routes are replayed against the contract. Queries yield a feasible witness, certified infeasibility, or unknown; preference queries additionally certify minima when possible. Evidence is relative to the represented search domain (Appendix B.3).

3.3.3 Verification and Evidence-guided Repair

Verification applies the same rule relations to adopted commitments. A violated or unresolved required condition blocks acceptance; unknown does not establish a contradiction. Failed checks identify implicated commitments, source phrases, supporting premises, and dependent conclusions. Retained premises determine a correction when possible; otherwise the VLM revises the implicated semantic judgments using this evidence and u. Replacing commitments Δ^- with Δ^+ gives

$$\mathcal{K}' = \left[\mathcal{K} \setminus (\Delta^- \cup \mathrm{Dep}(\Delta^-))\right] \cup \Delta^+, \tag{8}$$

where Dep includes transitive dependents. Affected grounding, binding, and route queries are recomputed and checked (Appendix B).

Figure 3 illustrates how global propagation of order and scoped avoidance supports verification, repair, and target selection. Verification rejects $\mathrm{BEFORE}(v, v)$ because it requires $j_v < j_v$. Repair removes this order while retaining scoped avoidance. Replayed route queries exclude R9/R10 because reaching them crosses the avoided hallway before the bedroom visit, while R4 is feasible. The route may enter the hallway after this visit, when the avoidance scope has ended. Appendix B.4 gives the full trace.

4 EXPERIMENTS

We construct tasks matching the proposed setting, evaluate performance, analyze information and reasoning beyond the main time limit, and ablate task modeling and the Determination Engine before a real-robot demonstration.

4.1 BENCHMARK

Table 1(a) contrasts existing benchmarks for route following, object grounding, successive goals, and spatial instructions (Anderson et al., 2018; Jain et al., 2019; Ku et al., 2020; Krantz et al., 2020; Qi et al., 2020; Khanna et al., 2024; Song et al., 2025; Yang et al., 2026). Our task additionally needs scoped avoidance and counts, explicit preferences, and annotations of how these requirements resolve target identity; goal success or reference-path similarity alone is insufficient.

We construct **192 tasks across nine HM3DSem scenes**, with RGB-D rendered in Habitat (Yadav et al., 2023; Savva et al., 2019). Symbolic tasks are grounded in scene entities, validated with feasible route witnesses, and expressed as natural-language instructions. Each includes an exact start pose and evaluator-only requirements for scoring the complete route; no tasks are used for training. Hard constraints determine the target in **51 tasks** and preferences in **38**; the other **103** use local qualifiers or direct references as controls. Table 1(b) shows both task-family and target-resolution diversity. Intermediate visits, non-global scopes, and cross-floor routes occur in 55.73%, 30.21%, and 28.65% of tasks, respectively. Appendix C details construction.

Table 1: **Benchmark positioning and diversity.** (a) Instruction and evaluation focus. (b) Task families and target-resolution categories each partition our 192 tasks.

(a) Comparison with existing benchmarks

R2R / R4R / RxR / VLN-CE	Route descriptions; goal success and route following
REVERIE	Remote object descriptions; navigation and object grounding
GOAT-Bench	Successive multimodal goals; goal success and memory reuse
LHPR-VLN	Multiple navigation subtasks; completion and progress
NavSpace	Spatial instructions; navigation across spatial skills
Ours	**Visits, scoped avoidance/counts, preferences; target selection and complete-route satisfaction**

(b) Task composition and target resolution

Task family	Tasks	Target resolved by	Tasks
Goal grounding	36	Hard constraints	51
Visits / order	36	Preferences	38
Avoidance	21	Relational qualifiers	89
Distinct-entity counts	38	Direct / unique	14
Preferences	34		
Compositions	27		

4.2 EVALUATION SETUP

Table 2 compares six VLN methods with their original interfaces and four +ASG adaptations retaining their interpretation and planning logic. ReCoNav and +ASG methods receive ground-truth

scene graphs, without reference contracts, bindings, or routes. Replaceable generative components use Qwen3.8-27B-FP8 (Qwen Team, 2026); trained modules retain released checkpoints. *Direct* predicts a grounded contract for the solver; *Direct-Feedback* adds validation and solver feedback. Both share ReCoNav's ASG, backbone, and solver, isolating determination from information access and constraint planning. Appendix C.3 details the adaptations.

Metrics and budgets. **Hard NS** requires a physically valid route satisfying required visits, orders, and hard constraints; **Full IS** additionally requires all explicit preferences. **Prefer** averages the fraction of satisfied preference clauses over applicable tasks. The main comparison applies a **300 s limit**; timeouts, exhausted budgets, and runtime failures score zero. Completion means a scored outcome, not necessarily success. Fixed task denominators appear in Table 2; token and time means use completed runs. Appendix C.2 specifies accounting.

4.3 Main Results

ReCoNav achieves **80.21% Full IS**; existing methods remain below 8% even with the ASG (Table 2). NavGPT and NavGPT+ASG time out on 80.73% and 95.31% of runs, respectively. With the same ASG, backbone, and solver, Direct-Feedback reaches 42.71%, using 85.15k tokens per completed run versus ReCoNav's **6.54k**. We next remove the added wall-time limit on Hard44 to examine reasoning beyond timeouts, then ablate task modeling and determination.

Table 2: **Navigation and reasoning baselines (300 s).** Rates (%). Full IS / Hard NS: 178 native, 192 ASG tasks; Prefer: 51/54 tasks, respectively. Completion (Comp.) / timeout (TO): all 192 tasks. Costs: completed runs; n/a: uninstrumented.

Method	Full IS↑	Hard NS↑	Prefer↑	Comp.↑	TO↓	Tokens (k)↓	Time (s)↓
Original interfaces							
NavGPT	1.12	1.69	10.78	9.38	80.73	13.44	216.6
MapGPT	7.30	9.55	49.02	90.10	2.60	29.42	141.1
CA-Nav	1.12	2.25	34.31	92.71	0.00	1.28	157.6
GC-VLN	0.00	0.00	17.65	46.88	45.83	0.33	129.3
SNav	3.37	4.49	37.25	92.71	0.00	n/a	42.5
DualVLN	1.69	3.93	37.25	92.71	0.00	n/a	58.6
With ground-truth ASG							
NavGPT+ASG	2.60	2.60	9.26	4.69	95.31	89.62	227.5
GC-VLN+ASG	3.13	3.13	32.41	83.33	16.67	0.34	74.5
Lang2LTL-2+ASG	7.29	7.29	48.15	100.00	0.00	2.22	17.0
LIMP+ASG	5.21	7.81	40.74	98.44	0.00	1.09	3.7
Shared ASG and constraint solver							
Direct	29.69	29.69	25.93	96.35	2.60	47.74	99.4
Direct-Feedback	42.71	46.35	37.96	67.19	32.81	85.15	143.2
ReCoNav	**80.21**	**83.33**	**74.07**	94.27	4.69	6.54	71.5

4.4 Global Context and Constraint Reasoning

Hard44 contains all 44 hard-constraint-resolving tasks with starts supported by every compared method, selected independently of outcomes. Hard route requirements leave one binding among locally matching entities. Removing the added wall-time limit while retaining native resource bounds lets us examine these tasks beyond the main-table timeout cutoff.

