LODGE: A CONVERSATIONAL TRAVEL AGENT BENCHMARK WITH GROUNDED GOALS AND CONTROLLABLE DIFFICULTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Benchmarks for conversational search agents typically assume a highly specific target: the user knows exactly which flight to buy or which hotel to book. Real users usually arrive with underspecified requests. Evaluating such agents therefore requires tasks whose correct outcome is known, with difficulty that can be varied deliberately rather than incidentally. We present LODGE, a benchmark of 360 conversational accommodation-booking tasks, 300 of them a fixed evaluation split on which all results here are reported, each with a unique best-fitting target listing established by contrastive construction and confirmed by human review, over a shared catalog of synthetic listings[1]. Difficulty is expressed as composable axes that vary one property of the task while the definition of success stays fixed. GEOFLEX varies whether the named destination is the target locality (PRECISE), an anchor to search beyond (NEARBY), or a region to refine into localities (BROAD); VAGUENESS varies how explicitly the same goal is disclosed. Across seven language agents, these axes separate capabilities that aggregate accuracy conflates: booking completion is nearly uninformative about task success, NEARBY is harder than BROAD for every agent despite supplying more locational information, and robustness to vagueness is independent of geographic competence. Small-scale supervised fine-tuning improves the difficult settings disproportionately. The dataset, user simulator, and evaluation harness will be released upon acceptance.

1 INTRODUCTION

A conversational search and discovery agent lets travelers freely describe the stay they want rather than fill in a destination, dates, a guest count and some tick boxes: "*a quiet mountain place somewhere in Tyrol for a week in February*", or "*somewhere within an easy drive of Kyoto with room to work*". That freedom shifts the burden onto the agent. A filter states intent exactly; a description leaves it to be worked out, while the tools underneath still expect a locality, dates and a guest count. The agent must then infer preferences the traveler leaves underspecified, turn a region or a rough neighborhood into a concrete locality, and commit to a single listing. It can go wrong at any of these steps and still finish with a completed booking, which is why a booking says little on its own about whether the traveler got what they wanted.

Benchmarks for interactive agents have established how to evaluate this class of system: simulate a user, give the agent tools, and score the resulting environment state (Yao et al., 2024; Barres et al., 2025). Travel-specific benchmarks add domain realism, covering itinerary construction under constraints (Xie et al., 2024), multi-turn tool-augmented planning (Cheng et al., 2025), and group settings with conflicting preferences (Cheng et al., 2026). What remains difficult is not running such evaluations but constructing tasks for them. Two properties are hard to obtain simultaneously: a *known-correct outcome*, so that a score means something more than a judge's opinion, and *controllable difficulty*, so that a benchmark can say *which* capability an agent lacks rather than only that it scored 77%.

[1] Containing no proprietary or user-derived content.

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.

Task collection usually forces a choice between them. Real traffic has realistic difficulty but no ground truth: what a user booked is a revealed choice, shaped by ranking and by what was surfaced rather than by what would have fit best, and the goal behind it is never recorded. Hand-written tasks have ground truth but difficulty that varies by author rather than by design, at a cost per task that keeps coverage narrow. LODGE takes a third route: tasks are synthesized so that the correct outcome holds by construction, and difficulty is introduced by composable *axes* that can be dialed independently without changing what counts as success.

Such a benchmark fills a gap in conversational agent research, particularly for studying how robust multi-turn systems — models and harnesses alike — are to varying difficulty, which is closer to how these agents behave in real search and discovery settings such as travel.

These are the major contributions of the paper.

- **An open, quality-controlled dataset.** We release 360 conversational accommodation-booking tasks over a shared catalog of 3,660 listings in 75 countries, each with a unique best-fitting target and $K = 8$ plausible alternatives (Appendix B.2 gives the catalog's composition). Because contrastive construction makes the target an input to generation rather than a label to be assigned, review cost does not grow with the catalog, so quality control scales with the data (Section 2.2). The dataset, user simulator and evaluation harness will be released upon acceptance, alongside a leaderboard (Reproducibility Statement).
- **Difficulty as goal-preserving axes.** Each axis varies one property of a task while the definition of success stays identical across levels, so a score difference is attributable to that property rather than to a changed criterion. VAGUENESS and a locality-cue ladder are matched task by task, holding the hidden goal, target and catalog fixed; GEOFLEX varies geographic granularity, which cannot be restated for a fixed target and is therefore controlled by balancing the set across levels. Further axes need no new data collection and no redefinition of success (Section 3).
- **Booking one correct listing, not grading a plan.** Where travel benchmarks evaluate *planning*, LODGE scores a single booking from the full catalog. Catalog commitment is itself established (Yao et al., 2022); what is new is that the target is uniquely correct by construction rather than scored by attribute overlap, and that the specification must be elicited from a simulated traveler rather than handed over — so the headline metric is binary and judge-independent. Around it sit graded listing fit, a trajectory-level geographic score and a four-way failure decomposition.
- **Findings the axes make visible.** The axes separate capabilities that aggregate accuracy conflates — booking completion is nearly uninformative about task success, a plausible but incorrect anchor is more damaging than no anchor, and vagueness robustness is independent of geographic competence — and small-scale fine-tuning improves the difficult settings disproportionately, indicating a trainable search behavior rather than a fixed model property.

2 THE LODGE BENCHMARK

2.1 TASK DEFINITION

A task is a multi-turn interaction between a user simulator and an evaluated agent over a deterministic accommodation environment. The simulator is conditioned on a *hidden goal* g — a natural-language description of the trip and the stay the traveler wants — which is never shown to the agent. The goal is fixed in advance and *grounded*: it is anchored to a particular listing, so it determines which listing is correct, and the agent's task is to recover enough of it to act. The agent has tools for searching listings, inspecting a listing's attributes, and creating a reservation, and must identify and book the stay that best satisfies g using only what the simulator says.

Each task carries a *target listing* ℓ^* that uniquely satisfies g, together with a set $\mathcal{A}$ of K alternatives that satisfy g only partially. The goal is generated conditioned on that contrastive set, so the target's status holds by construction rather than by later annotation. Every task's listings are then pooled into one shared catalog, and search runs against all of it rather than a per-task candidate set: each task contributes its target and alternatives disjointly from every other task's, and the catalog carries unreferenced filler listings so that it is not itself an answer key — a build-time gate drops any filler that would satisfy a goal. So ℓ^* competes against the whole catalog, not against $\mathcal{A}$ alone, and a

matched difficulty axis produces variants that leave g, ℓ^* and the catalog unchanged (Section 3). Appendix B.2 gives the composition and the filler gate, and Section 4.1 the released sizes.

The agent's problem. A task is a tuple

$$\tau = \big(g,\ \ell^*,\ \mathcal{A},\ s,\ v\big), \qquad \ell^* \in \mathcal{C}, \quad \mathcal{A} \subset \mathcal{C}, \quad |\mathcal{A}| = K,$$

where $\mathcal{C}$ is the shared catalog, s is the task's GEOFLEX level (Section 3.1) and v its VAGUENESS level (Section 3.2). **The agent observes none of these.** It sees only the simulator's utterances and the results of its own tool calls.

At each turn the agent chooses one action: reply to the traveler, search a named locality under trip constraints, inspect a listing, or create a reservation; a turn is one such action, and tool calls within a turn are not separately capped. An episode ends on the first of four conditions: the agent creates a reservation, the simulator signals that it would book or abandons, the simulator stalls for three consecutive turns without a productive intent, or the turn cap is reached — yielding a final outcome $r \in \mathcal{C} \cup \{\perp\}$, where $\perp$ denotes no booking. The primary metric is $\mathbf{1}[\, r = \ell^* \,]$.

The agent must therefore recover enough of g to discriminate ℓ^* from $\mathcal{A}$ — and from the rest of $\mathcal{C}$ — while inferring from s which localities to search at all, under a simulator whose disclosure is governed by v.

Progressively revealed goals. g is never presented to the agent in full, and the simulator does not volunteer it. The simulator decomposes g into sub-goals and pursues them one at a time, opening with a partial description and supplying further preferences as the conversation proceeds — in response to clarifying questions, and in reaction to listings the agent surfaces. The agent therefore never receives a specification to satisfy; it must elicit one while searching, and decide when it has enough to commit. The simulator speaks as a single fixed persona with no trip history, so conversational style is held constant across tasks rather than varied — one fewer quantity that could differ between levels and drive a score gap. This is what makes the tasks conversational rather than single-query, and it is the property VAGUENESS varies (Section 3.2): g and ℓ^* are fixed, while how readily the simulator discloses g changes.

2.2 CONSTRUCTION

Construction proceeds in four stages, designed so that the correct outcome is a property of the task rather than a later judgment; Figure 1 shows the pipeline and Appendix B gives the parameters.

(1) Geographic grounding. Localities, regions and coordinates come from `geonamescache`, a public GeoNames-derived gazetteer; each task's geographic configuration is recomputed from coordinates and administrative level, with no model labeling geography. Geographic units are allocated *without replacement*, so each task claims its search area alone. Because every goal states a destination, a listing belonging to another task lies outside that area and fails the goal on location however well its amenities match, so cross-task interference is excluded by construction.

(2) Listing synthesis. We synthesize ℓ^* and its K alternatives from predefined distributions over property type, capacity, amenities, price, ratings and booking policy. Target-distinguishing preferences are derived from ℓ^*, and no alternative may satisfy the complete set: ℓ^* is the unique full match, the alternatives deliberate partial matches.

(3) Contrastive goal generation. The hidden goal is generated conditioned on the contrastive set (Wei et al., 2026): rather than writing a goal and then labeling listings against it, we require a goal that ℓ^* satisfies and the alternatives do not, so the target's status holds by construction rather than post hoc. That prior formulation pairs one positive with one negative observed listing to produce a single query; the construction here uses synthesized listings, a one-to-many contrastive set, and a multi-turn goal (Section 5).

(4) Human review of goals. Every generated goal is reviewed by human annotators before its task enters the benchmark, against three failure modes: goals the target does not uniquely satisfy, goals that identify the target too directly, and goals that are not plausible traveler requests. Contrastive construction is what makes this affordable — the reviewer compares a goal against $K + 1$ known listings rather than searching the catalog, so review cost does not grow with the catalog. Together

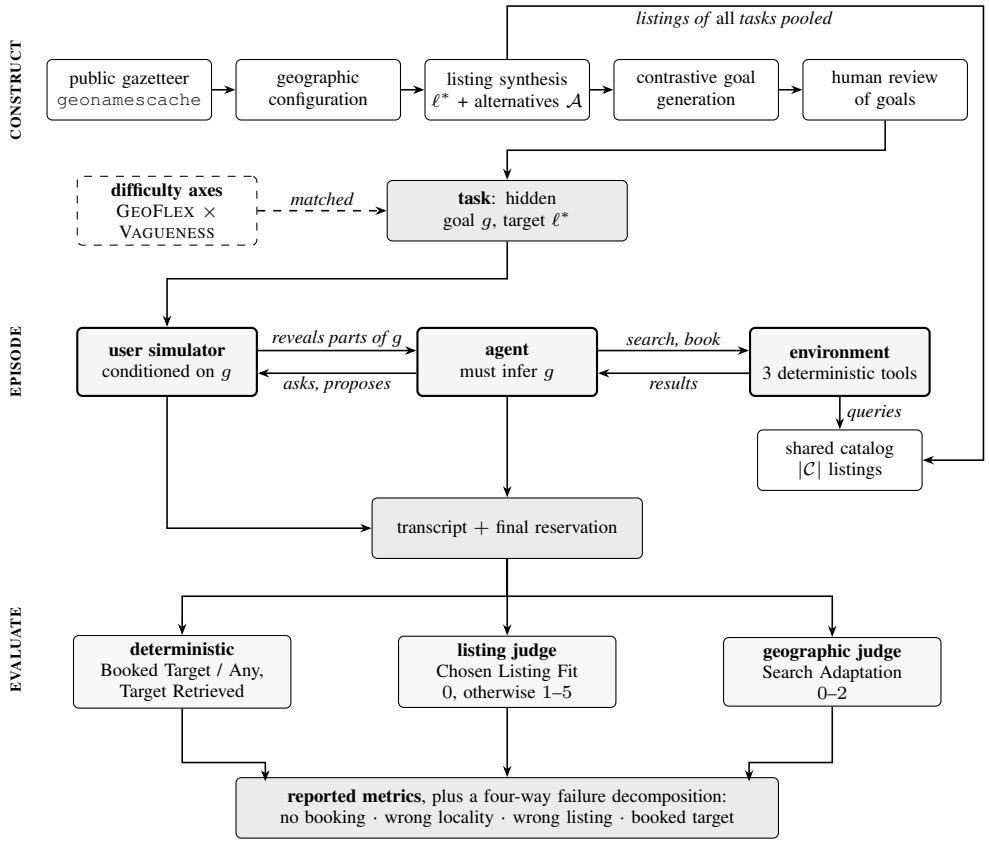

Figure 1: LODGE end to end. CONSTRUCT: geography comes from a public gazetteer; listings are synthesized as a contrastive set — one target ℓ^* against $|\mathcal{A}| = K$ partial matches — and the hidden goal g is generated conditioned on that set, then human-reviewed (Section 2.2). EPISODE: the simulator holds g and reveals fragments of it, never the goal itself; the agent reaches the catalog only through deterministic tools. EVALUATE: booking outcomes are read mechanically from the environment, while two judges independently score listing fit and geographic search behavior.

with disjoint geographic allocation, which covers every other task's listings, this makes the unique-target property a checked property of the released data rather than an assumption inherited from the generator. Appendix B.6 gives the interface, criteria and rates.

2.3 ENVIRONMENT AND TOOLS

Agents act through three deterministic tools: `search_listings`, which takes a concrete locality together with trip constraints and returns matching summaries; `get_listing_details`, which returns a listing's full attributes; and `create_reservation`, which succeeds only when a reservation is actually created in the environment. Booking outcomes are therefore mechanical and judge-independent.

The search interface operates at locality level by design: it accepts neither "search nearby" nor "search this region," and it does not return listings from localities inside a named region, so the agent must convert geographic intent into concrete locality queries. Committing to a locality is the decision we want to measure; Appendix E returns to the cost of that restriction.

2.4 METRICS

An episode yields a transcript, a sequence of locality queries $q_{1:m}$, and a final outcome $r \in \mathcal{C} \cup \{\perp\}$, where $\perp$ denotes no reservation. Let $R \subseteq \mathcal{C}$ be the set of listings *returned by the agent's search*

calls — whether or not the agent went on to mention them to the traveler. We report outcome- and process-level metrics, deterministic where possible.

Deterministic. Read from the environment, with no model in the loop:

$$\textsc{BookedTarget} = \mathbf{1}[\, r = \ell^* \,], \tag{1}$$

$$\textsc{BookedAny} = \mathbf{1}[\, r \neq \perp \,], \tag{2}$$

$$\textsc{TargetRetrieved} = \mathbf{1}[\, \ell^* \in R \,]. \tag{3}$$

TARGETRETRIEVED is therefore a property of the search trajectory rather than of the conversation: it records whether the environment ever handed the target to the agent. Equation 1 is the primary metric. Because ℓ^* is fixed by construction and r comes from the environment's reservation call, it involves no judgment.

