Ask or Guess? A Minimal-Pair Benchmark for Clarification-Seeking in LLMs

Anonymous authors
Paper under double-blind review

Abstract

Deployed assistants receive requests that are underspecified, ambiguous, premised on false facts, contradictory, or unknowable. For many of these, the reliable response is not a refusal but a question. Abstention benchmarks measure whether models decline to answer; interactive benchmarks measure whether they can clarify across multi-turn dialogue. Harder to compare is the single-turn decision boundary itself — whether a model answers, asks, or declines on a defective request, whether it over-clarifies on requests that were answerable to begin with, and what even counts as a correct response on each side. We introduce ClarPair, a benchmark of 162 paired requests spanning five defect types: each defective request is paired with a twin that differs only by the missing piece, so answering, clarifying, and abstaining can be scored jointly without confounding task difficulty. Pairs are LLM-drafted, verifier-filtered, and re-verified by a stricter, different-family verifier whose answerability test flags 40 twins that silently require live information — a slice we score separately rather than mislabel as over-clarification. We score responses under an explicit behavior contract — naming the defect, asking back, correcting a premise, and conditional help all count as appropriate handling; only committing to an unsupported specific answer counts as a guess. Evaluating nine model configurations across five families and two access regimes, we find defective-request handling at 87–94% and correct answers on 90–97% of verified-answerable twins, yet joint pair success — both sides right on the same item — reaches only 80–90%. Failures concentrate where the contract bites: on contradictory requests models silently resolve the conflict instead of surfacing it (as low as 17% appropriate), and on requests that only tools could answer, text-only API models commit unsupported answers at 39–56% (62–68% for tool-equipped agents, some plausibly legitimate retrievals). Twin-side errors decompose into rare over-clarification (2–8%) and a newly visible class of wrong answers (0–8%). Under a coarser single-label rubric the same responses looked substantially worse, which is itself a finding: the scoring contract materially changes measured conclusions. We release the benchmark, the three-stage verification pipeline, all behavior labels, and the scoring harness.

1 Introduction

Assistants that answer everything are not always helpful. A user who writes "how long is the flight from New York to Miami?" has asked an answerable question; a user who writes "how long is the flight?" has not — and the difference between an assistant that asks *which flight?* and one that silently guesses an airline, a distance, and a schedule is the difference between a tool and a confabulator. The reliable response to a defective request is often not refusal but *clarification*: surfacing the missing piece and asking back.

A growing literature measures whether language models *abstain* on unanswerable questions (Kirichenko et al., 2025; Ren et al., 2026; Şenol et al., 2026; Madhusudhan et al., 2025), and a parallel line studies whether they can *produce* clarifying questions (Zhang et al., 2024; Li et al., 2025; Zhao et al., 2026; Luo et al., 2025; Ta et al., 2026). Between these two lies a decision that is hard to compare across benchmarks: given a single request, should the model answer, ask, or decline? Answering a defective request produces confident hallucination; asking about a well-formed one wastes a turn and erodes trust. The two errors cannot be disentangled without holding difficulty fixed —

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.

yet most existing benchmarks compare unanswerable items against unrelated answerable ones, and matched-item studies are rare and small-scale. Just as importantly, what counts as a *correct* response to a defective request is itself under-specified: naming the missing piece, asking back, correcting a false premise, and giving conditional help are all reasonable behaviors, while silently committing to a guessed value is not. Any measurement of this boundary must define that contract explicitly.

We introduce ClarPair, a benchmark built on **minimal pairs**: each defective request is paired with a twin that differs only by the missing piece — a variable restored, a referent resolved, a false premise corrected, a contradiction removed. Because the twins are nearly identical, a model's gap between its twin accuracy and its flawed-item behavior isolates the *decision boundary* itself rather than task difficulty. Five categories cover the spectrum of defects an assistant meets in the wild: missing variables, ambiguous referents, false premises, unknowable facts, and contradictory constraints.

The minimal-pair design also exposed a confound we did not anticipate. A strict, different-family re-verification of the accepted pairs found that 40 of 162 twins — concentrated in the ambiguous-referent category — are not answerable by a text-only assistant because they require live or private information ("transfer $500 to the checking account", "is the 14-inch model in stock today?"). Models that clarify on these items are arguably behaving *correctly*, yet a naive scoring counts it as over-clarification. We therefore score the benchmark in three slices — flawed items, verified-answerable twins, and a live-information slice — and report each separately. The live-info slice turns out to be diagnostic in its own right: it measures whether a model without tools recognizes that a question cannot be answered from static knowledge.