Global scene information matters: the ASG improves NavGPT from 4.55% to 29.55% Full IS and GC-VLN from 0% to 6.82%. Yet information alone is insufficient. The +ASG adaptations retain native task models. In particular, geometric waypoint constraints or subinstruction completion checks alone do not enforce scoped avoidance and counts along all connecting segments (Section 2). These route clauses must be explicitly modeled and propagated through target selection and planning. Even with our scoped contract and solver, VLM predictions remain fallible: ReCoNav reaches **81.82% Full IS** (36/44), versus 20.45% for Direct, and 38.64% for Direct-Feedback (both Qwen) (Table 3). NavGPT+ASG and Direct-Feedback average 382.05k and 120.82k tokens per completed run, versus

Table 3: **Hard44 without the added wall-time limit.** Rates (%) divide by all 44 tasks (Prefer: four); failures score zero. Token and time means use completed runs.

Method	Full IS↑	Hard NS↑	Prefer↑	Comp.↑	Tokens (k)↓	Time (s)↓
NavGPT	4.55	4.55	25.00	95.45	37.41	534.4
MapGPT	13.64	13.64	50.00	100.00	29.14	134.3
CA-Nav	0.00	0.00	50.00	100.00	1.27	157.6
GC-VLN	0.00	0.00	25.00	100.00	0.34	539.8
SNav	4.55	4.55	50.00	100.00	n/a	62.4
DualVLN	4.55	4.55	50.00	100.00	n/a	79.0
NavGPT+ASG	29.55	29.55	50.00	97.73	382.05	782.7
GC-VLN+ASG	6.82	6.82	25.00	100.00	0.34	236.2
Lang2LTL-2+ASG	2.27	2.27	25.00	100.00	2.21	15.8
LIMP+ASG	9.09	9.09	50.00	97.73	1.08	3.0
Direct	20.45	20.45	0.00	100.00	47.06	109.3
Direct-Feedback	38.64	38.64	25.00	100.00	120.82	273.5
ReCoNav	**81.82**	**81.82**	**100.00**	97.73	7.17	105.5

ReCoNav's 7.17k. Cross-method time is descriptive because service and decoding settings differ; Prefer covers only four tasks.

4.5 Mechanism Ablations

We use the same 192 tasks and GT-ASGs with GPT-5.6 Sol (OpenAI, 2026), Gemini 3.1 Pro Preview (Google DeepMind, 2026a), Qwen3.8-27B (Qwen Team, 2026), and Gemma4-31B (Gemma Team, 2026). Without an added wall-time limit, completion is 98.44–100%. Quality and cost use completed runs; Appendix C.4 reports the full baseline comparison and remaining metrics.

Table 4: **Task modeling and engine ablations.** No added wall-time limit. (a) Full IS (%) and tokens (k) on completed runs. (b) Full IS successes / jointly completed tasks, Forced-Early → ReCoNav; full groups: 89 resolving, 103 controls.

	GPT		Gemini		Qwen		Gemma	
(a) Overall quality and cost								
Variant	Full IS↑	Tokens↓	Full IS↑	Tokens↓	Full IS↑	Tokens↓	Full IS↑	Tokens↓
Direct-ID-Sequence	86.91	31.13	82.11	45.64	42.63	42.15	55.03	42.40
Forced-Early	68.23	16.05	67.54	38.32	56.54	9.96	64.21	9.94
No Verify/Repair	98.96	**9.19**	96.35	**16.05**	73.82	**5.68**	93.19	**6.37**
ReCoNav	**100.00**	9.27	**98.44**	17.08	**82.11**	7.03	**96.34**	6.87
(b) Paired target-resolution analysis								
Resolving tasks	28/89→89/89		28/88→87/88		19/88→69/88		25/88→87/88	
Control tasks	103/103→103/103		101/103→101/103		89/102→87/102		97/102→97/102	

Explicit instruction contracts. *Direct-ID-Sequence* predicts entity IDs and connects them geometrically, without enforcing route clauses during connection. Its Full IS is lower than ReCoNav across all four backbones: 42.63% versus 82.11% with Qwen. This exposes the limitation of ID-sequence planning and supports explicitly modeling scoped requirements for both target selection and route construction.

Global propagation for target selection. *Forced-Early* retains local grounding and repair, omitting global feasibility and preference queries. Table 4(b) contrasts 89 resolving tasks with 103 controls. GPT successes rise from **28/89 to 89/89**, with controls unchanged. Gains concentrate on resolving tasks across backbones; controls are unchanged except for two fewer Qwen successes.

Verification and propagation after repair. *No Verify/Repair* retains derivation and joint binding. Restoring verification and repair raises Full IS from 98.96% to 100% for GPT and from **73.82% to 82.11%** for Qwen, whose tokens rise from 5.68k to 7.03k. Repairs propagate to dependent grounding and route decisions.

4.6 Real-Robot Demonstration

Figure 4 shows a scanned scene, its ASG, and a 14.53 m planned route satisfying a specified contract: visit a bin before the washing machine while avoiding the office. Avoidance excludes the office bin; robot photographs document the corresponding task locations.