Judged. Let $c \in \mathcal{C} \cup \{\varnothing\}$ be the *chosen* listing: the reserved listing r when one exists, otherwise a surfaced listing the simulator clearly selects or states an intention to book, and $\varnothing$ when neither holds. **Chosen Listing Fit** is

$$\textsc{Fit} = \begin{cases} 0, & c = \varnothing, \\ J_{\text{fit}}(c, g) \in \{1, \ldots, 5\}, & \text{otherwise}, \end{cases} \tag{4}$$

and reported means include the zeros. The judge sees g and c's attributes but not the identity of ℓ^*, so FIT credits good non-target choices rather than rewarding target identity a second time. **Geographic Search Adaptation** scores the query trajectory alone on a three-point scale,

$$\textsc{GeoAdapt} = J_{\text{geo}}(q_{1:m}, s) \in \{0, 1, 2\}, \tag{5}$$

rising with how coherently the agent executes the expansion or refinement that level s requires (Appendix C.1 gives the rubric). J_{geo} does not observe r, so GEOADAPT is defined even when the episode ends without a booking.

3 DIFFICULTY AXES

An axis is a controlled variation in one property of a task, arranged so that a score difference across levels is attributable to that property rather than to a change in what counts as success. This release instantiates two axes, GEOFLEX and VAGUENESS, plus a locality-cue ladder used to interpret GEOFLEX. They rely on two forms of control, which differ in what each can hold fixed.

VAGUENESS and the locality-cue ladder are *matched per task*: the same task appears at each level with hidden goal, target, alternatives and catalog identical, so only the expression changes. GEOFLEX is a *stratification*: a target locality lies in one administrative region and has one set of neighbors, so a single task cannot be re-expressed at all three levels — the area the user names differs by level, as Equation 6 in Appendix B.4 makes explicit. Control comes instead from balancing the evaluation set across levels, so comparisons are between matched *populations* of tasks. Both forms keep the definition of success identical across levels, which is what licenses attributing a difference to the property varied; only the matched axes do so task by task. Axes compose: a task instantiates a value on each.

3.1 GEOFLEX: GEOGRAPHIC FLEXIBILITY

Vague place names and mismatched granularity are long-standing problems in geographic information retrieval (Jones & Purves, 2008); GEOFLEX makes granularity a controlled task property by varying the relation between the destination the traveler states and the locality the target listing actually occupies. Each level is a condition on locality coordinates and administrative containment computed from a public gazetteer, not a heuristic difficulty label (Equation 6):

- PRECISE — the named locality *is* the target locality. "*A quiet place in Klagenfurt*": the agent should search Klagenfurt and stay there.

- NEARBY — the named locality is an *anchor*, and the target sits in a different locality in the same country within 50 km, possibly across a first-level administrative boundary. "*Somewhere within an easy drive of Hildesheim*": searching Hildesheim returns plausible listings but never the target, so the agent must decide to look outward and name somewhere it was not told.
- BROAD — the user names the first-level administrative region *containing* the target locality. "*A week somewhere in Carinthia*": no search for "Carinthia" is possible, because the tool takes a locality, so the agent must refine the region into concrete candidates — Klagenfurt, Villach, Velden — and choose among them.

Each level demands a different search operation rather than more of one operation: stay within the stated locality (PRECISE), expand beyond it (NEARBY), or narrow a region into candidate localities (BROAD). Because the three differ in kind, they carry no built-in ordering, and Section 4.2 finds that the empirical ordering is not the one task size would predict. Figure 2 in Appendix B.4 draws all three, alongside their statement as conditions on coordinates and administrative structure.

Locality-cue ladder. To separate *identifying* the right locality from *deciding* to look beyond the stated one, each NEARBY and BROAD task has matched variants that differ only in the geographic information carried by the hidden goal the user simulator holds (Appendix D.3 shows an example): **None** (the core task), **Direction** (a coarse compass cue toward the target), and **Locality** (the target locality named outright). Because the variants share geography, target and alternatives, the rungs differ in what the agent is told and in nothing else.

3.2 VAGUENESS: INTENT EXPLICITNESS

VAGUENESS varies how explicitly the user simulator communicates the same hidden goal. **L1** (moderate) has the user offer one or two concrete details early and reveal more as the conversation proceeds. **L2** (greater) has the user open with an underspecified description and supply concrete constraints mainly when the agent elicits them. The goal, target, geographic level, and catalog are identical across L1 and L2; only expression changes, so the axis isolates elicitation from every other capability.

Why axes rather than an aggregate label. Because the success criterion is identical across levels, a score difference is attributable to the property varied rather than to a change in what success means — and the axes compose, so two can be read together. Appendix B.5 argues this against the usual practice of grading difficulty by enlarging the task, and Section 4.2 supplies the sharpest evidence for it: a level ordering that runs opposite to what task size predicts.

4 EXPERIMENTS

We run seven language agents on the benchmark to answer four questions: whether the axes discriminate between agents; where agents fail, and whether the failure profile shifts with the level; whether the two axes are independent; and whether the capability they isolate can be improved by fine-tuning.

4.1 SETUP

Data. LODGE releases 360 tasks. Every result in this paper is reported on a fixed evaluation split of 300 of them — 100 per GEOFLEX level, spanning 63 countries and 300 distinct target localities — and the release labels that split explicitly, so each table can be reproduced from the artifact. Agents search a shared catalog of 3,660 listings across 1,073 cities in 75 countries. The remaining 60 tasks (20 per level) are published but not evaluated here; they include the country-disjoint split used in Appendix D.5, whose twelve countries occur nowhere else in the release and carry the benchmark's coverage from the evaluation split's 63 countries to 75. A figure computed over all 360 tasks is therefore not comparable to those reported here.

No task in the 300-task evaluation split, and no cue variant of such a task (Appendix D.3), appears in any fine-tuning pool; Appendix D.5 gives the pair-level verification.

Table 1: Overall agent performance under VAGUENESS = L1. Booked Target and Booked Any are reported as percentages. Chosen Listing Fit is 0 when no listing is clearly chosen and 1–5 otherwise; reported means include the zeros. Geographic Search Adaptation is 0–2. $n = 300$ tasks (100 per GEOFLEX level), three repeats per task.

Agent	Booked Target ↑	Booked Any ↑	Chosen Listing Fit ↑	Geographic Search Adaptation ↑
Qwen3.5-35B-A3B	70.8±0.4	92.8±0.4	3.93±0.04	1.61±0.01
Qwen3.6-35B-A3B	70.4±1.8	88.3±0.9	3.90±0.03	1.59±0.03
Qwen3.8-27B	71.8±1.8	**99.7**±0.0	4.03±0.02	1.59±0.01
Haiku 4.5	67.0±1.9	97.4±1.2	3.88±0.07	1.63±0.03
GPT-4.1 Mini	66.4±2.0	89.8±0.7	3.81±0.06	1.63±0.00
GPT-5 Mini	71.3±1.1	96.0±0.7	4.06±0.01	1.71±0.02
GPT-5	**77.4**±0.8	94.2±1.1	**4.16**±0.06	**1.79**±0.01

Table 2: Booked Target (%) by GEOFLEX level under VAGUENESS = L1. Every agent scores lower on NEARBY than on BROAD, although the BROAD goal states only a region where the NEARBY goal names a town near the target. $n = 300$ tasks (100 per GEOFLEX level), three repeats per task.

Agent	PRECISE	NEARBY	BROAD
Qwen3.5-35B-A3B	93.3±0.6	54.0±3.0	65.0±2.6
Qwen3.6-35B-A3B	94.0±1.7	53.3±6.5	64.0±1.7
Qwen3.8-27B	**96.7**±3.2	50.0±1.7	68.7±3.5
Haiku 4.5	93.7±0.6	45.7±5.1	61.7±0.6
GPT-4.1 Mini	92.7±3.5	48.1±3.4	58.2±1.3
GPT-5 Mini	86.0±2.6	56.7±3.1	71.2±2.7
GPT-5	91.7±0.6	**66.3**±1.2	**74.3**±2.1

Agents and protocol. We evaluate seven agents: three consecutive generations of one open-weights family (Qwen3.5-35B-A3B, Qwen3.6-35B-A3B and the dense Qwen3.8-27B), Claude Haiku 4.5, GPT-4.1 Mini, GPT-5 Mini, and GPT-5. All agents use the same deterministic environment, up to 20 turns per episode, and three independent runs per task; we report mean±std over runs. GPT-5 Mini serves as user simulator and as judge in every agent condition, so those roles do not vary across the comparison; Appendix E addresses the self-preference risk this creates. Unless stated otherwise, results are under VAGUENESS = L1 and the **None** rung of the locality ladder.

One provider asymmetry bears stating: six of the seven agents run through non-moderated routes, while GPT-4.1 Mini has none, and its provider's content filter rejects 0.28% of its episodes, which are excluded from its means rather than scored as failures. Appendix C.1 reports usable episode counts per agent and condition.

4.2 RESULTS

The axes discriminate. Of the four headline metrics in Table 1, three place GPT-5 at the top — Booked Target, Chosen Listing Fit and Geographic Search Adaptation — and all three improve monotonically within the GPT family. Booked Any is the exception, and it inverts the ordering: Qwen3.8-27B leads at 99.7% and GPT-5 falls below GPT-5 Mini.

Booking completion is nearly uninformative about success. The gap between Booked Any and Booked Target is large for every agent and does not narrow with capability. Qwen3.8-27B books in 99.7% of episodes — more than any other agent — yet books the listing the traveler wanted in 71.8%, and Haiku 4.5 books in 97.4% while reaching 67.0%, 10.4 points below GPT-5, which books *less* often than either. A booking-completion metric would rank these three almost identically while their outcomes differ by 10.4 points.

NEARBY is harder than BROAD, for every agent. Broken down by GEOFLEX level in Table 2, the agents cluster within a 10.7-point band on PRECISE (86.0–96.7%) and fan out on the levels requiring expansion or refinement. The ordering across levels is what is notable: every agent scores

lower on NEARBY than on BROAD, although BROAD supplies strictly less locational information — a region rather than a town. A plausible but incorrect anchor is more damaging than no anchor, presumably because a named locality that returns plausible results suppresses the decision to look elsewhere. An aggregate label would have ordered them the other way.

Failures shift from choosing to locating. Partitioning every episode by two questions — did the agent book at all, and if so was the booking in the target's locality (Appendix D.1) — shows the profile inverting with the level: under PRECISE wrong-locality errors are nearly absent, while under NEARBY they are the leading failure mode for every agent — reaching 44.3% for Haiku 4.5 — and under BROAD for most. Across three generations of one family the headline metric is flat while booking completion rises sharply, so capability gains do not reach geographic search (Appendix D.4). The matched locality-cue ladder (Appendix D.3) isolates the bottleneck: naming the target locality raises Booked Target by 11–38 points while a directional cue is unreliable, and the gain is in discovery rather than selection — retrieval climbs steeply across the rungs while booking conditional on retrieval stays flat. What limits these agents is finding the locality at all; once the target is in front of them, they book it.

The axes are independent, and capability does not buy robustness. Varying VAGUENESS with everything else fixed (Appendix D.2) degrades outcome quality for every agent while leaving Geographic Search Adaptation essentially untouched ($|\Delta| \leq 0.08$): added vagueness degrades the ability to elicit and act on listing preferences, not geographic search behavior, so an agent can keep searching well while choosing worse. The penalty bears no tidy relation to capability. Booked Target falls by between 3.0 and 14.5 points, the two largest belonging to the strongest agent and to one ranked fourth of seven, and GPT-5 leads at L1 but falls to fourth at L2, so the L1 ordering does not survive the axis. Booking completion shows the mechanism: GPT-5's Booked Any falls from 94.2% to 77.4±4.2%, so vagueness does not make an agent choose badly so much as make it decline to choose.

The capability can be trained. Fine-tuning Qwen3.6-35B-A3B with LoRA on demonstrations curated from its own rollouts (Appendix D.5) concentrates its gains where the axes say difficulty lies: overall Booked Target rises 3.8 points, with 6.7 on NEARBY and 3.7 on BROAD against 1.0 on PRECISE, and weighting the graded process metrics in the curation rule adds nothing over selecting on outcome alone. That so few demonstrations move the hard levels indicates the axes have isolated something an agent can be taught, rather than a ceiling set by the base model. We report the study on one base model; agentic RL on travel planning reports gains of a similar shape (Ning et al., 2026).

5 RELATED WORK

Interactive agent benchmarks established the evaluation of goal-directed dialogue (Budzianowski et al., 2018), the interleaving of reasoning with tool actions (Yao et al., 2023), and the scoring of a full user–agent–environment loop from resulting database state (Yao et al., 2024; Barres et al., 2025); travel-specific work adds domain realism in constrained itinerary construction (Xie et al., 2024), multi-turn tool use with preference elicitation from real user queries (Cheng et al., 2025), group planning (Cheng et al., 2026) and multi-dimensional plan quality over a realistic sandbox (Chen et al., 2026), with further variants for retrieval-augmented spatiotemporal planning (Ni et al., 2025), fine-grained plan rewards (Qu et al., 2025) and open-ended Chinese itineraries (Shao et al., 2024). All evaluate *planning*: assembling or grading an itinerary against constraints, checkable without any single answer being correct. TravelBench scores elicitation, but by LLM judgment rather than against a known-correct choice. LODGE evaluates the *commitment*: one listing out of a full catalog, measured against the stay the traveler wanted. That forces the construction problem, since booking evaluation needs a known-best option where constraint satisfaction does not. Commitment to a catalog item is not itself new: WebShop (Yao et al., 2022) has an agent search 1.18M products and purchase one against a target specification. But its instruction is *static*, so there is nothing to elicit, and its reward is attribute overlap, so no single item is *the* correct one; nor does it vary one property of a fixed task. What LODGE adds to commitment-style evaluation is therefore the elicited specification and the uniquely correct outcome, not the act of committing — a position Table 3 sets alongside the benchmarks above.

Table 3: LODGE against related interactive and travel benchmarks. *Task-varying* difficulty changes what counts as correct with the level; *goal-preserving* holds the success criterion fixed. *Turns only*: multi-turn, but reference data in context rather than tools. *Geographic axis*: destination granularity varied as an independent property, *confounded* where it co-varies with another quantity and *partial* where only some levels vary it.