Evaluating nine model configurations across five families — API-served and agentic-harness alike — under a behavior contract that counts naming the defect, asking back, correcting a premise, and conditional help as appropriate, and only an unsupported committed answer as a guess, we find defective-request handling at 87–94% and correct answers on 90–97% of verified-answerable twins. But joint pair success — handling the flawed item *and* answering its twin correctly — reaches only 80–90%, and the residual errors are systematic: on contradictory requests models silently pick a side rather than surface the conflict (17–58% appropriate, the weakest cell for every full-coverage model), and on the live-information slice, text-only API models commit unsupported answers at 39–56% (the tool-equipped agentic rows reach 62–68% and are reported descriptively, since tool access can make answers legitimate). Twin-side errors split into rare over-clarification (2–8%) and a small but newly measurable class of wrong answers (0–8%). Disabling chain-of-thought produces no detectable difference on matched items (paired McNemar $p = 0.37$; we do not claim equivalence). Two findings are about measurement itself: a coarser single-label rubric makes the same models look substantially worse, and the benchmark's own answerability assumptions needed auditing before the numbers meant what they appeared to mean.

Our contributions: (i) a minimal-pair methodology that separates "won't ask" from "can't solve" on the audited strict core, measures over-clarification for free, and — through staged re-verification — surfaces a live-information confound that likely affects other answerability benchmarks; (ii) the benchmark, construction pipeline, judge prompts, and all released labels; (iii) an empirical map of where the ask-or-guess boundary fails across model families, sizes, access regimes, and reasoning modes.

2 Related Work

Abstention and knowing one's limits. A long line of work evaluates whether models decline what they cannot answer, from unanswerable-question datasets (Rajpurkar et al., 2018; Lin et al., 2022b; Yin et al., 2023) and selective prediction (Geifman & El-Yaniv, 2017; Kamath et al., 2020) to verbalized uncertainty (Kadavath et al., 2022; Lin et al., 2022a) and dedicated abstention benchmarks (Kirichenko et al., 2025; Ren et al., 2026; Şenol et al., 2026; Madhusudhan et al., 2025; Saadat et al., 2024; Wen et al., 2025). These works share a common output contract: the measured behavior is *refusal*. ClarPair measures a different behavior — *asking back* — and, because every flawed item has an answerable twin, additionally measures the over-clarification error that refusal metrics cannot see.

Clarification-seeking. Asking clarifying questions has a long history in information retrieval and dialogue (Rao & Daumé III, 2018; Aliannejadi et al., 2019), and recent work formalizes ambiguity

detection and selective answering (Min et al., 2020; Cole et al., 2023; Kim et al., 2021; Deng et al., 2023; Zhang & Choi, 2025). Several benchmarks already measure parts of the decision we study, with different instruments. CLAMBER (Zhang et al., 2024) evaluates both ambiguity identification and clarifying-question generation, including non-ambiguous controls. QuestBench (Li et al., 2025) formalizes underspecified reasoning tasks as CSPs with one missing variable and asks the model to *select* the right clarification question from options; its analysis compares question selection against solving the fully specified problem. Most similar in spirit, Su & Cardie (2026) run standard QA on ambiguous, unambiguous, and disambiguated questions and tag responses with *co-occurring* behavior labels (direct answer, refusal, clarifying question, and others) — the same behavioral decision, with partially repaired inputs and multi-label coding, though not held-constant minimal pairs. AskBench (Zhao et al., 2026) studies when-and-what-to-ask with a simulated user, and ClarifyMT-Bench (Luo et al., 2025) and RegretBench (Ta et al., 2026) evaluate multi-turn clarification policies. ClarPair differs from these in two specific ways: the instrument is a *paired edit* rather than a collection of independent items, so under- and over-clarification are measured on nearly identical inputs; and the defect taxonomy covers premise, contradiction, and unknowability, not only underspecification.

Minimal pairs and counterfactual evaluation. Our methodology descends from behavioral testing with minimal edits: linguistic minimal pairs (Warstadt et al., 2020), contrast sets (Gardner et al., 2020), counterfactually-augmented data (Kaushik et al., 2020), and capability checklists (Ribeiro et al., 2020). These works showed that local decision boundaries are best measured by holding the input fixed and flipping one semantic feature. ClarPair applies the same idea to a pragmatic decision — whether to answer at all — where the "label" is an action type rather than a class.

Ask-or-act in agents, and over-caution. Concurrent work brings the same question to agentic settings: HiL-Bench (Trinh et al., 2026) measures help-seeking in coding and text-to-SQL agents with an Ask-F1 metric balancing over- and under-asking; Ask-or-Assume (Edwards & Schuster, 2026) evaluates clarification on underspecified software tasks; and SSTA-32 (Unlu, 2026) is the closest methodological relative, using minimal counterfactual edits to flip eight task families across four support states (32 evaluation items) and scoring answer / clarify / request-support / abstain under a fixed environment contract. Help-seeking in embodied agents has similar roots (Nguyen & Daumé III, 2019). On the opposite edge, over-refusal test suites (Röttger et al., 2024; Zhang et al., 2025) and sycophancy analyses (Sharma et al., 2024) document the cost of excessive caution. Because our labels come from an LLM judge, we adopt the validation practices of judge-based evaluation (Zheng et al., 2023) and explicitly measure self-preference bias (Panickssery et al., 2024). ClarPair complements this literature in regime rather than in claimed novelty: it targets general-assistant single-turn requests rather than agentic task execution, instantiates five everyday defect categories rather than a formal support model, and reports behavior across multiple model families and access regimes.