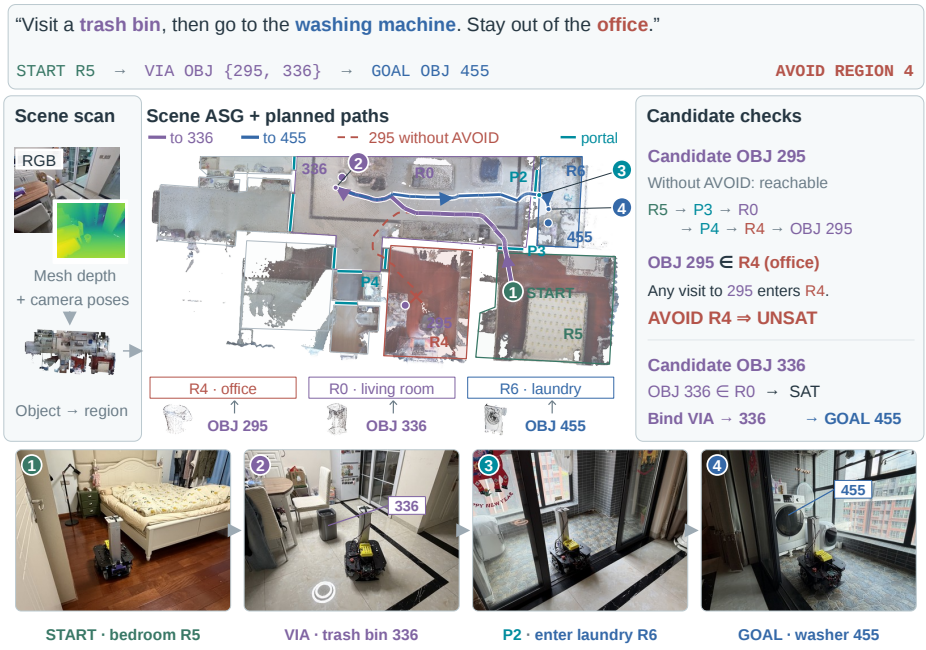


Figure 4: **Real-robot demonstration.** P: portal (P2: portal 2); OBJ: object ID; RGB: color images. START/VIA/GOAL: start, intermediate visit, final destination. The map shows planned paths; numbers 1–4 link task locations to the robot photographs.

5 Limitations and Future Work

ReCoNav verifies conclusions relative to an extracted instruction contract and a symbolic scene representation. A semantically incorrect but internally consistent contract may escape verification (Ye et al., 2023). Likewise, symbolic route guarantees depend on the ASG and navigation graph faithfully capturing relevant entities and traversability; imperfect object and room recognition has been observed in scene-graph navigation (Werby et al., 2024). Our controlled evaluation uses fixed, ground-truth ASGs to isolate task reasoning. A next step is to integrate online 3D semantic mapping (Deng et al., 2026; Zhai et al., 2026) and incremental scene-graph construction (Hughes et al., 2022), recomputing dependent conclusions as evidence changes.

6 Conclusion

ReCoNav couples target selection with whole-route satisfaction through an instruction contract. Its Determination Engine propagates route constraints to derive decisions, verify judgments, and guide repair using scene and task evidence. Across 192 tasks and four backbones with shared ground-truth ASGs, it improves complete instruction satisfaction while reducing model usage relative to direct generation and feedback controls.

AI USE STATEMENT

Generative AI assisted writing, literature search, coding and auditing, baseline adaptation, experiments and analysis, and figure preparation. Gemini-3.7-Flash and GPT-5.6-Sol generated benchmark instructions from author-defined symbolic tasks. The authors developed the research question, core method, formalization, and benchmark design, reviewed AI-assisted outputs, and take responsibility for the final text, code, data, and results. Language and vision-language models are also components of the evaluated systems.

REPRODUCIBILITY STATEMENT

Section 3 and Appendices A–C document the formulation, reasoning and search procedures, scene construction, evaluation protocols and settings, and complete ablation results. We plan to release a cleaned implementation and benchmark.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 3674–3683. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00387. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Anderson_Vision-and-Language_Navigation_Interpreting_CVPR_2018_paper.html.

Neel P. Bhatt, Yunhao Yang, Rohan Siva, Pranay Samineni, Daniel Milan, Zhangyang Wang, and Ufuk Topcu. VLN-Zero: Rapid exploration and cache-enabled neurosymbolic vision-language planning for zero-shot transfer in robot navigation. In *3rd International Conference on Neuro-Symbolic Systems (NeuS)*, 2026.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Kenneth Wong. MapGPT: Map-Guided Prompting with Adaptive Path Planning for Vision-and-Language Navigation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 9796–9810. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.529. URL https://doi.org/10.18653/v1/2024.acl-long.529.

Kehan Chen, Dong An, Yan Huang, Rongtao Xu, Yifei Su, Yonggen Ling, Ian D. Reid, and Liang Wang. Constraint-Aware Zero-Shot Vision-Language Navigation in Continuous Environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(11):10441–10456, 2025. doi: 10.1109/TPAMI.2025.3594204.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 16516–16526. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01604.

Giuseppe De Giacomo and Moshe Y. Vardi. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In Francesca Rossi (ed.), *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pp. 854–860. IJCAI/AAAI, 2013. URL http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6997.

Leonardo Mendonça de Moura and Nikolaj S. Bjørner. Z3: An Efficient SMT Solver. In C. R. Ramakrishnan and Jakob Rehof (eds.), *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pp. 337–340. Springer, 2008. doi: 10.1007/978-3-540-78800-3_24. URL https://doi.org/10.1007/978-3-540-78800-3_24.

Zilong Deng, Federico Tombari, Marc Pollefeys, Johanna Wald, and Daniel Barath. OVI-MAP: Open-vocabulary instance-semantic mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12606–12616, June 2026.

Jon Doyle. A Truth Maintenance System. *Artif. Intell.*, 12(3):231–272, 1979. doi: 10.1016/0004-3702(79)90008-0.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Google DeepMind. Gemini 3.1 Pro, 2026a. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/.

Google DeepMind. Gemini 3.7 Flash, 2026b. URL https://deepmind.google/models/model-cards/gemini-3-7-flash/.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 5021–5028. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610243.

Zebin Han, Xudong Wang, Baichen Liu, Qi Lyu, Zhenduo Shang, Jiahua Dong, Lianqing Liu, and Zhi Han. SeqWalker: Sequential-Horizon Vision-and-Language Navigation with Hierarchical Planning. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 18279–18287. AAAI Press, 2026. doi: 10.1609/AAAI.V40I22.38891.

Yilun Hao, Yang Zhang, and Chuchu Fan. Planning Anything with Rigor: General-Purpose Zero-Shot Planning with LLM-based Formalized Programming. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=0K1OaL6XuK.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual Language Maps for Robot Navigation. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*, pp. 10608–10615. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10160969.

Sheng-Yu Huang, Jaesung Choe, Yu-Chiang Frank Wang, and Cheng Sun. OpenVoxel: Training-free grouping and captioning voxels for open-vocabulary 3D scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16734–16745, 2026.

Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization. In Kris Hauser, Dylan A. Shell, and Shoudong Huang (eds.), *Robotics: Science and Systems XVIII, New York City, NY, USA, June 27 - July 1, 2022*, 2022. doi: 10.15607/RSS.2022.XVIII.050.

Vihan Jain, Gabriel Magalhães, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the Path: Instruction Fidelity in Vision-and-Language Navigation. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 1862–1872. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1181.

Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Théophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. GOAT-Bench: A Benchmark for Multi-Modal Lifelong Navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 16373–16383. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01549.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the Nav-Graph: Vision-and-Language Navigation in Continuous Environments. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVIII*, volume 12373 of *Lecture Notes in Computer Science*, pp. 104–120. Springer, 2020. doi: 10.1007/978-3-030-58604-1_7. URL https://doi.org/10.1007/978-3-030-58604-1_7.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual Vision-and-Language Navigation with Dense Spatiotemporal Grounding. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online,*

November 16-20, 2020, pp. 4392–4412. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.356. URL https://doi.org/10.18653/v1/2020.emnlp-main.356.

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

Jason Xinyu Liu, Ankit Shah, George Konidaris, Stefanie Tellex, and David Paulius. Lang2LTL-2: Grounding Spatiotemporal Navigation Commands Using Large Language and Vision-Language Models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pp. 2325–2332. IEEE, 2024. doi: 10.1109/IROS58592.2024.10802696.

OpenAI. GPT-5.6 Sol, 2026. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol.

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 9979–9988. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01000. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Qi_REVERIE_Remote_Embodied_Visual_Referring_Expression_in_Real_Indoor_Environments_CVPR_2020_paper.html.

Benedict Quartey, Eric Rosen, Stefanie Tellex, and George Konidaris. Verifiably Following Complex Robot Instructions with Foundation Models. In *IEEE International Conference on Robotics and Automation, ICRA 2025, Atlanta, GA, USA, May 19-23, 2025*, pp. 1–8. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11127418.

Qwen Team. Qwen3.8-27B-FP8, 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B-FP8.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. SayPlan: Grounding large language models using 3D scene graphs for scalable robot task planning. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 23–72. PMLR, 2023. URL https://proceedings.mlr.press/v229/rana23a.html.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=Ha6RTeWMd0.

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 *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 9338–9346. IEEE, 2019. doi: 10.1109/ICCV.2019.00943.

Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. Towards Long-Horizon Vision-Language Navigation: Platform, Benchmark and Method. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 12078–12088. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01128. URL https://openaccess.thecvf.com/content/CVPR2025/html/Song_Towards_Long-Horizon_Vision-Language_Navigation_Platform_Benchmark_and_Method_CVPR_2025_paper.html.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground Slow, Move Fast: A Dual-System Foundation Model for Generalizable Vision-Language Navigation. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust (eds.), *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. proceedings.iclr.cc, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/14da7aea05debb963b3d8d46449d51a0-Abstract-Conference.html.

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado (eds.), *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.077.

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Théophile Gervet, John M. Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-Matterport 3D Semantics Dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 4927–4936. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00477.

Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. MaskClustering: View Consensus Based Mask Graph Clustering for Open-Vocabulary 3D Instance Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 28274–28284. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02671.

Haolin Yang, Yuxing Long, Zhuoyuan Yu, Zihan Yang, Minghan Wang, Jiapeng Xu, Yihan Wang, Ziyan Yu, Wenzhe Cai, Lei Kang, and Hao Dong. NavSpace: How navigation agents follow spatial intelligence instructions, 2026. URL https://arxiv.org/abs/2510.08173.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. SatLM: Satisfiability-Aided Language Models Using Declarative Prompting. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Hang Yin, Haoyu Wei, Xiuwei Xu, Wenxuan Guo, Jie Zhou, and Jiwen Lu. GC-VLN: Instruction as graph constraints for training-free vision-and-language navigation. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1809–1824. PMLR, 2025. URL https://proceedings.mlr.press/v305/yin25a.html.

Hongjia Zhai, Qi Zhang, Xiaokun Pan, Xiyu Zhang, Yitong Dong, Huaqi Zhang, Dan Xu, and Guofeng Zhang. OnlinePG: Online open-vocabulary panoptic mapping with 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33269–33279, June 2026.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado (eds.), *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.079.

Jiwen Zhang, Zejun Li, Siyuan Wang, Xiangyu Shi, Zhongyu Wei, and Qi Wu. SpatialNav: Leveraging Spatial Scene Graphs for Zero-Shot Vision-and-Language Navigation. *CoRR*, abs/2601.06806, 2026. doi: 10.48550/ARXIV.2601.06806.

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A Novel Memory Representation via Annotated Semantic Maps for VLM-based Vision-and-Language Navigation. In

Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 13032–13056. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.638.

Gengze Zhou, Yicong Hong, and Qi Wu. NavGPT: Explicit Reasoning in Vision-and-Language Navigation with Large Language Models. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 7641–7649. AAAI Press, 2024. doi: 10.1609/AAAI.V38I7.28597. URL https://doi.org/10.1609/aaai.v38i7.28597.

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 *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 8120–8132. IEEE, 2025. doi: 10.1109/ICCV51701.2025.00761.

A CONTRACT SEMANTICS

The contract's visits, order, scoped avoidance and counts, and preferences are all *route constraints*. They share one target assignment and event realization. Their consequences can therefore constrain interpretation and grounding as well as the final route.

Waypoint occurrences. The waypoint sequence includes the selected visit occurrences. A connecting motion may be split at an occurrence without changing the physical path. Each realized event has its own waypoint index j_i; successive events may share a physical pose and are then represented by separate entries. START has index 0 and GOAL has index n.

Descriptions and realizations. Descriptions are fields of events and constraint clauses in $\mathcal{U} = (E, O, H, P)$. Each retains a nominal phrase, spatial qualifiers, and any coreference relation. Event grounding selects $\beta(e_i) \in \mathcal{D}_i \subseteq N$; AVOID and COUNT descriptions denote sets of matching entities. The start s_0 anchors start-relative distance comparisons. If a set description uses a resolved event target as a spatial reference, its denotation uses that same target. A realization ω records these consistent target choices and one selected occurrence for each realized event. An optional preferred visit may have no selected occurrence. Two visits to the same entity remain distinct events and may be ordered independently. All predicates use this shared realization.