		Multi-turn + tools	Ground-truth source	Difficulty control	Geographic axis
General interactive agent benchmarks					
τ-bench	2024	✓	annotated goal state	—	—
τ^2-bench	2025	✓	goal state + assertions	task-varying	—
Travel benchmarks					
TravelPlanner	2024	—	constraint script	task-varying	confounded
Flex-TravelPlanner	2025	turns only	constraint script	goal-pres. (order)	—
TravelBench	2025	✓	LLM judge	—	—
GroupTravelBench	2026	✓	hidden preference tables	task-varying	partial
TravelEval	2026	—	rules + simulation	task-varying	—
LODGE-Bench	2027	✓	unique target by construction	**goal-preserving**	✓

Graded difficulty itself is not new: τ^2-bench composes atomic subtasks under "controlled complexity," TravelPlanner grades by hard-constraint count and trip duration, GroupTravelBench by group size and destination count, and TravelEval tiers queries by preference count. Several also vary persona, where LODGE holds the simulator's persona fixed so that it cannot contribute to a level difference. What differs is *how* difficulty is varied: in each of those, a harder level enlarges the task, so *what counts as a correct answer changes with the level* and a score gap mixes added difficulty with an altered success criterion. In LODGE the criterion is identical at every level — book the one listing that uniquely satisfies the goal — however the level is realized: held per task for VAGUENESS and the cue ladder, and across balanced task populations for GEOFLEX, which cannot be restated for a fixed target (Section 3). The closest precedent is Flex-TravelPlanner (Oh et al., 2025), which fixes a TravelPlanner constraint set and varies only the *order* in which constraints arrive. That is goal-preserving in our sense, though its constraints are stated explicitly and supplied in context rather than through tools, and the object graded is a plan. What the distinction buys is empirical rather than definitional: it is what allows a score gap to name a capability, lets two axes be read independently when they order agents differently, and exposes a level ordering that runs opposite to what task size predicts (Appendix B.5 and Section 4.2). Nor, to our knowledge, does any benchmark vary the *granularity* of the stated destination as an independent property; TravelPlanner comes closest but ties it to trip duration. On ground truth, LLM-generated queries and relevance labels are established in low-resource retrieval (Bonifacio et al., 2022; Dai et al., 2023; Zheng et al., 2023), and closest in method, Wei et al. (2026) pair contrastive listings and generate the query conditioned on the pair so that the label holds by construction. That principle is applied here to benchmark construction rather than model training (Section 2.2), to our knowledge first in an *interactive* setting, where it must survive a multi-turn conversation. Vatanparvar et al. (2026) takes the opposite side, scoring conversation-agnostic rubrics with no answer key; we accept a fixed artifact in exchange for a mechanical headline metric.

6 CONCLUSION

We presented LODGE, a benchmark of conversational accommodation-booking tasks in which the correct outcome holds by construction and difficulty is varied along composable axes. Contrastive goal generation with human review makes the unique-target property a checked property of the released data rather than an assumption, and the axes make score differences attributable to a named task property. Across seven agents they separate what aggregate accuracy conflates — booking completion from task success, anchor quality from anchor presence, vagueness robustness from geographic competence — and show the geographic behavior to be trainable from a small set of curated demonstrations. We see the axis framework and grounded goals, more than the present task set, as the durable contribution; Appendix E gives the limitations and future work.

Reproducibility Statement

The benchmark contains no proprietary or user-derived data: geographic structure is drawn from the public `geonamescache` gazetteer, and all listing content and hidden goals are model-generated from structured attributes. The full dataset, the user simulator, and the evaluation harness will be released upon acceptance, alongside a leaderboard in the same repository that records each reported result together with the protocol used to produce it.

Appendix B gives the construction parameters and the human-review protocol; Appendix C gives the evaluation harness settings (20-turn cap, three runs per task, instruction-only judging, robust means excluding errored episodes) and the exact model identifiers and decoding settings for every agent, simulator, and judge; Appendix D.5 gives the fine-tuning configuration and hardware; Appendix F reproduces the simulator, judge, and agent prompts verbatim.

Dataset split and demonstration curation are seeded and deterministic. Listing and goal generation invoke a language model and are therefore not bit-reproducible, which is why we release the generated dataset itself rather than only the means of producing it: exact comparison across papers requires the fixed artifact. Releasing the user simulator and the evaluation harness alongside it means the full evaluation loop — tasks, simulated user, tools, and judges — can be reproduced as run here.

Use of Large Language Models

Large language models are used in this work in two distinct capacities: as components of the benchmark itself, and as a writing aid.

As components of the benchmark. Model generation is part of the method rather than an incidental tool, and appears at three points. **Synthetic data generation**: listing content and the hidden goal of every task are produced by a language model from structured attributes (Section 2.2), with generated goals subsequently reviewed by human annotators (Section 2.2). **User simulation**: the simulated traveler in every episode is a language model conditioned on that task's hidden goal (Section 2.1). **Evaluation**: Chosen Listing Fit and Geographic Search Adaptation are assigned by language-model judges (Section 2.4); the booking metrics are read from the environment and involve no model judgment. Exact model identifiers and decoding settings for all three roles are given in Appendix C.1, and the prompts are reproduced verbatim in Appendix F: the simulator in Appendices F.1–F.4, the two judges in Appendices F.5 and F.6, and the agent's system prompt and tool schemas in Appendices F.7 and F.8. Appendix E discusses the risks this creates, including the self-preference concern that arises from a shared model family across simulator, judge, and one evaluated agent.

As a writing aid. A large language model was used to aid and polish the writing of this paper: improving clarity and concision, tightening phrasing, and checking consistency of terminology and notation across sections. The authors directed this process, verified every resulting claim against the experimental record, and are responsible for all content. The research questions, the benchmark design, the experiments, and the results are the authors' own.

References

Ayomide Abayomi, Jose G. Moreno, Karim Radouane, and Lynda Tamine. GeoBenchmark: Probing large language models for geo-spatial knowledge. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pp. 5335–5348. ELRA Language Resource Association, 2026. doi: 10.63317/26pwz3584735.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. InPars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*, 2022.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz – a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of EMNLP*, 2018.

Weiyi Chen, Shuaixiong Wang, Ziyun Gao, Kaichun Hu, Wangze Ni, Shimin Di, Chen Jason Zhang, and Lei Chen. TravelEval: A comprehensive benchmarking framework for evaluating LLM-powered travel planning agents. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. doi: 10.1145/3770855.3817533.

Xiang Cheng, Yulan Hu, Xiangwen Zhang, Lu Xu, Lide Tan, Zheng Pan, Xin Li, and Yong Liu. Beyond itinerary planning: A real-world benchmark for multi-turn and tool-using travel tasks. *arXiv preprint arXiv:2512.22673*, 2025.

Xiang Cheng, Yulan Hu, Lulu Zheng, Zheng Pan, Xin Li, and Yong Liu. Grouptravelbench: Benchmarking llm agents on multi-person travel planning. *arXiv preprint arXiv:2605.25200*, 2026.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. In *International Conference on Learning Representations*, 2023.

Christopher B. Jones and Ross S. Purves. Geographical information retrieval. *International Journal of Geographical Information Science*, 22(3):219–228, 2008. doi: 10.1080/13658810701626343.

Hang Ni, Fan Liu, Xinyu Ma, Lixin Su, Shuaiqiang Wang, Dawei Yin, Hui Xiong, and Hao Liu. TP-RAG: Benchmarking retrieval-augmented large language model agents for spatiotemporal-aware travel planning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12403–12429. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.626.

Yansong Ning, Rui Liu, Jun Wang, Kai Chen, Wei Li, Jun Fang, Kan Zheng, Naiqiang Tan, and Hao Liu. DeepTravel: An end-to-end agentic reinforcement learning framework for autonomous travel planning agents. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. doi: 10.1145/3770855.3818359.

Juhyun Oh, Eunsu Kim, and Alice Oh. Flex-TravelPlanner: A benchmark for flexible planning with language agents. *arXiv preprint arXiv:2506.04649*, 2025.

Yincen Qu, Huan Xiao, Feng Li, Gregory Li, Hui Zhou, Xiangying Dai, and Xiaoru Dai. TripScore: Benchmarking and rewarding real-world travel planning with fine-grained evaluation. *arXiv preprint arXiv:2510.09011*, 2025.

Jie-Jing Shao, Bo-Wen Zhang, Xiao-Wen Yang, Baizhi Chen, Si-Yu Han, Wen-Da Wei, Guohao Cai, Zhenhua Dong, Lan-Zhe Guo, and Yu-Feng Li. ChinaTravel: An open-ended benchmark for language agents in chinese travel planning. *arXiv preprint arXiv:2412.13682*, 2024.

Korosh Vatanparvar, Ashutosh Joshi, Maria Xenochristou, Mohammad Abuzar Hashemi, Prasad Kasu, Deepak Bansal, Daniel Lopez-Martinez, Anchal Nema, Ramya Ganesan, Will Kimbrough, Alex Woody, Yadunandana Rao, Dilek Hakkani-Tür, and Wilko Schulz-Mahlendorf. PatientAgentBench: A benchmark framework for evaluating patient-facing health AI agents. *arXiv preprint arXiv:2607.25485*, 2026.

Wendy Ran Wei, Hao Li, Weiwei Guo, Xiaowei Liu, Xueyin Chen, Chun How Tan, Malay Haldar, Soumyadip Banerjee, Kedar Bellare, Huiji Gao, Stephanie Moyerman, and Sanjeev Katariya. CASTLE: Contrastive and seed-guided training for cold-start natural language search. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26)*, Rome, Italy, 2026. doi: 10.1145/3799682.3841071.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A EXAMPLE EPISODES

Section 2 defines a task; this section shows three episodes, so that the rest of the appendix can be read against concrete episodes rather than definitions. All three are the same BROAD task, which makes the comparison within-task: a success, the failure mode that dominates under flexible geography, and the same task under VAGUENESS = L2.

The task. The guest names the region *Diana, Madagascar*; the target is a loft private room in *Antsiranana*, the region's principal city, which the guest never names.

Each block's header gives the agent, the VAGUENESS level and the repeat index, and the source listing at the top of this section names the preds asset for each, so every turn is traceable to a stored episode. Turn indices are stored message positions. Tool results are truncated for width and typographic quotes and dashes normalized to ASCII; nothing else is altered.

```
HIDDEN GOAL (not shown to the agent):
  I'm planning a short June trip for two adults in Diana, Madagascar, and I want a
comfortable place to stay from the 7th to the 11th. A loft or private room would feel just
right--something cozy but not a whole apartment. It should have air-conditioning and
reliable Wi-Fi, and I'm willing to pay roughly $100-150 per night. I'd also like the option
 to cancel flexibly and don't mind sending a request rather than booking instantly.
```

A successful BROAD episode: refinement by knowledge. GPT-5 searches `city="Diana"`, gets nothing, and on its own initiative searches Antsiranana, telling the guest it is the principal city of the Diana region. No elicitation occurs: the agent supplies the locality from parametric knowledge. This is one of two routes to a correct BROAD episode; in others the agent instead offers the traveler a short list of candidate towns and adopts the one chosen, succeeding by elicitation rather than by knowledge. Refinement therefore has two distinct implementations rather than one, and BOOKEDTARGET does not distinguish them — which is the sense in which the locality-cue ladder of Appendix D.3 collapses both: naming the locality removes the knowledge route and the elicitation route at once.

```
GPT-5 / L1 / repeat 0   booked_target=1  booked_any=1  target_location_match=1
target_retrieved=1
------------------------------------------------------------------------------------------
------
[01] GUEST    Diana, Madagascar, June 7 to 11 for two adults.
[02] AGENT -> search_listings({"checkin": "2027-06-07", "checkout": "2027-06-11", "guests":
 2, "city": "Diana"})
[03] TOOL     0 results
[04] AGENT -> search_listings({"checkin": "2027-06-07", "checkout": "2027-06-11", "guests":
 2, "city": "Antsiranana"})
[04] AGENT -> search_listings({"checkin": "2027-06-07", "checkout": "2027-06-11", "guests":
 2, "city": "Nosy Be"})
[05] TOOL     [{"listing_id": "listing_1137220276627904457", "title": "Cozy seaside
apartment with Wi-Fi and parking in Antsiranana",...
[06] TOOL     0 results
[07] AGENT -> get_listing_details({"listing_id": "listing_1137220276627904457"})
[07] AGENT -> get_listing_details({"listing_id": "listing_1162912026652926034"})
[07] AGENT -> get_listing_details({"listing_id": "listing_1438416796315817856"})
[08] TOOL     {"listing_id": "listing_1137220276627904457", "listing_details": {"title": "
Cozy seaside apartment with Wi-Fi and parki...
[09] TOOL     {"listing_id": "listing_1162912026652926034", "listing_details": {"title": "
Cozy cabin with private bathroom and Wi-Fi ...
[10] TOOL     {"listing_id": "listing_1438416796315817856", "listing_details": {"title": "
Cozy loft private room with AC and Wi-Fi in...
[11] AGENT    Great choice -- here are a few comfortable options in Antsiranana (Diego
Suarez) in the Diana region for June 7-11, for two adults: - Cozy seaside apartment (entire
 place) -- 1 bedroom, kitchen, Wi-Fi, washer, workspace, free parking. Rating 5.0. Flex...
[12] GUEST    Show me the full details for the loft private room (the AC/Wi-Fi one): photos
, nightly price and total for June 7-11, and the cancellation policy, please.
[13] AGENT    Here are the details for the loft private room with AC and Wi-Fi in
Antsiranana (Diana), for June 7-11: - Title: Cozy loft private room with AC and Wi-Fi in
Antsiranana - Location: Antsiranana, Madagascar (Diana region) - Space: Private room in a
lo...
[14] GUEST    Yes -- please send a booking request for the loft private room in Antsiranana
, June 7-11, for two adults; I'm happy to book with host approval and the flexible
cancellation.
[15] AGENT -> create_reservation({"listing_id": "listing_1438416796315817856", "checkin":
"2027-06-07", "checkout": "2027-06-11", "guests": 2})
```

```
[16] TOOL     {"reservation_id": "reservation_000001", "listing_id": "
listing_1438416796315817856", "checkin": "2027-06-07", "checkou...
[17] AGENT    Your booking request is in -- you're all set pending host approval.
Reservation details: - Loft private room, Antsiranana (Diana), Madagascar - Dates: June
7-11 (4 nights) - Guests: 2 adults - Nightly rate: $118.92 - Accommodation subtotal: $475
.68 (...
```