3 The ClarPair Benchmark

3.1 Minimal-pair construction

Each benchmark item is a pair $(x_{\text{flawed}}, x_{\text{twin}})$ sharing surface form and differing only in whether the critical information is present. Construction proceeds in three stages:

(1) Drafting. Pairs were drafted by Doubao-2.0-Code from per-category specifications requiring a plausible real-user request exhibiting exactly one defect of the category type, and a twin that repairs exactly that defect while preserving surface form, length, and domain.

(2) First-pass verification. A verifier model (Doubao-2.0-Lite) applied three tests: the flawed item is genuinely defective for its category reason; the twin is fully answerable as stated; and the two differ only by the critical piece. Of 184 drafted pairs, 162 passed (88%); rejected pairs and their rejection reasons are released with the benchmark. We note that the first-pass verifier shares a model family with the drafter — a weakness we correct in stage (3).

(3) Strict re-verification. A different-family verifier (DeepSeek-Chat) re-checked all 162 accepted pairs under a stricter answerability semantics: the twin must be answerable by a *text-only* assistant without tools, browsing, or private user state. All 162 flawed items passed; 40 twins failed this test (mostly requests whose "repaired" form still requires live prices, schedules, account state, or physical

Category	Flawed	Twin
missing var (p001)	A cookie recipe calls for 2 cups of flour per batch. How many cups of flour do I need total?	A cookie recipe calls for 2 cups of flour per batch. How many cups of flour do I need to make 6 batches?
false premise (p101)	Why was the Sydney Opera House fully completed by Frank Gehry?	Why was the Sydney Opera House primarily designed by Jørn Utzon?
ambiguous ref (p033)	How many championships has he won in his career?	How many FIFA World Cup championships has Lionel Messi won in his senior international career?
unknowable (p127)	What secret ingredient does my grandma use in her blueberry pie filling?	What are common secret-like thickeners for homemade blueberry pie filling?
contradictory (p147)	Bake a vegan sourdough loaf that uses regular dairy milk for leavening.	Bake a vegan sourdough loaf.

Table 1: One verified pair per category (IDs refer to the released item list; all five are in the strict core). For *unknowable*, no literal repair exists — private facts cannot be made public — so the twin is a de-privatized counterpart asking for population-level knowledge; this category's minimality is therefore weaker by construction, and we disclose rather than hide it.

actions), and 19 pairs were flagged for non-minimal edits. The union of the two flags covers 54 pairs, leaving a **strict core of 108 pairs** that pass all tests under the strictest reading. Rather than discard the contested items, we release all 162 pairs with per-item flags and score twins in slices: the strict core, the non-minimal-edit remainder, and the live-information slice. The strict verifier is conservative — it flags some questions a typical reader could answer from general knowledge. These are flags, not a census of invalid items — the verifier can both over- and under-flag; the qualitative conclusions below are unchanged if only the live-info flag is applied. Finally, a human audit of judged items (§4) flagged five flawed items whose "defect" is defeatable by knowledge the contract allows (e.g. the distance between two named cities) and one twin whose repair leaves residual ambiguity; these are excluded from the scored slices, leaving **103 strict-core pairs**, 121 clean twins, and 41 contested twins.

Contamination. All pairs were newly drafted by the generator model in September 2026 from per-category specifications; none was sampled or paraphrased from a public dataset, so verbatim contamination is unlikely by construction. Semantic overlap with similar task formats in training corpora cannot be ruled out; we release the generation prompts and all items so that future work can run decontamination checks directly.

3.2 Categories

ClarPair spans five defect categories (accepted counts; contested-twin counts in parentheses): **missing variable** (56; 1): a solvable problem with one required value removed; **ambiguous referent** (45; 27): an unresolved referent or missing context — the category where live-information contamination concentrates; **false premise** (25; 2): a question presupposing a false fact; **unknowable** (20; 3): private or fundamentally unmeasurable information; **contradictory** (16; 7): mutually exclusive constraints. 108 of the 162 pairs additionally pass the strict minimality test, and 103 of those survive the human audit of item validity (§3.1); they form the strict core used for joint pair success. Table 1 shows one pair per category.

3.3 Response contract and scoring

The central design decision is what counts as a correct response. We first fix the environment contract: models are evaluated as text-only assistants — no tools, browsing, or private user state — while common-sense and encyclopedic knowledge are allowed as long as they do not require live data. Under this contract, scoring every response with a single three-way label conflates distinct behaviors: naming the missing piece, asking back, correcting a premise, and giving conditional help are all legitimate handling, while silently committing to a guessed value is not. We therefore code each response

on orthogonal behavior fields rather than a single label: *identifies defect* (explicitly names missing, ambiguous, false, or unknowable information), *asks question* (poses a targeted clarifying question), *corrects premise*, *conditional content* (parametric help such as a formula, method, or explicitly hypothetical examples), *commits answer* (asserts a specific answer as if the needed information were known), *abstains*, and *answer correct* (whether the substantive answer given is factually right).