Table 5: Contract operators. Ordering uses requested event witnesses; scoped conditions use the physical encounter trace.

Operator	Semantics
VIA / GOAL	A visit witness / a terminal visit witness for the target.
BEFORE / AFTER	Strict order between two requested occurrences.
NEXT	Consecutive requested milestone witnesses, allowing physical motion between them.
BETWEEN–ONLY	Exactly the named milestones occur between the boundary witnesses.
BETWEEN–WITHOUT	The named milestones may not occur between the boundary witnesses.
AVOID	No target-entity encounter strictly within the scope.
COUNT	The number of distinct target identities in the scope satisfies $\leq$, $=$, or $\geq k$.
PREFER	A visit, order, avoidance, or count predicate incurs its violation cost if unsatisfied.

Order predicates. Let $E_\omega \subseteq E$ be the realized requested events. For realized boundary witnesses with $j_a < j_b$, define $B_{ab}(\omega) = \{e_i \in E_\omega : j_a < j_i < j_b\}$. BEFORE$(e_a, e_b)$ and AFTER(e_b, e_a) both require $j_a < j_b$. For a named event set $F \subseteq E$, the remaining predicates require

$$\begin{aligned} \text{NEXT}(e_a, e_b): &\quad j_a < j_b \ \wedge \ B_{ab}(\omega) = \varnothing, \\ \text{BETWEEN-ONLY}(e_a, e_b, F): &\quad j_a < j_b \ \wedge \ B_{ab}(\omega) = F, \\ \text{BETWEEN-WITHOUT}(e_a, e_b, F): &\quad j_a < j_b \ \wedge \ B_{ab}(\omega) \cap F = \varnothing. \end{aligned}$$

These predicates concern requested event witnesses; incidental physical encounters are evaluated by the scoped constraints.

Encounter trace. Let $\tau(\pi) = ((v_\ell, \xi_\ell))_{\ell=1}^{L}$ be the ordered physical encounters, where $v_\ell \in N$ and $\xi_\ell \in [0, n]$ records an encounter's position along the route. Waypoint q_j has position j; positions strictly between j and $j+1$ lie on their connecting motion. For $\sigma = (e_a, e_b)$,

$$J_\sigma(\pi, \omega) = \{v_\ell : j_a < \xi_\ell < j_b\}.$$

The geometry of each entity determines its encounter condition:

Entity	Visit geometry $D(v)$	Encounter recorded in $\tau(\pi)$
Object	Approach neighborhood of its footprint	Entry into that neighborhood
Region	Navigable region interior	Entry from outside the region
Portal	Registered passage interface	Crossing between its two sides
Level	Navigable extent of its regions	Transition into the level

Encounters are computed along complete waypoint segments, so unrequested objects still affect AVOID and COUNT. Initial occupancy does not create a new entry. Leaving and re-entering creates another occurrence but does not increase a distinct-identity count. Only the encounter at an actual scope boundary is excluded; an earlier encounter with the eventual goal remains relevant. A local planning-leg endpoint does not create a new scope boundary. The boundaries must have selected occurrences with $j_a < j_b$. Omitted boundaries default to START and GOAL.

Conditional orders and preferences. Orders involving optional preferred visits apply when all their referenced events are realized. An unselected preferred visit incurs its violation cost. Each preferred clause incurs its cost at most once, regardless of repeated violations. The minimization in Equation 4 uses one ω for all clauses; it cannot select different target identities or occurrences to satisfy individual clauses independently. Distinct-entity counts over neighboring scopes cannot generally be added, because their identity sets can overlap.

B Constraint Propagation, Verification, and Repair

The Determination Engine applies the same scene and task relations to derive decisions, verify adopted judgments, and provide evidence for repair. Target selection is one application of this shared mechanism.

B.1 Commitments and Conditional Propagation

Let $\mathcal{K}$ contain the current commitments and $\mathcal{R}$ the applicable rules. A VLM prediction enters $\mathcal{K}$ as a revisable premise. A rule application records its premises, rule, and conclusion, using either form in Equation 5: a new commitment or a conflict $\bot$. A derived conclusion remains conditional on its supporting premises; an uncertain premise does not become certain merely through propagation. A conflict is recorded for verification and revision.

Local rules combine admissible entity kinds, semantic matches, membership, and geometry to restrict candidate domains. Route queries then propagate the contract's requirements across interpretation, grounding, and planning. This propagation preserves each clause's scope: a requirement restricted to one route portion can affect an upstream decision without becoming a whole-mission restriction.

Target selection under retained premises. For a candidate hypothesis K_1, the conflict $K_1 \wedge K_2 \overset{R}{\Longrightarrow} \bot$ establishes exclusion relative to the retained premises K_2. Query evidence depends on the interpreted contract and scene representation; changes to either invalidate its use for candidate exclusion.

Incomplete route evidence. For the joint binding domain $\mathcal{B}_0$, let $\mathcal{B}_{\mathrm{sat}}$ contain assignments with verified feasible witnesses, $\mathcal{B}_{\mathrm{unsat}}$ those with infeasibility certificates, and $\mathcal{B}_?$ the remaining assignments. These sets partition $\mathcal{B}_0$, and valid query evidence gives

$$\mathcal{B}_{\mathrm{sat}} \subseteq \mathcal{B}_H \subseteq \mathcal{B}_{\mathrm{sat}} \cup \mathcal{B}_?. \tag{9}$$

The first inclusion follows from the witnesses; the second follows because an infeasibility certificate rules out every feasible route for that assignment in the search domain. Consequently, one feasible assignment determines the binding only after all competitors have been excluded. A timeout leaves its assignment in $\mathcal{B}_?$.

If several assignments are feasible, their certified preference minima define $\mathcal{B}_*$ in Equation 7. A unique minimum determines a binding; equal minima preserve alternatives. Path length and entity ID do not resolve this semantic ambiguity. All conclusions remain conditional on the current scene representation and instruction interpretation.

B.2 Verification, Evidence, and Repair

Verification. A check a applies a rule relation to adopted commitments. Its result $v_a = \mathrm{Verify}_a(\mathcal{K}) \in \{\top, \bot, ?\}$ denotes an established obligation, a demonstrated violation, or insufficient

evidence. Let $b_a = 1$ when establishing this obligation is required at the current stage:

$$\text{status}(a) = \begin{cases} \text{PASS}, & v_a = \top, \\ \text{BLOCK}, & v_a \neq \top \ \wedge \ b_a = 1, \\ \text{WARNING}, & \text{otherwise.} \end{cases} \tag{10}$$

An unknown required check blocks acceptance without proving a contradiction.