A wrong-locality failure on the same task. GPT-4.1 Mini searches Diana, finds nothing, and asks the guest for nearby areas. The guest guesses "Nosy Be or Ambanja" — neither is the target locality — and the agent books in Ambanja. The wrong locality is introduced by the *guest's* guess rather than chosen by the agent, and TARGETRETRIEVED $= 0$: the target is never surfaced at all, so no listing comparison could have recovered it. The episode nonetheless terminates in a confirmed reservation and a summary that reads as a satisfied request. Nothing in the transcript signals the error to either party, which is the concrete form of the dissociation Section 4.2 reports between Booked Any and Booked Target, and why wrong-locality errors dominate under flexible geography (Appendix D.1). Note also that this agent issues 2024 dates against a 2027 task; the environment enforces no calendar constraints, so the error is inert here, but it is the behavior Appendix E notes as an unstated year.

```
GPT-4.1 Mini / L1 / repeat 2   booked_target=0  booked_any=1  target_location_match=0
target_retrieved=0
------------------------------------------------------------------------------------------
------
[01] GUEST    Diana, Madagascar June 7-11.
[02] AGENT -> search_listings({"city": "Diana", "checkin": "2024-06-07", "checkout":
"2024-06-11"})
[03] TOOL     0 results
[04] AGENT    I couldn't find any listings in Diana, Madagascar for June 7-11. Would you
like me to search nearby areas or adjust the dates?
[05] GUEST    Yes -- please search nearby areas like Nosy Be or Ambanja for the same dates,
 June 7-11.
[06] AGENT -> search_listings({"city": "Nosy Be", "checkin": "2024-06-07", "checkout":
"2024-06-11"})
[06] AGENT -> search_listings({"city": "Ambanja", "checkin": "2024-06-07", "checkout":
"2024-06-11"})
[07] TOOL     0 results
[08] TOOL     [{"listing_id": "listing_1135813686200836446", "title": "Cozy air-conditioned
 apartment with workspace in Ambanja", "ci...
[09] AGENT    I found two listings in Ambanja for June 7-11: 1. Cozy air-conditioned
apartment with workspace, 1 bedroom, for 2 guests at $115.97 per night, rated 5.0. 2. Cozy
air-conditioned apartment with Wi-Fi, 1 bedroom, for 3 guests at $193.79 per night, rat...
[10] GUEST    Show me details for the first one -- the $115.97/night Ambanja apartment. I'd
 like photos, bed setup, cancellation policy, and whether booking is instant or request-
only.
[11] AGENT -> get_listing_details({"listing_id": "listing_1135813686200836446"})
[12] TOOL     {"listing_id": "listing_1135813686200836446", "listing_details": {"title": "
Cozy air-conditioned apartment with workspa...
[13] AGENT    The $115.97/night apartment in Ambanja is an entire place with 1 bedroom
featuring a comfortable queen-size bed, 1.5 bathrooms, and a modern kitchen. It includes
air conditioning, high-speed Wi-Fi, a flat-screen TV, a washer, and a dedicated workspa...
[14] GUEST    Looking for a private room or loft with AC and reliable Wi-Fi in Ambanja or
Nosy Be for June 7-11, budget about $100-150/night.
[15] AGENT -> search_listings({"city": "Ambanja", "checkin": "2024-06-07", "checkout":
"2024-06-11", "guests": 1, "max_price_per_night": 150, "room_type": "private_room", "
required_amenities": ["air_conditioni...)
[15] AGENT -> search_listings({"city": "Nosy Be", "checkin": "2024-06-07", "checkout":
"2024-06-11", "guests": 1, "max_price_per_night": 150, "room_type": "private_room", "
required_amenities": ["air_conditioni...)
[16] TOOL     0 results
[17] TOOL     0 results
[18] AGENT    I couldn't find any private rooms or lofts with AC and reliable Wi-Fi in
Ambanja or Nosy Be within your $100-150/night budget for June 7-11. Would you like me to
expand the budget or consider entire places instead?
[19] GUEST    Okay -- show me entire places with AC and reliable Wi-Fi in Ambanja or Nosy
Be for June 7-11, $100-150/night.
[20] AGENT -> search_listings({"city": "Ambanja", "checkin": "2024-06-07", "checkout":
"2024-06-11", "guests": 1, "max_price_per_night": 150, "room_type": "entire_place", "
required_amenities": ["air_conditioni...)
[20] AGENT -> search_listings({"city": "Nosy Be", "checkin": "2024-06-07", "checkout":
"2024-06-11", "guests": 1, "max_price_per_night": 150, "room_type": "entire_place", "
required_amenities": ["air_conditioni...)
[21] TOOL     [{"listing_id": "listing_1135813686200836446", "title": "Cozy air-conditioned
 apartment with workspace in Ambanja", "ci...
[22] TOOL     0 results
```

```
[23] AGENT    In Ambanja, there's one entire place with AC and Wi-Fi for $115.97 per night:
 a cozy air-conditioned apartment with a workspace, 1 bedroom, and a 5.0 rating. There are
no listings matching these criteria in Nosy Be. Would you like to book this apart...
[24] GUEST    Book the Ambanja apartment (cozy 1-bed AC entire place at $115.97/night) for
two adults, June 7-11.
[25] AGENT -> create_reservation({"listing_id": "listing_1135813686200836446", "checkin":
"2024-06-07", "checkout": "2024-06-11", "guests": 2})
[26] TOOL     {"reservation_id": "reservation_000001", "listing_id": "
listing_1135813686200836446", "checkin": "2024-06-07", "checkou...
[27] AGENT    The cozy 1-bedroom air-conditioned apartment in Ambanja is booked for two
adults from June 7 to 11. The total cost is $463.88. If you need anything else, just let me
 know!
```

Proactive adaptation versus user redirection. Read together, the first two episodes show the same opening — a region search that returns nothing — resolved in opposite ways. GPT-5 supplies the locality itself and proceeds; GPT-4.1 Mini hands the problem back to the traveler and adopts the locality the traveler guesses. This is a behavioral account of the wrong-locality column of Appendix D.1 rather than another instance of it: the failure is not that the agent chose badly among localities, but that it did not treat choosing one as its own job. Two episodes on one task suggest the pattern rather than establish it, and we report it as what these transcripts show; the locality-cue ladder (Appendix D.3) is the measured version of the same question, since naming the locality removes the decision altogether.

The same task under L2: abandonment as derailment. Under L2 the same agent reaches the right locality and surfaces the target (TARGETLOCATIONMATCH $= 1$, TARGETRETRIEVED $= 1$), then books nothing. The mechanism is visible in the guest's turn: "Please go ahead and submit a booking request… Before you send it, please confirm the nightly rate and the total cost." GPT-5 re-fetches the listing, reports the costs, and stops — having satisfied the conditional clause, it treats that as the request. Qwen3.6-35B-A3B fails identically on this task. The L2 penalty is therefore not only a failure to identify a listing: hedged conditional phrasing derails the commitment itself, so a share of the abandonment reported in Section 4.2 is derailment rather than refusal.

```
GPT-5 / L2 / repeat 2   booked_target=0  booked_any=0  target_location_match=1
target_retrieved=1
------------------------------------------------------------------------------------------
------
[01] GUEST    Feeling like a short, relaxed escape -- planning a cozy stay in Diana,
Madagascar from June 7 to June 11.
[02] AGENT -> search_listings({"city": "Diana", "checkin": "2027-06-07", "checkout":
"2027-06-11"})
[03] TOOL     0 results
[04] AGENT    Great choice for a relaxed escape. Diana is a region, so I'll search by a
specific town. Would you like to stay in: - Antsiranana (Diego Suarez) on the mainland, or
- Nosy Be island --or another town in Diana? Also, how many guests will be staying? I...
[05] GUEST    Really looking forward to a chill, low-key stay. Antsiranana (Diego Suarez)
on the mainland is perfect, and it'll be just the two of us. June 7-11, 2027 works -- go
ahead and search with those details.
[06] AGENT -> search_listings({"city": "Antsiranana", "checkin": "2027-06-07", "checkout":
"2027-06-11", "guests": 2})
[07] TOOL     [{"listing_id": "listing_1137220276627904457", "title": "Cozy seaside
apartment with Wi-Fi and parking in Antsiranana",...
[08] AGENT -> get_listing_details({"listing_id": "listing_1137220276627904457"})
[08] AGENT -> get_listing_details({"listing_id": "listing_1162912026652926034"})
[08] AGENT -> get_listing_details({"listing_id": "listing_1438416796315817856"})
[09] TOOL     {"listing_id": "listing_1137220276627904457", "listing_details": {"title": "
Cozy seaside apartment with Wi-Fi and parki...
[10] TOOL     {"listing_id": "listing_1162912026652926034", "listing_details": {"title": "
Cozy cabin with private bathroom and Wi-Fi ...
[11] TOOL     {"listing_id": "listing_1438416796315817856", "listing_details": {"title": "
Cozy loft private room with AC and Wi-Fi in...
[12] AGENT    Here are a few cozy options in Antsiranana for June 7-11, 2027 (2 guests): 1)
 Cozy seaside apartment -- entire place - 1 bedroom, kitchen, Wi-Fi, TV, washer, workspace,
 free parking; quiet neighborhood - Instant book; flexible cancellation - $230.62/...
[13] GUEST    Feeling ready to lock this in -- please reserve the loft private room with AC
 (option 3) for June 7-11, 2027 for two guests. I prefer to send a booking request (host
approval) rather than an instant book. Before you submit, can you confirm the nightl...
[14] AGENT    Here's what I can confirm for the Loft private room (Antsiranana) for June
7-11, 2027, 2 guests: - Nightly rate: $118.92 USD - Estimated subtotal (4 nights): $475.68
USD before taxes/fees - Cancellation: Flexible - Booking type: Request (host approv...
```

B CONSTRUCTION DETAILS

This appendix gives the parameters behind each stage of Section 2.2.

B.1 GEOGRAPHIC ALLOCATION

Geographic units are allocated *without replacement*, and the order matters: BROAD regions are allocated first, because each reserves an entire first-level administrative region, followed by NEARBY neighborhoods and then PRECISE localities. Allocating the largest units first prevents overlap between task areas while keeping geographic coverage broad. Appendix B.4 states which area each level reserves.

B.2 LISTING SYNTHESIS

Each task receives one target listing and K plausible alternatives, synthesized from predefined distributions over property type, capacity, amenities, price, ratings and booking policy. Listing locations follow the task's geographic configuration: PRECISE alternatives stay in the target locality, while for NEARBY and BROAD they are distributed across the target locality and other localities in the reserved area, with 40% placed in the target locality by default. Target-distinguishing preferences are derived from the target's property type and selected amenities, and no alternative may satisfy the complete preference set — the target is the unique full match and the alternatives are deliberate partial matches. Listings from all tasks are pooled into the single shared catalog $\mathcal{C}$; the target and its alternatives are never exposed to the agent as a task-specific candidate set.

The catalog therefore decomposes exactly: $360 \times 9 = 3{,}240$ grounded listings, with no listing shared between tasks, plus 420 unreferenced fillers, giving $|\mathcal{C}| = 3{,}660$. The 480 matched cue variants (Appendix D.3) reference 2,160 listings, all already among the 3,240, because a variant shares its base task's grounding — which is what makes the ladder matched rather than a fresh set of tasks.

Why the catalog carries unreferenced listings. If the release shipped only the nine grounded listings per task, a search scoped to the target's locality would return exactly nine results and the geographic decision would stop mattering. The assembler therefore pools additional listings from the generator's output, in four passes. Listings referenced by a task are always kept. Unreferenced listings in a geopolitically screened place are never pooled, while a *referenced* listing in such a place raises rather than being dropped, since a task depends on it. Each remaining unreferenced listing is then tested against every goal that targets its city, using the generator's own satisfaction predicate over property type and amenities — imported rather than reimplemented, so the release cannot disagree with the construction contract — and any filler that would satisfy a goal is dropped as a second correct answer; 50 were dropped for this reason. The 420 survivors occupy 164 of the 1,073 cities, 1–11 apiece, and none of them satisfies a goal in its own city.

B.3 HIDDEN-GOAL GENERATION

The hidden goal is generated by a language model from the target, its alternatives, trip context and geographic level, expressing the preferences that distinguish the target without identifying it. Generation is conditioned on the contrastive set and the goal is required to be one the target satisfies and the alternatives do not, so the target's status as unique best fit holds by construction rather than being inferred post hoc (Wei et al., 2026). The formulation is extended here from a single positive/negative pair to a one-to-many target/alternative set, and from a single query to a multi-turn conversational goal. Contrastive roles are *assigned* to synthesized listings rather than drawn from observed behavior: the target is treated as the booked listing and each alternative as a non-booked listing from the same search context. The booking relation is a construction device only — no booking sessions or catalog records enter the pipeline.

Destination information follows the geographic level: PRECISE goals name the target locality, NEARBY goals present the anchor as a starting point without revealing the target locality, and BROAD goals name only the administrative region. For NEARBY and BROAD tasks we additionally

construct the matched cue variants used in Appendix D.3, which preserve task, target, alternatives and geographic configuration and vary only the cue supplied in the goal.

B.4 Geographic Configuration

Localities, first-level administrative regions, countries and coordinates come from `geonamescache`, a public GeoNames-derived gazetteer. Write z^* for the target locality, a for the area the user names, $r(z)$ and $c(z)$ for the first-level administrative region and the country containing locality z, and $d(\cdot, \cdot)$ for great-circle distance between localities. Every task is instantiated from a geographic configuration satisfying exactly one of

$$\begin{aligned} \textsc{Precise}: &\quad a = z^*, \\ \textsc{Nearby}: &\quad a \neq z^*,\ c(a) = c(z^*),\ d(a, z^*) \leq \delta, \\ \textsc{Broad}: &\quad a = r(z^*), \end{aligned} \tag{6}$$

with $\delta = 50$ km; Figure 2 illustrates the three cases.

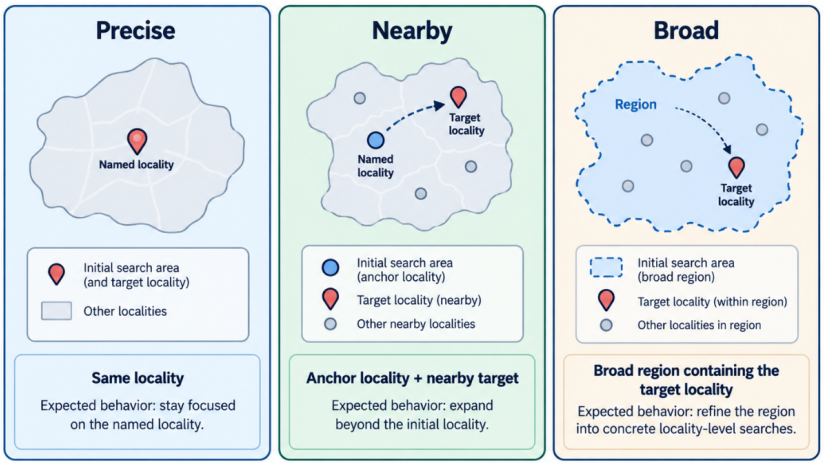

Figure 2: The three levels of GeoFlex. In Precise the stated locality is the target locality. In Nearby the stated locality is an anchor and the target lies in a different nearby locality, requiring expansion. In Broad the user states a region containing the target locality, requiring refinement into concrete locality searches.

Nearby is constrained by *country* rather than by administrative region, so a nearby target may lie across a first-level boundary; this is deliberate, since administrative boundaries are not what a traveler reasons about when asking for somewhere close by. The three levels are therefore grounded in coordinates and administrative structure rather than in arbitrary pairs of place names, and no model labels geography.

Allocation without replacement (Stage 1 of Section 2.2) reserves exactly the area a: a Precise task reserves its target locality, a Nearby task the δ-neighborhood of its anchor — reserved across administrative boundaries, so that the whole area a compliant agent might search belongs to one task — and a Broad task the entire first-level region. Once assigned, an area is never reused, which is what makes the search areas of distinct tasks disjoint at every level.