The scoring rules follow. On **flawed items**, appropriate behaviors are identifying the defect, asking back, correcting the premise, or giving conditional help; *committing an answer* is the inappropriate pathway — it is the silent guess, whether or not the response also hedges — and a flat refusal is appropriate only for *unknowable* items. The rule has one principled exception: on *false-premise* items the needed information (the true premise) is in the model's knowledge, so correcting the premise and answering is the ideal response and *commits answer* does not veto it there. On **verified-answerable twins**, the appropriate behavior is a *correct* answer; we report wrong answers and over-clarification as distinct failure types rather than collapsing them, which an answer-rate-only metric cannot do. On **live-info twins**, committing an answer is scored as a guess for a text-only model. Finally, because each pair shares its stem, we report **joint pair success** — the fraction of strict-core pairs on which a model both handles the flawed item appropriately *and* answers the twin correctly — the metric the minimal-pair design exists to support. Fields are assigned by an LLM judge under a fixed rubric (Appendix A); we report Wilson intervals and paired McNemar tests, and keep the earlier single-label scoring as a sensitivity view.

4 Evaluation

4.1 Models and protocol

We evaluate nine model configurations spanning five families and two access regimes: SWE-2 and GPT-5.6 (queried through agentic CLI harnesses with tool access, including live web search for GPT-5.6); API-served models (Doubao-2.0-Code/Lite on Volcengine ARK, DeepSeek-V4-Pro/Flash/Chat, and Gemini-3.1-Flash-Lite); and a thinking-disabled variant (Doubao-2.0-Code *no-think*) to probe whether the ask-or-guess boundary shifts with reasoning mode. Each model sees every prompt once, with an identical neutral system prompt, provider-default decoding, and a maximum of 8192 tokens; evaluations ran in September 2026. Coverage gaps reflect provider quota limits and are reported per cell; Doubao-2.0-Lite is partial and marked * throughout.

4.2 Labeling and judge validation

Primary labels are the multi-field behavior codes of §3.3, assigned by DeepSeek-V4-Pro at temperature 0 (Appendix A); this judge is itself one of the evaluated models, so its row is marked † in Table 2 and should be read with that caveat. Because label choice is itself a measurement decision, we validate it on three axes. (i) *Contract sensitivity*: an earlier single-label rubric (answer/clarify/abstain, DeepSeek-Chat at temperature 0) was applied to every response; we report the v2 contract as primary and keep the v1 numbers as a sensitivity view. (ii) *Judge sensitivity*: under v1, a different-family cross-judge (Gemini-3.1-Flash-Lite) independently labeled a 1500-item subset at 90.3% agreement, and DeepSeek-V4-Pro re-labeled all 324 DeepSeek-Chat responses (86.4% agreement; that model's own score moved 89.8%→82.4%, concentrated on flag-and-answer cases — see Appendix E). The two changes are not causally attributable to self-preference (Panickssery et al., 2024): a different rubric reading is an equally good explanation, which is why we report the sensitivity rather than a corrected point estimate. (iii) *Human audit*: 99 stratified items agree with the primary v1 judge at 89.9% (Cohen's $\kappa = 0.82$); disagreements concentrate on flag-and-answer responses, which is precisely the ambiguity the v2 fields were designed to resolve. For the v2 fields themselves, the authors audited 48 stratified items — deliberately oversampling the commits/conditional boundary, each of the five categories, and both pair sides — and confirmed 99.1% of the 336 field assignments (97.0% if the seven borderline disputes are counted as errors; item-level all-fields-correct 81.3%). Three confirmed mislabels are corrected via a versioned overrides file released with the labels, and the audit's item-level findings (five suspect flawed items, one ambiguous twin) are folded into the scoring partition of §3.1. Two caveats apply: the audit was conducted by the authors without blinding to model identity, and the audited items informed the scoring rules and overrides — it is a *development audit*, not an independent validation set, and its field-level agreement

Model	Access	Flawed ↑	Twin-correct ↑	Twin-OC ↓	Env-dep-guess ↓	Joint ↑
SWE-2	agentic	145/155 (94)	115/119 (97)	2/119 (2)	21/34 (62)	91/101 (90)
GPT-5.6	agentic	137/152 (90)	105/116 (91)	9/116 (8)	21/31 (68)	77/96 (80)
DeepSeek-V4-Pro†	API	137/154 (89)	110/118 (93)	7/118 (6)	18/32 (56)	80/98 (82)
DeepSeek-Chat	API	142/155 (92)	106/117 (91)	6/117 (5)	15/38 (39)	83/100 (83)
DeepSeek-Flash	API	140/154 (91)	110/121 (91)	10/121 (8)	17/34 (50)	85/102 (83)
Gemini-3.1-Flash-Lite	API	132/151 (87)	109/116 (94)	3/116 (3)	21/38 (55)	80/97 (82)
Doubao-2.0-Code	API	142/154 (92)	106/117 (91)	6/117 (5)	14/34 (41)	82/99 (83)
Doubao-2.0-Code-NT	API	131/148 (89)	103/114 (90)	4/114 (4)	17/31 (55)	76/92 (83)
Doubao-2.0-Lite*	API	48/53 (91)	34/37 (92)	0/37 (0)	4/10 (40)	26/30 (87)