Suppose K_1 and K_2 are both revisable judgments and $K_1 \wedge K_2 \overset{R}{\Longrightarrow} \bot$. The rule establishes that they cannot both hold; it does not establish which judgment is wrong. For example, extracted orders $e_a \prec e_b$ and $e_b \prec e_a$ imply $j_a < j_b < j_a$, but either order may have been misinterpreted. Likewise, requiring at least three distinct entities from a grounded set of size two exposes an inconsistency whose source may be the requested number or the grounding. These illustrative checks do not by themselves determine a semantic replacement.

Evidence from connected derivations. Let $\mathcal{T}$ contain the current rule applications, with $\text{prem}(d)$ and $\text{out}(d)$ denoting the premise set and conclusion of application d. For implicated commitments S, define their directly connected derivations as

$$\mathcal{E}(S) = \{d \in \mathcal{T} : S \cap (\text{prem}(d) \cup \{\text{out}(d)\}) \neq \varnothing\}. \tag{11}$$

For the conflict above, $S = \{K_1, K_2\}$. Either commitment may occur as a premise or conclusion in either rule form of Equation 5. Incoming derivations expose the support for a judgment; outgoing derivations expose its consequences and further conflicts. Following these links supplies additional upstream support and downstream dependencies when needed. A predicted premise with no incoming derivation retains its instruction phrase or observation as its source.

Repair evidence combines these rule relations with source phrases, relevant scene facts, the failed check, and admissible alternatives. A count diagnostic links the requested number to the grounded identities and the filters that produced that set. A route diagnostic links an offending encounter to its active scope and clause. The implementation passes a localized projection of this information to the VLM: implicated semantic fields, current interpretations and grounding, diagnostics, and legal label alternatives. The notation describes the dependency semantics; it does not require serializing an exhaustive proof or search tree. Evidence can guide revision while leaving the correct interpretation unresolved.

Determined and predicted repairs. Let Δ^- contain the commitments selected for revision and let

$$\mathcal{K}^- = \mathcal{K} \setminus \big(\Delta^- \cup \text{Dep}(\Delta^-)\big). \tag{12}$$

A rule may commit a replacement K^+ only when retained premises derive it, $\mathcal{K}^- \vdash_{\mathcal{R}} K^+$, and the applicable rule uniquely determines that replacement. Here $\vdash_{\mathcal{R}}$ denotes supported applications of the registered rules. The replacement cannot rely on the removed judgment or its dependent conclusions.

Otherwise, the VLM proposes revisions to the implicated semantic clauses or label correspondences using the original instruction and this evidence. Such revisions remain premises to be checked. A backend failure or an inconclusive query alone does not justify changing instruction semantics; warnings and preference violations alone do not trigger repair.

Invalidation and replay. Equation 8 adds the replacements to $\mathcal{K}^-$. Dependencies are transitive: changing a semantic match can change a candidate domain, invalidate binding evidence, and alter the resulting route. Unaffected premises are retained; affected grounding, binding queries, and route checks are recomputed. Thus propagation also carries the consequences of a repair through the task. Maintaining supports and revising dependent beliefs follows the truth-maintenance principle (Doyle, 1979); here, scene and route queries supply task-specific evidence. Passing the implemented checks establishes consistency relative to the retained premises and symbolic abstraction. An internally consistent but semantically incorrect contract can still escape these checks.

B.3 Route Search and Query Evidence

Route queries supply evidence for target determination, verification of adopted routes, and diagnosis of conflicts under the current contract. For a fixed binding, the solver augments G_{nav} with the start and target positions. Within regions, collision-free visibility edges connect navigable positions; portal and stair transitions connect regions. The search domain is finite, and physical edges are evaluated together with task progress.

State and transitions. A conceptual search state is

$$z = \big(x, \eta, \{C_\sigma\}_{\sigma \text{ active}}, \rho\big), \tag{13}$$

where x is the current navigation vertex, η records requested-event and order/scope progress, C_σ stores the encountered identities needed by each scope's count clauses, and ρ tracks preference satisfaction and incurred penalties. Selected event occurrences are recovered with the route.

Traversing an edge produces an ordered trace of entity entries and portal crossings over the whole segment. The monitor updates events, scopes, identity sets, and preferences in trace order using Appendix A. Requested occurrences must meet their visit and ordering conditions; incidental encounters still affect scoped clauses. Hard avoidance violations and exceeded count upper bounds reject transitions immediately. Lower bounds and count equality are checked when the scope closes. A terminal state must realize the required events, close their scopes, and end at the bound goal. Unrealized preferred visits incur their declared costs.

Why event monitoring is sufficient. After each transition, the task state summarizes the processed physical trace and selected event occurrences. This invariant holds initially at START. Each edge update processes every new encounter and applies the same visit, scope, and count semantics, so it preserves the invariant. Thus an accepting state supplies a route–realization pair satisfying the grounded hard requirements. Distinct identities must be retained: counting transitions alone would count a revisited entity more than once.

Temporal monitors implement finite-trace visit and order patterns (De Giacomo & Vardi, 2013), supplemented with identity-set memory for counts. Z3 (de Moura & Bjørner, 2008) separately checks order and scope consistency using integer event positions. These symbolic checks can reject inconsistent contracts before physical route search.

Feasibility and optimality evidence. A replay-checked witness establishes feasibility. Infeasibility requires exhausting the relevant finite search or a valid certificate from a relaxation. Connectivity and region-topology relaxations can exclude an assignment when even the relaxed problem has no route; a route in a relaxation alone does not establish physical feasibility.

Each preference contributes a nonnegative cost at most once. A zero-cost witness therefore certifies the minimum. A positive minimum requires completed optimization or a matching lower bound excluding smaller costs. Interrupted search without such evidence returns an unknown result for the unresolved claim. Predecessor links reconstruct waypoint segments and event occurrences; replay evaluates the complete route against the contract. These certificates concern the represented search domain.

B.4 Worked Determination Example

Figures 2 and 3 use one recorded run with the instruction:

> Go directly to the bedroom, avoiding the hallway on the upper floor before reaching the bedroom, then go directly to the chair with a nightstand nearby.