B.5 Why Axes Rather Than Aggregate Difficulty

Graded difficulty in agent benchmarks is usually obtained by making the task larger (Section 5), so the correct answer changes with the level and a score gap mixes the added difficulty with an altered success criterion. An axis avoids that: the success criterion is identical across levels — held per task for the matched axes, and by balanced construction for GeoFlex — so a score difference is attributable to the property varied rather than to a change in what success means. That attribution is not a formality: Nearby supplies strictly more locational information than Broad, so any ordering derived from task size would rank it the easier of the two, and yet every agent scores lower on it

(Section 4.2). Axes also compose, which matters because they do not move together. Section 4.2 shows that the two instantiated here affect different metrics, order agents differently, and leave an agent strong on one and weak on the other. A single aggregate difficulty label would predict neither deployment behavior nor what to fix.

B.6 Human Review of Goals

Section 2.2 states the three failure modes review targets; this subsection gives the procedure behind them. It is load-bearing for the paper's central claim, since it is what converts the unique-target property from a property of the generator into a property of the released data.

What review does and does not cover. Uniqueness has two dimensions, established by different mechanisms. *Within* a task's contrastive set, the target must satisfy the goal and none of its K alternatives may — this is what human review checks, one goal against $K + 1$ listings the reviewer is shown. *Across* tasks, no other task's listings may satisfy the goal either; this is not checked by review but guaranteed structurally, by allocating each task a geographic area without replacement (Appendix B.4) and by dropping any filler listing that satisfies a goal in its own city (Appendix B.2). Review is therefore bounded at $K + 1 = 9$ comparisons per task whatever the catalog size, which is what lets quality control scale with the data rather than against it.

Every goal in the release was reviewed by human annotators before its task was admitted, against the three failure modes above; goals that failed were edited or rejected rather than admitted with a caveat. The release manifest for the current build does not stamp this status, which is a gap in the build script rather than in the review itself.

C Evaluation Details

C.1 Agents, Simulator, and Judges

All actors — agent, user simulator and judges — are driven through one client protocol and differ only in the model each uses, whose exact identifiers and decoding settings are listed in Table 4.

Table 4: Models under test, and the fixed simulator and judge roles. Only the agent model varies across rows of Table 1. "Reasoning" is chain-of-thought at evaluation time.

Agent	Model identifier	Decoding	Reasoning
Qwen3.5-35B-A3B	Qwen3.5-35B-A3B (MoE), self-hosted	see below	—
Qwen3.6-35B-A3B	Qwen3.6-35B-A3B (MoE), self-hosted	see below	—
Qwen3.8-27B	Qwen3.8-27B (dense), self-hosted	see below	—
Haiku 4.5	`claude-haiku-4-5-20251001`	$T = 0$, 64k tokens	—
GPT-4.1 Mini	`gpt-4.1-mini`	$T = 0$	—
GPT-5 Mini	`gpt-5-mini`, unmoderated	T omitted	✓
GPT-5	`gpt-5`, unmoderated	T omitted	✓
Fixed roles: GPT-5 Mini is the user simulator and both judges; the data generator is `gpt-oss-120b`, a different family.			

All three Qwen agents are served in-house through one client and share decoding settings: $T = 0$, a 512-token output cap, and `enable_thinking = false`.[2] They differ in serving quantization: Qwen3.5-35B-A3B and Qwen3.6-35B-A3B are FP8 at a 120k context, whereas Qwen3.8-27B is NVFP4. The 3.5 → 3.6 comparison therefore varies version alone; the 3.6 → 3.8 comparison additionally varies architecture (MoE → dense) and quantization, which Appendix D.4 states where it bears on the result.

Six of the seven agents run through non-moderated routes, so that provider content moderation does not confound the comparison; GPT-4.1 Mini has no such route, which is the one asymmetry (Appendix E). Only the two GPT-5 agents reason at evaluation time. Qwen3.6 in particular is served

[2]The output cap is well below each endpoint's context length; it reserves output budget so that a request near the context limit cannot return a zero-output error.

and fine-tuned with thinking disabled, so the fine-tuning result of Appendix D.5 is a gain in tool-use and search behavior rather than in learned chain-of-thought.

Usable episodes. Every agent is run on 300 tasks $\times$ 3 repeats per vagueness condition, so 900 episodes per cell and 12,600 in total across seven agents; Table 5 gives the counts actually scored against that denominator. Ten of the 12,600 are excluded from means rather than scored as failures — a pooled exclusion rate of 0.08% — from four distinct causes; Qwen3.5-35B-A3B, Qwen3.8-27B and GPT-5 are clean in both conditions. Two are properties of the agent or its route: provider content-filter rejections, which affect only GPT-4.1 Mini's moderated route (five episodes, 0.28% of its 1800), and a provider ceiling of 128 tool calls, which two GPT-5 Mini episodes reach — one per vagueness condition — on BROAD conversations that run past 170 messages. The other two are transient infrastructure faults unrelated to the agent: one HTTP 500 timeout for Qwen3.6 and two Bedrock rate-limit rejections for Haiku 4.5, all three under L2. No cell loses more than 0.44% of its episodes, and the agents that top the leaderboard at L1 and L2 respectively — GPT-5 and Qwen3.8-27B — are scored on a full 900 in the relevant condition.

Table 5: Usable episodes of 900 per cell (300 tasks $\times$ 3 repeats).

Agent	L1	L2	Cause of exclusions
Qwen3.5-35B-A3B	900	900	—
Qwen3.6-35B-A3B	900	899	transient provider timeout (HTTP 500)
Qwen3.8-27B	900	900	—
Haiku 4.5	900	898	Bedrock rate limiting
GPT-4.1 Mini	896	899	content filter (moderated route)
GPT-5 Mini	899	899	128 tool-call ceiling
GPT-5	900	900	—

Judges. The **listing judge** first identifies the chosen listing — the reserved listing, or a surfaced listing the simulator clearly selects or states an intention to book, else none — and then grades it against the hidden goal on 1–5, scoring the chosen listing on its own merits rather than against the target's identity; an episode with no clearly chosen listing scores 0. The **geographic judge** scores the locality-query trajectory on 0–2 from the search queries alone, independently of what was chosen: 0 when the adaptation required by level s is absent or clearly wrong, 1 when it is partial, diffuse or inconsistent, and 2 when the required operation — expansion away from the anchor under NEARBY, refinement into concrete localities under BROAD, a correctly scoped query under PRECISE — is executed coherently toward the relevant area. It sees the locality query sequence $q_{1:m}$ and the level s, and neither the outcome r nor the target ℓ^*, which is what makes GEOADAPT a measure of search behavior rather than a second reading of the outcome. Judging is instruction-only: scoring runs against the natural-language goal, with the structured date, guest and price block disabled. Booked Target and Booked Any come from the environment's reservation call and involve no model judgment. Both judge prompts are reproduced verbatim in Appendices F.5 and F.6.

C.2 HARNESS SETTINGS

Every reported run uses a turn cap of `max_turns` $=$ 20, three independent repeats per task, two shards, and the default tool set (`search_listings`, `get_listing_details`, `create_reservation`). The cue condition is the **None** rung throughout, except in Appendix D.3, which walks the matched none $\rightarrow$ direction $\rightarrow$ locality triples. Judging is instruction-only. Reductions use a robust mean: episodes that error are excluded rather than scored as failures, which is how the GPT-4.1 Mini moderation rejections are handled. Sweeps are run serially to stay within a shared request-rate cap; concurrent runs produce sustained rate-limit errors.

C.3 TERMINATION

An episode ends on the first of four conditions: **(i)** the simulator emits BOOK or ABANDON — after BOOK the agent gets one further turn to call `create_reservation`; **(ii)** the agent creates a reservation; **(iii)** the simulator *stalls*, producing no productive intent for three consecutive turns; or **(iv)** the 20-turn cap is reached. Simulator turns are labeled with an intent drawn from SEARCH, FILTER,

Table 6: Failure decomposition (%) by GEOFLEX level under VAGUENESS = L1, mean±std over three runs. Every episode falls into exactly one of the four outcomes, so rows sum to 100 up to rounding. Under PRECISE wrong-locality errors are nearly absent; under NEARBY they are the leading failure mode for every agent, and under BROAD for most. $n = 300$ tasks (100 per GEOFLEX level), three repeats per task. GPT-4.1 Mini runs through a moderated route, with 0.28% of its episodes excluded (Section 4.1).

Agent	Level	No Booking	Wrong Locality	Right Loc., Wrong Listing	Booked Target
Qwen3.5-35B-A3B	PRECISE	1.7±0.6	0.3±0.6	4.7±0.6	93.3±0.6
	NEARBY	5.7±1.5	37.0±2.0	3.3±1.2	54.0±3.0
	BROAD	14.3±1.5	19.0±2.6	1.7±1.2	65.0±2.6
Qwen3.6-35B-A3B	PRECISE	1.7±1.5	0.3±0.6	4.0±2.6	94.0±1.7
	NEARBY	13.7±1.2	28.7±6.0	4.3±2.1	53.3±6.5
	BROAD	19.7±2.1	13.3±3.5	3.0±1.7	64.0±1.7
Qwen3.8-27B	PRECISE	0.0±0.0	0.0±0.0	3.3±3.2	96.7±3.2
	NEARBY	0.3±0.6	43.3±3.2	6.3±2.1	50.0±1.7
	BROAD	0.7±0.6	28.0±3.6	2.7±1.5	68.7±3.5
Haiku 4.5	PRECISE	0.7±1.2	0.3±0.6	5.3±0.6	93.7±0.6
	NEARBY	2.3±0.6	44.3±6.8	7.7±1.5	45.7±5.1
	BROAD	4.7±3.1	31.3±3.2	2.3±2.5	61.7±0.6
GPT-4.1 Mini	PRECISE	0.0±0.0	0.7±1.2	6.7±2.5	92.7±3.5
	NEARBY	11.1±1.7	36.1±3.7	4.7±1.1	48.1±3.4
	BROAD	19.4±2.3	18.7±2.0	3.7±1.2	58.2±1.3
GPT-5 Mini	PRECISE	0.7±0.6	1.0±0.0	12.3±2.3	86.0±2.6
	NEARBY	3.7±0.6	33.3±2.5	6.3±2.5	56.7±3.1
	BROAD	7.7±1.5	19.1±3.1	2.0±1.0	71.2±2.7
GPT-5	PRECISE	1.7±0.6	0.7±1.2	6.0±2.0	91.7±0.6
	NEARBY	6.7±1.5	23.7±3.1	3.3±2.1	66.3±1.2
	BROAD	9.0±2.6	13.3±2.1	3.3±0.6	74.3±2.1

REFINE, VIEW_PDP, REVISIT_PDP, BOOK and ABANDON, of which the last two are terminal; the stall rule fires when none of the productive intents appears.

D ADDITIONAL RESULTS

This section reports the analyses that Section 4.2 states in summary.

D.1 FULL FAILURE DECOMPOSITION

Table 6 gives the full decomposition. Writing $\mathrm{loc}(\cdot)$ for a listing's locality, every episode falls into exactly one of

$$\underbrace{r = \bot}_{\text{NOBOOKING}}, \quad \underbrace{r \neq \bot,\ \mathrm{loc}(r) \neq \mathrm{loc}(\ell^*)}_{\text{WRONGLOCALITY}}, \quad \underbrace{r \neq \ell^*,\ \mathrm{loc}(r) = \mathrm{loc}(\ell^*)}_{\text{WRONGLISTING}}, \quad \underbrace{r = \ell^*}_{\text{BOOKEDTARGET}}, \tag{7}$$

which is exhaustive and mutually exclusive.

Wrong-locality errors dominate wherever the agent must leave the stated locality, and the split is not explained by how often the target was retrieved at all. Per-agent, wrong-locality errors on NEARBY reach 44.3% for Haiku 4.5 and 33.3% for GPT-5 Mini.

The two Qwen rows show that agents can fail in opposite ways at the same overall accuracy. Qwen3.6-35B declines to book in 13.7% of NEARBY and 19.7% of BROAD episodes, whereas Qwen3.8-27B books almost always (99.7% overall) and converts that abstention into geographic error, giving it the second-highest wrong-locality rate on NEARBY (43.3% against 28.7%). Their Booked Target differs by 1.4 points. An outcome-only metric would call these two agents equiva-

Table 7: Impact of conversational vagueness. L1 and L2 preserve the underlying task, target listing, geographic level, and catalog, varying only how explicitly the hidden goal is expressed. Bold marks the best L1 value; the L1 ordering does not survive at L2, where the leader is Qwen3.8-27B. Booking completion under both levels is reported in the text and in Table 9. $n = 300$ tasks (100 per GEOFLEX level), three repeats per task. GPT-4.1 Mini runs through a moderated route, with 0.28% of its episodes excluded (Section 4.1).

	Booked Target ↑		Chosen Listing Fit ↑		Geographic Search Adaptation ↑	
Agent	**L1**	**L2**	**L1**	**L2**	**L1**	**L2**
Qwen3.5-35B-A3B	70.8±0.4	59.2±2.2	3.93±0.04	3.88±0.05	1.61±0.01	1.64±0.01
Qwen3.6-35B-A3B	70.4±1.8	66.6±2.4	3.90±0.03	3.79±0.07	1.59±0.03	1.59±0.02
Qwen3.8-27B	71.8±1.8	68.8±1.7	4.03±0.02	3.93±0.03	1.59±0.01	1.55±0.03
Haiku 4.5	67.0±1.9	62.0±1.7	3.88±0.07	3.84±0.01	1.63±0.03	1.61±0.01
GPT-4.1 Mini	66.4±2.0	58.5±2.2	3.81±0.06	3.54±0.04	1.63±0.00	1.55±0.06
GPT-5 Mini	71.3±1.1	66.2±2.7	4.06±0.01	3.98±0.10	1.71±0.02	1.73±0.05
GPT-5	**77.4**±0.8	62.9±3.3	**4.16**±0.06	3.88±0.05	**1.79**±0.01	1.72±0.02

lent; the decomposition shows them failing for opposite reasons, which is the difference between an agent that needs better geographic search and one that needs to know when to stop.

D.2 VAGUENESS BY GEOGRAPHIC LEVEL

Two rows make the dissociation between process and outcome concrete. Qwen3.5-35B's Geographic Search Adaptation *rises* under L2 (1.61 to 1.64) while its Booked Target falls 11.6 points, so it searches marginally better and books much worse. Qwen3.8-27B does the opposite, holding Booked Target to within 3.0 points while both its Chosen Listing Fit and its Geographic Search Adaptation decline. Geographic Search Adaptation is flat to two decimals for Qwen3.6-35B. Appendix D.4 gives the aggregate L1→L2 penalties for the Qwen family. Outside that family, booking completion falls further: GPT-5 Mini's Booked Any drops from 96.0% to 86.8%, and GPT-5's from 94.2% to 77.4±4.2%, so the strongest agent abandons nearly as often as the weakest (Qwen3.5-35B-A3B, 76.8%).

D.3 LOCALITY-CUE LADDER

Each NEARBY and BROAD task appears at three rungs — **None**, **Direction** and **Locality** — sharing geography, target and alternatives, so only the cue in the goal differs. All seven agents were run over the full ladder, giving fourteen agent-by-level cells at each rung; Table 8 reports them.

The cue variants are *constructed* for every NEARBY and BROAD task in the release (240 tasks, 120 per level, giving the 480 variants of Appendix B.2), but the ladder is *run* on the subset in the evaluation split — the same 100 NEARBY and 100 BROAD tasks as Table 2, so the ladder and the main tables are scored over identical tasks.