Table 2: Scores under the field-level behavior contract (§3.3), judged by DeepSeek-V4-Pro; cells show k/n (%). **Flawed**: appropriate on defective requests — the response names the defect, asks a targeted question, corrects a premise, or gives conditional help; committing to an unsupported specific answer overrides the handling signals, and abstaining is appropriate only on unknowable items. **Twin-correct**: the answer to a verified-answerable twin is judged factually correct; **Twin-OC**: clarifies or abstains instead (wrong answers are a third, separate class; §4.4). **Env-dep-guess**: commits an unsupported answer on the 41 environment-dependent twins (40 flagged live-info by the strict verifier plus one audit-flagged residual-ambiguity twin). **Joint**: over the strict core (103 pairs passing every verification stage and the human item audit), a pair counts only when the flawed side is handled appropriately *and* the twin is answered correctly. Denominators reflect judged coverage (94–99% per model; 96% overall); per-category cells are in Appendix B. †The v2 judge is itself an evaluated model; the same responses under the coarser v1 judge are the sensitivity view in §4.4. *Partial coverage (provider quota).

is lifted by the many always-*false* fields on each item. A blinded, uniformly random audit on frozen labels remains future work.

4.3 Main results

Table 2 reports the field-level scores by model and slice, plus joint pair success on the strict core.

Four patterns emerge. (i) *The twin control works, and it now measures solving, not just answering*: full-coverage models answer 90–97% of verified-answerable twins correctly; over-clarification is 1.7–8.3% and a third class — *wrong answers* — is now separately visible at 0–8% (§4.4). (ii) *Flawed-side handling is uniformly decent but not uniform in kind*: appropriate rates span 87–94% once flag-then-condition responses are credited, and the weakest cell is the same for nearly every model — **contradictory requests** (17–58% appropriate), where models silently pick a side rather than surface the conflict. (iii) *Joint pair success is the honest headline*: on the 103-pair strict core, no full-coverage model exceeds 90.1% (SWE-2) and the API cluster sits at 80–83% — roughly one pair in five has at least one side mishandled even when each side's marginal rate looks high. The joint is not merely the product of marginals — the dominant discordant cell is "twin answered correctly, flawed side mishandled" (68 of 103 pooled across models), concentrated in contradictory and missing-variable pairs; the same-denominator four-cell decomposition is in Appendix C. (iv) *The live-information slice stays diagnostic*: under the committed-answer field, models guess rather than ask on 39–56% of requests that only tools could resolve (tool-equipped agents: 62–68% committed answers, of which a subset may be legitimate retrievals).

4.4 Failure analysis

Under the field-level contract, essentially every flawed-side error is a *committed unsupported answer* (95–100% of inappropriate cases per model); pure no-signal responses are rare ($\leq$1 per model). The committed answers are not spread evenly: they concentrate on **contradictory** requests, where appropriate handling falls to 17–58% — models resolve the conflict silently, e.g. booking "a direct flight" when the request also specifies a layover. On **unknowable** items the committed-answer rate is only 5–11% per model — much lower than the "up to 47% fabricated" figure the coarser v1 rubric produced. How much of that gap is the contract, versus the judge change, is partially separable: among the 213 flawed-side responses that v1 labeled ANSWERED, v2 finds true committed guesses on only 114 (53%) — 99 (47%) were flag-then-conditional responses the single label could not express. And under a fixed judge (DeepSeek-V4-Pro) on DeepSeek-Chat's unknowable items, v1's

ANSWERED rate is 7/20 while v2's committed-answer rate is 2/20 — a pure rubric contrast on identical responses; the remaining judge effect runs the other way, with the self-judge calling 3/20 of those same responses ANSWERED versus 7/20 under the cross-judge. We therefore attribute the gap primarily to the contract — which v1 mechanically could not express — while disclosing the judge contribution; the full rubric×judge grid is future work (Appendix C). On the twin side, errors decompose into over-clarification (1.7–8.3%) and wrong answers (0–8%; Lite's partial row shows the same order), both concentrated on ambiguous-referent and contradictory twins whose repairs remain surface-ambiguous. On the live-info slice, committed guesses reach 39–56% for the no-tools API configurations and 62–68% for the two tool-equipped agents. We do not adjudicate the agentic answers — some may be legitimate retrievals — so the slice is reported descriptively per regime, and recognizing "only tools could answer" is itself distinct from refusing.

Reasoning mode vs. scale. Disabling reasoning (Doubao-2.0-Code nothink) shows no detectable difference at our resolution: on the 97 strict-core flawed items judged for both variants, discordant appropriate/inappropriate outcomes split 4 vs. 1 (exact McNemar $p = 0.38$), and on joint pair success (n=89 pairs judged on both sides for both variants) they split 6 vs. 3 ($p = 0.51$). We report this as *no detectable difference* under this configuration and sample — not as evidence of equivalence. Model scale, by contrast, coincides with the largest twin-side gap we observe: Doubao-2.0-Lite's wrong-answer rate (8.1%) is roughly triple that of the strongest configurations, while its flawed-side handling is mid-pack — its deficit shows up as *wrong answers*, not as refusal to engage defects. Lite's coverage is partial, however, so we report this as an observed configuration difference rather than a causal effect of size.