Verification and semantic repair. The initial contract contains the correct avoidance scope $\sigma = (\mathrm{START}, v)$ and an incorrect self-order $\mathrm{BEFORE}(v, v)$. The strict-order rule gives $j_v < j_v$, so verification blocks before dependent route queries. Using the source phrase, VLM repair removes this order while preserving the visits, avoidance clause, and preferences. The remaining order list is empty; the required visit still precedes the terminal goal through their event roles. The failed rule localizes the inconsistency, while the source phrase guides its semantic correction.

Replayed target determination. Replayed grounding yields bedroom regions R4/R9/R10, avoided hallway R7, and goal chair 50 in R10. Let β_i bind v to Ri with the chair fixed. The feasible prefix for β_4 is R5–R0–R1–R4. R7 separates the start from R9 and R10 in the region topology, so reaching either candidate before v contradicts the avoidance requirement. Hence

$$\mathcal{B}_{\text{sat}} = \{\beta_4\}, \qquad \mathcal{B}_{\text{unsat}} = \{\beta_9, \beta_{10}\}, \qquad \mathcal{B}_? = \varnothing.$$

Equation 9 therefore determines R4. The rejected rows in Figure 3(b) show necessary topological passages, while the R4 row shows a feasible route prefix.

Complete-route verification. The full region sequence is R5–R0–R1–R4–R1–R0–R7–R10. Replay checks that R7 is encountered only after v, when its avoidance scope has ended. Visit, order, and avoidance requirements therefore support both the earlier verification and repair and the subsequent binding and route decisions, with the avoidance scope preserved. It uses a ground-truth ASG; the overview's RGB-D views illustrate the scene input rather than a reconstruction from those two views alone.

B.5 Scene Instantiation

Our ASG instantiation uses a registered pre-exploration RGB-D scan. MaskClustering (Yan et al., 2024) supplies 3D object instances. VLM recognition uses mask-conditioned RGB views following OpenVoxel (Huang et al., 2026), retaining surrounding context while highlighting the intended instance. Floor and wall geometry partition regions, which VLMs label using views and contained objects.

Measured wall openings and SAM 2 masks (Ravi et al., 2025) propose portals; identified stairs propose cross-level transitions. Geometric checks establish passage interfaces and navigable anchors. Containment supplies membership, and portal incidence supplies region connectivity. Entity encounter geometry is retained separately from collision geometry, allowing an object neighborhood to affect a task constraint without becoming a physical obstacle.

C Evaluation Details

C.1 Benchmark Construction

Table 1 gives benchmark positioning and the task-family and target-resolution counts. Explicit preferences occur in 28.13% of tasks; the other overlapping properties are reported in Section 4.1.

Tasks are constructed from scene entities, candidate descriptions, and route requirements. Symbolic validation retains a feasible route witness and checks how local qualifiers, hard constraints, or preferences resolve the target. A witness certifies feasibility; it need not be the shortest route. Candidate diagnostics concern the alternatives remaining after semantic and spatial matching. Requirement annotations and these diagnostics are available only to the evaluator.

The nine HM3DSem scenes contribute 26, 33, 18, 30, 9, 10, 27, 27, and 12 tasks, respectively, for scene IDs 00824, 00829, 00843, 00848, 00849, 00853, 00861, 00873, and 00891. Natural-language instructions are generated from the symbolic tasks using Gemini-3.7-Flash (109) (Google DeepMind, 2026b) and GPT-5.6-Sol (83) (OpenAI, 2026). The 89 constraint/preference-resolving tasks contain 42 preference tasks; the 103 controls contain 12.

Task settings. The benchmark restricts GOAL to objects or regions and uses a $1\,\mathrm{m}$ threshold for object approach and encounter. Each violated preference contributes one unit of cost, at most once. For a fixed binding, equal-penalty routes are ranked by length; length does not resolve ties between target bindings. The description vocabulary uses membership, portal incidence, and geometric NEAR, NEAREST, and FARTHEST predicates. These predicates are evaluated from geometry rather than stored as ASG edges; the general formulation permits additional geometric predicates.

C.2 Scoring and Budgets

Evaluation checks the physical waypoint trace, including intermediate encounters, event order, scoped avoidance, and distinct-entity counts. Full IS and Hard NS are task success rates. Prefer

first computes the satisfied fraction of explicit preference clauses for each applicable task, then averages across those tasks; it does not measure preference optimality. Preference-free tasks contribute to Full IS / Hard NS but not Prefer.

Main comparison. Main-table quality uses fixed denominators of 178 supported native-interface tasks and 192 ASG-based tasks (including Direct and Direct-Feedback), with 51 and 54 preference tasks. Unsupported starts are identified before evaluation and are not moved. All subsequent timeouts, compute-budget failures, and runtime exceptions contribute zero. Completion and timeout rates divide by all 192 issued tasks. An ordinary incorrect decision or native step-limit termination remains a scored outcome. Table 6 gives the outcome counts.

Table 6: **Run outcomes under the main-table budget.** Each finalized row sums to 192. Resource denotes token/solver-budget exhaustion; error denotes a runtime exception.

Method	Completed	Timeout	Resource	Error	Unsupported
NavGPT	18	155	0	5	14
MapGPT	173	5	0	0	14
CA-Nav	178	0	0	0	14
GC-VLN	90	88	0	0	14
SNav	178	0	0	0	14
DualVLN	178	0	0	0	14
NavGPT+ASG	9	183	0	0	0
GC-VLN+ASG	160	32	0	0	0
Lang2LTL-2+ASG	192	0	0	0	0
LIMP+ASG	189	0	1	2	0
Direct	185	5	2	0	0
Direct-Feedback	129	63	0	0	0
ReCoNav	181	9	2	0	0

The main comparison treats every run exceeding 300 s as a timeout. For previously uncapped runs, including Direct, Direct-Feedback, and ReCoNav, this cutoff is applied retrospectively. No eventual success after the cutoff is credited.

Hard44 beyond the main time limit. The Hard44 analysis removes this extra limit while retaining each method's native resource bounds. It uses the complete predeclared pool of 44 hard-constraint-resolving tasks within the 178 common supported starts: 14 avoidance, 22 counting, and eight composite tasks. Full IS, Hard NS and completion divide by 44; Prefer divides by the four preference-containing tasks. Final budget failures and exceptions score zero. For existing methods, the 24 originally uncapped records are retained, including failures, and the other 20 tasks are uniformly rerun without the additional wall-time cap. No best-of selection is used. Direct, Direct-Feedback, and ReCoNav use their saved Qwen uncapped records on the same 44 task IDs. These extension results do not replace the original records in the 300 s main table.