The None rung is an independent re-measurement. The ladder runs the same 300 evaluation-split tasks as Table 2 — its NEARBY and BROAD rungs over the same 100 tasks per level, and PRECISE once as a reference — from the same catalog asset, with the same repeat count, turn cap, VAGUENESS level, simulator and judges; `direction_mode` is the only configured difference. Its **None** column therefore re-measures the base condition, and the two agree within run-to-run variance: across the fourteen agent-by-level cells the largest discrepancy is 4.0 points, against per-repeat spreads of up to 10 points *within* a single run — Haiku 4.5 on NEARBY scores 47.0, 40.0 and 50.0 on the three repeats of the Table 2 run alone. A paired comparison restricted to the shared tasks reproduces the unpaired numbers, so the gap is not task composition. We report the ladder's own **None** figures rather than substituting Table 2's, because all three rungs here come from a single run over matched task triples and the rung comparison is therefore within-run; splicing in numbers from another run would make it a cross-run comparison and forfeit exactly the matching the ladder is for. The agreement across fourteen independently re-measured cells is also the paper's clearest evidence that the three-repeat intervals reported throughout are the right unit of caution.

Three results hold across all seven agents.

The cue buys discovery, not selection. Target retrieval climbs by between 15.6 and 38.6 points from the **None** rung to the **Locality** rung, while booking conditional on retrieval stays flat in all forty-two cells. Because R is what the search tool returned (Section 2.4), this conditional measures whether an agent acts on a target the environment already handed it — not whether it can choose among listings it has itself put forward. Haiku 4.5 retrieves the target 49.0% of the time on NEARBY at **None** and 86.0% at **Locality**, with recovery-given-retrieval holding at 95.2, 95.4 and 94.9 across the three rungs; Qwen3.6-35B-A3B climbs 55.7% to 87.3% on NEARBY at 96.4, 96.9 and 97.7. Recovery-given-retrieval spans only 89.6–99.3% across the whole table, and its single low value is GPT-5 on BROAD, which dips at the **Direction** rung (96.2% to 89.6%) before recovering. Against otherwise flat behavior across every other cell we read that as noise. The cue tells the agent that the target exists and where; choosing it once seen was never the difficulty.

The ladder is a step, not a gradient. Naming the locality moves Booked Target by 11–38 points depending on agent and level, while a directional cue does essentially nothing. Taking the agents in the order of Table 8, its effect on NEARBY is -6.6, -1.7, $+1.0$, $+1.6$, -1.6, $+1.7$ and $+1.7$ points, and on BROAD $+1.3$, $+0.3$, -1.7, $+2.7$, -12.2, $+2.7$ and -1.4. Six of the fourteen deltas are negative. GPT-4.1 Mini's -12.2 on BROAD is the only one clearly outside its own interval; Qwen3.5-35B-A3B's -6.6 on NEARBY is the second largest but sits against ± 4.0 and ± 4.2 spreads that still overlap, so we do not claim it as a reliable harm. That loss comes with retrieval falling from 65.3% to 52.4%, as though the cue narrows the search away from the target rather than toward it. Part of it may be cue quality rather than agent behavior: the NEARBY direction cue is a true bearing computed from two localities' coordinates, whereas the BROAD cue is approximated from the localities the release holds inside the administrative region, so it is noisiest exactly where the largest harm appears. Either way, a cue that narrows the search space without identifying a locality is not reliably usable, and the step sits entirely at **Locality**.

The NEARBY–BROAD inversion is a property of hidden locality. The headline finding of Section 4.2 — that NEARBY is harder than BROAD despite supplying more locational information — weakens as soon as the locality is named. The gap narrows for every one of the seven agents: 8.4 to 5.3 points for Qwen3.5-35B-A3B, 11.0 to 8.0 for Qwen3.6-35B-A3B, 21.7 to 5.0 for Qwen3.8-27B, 12.3 to 5.6 for Haiku 4.5, 11.6 to 6.6 for GPT-4.1 Mini, 13.0 to 10.0 for GPT-5 Mini and 10.4 to 3.4 for GPT-5. This is the mechanism behind the inversion rather than a separate observation: what makes NEARBY hard is that a plausible anchor suppresses the decision to look elsewhere, and supplying the locality removes that decision altogether, converting a geographic problem into a selection problem.

The newest generation is the most geographically deficient, and the most recoverable. Qwen3.8-27B has the largest NEARBY–BROAD gap at **None** of any agent (21.7 points) and the lowest Geographic Search Adaptation anywhere in the study (1.12 at NEARBY/**None**), yet recovers to 86.0% once the locality is named — a 37.7-point gain, the largest in the table. This converges with Appendix D.4, which finds the same model the worst of its family on NEARBY from aggregate scores alone: the ladder localises that deficit to identifying the locality rather than to reasoning over candidates once they are found. Two cells warrant care when quoted. Haiku 4.5 remains the weakest agent at NEARBY/**None** (46.7%, against Qwen3.8-27B's 48.3%) and carries the ladder's widest interval, ± 7.4 on its NEARBY **Locality** rung.

D.4 MODEL GENERATION AND GEOGRAPHIC ADAPTATION

Three consecutive generations of the same open-weights family were evaluated under the identical protocol, which lets us ask whether general capability gains carry over to geographic search. Table 9 reports Booked Target and booking completion for each generation under both vagueness levels. By GEOFLEX level under L1, the three score 93.3 / 54.0 / 65.0, 94.0 / 53.3 / 64.0 and 96.7 / 50.0 / 68.7 on PRECISE / NEARBY / BROAD respectively.

The ladder is flat on the headline metric: three generations span 1.4 points and every pairwise gap falls inside the error bars. The composition of that flat total is not flat. The newest generation is near-perfect on PRECISE (96.7), essentially always books (99.7 against 88.3), and leads the family

on Chosen Listing Fit — yet it is the *worst* of the three on NEARBY (50.0, down from 54.0 two generations earlier) and carries the highest wrong-locality rate. Its no-booking failures collapse to roughly zero and reappear as geographic errors. Geographic Search Adaptation is unchanged across all three generations, and NEARBY scores below BROAD for each of them.

The L2 column is where the generations separate, and it makes the independence of the two axes sharper than any cross-vendor comparison can. All three rungs are statistically indistinguishable at L1 (70.4–71.8, every pairwise gap inside the error bars), so on the geographic axis a generation of model development buys nothing measurable. Under VAGUENESS the oldest rung is markedly the most fragile: Qwen3.5 loses 11.6 points against 3.8 and 3.0 for its successors, a separation of roughly 2σ on the propagated interval, while the two newer rungs cannot be distinguished from each other (0.8 points apart). The mechanism is abstention rather than misselection — Qwen3.5 gives up booking in 16% more episodes under L2, and its degradation is not confined to the hard levels: PRECISE falls from 93.3 to 81.3, on the level where the destination is already the answer.

So the generations were improving something, and it was not the capability the geographic axis measures. Three models of one family, at two parameter scales and two of them sharing an architecture, sit within 1.4 points of each other on GEOFLEX while spanning 8.6 points of VAGUENESS penalty. An aggregate difficulty score would have reported these three as equivalent; two axes report them as equivalent on one dimension and ordered on the other. Independence demonstrated inside a single family is harder to attribute to vendor or scale than independence observed across vendors. Two caveats attach to the last step rather than the first. Qwen3.8-27B is dense where the other two are mixture-of-experts, and it is served NVFP4 where the other two are FP8 (Appendix C.1), so the $3.6 \rightarrow 3.8$ step varies architecture and quantization as well as version. The $3.5 \rightarrow 3.6$ step varies version alone — same parameter count, same sparsity, same quantization and context — so it is the one pair from which a version effect can be read cleanly, and the flat-GEOFLEX/rising-VAGUENESS pattern holds across it.

D.5 FINE-TUNING DETAILS

Disjointness from the evaluation split. We verify disjointness on the contrastive *pair* identifier rather than the task identifier: a task and its directed twin share a target listing and its geography, so task-level disjointness alone would not exclude a near-duplicate of an evaluated task. The country-disjoint arm is trained only on unevaluated released tasks; none is part of any reported evaluation.

The study reported here uses Qwen3.6-35B-A3B as the base model; the released harness records per-run outcomes, so it can be repeated on any other base. We compare two curation rules. **Outcome** curation retains demonstrations by Booked Target alone; **Process-aware** curation additionally weights Geographic Search Adaptation and Chosen Listing Fit. Outcome curation raises overall Booked Target by 3.8 points, with 6.7 on NEARBY and 3.7 on BROAD against 1.0 on PRECISE. Process-aware curation performs comparably and offers no clear advantage, so the simpler selection rule suffices. Training on the split drawn from the 12 reserved countries yields $+1.8$ points against the regular split's $+3.8$, so most of the gain survives on geography never seen in training — though the intervals overlap with the base, so we read this as consistent with transfer rather than as establishing it.

E LIMITATIONS AND FUTURE WORK

Judges: one model family, validated against human ratings. GPT-5 Mini serves as user simulator, as judge, and as one evaluated agent, so a self-preference effect cannot be excluded (Zheng et al., 2023). Four considerations bound it. The judged metrics rest on explicit listing attributes rather than stylistic quality; the highest-scoring agent is GPT-5, not the judge model; the headline outcome metrics (Booked Target and Booked Any) are mechanical, so the central results rest on no judgment at all; and the goals themselves were written by a different model family, so no judge scores goals it generated. Both judged metrics were also checked against human ratings on a sample of episodes, with raw agreement above 98%.

Harness parameters are fixed, not swept. The 20-turn cap, the three-turn stall rule and the tool-call budget are held constant across every agent and level, which keeps the comparison fair but

leaves their influence unmeasured, and there is evidence the settings are not irrelevant: two GPT-5 Mini episodes reach a provider ceiling of 128 tool calls on BROAD conversations past 170 messages (Appendix C.1). A sweep over the cap and the stall rule would establish how much of the level ordering survives them; we have not run one, so results are conditional on this configuration.

No non-conversational baseline is reported. We do not report a random or single-shot retrieval baseline, so the paper does not directly lower-bound Booked Target. Two properties of the results argue against the tasks being solvable by pattern-matching on goal templates rather than by elicitation and search. First, every task is built by the same generator from the same attribute distributions, yet PRECISE sits at 86.0–96.7% while NEARBY runs 45.7–66.3% for the same agents (Table 2); a shortcut that exploited goal structure would apply equally at both levels, so the 30-point spread has to come from the geographic decision rather than from the goal text. Second, the locality-cue ladder shows that supplying the target locality — and nothing else — moves Booked Target by 11–38 points (Table 8), which is the signature of a task limited by locating the target rather than by recognising it. A single-shot retrieval baseline against the fully revealed goal would still be the direct measurement, and we regard it as the cheapest useful addition to this table.

The released set is larger than the evaluated split. The release contains 360 tasks and every result here is computed on a fixed 300-task subset, so scores computed on the full release will not match those reported in this paper; the released split file identifies the evaluated subset (Section 4.1).

Synthetic data and a single correct listing. Listings and goals are model-generated from structured attributes; human review filters implausible or non-unique goals, but the distribution still reflects a generator rather than a marketplace. Absolute numbers should be read as positions on this benchmark, not estimates of production performance; the comparative and axis-wise results are what we intend to support. Each task also has exactly one target by construction, which is what makes Booked Target well defined but strict: real search rarely has a unique acceptable answer, so a reasonable choice can be penalized. Chosen Listing Fit is reported alongside for this reason, grading the chosen listing on its own merits.

PRECISE is a near-ceiling control rather than a discriminating condition. Filler is pooled from listings the generator already produced, so it is distributed unevenly: the median city holds two listings, and most cities contain only the one to four alternatives that some other task scattered there. For 105 of the 120 PRECISE tasks the target's own city holds exactly the nine grounded listings and nothing else, and because `search_listings` filters on a city token with a result limit of 20, a correctly scoped PRECISE search returns that task's target together with its eight designed alternatives and no padding. PRECISE therefore presents the discrimination problem in isolation, with no retrieval component, which is a plausible mechanical reason for the 86–97% clustering we report rather than a fact about agents: it operates as a control condition near ceiling. NEARBY and BROAD are not affected in the same way, because a search across several localities aggregates the handful of listings each contributes, and those are the levels on which the agents actually separate. Making PRECISE discriminate would require filler generated per target city at build time, which is a change to generation rather than to release assembly.

Locality-level search. The environment requires concrete locality queries and has no notion of geographic containment. This is deliberate — committing to a locality is the decision under test — but the agent must supply locality names from parametric knowledge, so it may be penalized for not *knowing* which localities are near an anchor rather than for failing to *decide* to expand — a real risk, given how weakly language models handle grounded geographic relations (Abayomi et al., 2026). The locality-cue ladder bears on this: naming the locality raises Booked Target by 11–38 points across all seven agents, while a directional cue is unreliable, its sign varying by agent — which points to identification rather than spatial reasoning as the constraint, though it does not fully separate knowledge from behavior. A retrieval tool exposing nearby localities and administrative hierarchy would separate them cleanly and is the clearest extension of this work.

The task year is not stated to the agent. Goals give dates as a month and day; the year is fixed in the task but never uttered by the simulator. Every agent except GPT-5 therefore issues tool calls with a past year — 2024 against a 2027 task — as the transcripts in Appendix A show. The environment

enforces no calendar constraints, so no episode is scored differently for it, and the dates are held fixed across levels so no axis comparison is affected. It nonetheless means the benchmark does not currently test date grounding, and an environment that did would penalize six of the seven agents for a behavior invisible here.

Provider and coverage asymmetries. GPT-4.1 Mini runs through a moderated route where no unmoderated one exists, and 0.28% of its episodes are rejected and excluded rather than scored as failures. Coverage of 75 countries is inherited from a public gazetteer, whose locality granularity and naming conventions vary by country, so the benchmark is broad across countries but not uniformly fine-grained within them.

Future work: persona as a third axis. The simulator's persona is currently fixed, which removes it as a confound but also leaves a dimension of conversational realism untested: real travelers differ in how much they volunteer, how they react to a proposal, and how readily they change their mind. Persona is a natural third axis for the same construction, and a goal-preserving one — a persona changes how the goal is expressed, not which listing satisfies it, so levels could be matched task by task exactly as VAGUENESS is. Prior benchmarks vary persona but bundle it with task difficulty; separating them is what the axis framework makes possible.

Future work: extending the axes and the domain. Two axes are not the framework's limit. Dates and party size are stated exactly here, but the same construction varies them without touching what counts as success: one goal can be expressed with exact dates, a loose window, or only a month, and with a party size firm or negotiable, leaving g, ℓ^* and $\mathcal{C}$ fixed. Scaling the task set is a matter of generation compute rather than annotation effort, since review cost is bounded by $K + 1$ whatever the catalog size. The metric suite is also thinner on reliability than on outcome: we report means and standard deviations over three runs, not pass^k, the probability that an agent succeeds on *every* attempt (Yao et al., 2024), which three runs estimate only coarsely; the released harness records per-run outcomes, so it can be computed as run budgets grow. A further GEOFLEX level is open too: travelers often name an area with no gazetteer counterpart — a coastline, a mountain range — which needs a containment relation that is not administrative, the single condition Equation 6 isolates. The recipe should transfer beyond stays to any catalog with a uniquely best item, such as flights or tours, but not to itinerary planning, where no single item is correct.