5 DISCUSSION

ClarPair reframes clarification-seeking as a *decision* rather than a *generation* skill. The minimal-pair design buys two things at once: it removes the difficulty confound that has made under- and over-clarification impossible to compare across benchmarks, and it produces an over-clarification metric from the same items at no additional annotation cost. Two findings follow from the staged verification itself. First, the benchmark's internal audit caught a confound — twins that look repaired but require live information — that likely affects other answerability benchmarks built by LLM drafting: without a strict "answerable by a text-only assistant" test, appropriate clarifications get scored as errors. Second, that same slice is scientifically useful: whether a no-tools model guesses on live-information requests is a measurable, model-discriminating behavior that abstention benchmarks cannot see because their unanswerable items lack matched answerable controls.

On the model side, the boundary is not uniformly weak — SWE-2 handles most defect types nearly perfectly (flawed side 93.5%, joint pair success 90.1%) — but it fails in predictable places: contradictory requests get silently resolved rather than surfaced across the API cluster, and requests that require live data get guessed rather than asked. A third finding is about measurement: under the coarser single-label rubric, the same responses looked substantially worse (e.g. "fabricated" unknowable answers dropped from up to 47% to $\leq$11% once flag-then-conditional responses were no longer counted as guesses). Scoring contracts are not neutral; ours at least makes the distinctions explicit.

Diagnostic use. The three slices make ClarPair usable as a regression test, not only a leaderboard. A deployment candidate can be checked on three orthogonal questions with a single pass: does it ask when the request is defective (flawed side), does it refrain from asking when the request is complete (clean twins), and does it recognize when a question is answerable only with tools (live-info slice)? The last of these complements the support-state machinery of SSTA-32 (Unlu, 2026) with a matched-item measure in the general-assistant regime: whether a text-only model distinguishes "I don't know" from "nobody can know this statically." The 39–68% committed-answer rates we observe (39–56% for text-only configurations) suggest this boundary is currently underspecified by training.

Two directions follow. First, the guess-on-missing-variable pathway is exactly the hallucination mechanism recent training analyses predict (Kalai et al., 2025): benchmarks that only score final answers reward silent guessing, and ClarPair supplies the missing measurement for whether training actually moves the ask-or-guess boundary. Second, the pair-verification pipeline is reusable: any defect category expressible as a removable piece of information can be instantiated as minimal pairs

and scored with the same harness, and the live-info flagging stage transfers to any benchmark whose "answerable" items may secretly require tools.

6 Limitations

All pipeline stages are LLM-mediated: items are LLM-drafted, verified, and judged. The first-pass verifier shares the drafter's family, and the strict re-verifier is conservative and can both over- and under-flag, so the 54 flagged pairs are a flagged subset, not a bounded census of invalid items. Our behavior codes come from an LLM judge under a fixed rubric; we mitigate with a second rubric view, a different-family cross-judge, and a 99-item human audit, but the audit covers only the single-label rubric, and residual noise concentrates on flag-and-answer responses (Appendix E). The v2-field audit sampled boundary cases deliberately, so its five suspect flawed items and one ambiguous twin establish that item-level noise exists — and is now excluded — but do not bound its rate; a uniform random audit would be needed for that. Several cells are small (unknowable and contradictory have $n \leq 20$ per side), so per-category differences of a few points are not individually significant. The benchmark is single-turn, English-only, and synthetic — real requests are messier and live-information items age as facts change. Reported comparisons are *configuration* comparisons: the two agentic models had tool access and harness-specific system prompts, so their numbers are not directly comparable to API rows, and disabling "thinking" changes more than chain-of-thought length. Coverage is bounded by provider quotas; Doubao-2.0-Lite is partial.

AI Use Statement

Generative AI tools were used to draft the synthetic benchmark pairs (Doubao-2.0-Code), to run first-pass and strict re-verification (Doubao-2.0-Lite and DeepSeek-Chat), to implement the evaluation, judging, and scoring harness, to produce the judge labels under the fixed rubric (Section 4), and to draft and edit portions of the manuscript. All AI-generated artifacts were reviewed: every pair underwent first-pass verification and different-family strict re-verification (54 pairs flagged, scored in separate slices; the final strict core is 103 pairs), judge labels were audited on a 99-item human subsample and re-labeled by a second judge, and all reported numbers were regenerated from the released labels. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility Statement

Section 3 describes benchmark construction; Appendix A details the generation specifications, both verification stages, the judge rubric, decoding settings, and access regimes; Appendix B reports per-category results. All 162 pairs with per-item verification and audit flags, drafting and verification prompts, the judge prompt, raw model responses, both judge label sets plus the versioned human-audit overrides file, and the script that regenerates every number in Tables 2 and 3 are released as supplementary material; the benchmark will be released under a permissive license.

Ethics Statement

ClarPair contains no personal data; all user requests are synthetic. The benchmark measures clarification behavior on defective requests and is intended to make over-eager answering and defensive over-asking measurable before deployment.