Ablation accounting. Component ablations report quality conditional on normal completion; the paired Forced-Early comparison uses tasks completed by both variants. Their denominators are reported explicitly.

Model usage and time. Token and time means use completed runs. Token counts cover instrumented generative calls, counting reasoning tokens once; they exclude non-generative perception and policy computation. Uninstrumented usage is marked n/a. Shared upstream calls are attributed to each ablation that requires them.

C.3 Baselines and Runtime

Original-interface baselines use their released decision procedures with simulator and scoring adapters. The +ASG variants additionally receive scene entities, relations, and the navigation graph: NavGPT uses scene context and graph connectivity; GC-VLN uses ASG entities and geometry in its spatial constraint graph; Lang2LTL-2 and LIMP retain their specification and planning procedures

with the graph as an execution backend. These adapters preserve native task models and reasoning procedures; they do not introduce ReCoNav's Determination Engine.

Simulated agents have radius 0.25 m and height 1.50 m. Native interfaces use a size-matched Habitat navmesh; ASG-based methods use G_{nav}. The backends differ in discretization and support 178 and 192 exact starts, respectively. SNav and DualVLN retain their trained checkpoints. Lang2LTL-2 retains its T5 translator and embedding model.

External-method generative calls and ReCoNav's Qwen variants use Qwen3.8-27B-FP8 with thinking disabled, under separately configured inference runtimes. ReCoNav and its ablations allow two feedback/repair rounds, 90 s solver calls, and a 360 s binding-resolution budget. External-method services run on RTX 4090 GPUs; some methods share an inference endpoint, so queueing affects latency. External timing includes task initialization, whereas ReCoNav timing excludes service startup and evaluation. These differences limit cross-method latency conclusions.

Unresolved bindings. After the semantic repair budget, a remaining ambiguity uses a seeded choice among assignments not excluded by the available evidence. When preference minima are completely certified, this choice is restricted to tied minimizers. Forced-Early and ReCoNav share this execution fallback, which is recorded separately from rule-determined binding. The chosen assignment must still yield a feasible route.

C.4 Additional Ablation Results

All variants share the task population and GT-ASG within each backbone. Direct and Direct-Feedback use the same contract solver as ReCoNav; Direct-ID-Sequence predicts entity IDs and connects them geometrically without enforcing route clauses during connection, exposing the limitations of this task representation. Forced-Early removes global feasibility and preference queries for target determination after semantic and spatial matching, while retaining repair and the shared unresolved-binding fallback. No Verify/Repair removes the additional verification and evidence-guided repair stage, including the resulting revision and recomputation, while retaining parser validity and route-feasibility checks.

Table 7 retains all six variants across four backbones: ReCoNav achieves the highest Full IS while using fewer tokens than Direct and Direct-Feedback in each backbone. Together with the paired analysis in Table 4(b), these comparisons examine task modeling, propagation into target decisions, and verification with repair. Tables 8 and 9 report additional quality and latency metrics. Quality is conditional on completed runs; preference denominators include only completed tasks with explicit preferences. Differences in aggregate rates therefore do not by themselves count how many individual tasks were repaired.

Table 7: **Complete reasoning baseline and ablation results.** Full IS (%) and tokens (k) are measured on completed runs, without an added wall-time limit. Best Full IS and lowest token usage are bold.

	GPT		Gemini		Qwen		Gemma	
Variant	Full IS↑	Tokens↓	Full IS↑	Tokens↓	Full IS↑	Tokens↓	Full IS↑	Tokens↓
Direct	91.62	40.32	26.70	63.88	30.00	47.87	26.56	48.45
Direct-Feedback	96.35	44.01	58.64	149.21	45.79	109.71	50.00	109.94
Direct-ID-Sequence	86.91	31.13	82.11	45.64	42.63	42.15	55.03	42.40
Forced-Early	68.23	16.05	67.54	38.32	56.54	9.96	64.21	9.94
No Verify/Repair	98.96	**9.19**	96.35	**16.05**	73.82	**5.68**	93.19	**6.37**
ReCoNav	**100.00**	9.27	**98.44**	17.08	**82.11**	7.03	**96.34**	6.87

Table 8: **Additional ablation quality.** Hard NS and Prefer are percentages on completed runs; n is the preference-task denominator.

	GPT		Gemini		Qwen		Gemma	
Variant	Hard NS	Prefer (n)	Hard NS	Prefer (n)	Hard NS	Prefer (n)	Hard NS	Prefer (n)
Direct	93.19	79.63 (54)	27.75	18.52 (54)	30.00	25.93 (54)	28.13	25.93 (54)
Direct-Feedback	97.92	90.74 (54)	61.26	47.22 (54)	50.00	41.67 (54)	53.16	42.45 (53)
Direct-ID-Sequence	91.10	82.08 (53)	87.89	79.25 (53)	50.53	59.43 (53)	60.32	66.98 (53)
Forced-Early	81.25	48.15 (54)	79.58	49.06 (53)	67.54	43.52 (54)	75.79	45.28 (53)
No Verify/Repair	98.96	98.15 (54)	96.35	98.15 (54)	78.01	70.37 (54)	93.19	90.74 (54)
ReCoNav	100.00	100.00 (54)	98.44	100.00 (54)	86.32	75.93 (54)	96.34	94.44 (54)

Table 9: **Ablation completion and latency.** Completed counts use 192 issued tasks per variant; time averages use completed runs.

	GPT		Gemini		Qwen		Gemma	
Variant	Completed	Time (s)	Completed	Time (s)	Completed	Time (s)	Completed	Time (s)
Direct	191	52.5	191	87.6	190	112.7	192	82.0
Direct-Feedback	192	57.4	191	208.5	190	257.9	190	203.0
Direct-ID-Sequence	191	12.8	190	25.1	190	21.4	189	18.7
Forced-Early	192	56.9	191	158.8	191	154.2	190	88.4
No Verify/Repair	192	31.9	192	48.3	191	65.8	191	46.9
ReCoNav	192	32.1	192	52.4	190	96.0	191	52.7

C.5 Demonstration Setup

The real-scene example uses calibrated RGB images, mesh-derived depth, and camera poses. Regions and portals are manually annotated; object labels are predicted and the illustrated objects are checked against their RGB views. The specified contract produces a 165-waypoint, 14.53 m route that avoids the office. Four robot photographs correspond to the start, selected bin, laundry doorway, and washing machine. They document task locations; no synchronized pose log is available to quantify execution success.