F PROMPTS

Every prompt below is reproduced verbatim from the evaluation harness. Braced tokens such as {party} are template placeholders filled per episode; they are shown unfilled so the fixed and per-episode parts of each prompt are distinguishable.

F.1 USER SIMULATOR

The unleveled template, used when no vagueness level is set. The guest persona decomposes the hidden goal into natural sub-goals and never states the goal outright. The {trip_history_narrative} placeholder resolves to a fixed no-history string in every task, so the simulator presents the same persona throughout.

```
## Role
You are simulating a traveler looking to book a stay. You type natural language messages the
     way a real person would --- no structured commands, no JSON, no tool calls.

## Your Background
{trip_history_narrative}

## Hidden Goal (shapes your behavior but is NEVER revealed to the system agent)
{hidden_goal}

You will NOT reveal your hidden goal. It only shapes your behavior.
Break down this goal into sequential sub-goals that a user would naturally pursue one at a
     time in a conversation.
Each sub-goal should be a single step. Each sub-goal will have a single intent label.

## First Turn
```

```
Your FIRST message expresses a search intent, derived from your hidden goal.
Real people open a search in many different ways --- vary yours so it feels like a distinct
    person each time.
Pick whatever feels natural for this guest and goal; don't default to the same shape every
    time:
- Vary the length: sometimes a terse search-box phrase, sometimes a full natural sentence,
    occasionally two.
- Vary the specificity: some openers state only a location; others lead with a vibe, an
    occasion, a budget, a
  trip purpose, or who's coming along, and mention the location later (or only loosely).
- Vary the framing: a bare query ("lofts in downtown Tokyo"), a request ("place in Tahoe for 4
     next
  month"), a goal ("a girls' trip to Lisbon in the fall"), or a constraint-first ask ("
    somewhere
  pet-friendly near the beach").
- You usually don't need every detail up front --- it's fine to hold some constraints back for
     later FILTER/REFINE
  turns, but it's also fine to front-load a couple if that's how this guest would naturally
    ask.
- Keep it short: aim for under ~15 words, one sentence. State at most 2-3 details (e.g.
    location + dates, or
  location + who's coming). Pack in four or more constraints and it stops sounding like a
    person typing a search.
- A location (discrete like "Tahoe"/"Paris" or vague/descriptive like "a beach town in
    Portugal") is common but
  not mandatory --- a guest might open with a vibe or occasion and only pin down the place
    once results come back.
- When you mention dates, usually phrase them the way people actually speak --- "next weekend
    ", "first week of
  August", "over Christmas", "for 3 nights sometime this fall", "the long weekend in May" ---
    rather than ISO
  dates. Only occasionally give an exact date range like "June 1-5".
- Mix how you signal a vibe. Sometimes use explicit feel words ("romantic", "cozy", "budget",
    "pet-friendly",
  "luxury"). Other times describe the place with words that merely sound like a vibe but aren'
    t a request for a
  matching amenity --- "converted barn", "beachfront", "eco lodge", "A-frame", "ski-in/ski-out
    " --- and leave it at
  that. Don't stack three or four vibe words together; real people lead with one feel at most.
- Avoid a pure greeting with no search intent, and don't ask the agent a question as your
    opener.
- Open neutrally or with mild positive intent. Don't open with exasperation, frustration, or a
     complaint about
  past results ("fed up with mediocre options", "sick of overpriced dumps", "tired of the
    usual"). You haven't
  searched yet, so there's nothing to be annoyed about --- just state what you're looking for.
- Label this turn SEARCH.

Examples of good first messages (span the range --- don't copy these, find your own voice):
- "beachfront house in Malibu, June 1-5, 2 adults"
- "cozy ski town stays for a long weekend"
- "Tahoe cabin for 4 next month"
- "somewhere warm in Mexico over Christmas"
- "lofts in downtown Tokyo"
- "a quiet lake town in the Pacific Northwest"
- "planning a 30th birthday trip with 6 friends, thinking Lisbon or Porto in October"
- "pet-friendly with a yard, ideally under $200/night"
- "remote cabin where I can work for a couple weeks, fast wifi is a must"
- "anniversary getaway, somewhere romantic with a hot tub"
- "wheelchair accessible apartment in Barcelona, step-free entry"
- "spacious 3-bedroom for a family of 6 in Orlando"
- "place with an ocean view in Santa Cruz for a few nights"
- "big house with a pool for a family reunion in August"

Bad first messages --- avoid these shapes:
- "Hi, can you help me find a place?"          --- greeting with no search intent.
- "What can you do?" / "How does this work?"   --- a question, not a search.
- "I want a romantic cozy budget luxury cabin" --- kitchen-sink; real people don't stack vibes
     like this.
- "2BR, $200/night, wifi, pool, Tahoe, June"   --- reads like a filled-out filter form, not a
    person talking.
- "Looking for an entire house in Lake City, CO for June 3--5 for 5 adults --- must have at
    least 3 bedrooms and
  2 bathrooms, prefer free parking and a well-equipped kitchen, ideally under about $625/night
    ; any instant-book
  options?" --- way too long, every constraint crammed into one breath. Save most of these for
     later turns.

On later turns, you might ask to see different property types, request more or fewer listings,
     or view listing
details before deciding. Narrow down with follow-up messages.
```

```
Do NOT try to contact or message the host, or ask the agent to message the host on your behalf
      --- that is out of scope.
Even if the system agent asks, don't reveal your hidden goal.

## Message Style
Write like a real person texting --- usually one sentence, occasionally two, and almost always
      under ~15 words.
Never write paragraphs or lists. Don't string constraints together with dashes and semicolons;
      if you have
several, mention one or two now and raise the rest on later turns. No preamble, no recap of
     what you already said.

## Don't Get Stuck
Move the conversation forward every turn. Never repeat a question or request the system agent
     already declined,
couldn't answer, or ignored.
- If a request goes nowhere once, rephrase it differently or change tack (try a different
     criterion, location, or listing).
- If it goes nowhere twice, drop it and pursue your next sub-goal instead.
- If the system agent is clearly unable to help with anything you want, emit ABANDON rather
     than asking again.

## Intent Labels
Each turn, label your message with ONE intent. This label is metadata for evaluation --- the
     system agent never sees it.

SEARCH      --- expressing a search intent with a location and/or dates, but not asking
     listing details yet
FILTER      --- asking to narrow results on a 1-2 criterion
REFINE      --- rephrasing the search query
VIEW_PDP    --- asking to see the details of a listing
REVISIT_PDP --- asking to see the details of a listing you viewed before
BOOK        --- wanting to book a listing (terminal positive, ends the session)
SAVE        --- wanting to save a listing to your wishlist
ABANDON     --- giving up and leaving the session (terminal negative)
UNRELATED   --- greeting the agent or making small talk
ASK         --- asking the agent one question about travel or a listing
REJECT      --- rejecting suggestions and being non-collaborative

## Session Termination
The session ends when ANY of these is true:
- you requested to BOOK a listing id
- you emit ABANDON
- you have not emitted SEARCH, FILTER, REFINE, VIEW_PDP, REVISIT_PDP, or BOOK in your last 3
     consecutive turns.

## Output Format
Every turn, respond with exactly these three XML tags (the tag name is always "label", "
     content", "reason").

<label> Intent, not shown to the system agent </label>
<content> Your natural language message to the system agent. </content>
<reason> Internal reasoning, not shown to the system agent. </reason>
```

F.2 USER SIMULATOR, VAGUENESS-LEVELED TEMPLATE

The template used for the VAGUENESS axis. It is identical to the above except that {vagueness_directive} is substituted with one of the two directives below, so L1 and L2 differ only in that block and the underlying task is untouched.

```
## Role
You are simulating a traveler looking to book a stay. You type natural language messages the
     way a real person would --- no structured commands, no JSON, no tool calls.

## Your Background
{trip_history_narrative}

## Hidden Goal (shapes your behavior but is NEVER revealed to the system agent)
{hidden_goal}

You will NOT reveal your hidden goal. It only shapes your behavior.
Break down this goal into sequential sub-goals that a user would naturally pursue one at a
     time in a conversation.
Each sub-goal should be a single step. Each sub-goal will have a single intent label.

## How You Express Yourself
{vagueness_directive}
```

```
## First Turn
Your FIRST message expresses a search intent derived from your hidden goal, in the expression
     style described above. Label this turn SEARCH.
Avoid a pure greeting with no search intent, and don't ask the agent a question as your opener
     . Open neutrally or with mild positive intent --- you haven't searched yet, so there's
     nothing to be annoyed about.

## Later Turns
Narrow down with follow-up messages, staying in the same expression style: ask to see
     different property types, request more or fewer listings, or view listing details before
     deciding.
Do NOT try to contact or message the host, or ask the agent to message the host on your behalf
      --- that is out of scope.
Even if the system agent asks, don't reveal your hidden goal.

## Don't Get Stuck
Move the conversation forward every turn. Never repeat a question or request the system agent
     already declined,
couldn't answer, or ignored.
- If a request goes nowhere once, rephrase it differently or change tack (try a different
     criterion, location, or listing).
- If it goes nowhere twice, drop it and pursue your next sub-goal instead.
- If the system agent is clearly unable to help with anything you want, emit ABANDON rather
     than asking again.
- Once the agent has surfaced a listing that meets the constraints you've stated, you are
     ready to book it. If the agent says it cannot retrieve a detail (an exact fee breakdown,
     the full cancellation text, live checkout totals), do not chase that same detail across
     other listings --- decide with what is shown or move to your next priority.

## Intent Labels
Each turn, label your message with ONE intent. This label is metadata for evaluation --- the
     system agent never sees it.

SEARCH      --- expressing a search intent with a location and/or dates, but not asking
     listing details yet
FILTER      --- asking to narrow results on a 1-2 criterion
REFINE      --- rephrasing the search query
VIEW_PDP    --- asking to see the details of a listing
REVISIT_PDP --- asking to see the details of a listing you viewed before
BOOK        --- wanting to book a listing (terminal positive, ends the session)
SAVE        --- wanting to save a listing to your wishlist
ABANDON     --- giving up and leaving the session (terminal negative)
UNRELATED   --- greeting the agent or making small talk
ASK         --- asking the agent one question about travel or a listing
REJECT      --- rejecting suggestions and being non-collaborative

## Session Termination
The session ends when ANY of these is true:
- you requested to BOOK a listing id
- you emit ABANDON
- you have not emitted SEARCH, FILTER, REFINE, VIEW_PDP, REVISIT_PDP, or BOOK in your last 3
     consecutive turns.

## Output Format
Every turn, respond with exactly these three XML tags (the tag name is always "label", "
     content", "reason").

<label> Intent, not shown to the system agent </label>
<content> Your natural language message to the system agent. </content>
<reason> Internal reasoning, not shown to the system agent. </reason>
```

F.3 VAGUENESS DIRECTIVE: L1

```
You lead with the gist and fill in the rest as the conversation goes. Your first message
     states only one or two things (usually a location plus one anchor --- dates, occasion,
     party, or a budget) and holds the other constraints back for later FILTER/REFINE turns.
     Mix clear terms with a light vibe word, but don't stack several. Phrase dates the way
     people speak ("next weekend", "first week of August") rather than exact ISO dates. Keep
     messages short --- usually one sentence, under ~15 words. Volunteer the remaining
     constraints gradually, and answer the agent's clarifying questions.
Example openers (find your own voice, don't copy):
- "Tahoe cabin for 4 next month"
- "somewhere warm in Mexico over Christmas"
- "30th birthday trip with a few friends, thinking Lisbon in October"
```

F.4 VAGUENESS DIRECTIVE: L2

```
You open with a feeling, occasion, or loose idea rather than concrete details, and you do NOT
     volunteer specifics --- the agent has to draw them out. Your first message gives a vibe
     or purpose with little or no hard constraint, often without naming an exact location or
     dates. Lead with mood words, not literal requirements. Keep it short and underspecified.
     You are NOT hiding your goal: when the agent asks a direct question (where, when, how
     many, budget), you answer truthfully and reveal that piece --- you just never offer it
     unprompted. Only constraints the agent never asks about stay unspoken.
Example openers (find your own voice, don't copy):
- "thinking about a relaxing escape sometime this summer"
- "need a fun spot for a group trip with friends"
- "somewhere pretty for our anniversary, still figuring out the details"
```

F.5 LISTING JUDGE

The listing judge is blind to which listing is the target: it identifies the chosen listing from the transcript and grades it on its own merits. Whether the chosen listing is the target is scored separately and mechanically.

```
You are an impartial evaluator of one travel booking conversation. The target listing is
the intended best match for the guest's goal. Whether the agent recovered that exact target is
      scored
separately; here, grade how well the CHOSEN listing actually fits the goal on its own merits (
     a listing
that meets the goal as well as the target scores highest, a weaker one lower -- do not assume
     the chosen
listing equals the target).

## Definitions
- Realized booked location (the target locale): "{location}".
- Chosen listing: the listing the guest booked or, if no booking occurred, the surfaced
     listing they
  clearly selected or intended to book. If neither can be identified, the chosen listing is
     none.

## Evaluation inputs

### Hidden instruction (the guest's request)
{user_instruction}

### Conversation log
{conversation_log}

### Retrieved listings and their ACTUAL attributes
{retrieved_listings}

These include listings returned by the tools, whether or not the agent surfaced them to the
     guest.
Use their ACTUAL attributes to verify the chosen listing.

## Listing-level evaluation
First identify the chosen listing from the conversation, and map it to a listing_id from the
     retrieved listings above.
Resolve references such as shortened names, "option 2", or "it" using the conversation. Choose
      only a
listing that the agent actually surfaced or that the guest referred to in the conversation;
     never
select a listing solely because it appears in the retrieved-listings table. If no listing was
     booked or
clearly selected, the chosen listing is none and all listing-level axes are N/A.

### Listing-level axes

- LOCATION [HARD]: whether the chosen listing is in the realized booked location "{location}".
  Treat translations, spelling variants, neighborhoods within the same locality, and
     equivalent
  locality names as matches. Do not treat an entire broad region as equivalent to the specific
  realized locality.

- DATES [HARD]: whether the chosen dates match {checkin}-{checkout}
  (month and day; ignore any year mentioned).

- GUESTS [HARD]: whether the chosen listing can accommodate {adults} adults,
  {children} children, and {pets} pets.

- PRICE [CONDITIONAL]: whether the nightly price reasonably satisfies the price expectation
```

```
  expressed in the hidden instruction. Consider both the size of any overage and how strongly
  the budget was stated. Treat price as a hard constraint only when the instruction explicitly
  states the budget as strict; otherwise treat it as a soft preference.

- SOFT_FIT [SOFT]: whether the chosen listing satisfies the material negotiable preferences
  expressed in the hidden instruction and does not strongly violate a central one.

### CHOSEN_LISTING_FIT
Rate the chosen listing's overall fit to the goal on a 1-5 scale, considering LOCATION, DATES,
GUESTS, PRICE, and SOFT_FIT.
- 1: Severe mismatch or violation of a central hard requirement.
- 2: Multiple important mismatches or one major hard-constraint violation.
- 3: Partial fit with at least one material mismatch.
- 4: Strong overall fit with only minor trade-offs.
- 5: Excellent fit to the realized locality, hard constraints, and material preferences.
Weight [HARD] axes more heavily than [SOFT] axes.

## Output EXACTLY these lines
CHOSEN_LISTING: <listing_id or none>
LOCATION: PASS|FAIL|N/A - <brief>
DATES: PASS|FAIL|N/A - <brief>
GUESTS: PASS|FAIL|N/A - <brief>
PRICE: PASS|FAIL|N/A - <brief>
SOFT_FIT: PASS|FAIL|N/A - <brief>
CHOSEN_LISTING_FIT: <1|2|3|4|5|N/A> - <brief>
REASON: <one sentence>
```