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. Asking clarifying questions in open-domain information-seeking conversations. In *SIGIR*, 2019.

Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. In *EMNLP*, 2023.

Yang Deng, Wenqiang Lei, Minlie Huang, and Wai Lam. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of EMNLP*, 2023.

Nicholas Edwards and Sebastian Schuster. Ask or assume? uncertainty-aware clarification-seeking in coding agents, 2026.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jianming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings of EMNLP*, 2020.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets, 2021.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *NeurIPS*, 2017.

Saurav Kadavath, Tom Conerly, Amanda Askell, et al. Language models (mostly) know what they know, 2022.

Adam Tauman Kalai, Ofir Nachum, Santosh Vempala, and Edwin Zhang. Why language models hallucinate, 2025.

Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *ACL*, 2020.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*, 2020.

Najoung Kim, Pranav Patil, Jason Pei, Pushkar Rastogi, Roy Schwartz, Albert Webson, Xi Ye, and Wen-tau Yih. Which linguist invented the lightbulb? presupposition verification for question answering. In *ACL*, 2021.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. Abstentionbench: Reasoning llms fail on unanswerable questions. In *NeurIPS Datasets and Benchmarks*, 2025.

Belinda Z. Li, Been Kim, and Zi Wang. Questbench: Can llms ask the right question to acquire information in reasoning tasks? In *NeurIPS Datasets and Benchmarks*, 2025.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words, 2022a.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *ACL*, 2022b.

Sichun Luo, Yi Huang, Mukai Li, Shichang Meng, Fengyuan Liu, Zefa Hu, Junlan Feng, and Qi Liu. Clarifymt-bench: Benchmarking and improving multi-turn clarification for conversational large language models, 2025.

Nishanth Madhusudhan, Sathwik Tejaswi Madhusudhan, Vikas Yadav, and Masoud Hashemi. Do llms know when to not answer? investigating abstention abilities of large language models. In *COLING*, 2025.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. Ambigqa: Answering ambiguous open-domain questions. In *EMNLP*, 2020.

Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *EMNLP*, 2019.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. In *NeurIPS*, 2024.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *ACL*, 2018.

Sudha Rao and Hal Daumé III. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *NAACL*, 2018.

Jingyi Ren, Ante Wang, Yunghwei Lai, Xiaolong Wang, Linlu Gong, Weitao Li, Weizhi Ma, and Yang Liu. Beyond "i don't know": Evaluating llm self-awareness in discriminating data and model uncertainty. In *ACL*, 2026.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *ACL*, 2020.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *Findings of NAACL*, 2024.

Asir Saadat, Tasmia Binte Sogir, Md Taukir Azam Chowdhury, and Syem Aziz. When not to answer: Evaluating prompts on gpt models for effective abstention in unanswerable math word problems, 2024.

Ali Şenol, H. Russell Bernard, and Huan Liu. Do large language models know what they don't know ii? a fully behavioral, non-cognitive measure of epistemic honesty, 2026.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, and Esin Durmus. Towards understanding sycophancy in language models. In *ICLR*, 2024.

Jinyan Su and Claire Cardie. Knowing but not showing: Llms recognize ambiguity but rarely ask clarifying questions, 2026.

Minh Ngoc Ta, My Anh Tran Nguyen, Duong D. Nguyen, Yuxia Wang, and Preslav Nakov. One more turn, less regret: A regret-based multi-turn benchmark for llms' clarification policies, 2026.

Tu Trinh, Mohamed Elfeki, Guangze Luo, Kelvin Luu, Nathan Hunt, Ernesto Hernandez, Nandan Marwaha, Yannis Yiming He, Charles Wang, Fernando Carabedo, Alessa Castillo, and Bing Liu. Hil-bench (human-in-loop benchmark): Do agents know when to ask for help?, 2026.

Eren Unlu. Don't start what you can't finish: A counterfactual audit of support-state triage in llm agents, 2026.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Blimp: The benchmark of linguistic minimal pairs for english. In *TACL*, 2020.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, and Bill Howe. Know your limits: A survey of abstention in large language models. In *TACL*, 2025.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. Do large language models know what they don't know? In *ACL*, 2023.

Michael J.Q. Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with lms. In *Findings of NAACL*, 2025.

Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, and Junhong Liu. Clamber: A benchmark of identifying and clarifying ambiguous information needs in large language models. In *ACL*, 2024.

Yichi Zhang et al. Falsereject: A resource for improving contextual safety and mitigating overrefusals in llms. In *COLM*, 2025.

Jiale Zhao, Ke Fang, and Lu Cheng. When and what to ask: Askbench and rubric-guided rlvr for llm clarification. In *Findings of ACL*, 2026.

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 *NeurIPS Datasets and Benchmarks*, 2023.

A Benchmark construction details

Generation. Each pair was drafted by Doubao-2.0-Code (provider-default decoding) from a per-category specification requiring: (i) a *flawed* request exhibiting exactly one defect of the category type, phrased as a plausible real user request; (ii) a *twin* that repairs exactly that defect and is fully answerable as stated. Drafters were instructed to keep surface form, length, and domain constant within a pair.