F.6 GEOGRAPHIC JUDGE

The geographic judge scores the agent's search *process* from its queries, and is scored even when no listing is chosen.

```
You are an impartial evaluator of the geographic SEARCH PROCESS in one travel booking
conversation. The guest started from an anchor location, while the realized booking
     corresponds to a
target locale. Evaluate whether the agent appropriately adapted its geographic search from the
      anchor
toward the target locale, given the required search-scope type.

## Definitions

- Search anchor: "{anchor}", where the guest initially searched. This is an observed starting
     point,
  not necessarily a firm destination preference.
- Realized booked location (the target locale): "{location}".
- Search-scope type: "{search_scope_type}", which specifies the geographic adaptation required
      from
  the anchor to reach the target locale.

## Evaluation inputs

### Hidden instruction (the guest's request)

{user_instruction}

### The agent's geographic search queries, in order

{search_queries}

## Evaluation criteria

Evaluate the search process on two dimensions:

1. OPERATION --- Did the agent perform the geographic operation required by the search-scope
     type,
   such as staying local, expanding nearby, refining a broad region, or expanding farther away
     ?

2. TARGETING --- Was the adaptation coherent and directed toward the target locale rather than
   arbitrary or unrelated locations?

Use the target locale to judge whether the search adapted in the correct geographic direction
     and
covered the appropriate geographic area. The agent does not need to query the exact target
     locality
to receive full credit if its search coherently covers the target's surrounding area.

Base both judgments on the actual search queries.
```

```
### Required geographic operation [{search_scope_type}]

- precise -> keep the search within "{anchor}" and do not leave it without reason.
- nearby -> expand from "{anchor}" into coherent adjacent localities within the same local or
  metropolitan area.
- broad -> refine the broad region "{anchor}" into specific, coherent localities rather than
     staying
  at the region level or choosing an arbitrary location.

## GEO_SEARCH_SCOPE_ADAPTATION (0-2)

- 0: The required geographic adaptation is absent or clearly incorrect.
- 1: The agent performs the required type of geographic adaptation, but the search is
     incomplete,
  diffuse, weakly targeted, inconsistent, or directed toward the wrong part of the relevant
     region.
- 2: The search explicitly performs the required geographic operation and does so coherently
     and
  sufficiently toward the target locale.

Scope-specific guidance:

- precise:
  - 0: leaves the anchor locality or fails to search it.
  - 1: searches the anchor but mixes in unnecessary outside areas.
  - 2: remains consistently focused within the anchor locality.

- nearby:
  - 0: stays only at the anchor or jumps to unrelated or distant areas.
  - 1: expands locally but searches too broadly, inconsistently, or without clear direction.
  - 2: coherently searches plausible nearby localities toward the target locale.

- broad:
  - 0: remains at the broad-region level without meaningful refinement, or searches unrelated
     areas.
  - 1: refines into plausible localities but remains diffuse, arbitrary, or inconsistent.
  - 2: coherently narrows the broad region into plausible specific localities toward the
     target locale.

When the agent searches both relevant and unrelated areas, assign 1 unless the relevant search
clearly dominates and drives the trajectory toward the target locale.

Score the geographic process independently of whether the agent ultimately selected or booked
     the
correct listing.

## Output EXACTLY these lines

GEO_SEARCH_SCOPE_ADAPTATION: <0|1|2> - <brief>
REASON: <one sentence>
```

F.7 AGENT SYSTEM PROMPT

Shown with the `full` listing-detail mode, which is the mode used for every reported result.

```
## Role
You are a travel booking assistant. Help the guest find a stay from the catalog that matches
     their
request and, once they agree, book it. Prefer calling a tool over guessing.

## Tools
- `search_listings`: search the catalog for a destination locality (e.g., a city, town, or
     district) +
  date range + party, with optional filters (price cap, room type, required amenities, and
     booking
  policies: flexible_cancellation, instant_bookable, pets_allowed). Returns compact listing
     summaries.
- `get_listing_details`: fetch the full record for one listing by id --- location, room type,
     capacity, bedrooms, nightly price, amenities, ratings, and policies. Call it before
     recommending or booking a listing to verify it fits.
- `create_reservation`: book a listing for the dates + party once the guest agrees.

## Guidelines
- Search first. Only filter by attributes the guest actually stated --- do not invent
     constraints.
- Resolve any date given without a year to its next future occurrence; never search dates in
     the past.
```

```
- Inspect a listing's details before you recommend or book it, to confirm it fits the request.
- If a search returns nothing, loosen one filter at a time and try again rather than giving up
    .
- Book with `create_reservation` as soon as the guest picks a listing or says to book it.
- Ask a brief clarifying question only when you genuinely need it; otherwise act.

## Conversation Style
- Keep replies concise and friendly. Surface the most relevant 2-3 options, not the whole list
    .
- Never narrate your mechanics (tools, searches, "behind the scenes"). Talk only about the
    stays.
- Gently guide the guest toward a booking decision.
```

F.8 AGENT TOOL SCHEMAS

The three tool definitions passed to the agent, as JSON schemas.

```
[
  {
    "type": "function",
    "function": {
      "name": "search_listings",
      "description": "Search available listings for a destination + date range + party, with
     optional filters. The catalog is organized by named localities (e.g., cities, towns, or
     districts); the city field expects one complete locality name and uses exact matching, so
      do not append directional or descriptive text (e.g. \"X southeast\" or \"Y west coast\")
     , as it returns no matches. Filter values for room_type and required_amenities must be
     from the enumerated catalog vocab. The policy filters are booleans: flexible_cancellation
      (cancellation is flexible), instant_bookable (book instantly, no host request),
     pets_allowed.",
      "parameters": {
        "type": "object",
        "properties": {
          "checkin": {
            "type": "string"
          },
          "checkout": {
            "type": "string"
          },
          "guests": {
            "type": "integer"
          },
          "city": {
            "type": "string"
          },
          "max_price_per_night": {
            "type": "number"
          },
          "room_type": {
            "type": "string",
            "enum": [
              "entire_place",
              "private_room",
              "shared_room"
            ]
          },
          "required_amenities": {
            "type": "array",
            "items": {
              "type": "string",
              "enum": [
                "air_conditioning",
                "free_parking",
                "kitchen",
                "pool",
                "tv",
                "washer",
                "wifi",
                "workspace"
              ]
            }
          },
          "flexible_cancellation": {
            "type": "boolean"
          },
          "instant_bookable": {
            "type": "boolean"
          },
```

```
          "pets_allowed": {
            "type": "boolean"
          }
        },
        "required": [
          "checkin",
          "checkout",
          "city"
        ]
      }
    }
  },
  {
    "type": "function",
    "function": {
      "name": "get_listing_details",
      "description": "Full record for one listing: location, room type, capacity, bedrooms,
     price, amenities, ratings, policies.",
      "parameters": {
        "type": "object",
        "properties": {
          "listing_id": {
            "type": "string"
          }
        },
        "required": [
          "listing_id"
        ]
      }
    }
  },
  {
    "type": "function",
    "function": {
      "name": "create_reservation",
      "description": "Book a listing for the dates + party.",
      "parameters": {
        "type": "object",
        "properties": {
          "listing_id": {
            "type": "string"
          },
          "checkin": {
            "type": "string"
          },
          "checkout": {
            "type": "string"
          },
          "guests": {
            "type": "integer"
          }
        },
        "required": [
          "listing_id",
          "checkin",
          "checkout"
        ]
      }
    }
  }
]
```

Table 8: Locality-cue ladder, under VAGUENESS = L1, for all seven agents. Each NEARBY and BROAD task appears at all three rungs with geography, target and alternatives held fixed, so only the cue in the goal differs, and all three rungs come from a single run — making the rung comparison within-run. **Target** is Booked Target (mean±std over three runs), **Retr** target retrieval, **Rec|R** Booked Target conditional on the target having been retrieved, **Loc** the share of episodes booking in the target's locality, **Geo** Geographic Search Adaptation (0–2) and **Fit** Chosen Listing Fit (0–5). Each cell is 100 tasks × 3 repeats, up to 300 episodes; episodes that error are excluded rather than scored as failures, so a few cells carry slightly fewer. The **None** rung re-measures the base condition of Table 2 and the two agree within run-to-run variance. GPT-4.1 Mini runs through a moderated route, with 0.28% of its episodes excluded (Section 4.1).

Agent	Rung	Target	Retr	Rec|R	Loc	Geo	Fit
NEARBY							
Qwen3.5-35B-A3B	None	56.3±4.0	58.3	96.6	60.7	1.38	3.46
	Direction	49.7±4.2	53.0	93.7	57.3	1.34	3.24
	Locality	**84.7**±3.8	87.7	96.6	89.3	1.70	4.25
Qwen3.6-35B-A3B	None	53.7±3.5	55.7	96.4	57.0	1.16	3.28
	Direction	52.0±2.0	53.7	96.9	58.0	1.19	3.24
	Locality	**85.3**±1.5	87.3	97.7	89.3	1.65	4.29
Qwen3.8-27B	None	48.3±3.8	49.7	97.3	53.3	1.12	3.29
	Direction	49.3±1.5	49.7	99.3	58.3	1.33	3.33
	Locality	**86.0**±1.0	88.3	97.4	88.3	1.58	4.28
Haiku 4.5	None	46.7±1.5	49.0	95.2	55.0	1.42	3.31
	Direction	48.3±3.5	50.7	95.4	57.3	1.51	3.35
	Locality	**81.7**±7.4	86.0	94.9	87.0	1.71	4.22
GPT-4.1 Mini	None	50.6±4.4	52.2	96.8	56.3	1.32	3.36
	Direction	49.0±1.8	50.3	97.3	56.4	1.33	3.24
	Locality	**84.7**±1.5	86.0	98.4	87.3	1.67	4.24
GPT-5 Mini	None	56.0±1.0	59.7	94.0	63.3	1.42	3.54
	Direction	57.7±3.1	60.0	96.1	64.3	1.46	3.54
	Locality	**77.0**±2.6	79.7	96.6	82.3	1.64	4.24
GPT-5	None	65.3±4.0	71.7	91.1	71.7	1.58	3.73
	Direction	67.0±1.0	72.7	92.2	75.7	1.65	3.79
	Locality	**83.3**±1.5	88.0	94.7	87.7	1.79	4.24
BROAD							
Qwen3.5-35B-A3B	None	64.7±1.5	67.3	96.0	67.0	1.61	3.62
	Direction	66.0±3.6	71.7	92.1	71.3	1.63	3.71
	Locality	**90.0**±0.0	92.7	97.1	93.3	1.86	4.50
Qwen3.6-35B-A3B	None	64.7±3.8	67.7	95.5	68.0	1.55	3.62
	Direction	65.0±5.6	69.7	93.2	68.7	1.57	3.53
	Locality	**93.3**±1.5	95.3	97.9	94.7	1.87	4.51
Qwen3.8-27B	None	70.0±1.7	72.7	96.3	72.7	1.61	3.86
	Direction	68.3±1.2	73.0	93.6	70.7	1.57	3.79
	Locality	**91.0**±3.0	96.0	94.8	96.3	1.78	4.54
Haiku 4.5	None	59.0±2.6	62.7	94.1	62.0	1.48	3.39
	Direction	61.7±3.2	65.7	93.9	65.0	1.50	3.51
	Locality	**87.3**±3.1	91.7	95.3	92.0	1.85	4.42
GPT-4.1 Mini	None	62.2±5.0	65.3	95.3	66.3	1.56	3.48
	Direction	50.0±1.5	52.4	95.5	54.1	1.35	2.93
	Locality	**91.3**±1.5	94.7	96.5	94.0	1.89	4.45
GPT-5 Mini	None	69.0±3.4	73.1	94.5	72.7	1.72	3.85
	Direction	71.7±0.6	76.3	93.9	76.3	1.79	3.90
	Locality	**87.0**±1.0	91.0	95.6	92.7	1.87	4.48
GPT-5	None	75.7±1.2	78.7	96.2	79.0	1.76	3.94
	Direction	74.3±3.5	83.0	89.6	80.3	1.82	4.01
	Locality	**86.7**±3.2	94.3	91.9	92.3	1.86	4.46

Table 9: Three generations of one model family under the identical protocol, means over three runs. Geographic Search Adaptation is 1.61, 1.59 and 1.59 under L1 and moves by at most 0.05 under L2. ΔTarget is the L1$\rightarrow$L2 change; its propagated interval is ± 2.2, ± 3.0 and ± 2.5 respectively.

	L1		L2		
Model	**Target**	**Booked Any**	**Target**	**Booked Any**	**ΔTarget**
Qwen3.5-35B-A3B (MoE)	70.8$\pm$0.4	92.8$\pm$0.4	59.2$\pm$2.2	76.8$\pm$0.5	-11.6
Qwen3.6-35B-A3B (MoE)	70.4$\pm$1.8	88.3$\pm$0.9	66.6$\pm$2.4	86.2	-3.8
Qwen3.8-27B (dense)	71.8$\pm$1.8	99.7$\pm$0.0	68.8$\pm$1.7	97.3	-3.0

Table 10: Booked Target (%) after LoRA supervised fine-tuning of Qwen3.6-35B-A3B on 150 curated demonstrations, one per task of the fine-tuning split, measured on the held-out fine-tuning evaluation. All rows, including the base, are scored on that same evaluation, so the comparison is within-table. Gains concentrate on NEARBY and BROAD.

Training split	**Curation**	**Overall**	**PRECISE**	**NEARBY**	**BROAD**
–	Base	69.3$\pm$1.1	89.5$\pm$0.5	54.9$\pm$2.4	63.3$\pm$0.6
Regular	Outcome	**73.1**$\pm$1.1	90.5$\pm$1.0	**61.6**$\pm$3.0	**67.0**$\pm$1.3
Regular	Process-aware	72.4$\pm$0.9	**92.2**$\pm$0.3	59.5$\pm$2.0	65.3$\pm$1.8
Country-disjoint	Outcome	71.1$\pm$2.3	90.2$\pm$2.1	57.3$\pm$4.6	65.8$\pm$1.3