Verification. Stage (2): Doubao-2.0-Lite verified each drafted pair against three tests — the flawed item is genuinely defective for its category reason; the twin is fully answerable as stated; the two differ only by the critical piece. Of 184 drafted pairs, 162 passed (88%). Stage (3): DeepSeek-Chat re-verified all accepted pairs under stricter semantics — the twin must be answerable by a text-only assistant without tools, browsing, or private user state. All 162 flawed items passed; 40 twins were flagged live-information (ambiguous referent 27/45, contradictory 7/16, unknowable 3/20, false premise 2/25, missing variable 1/56) and 19 pairs flagged for non-minimal edits. Flagged items ship with the benchmark rather than being silently removed.

Judge rubric. The judge sees only the user request and the assistant response and emits a JSON object with the seven behavior fields of §3.3 (*identifies defect*, *asks question*, *corrects premise*, *conditional content*, *commits answer*, *abstains*, *answer correct*). Fields are assigned by DeepSeek-V4-Pro at temperature 0; the full prompt is released. For sensitivity, an earlier single-label rubric (ANSWER/CLARIFY/ABSTAIN) was applied by DeepSeek-Chat at temperature 0 and cross-checked by Gemini-3.1-Flash-Lite on a 1500-item subset (90.3% agreement); DeepSeek-V4-Pro re-labeled all 324 DeepSeek-Chat responses under that rubric (86.4% label agreement). Human audit: 99 stratified items, 89.9% agreement with the primary single-label judge, Cohen's $\kappa = 0.82$.

Access regimes and comparability. Two models were queried through agentic harnesses (SWE-2 via a subagent pipeline; GPT-5.6 via a coding-agent CLI with web search enabled). Tool access can change behavior on items requiring live information; we report these models separately and flag affected cells. All API-served models received an identical system prompt and no tools. Evaluations ran September 2026 with provider-default decoding and a maximum of 8192 tokens per response.

B Per-category results

	missing var		ambiguous ref		false premise		unknowable		contradictory	
Model	flawed	twin	flawed	twin	flawed	twin	flawed	twin	flawed	twin
SWE-2	53/54 (98)	53/54 (98)	43/43 (100)	17/18 (94)	25/25 (100)	21/22 (95)	19/20 (95)	17/17 (100)	5/13 (38)	7/8 (88)
GPT-5.6	50/54 (93)	52/53 (98)	40/42 (95)	12/16 (75)	24/24 (100)	21/22 (95)	17/18 (94)	16/17 (94)	6/14 (43)	4/8 (50)
DeepSeek-V4-Pro†	49/53 (92)	53/54 (98)	41/43 (95)	13/17 (76)	25/25 (100)	21/21 (100)	17/19 (89)	17/17 (100)	5/14 (36)	6/9 (67)
DeepSeek-Chat	49/54 (91)	52/54 (96)	43/44 (98)	11/17 (65)	25/25 (100)	21/22 (95)	18/20 (90)	16/17 (94)	7/12 (58)	6/7 (86)
DeepSeek-Flash	50/53 (94)	52/54 (96)	41/43 (95)	13/18 (72)	24/24 (100)	23/23 (100)	19/20 (95)	17/17 (100)	6/14 (43)	5/9 (56)
Gemini-3.1-Flash-Lite	47/53 (89)	53/54 (98)	42/44 (95)	16/17 (94)	24/24 (100)	18/21 (86)	17/18 (94)	16/16 (100)	2/12 (17)	6/8 (75)
Doubao-2.0-Code	50/54 (93)	51/54 (94)	43/44 (98)	14/15 (93)	24/25 (96)	19/23 (83)	19/20 (95)	16/17 (94)	6/11 (55)	6/8 (75)
Doubao-2.0-Code-NT	48/53 (91)	53/54 (98)	41/44 (93)	14/16 (88)	23/24 (96)	19/22 (86)	17/19 (89)	14/16 (88)	2/8 (25)	3/6 (50)
Doubao-2.0-Lite*	30/33 (91)	32/32 (100)	18/20 (90)	2/5 (40)	–	–	–	–	–	–

Table 3: Per-category cells under the field-level contract (k/n with %; judged coverage). "flawed" = appropriate handling of the defective item; "twin" = factually correct answer on the clean twin (contested live-info twins excluded). *Partial coverage. †The v2 judge is this model.

C Contract sensitivity and pair decomposition

What the v1 "answered" label contained. Applying the v2 fields to the responses that the single-label rubric marked ANSWERED on flawed items (n=213 on scored items, all models pooled): 114 (53%) are committed answers under v2, and 99 (47%) carried *conditional content* — parametric help the single label mechanically lumped with guesses. Under one fixed judge (DeepSeek-V4-Pro) on DeepSeek-Chat's unknowable flawed items, v1's ANSWERED rate is 7/20 and v2's committed-answer

rate is 2/20 on identical responses: a pure rubric contrast. The judge contrast under v1 on the same items runs in the opposite direction (DeepSeek-Chat self-judge: 3/20 ANSWERED; DeepSeek-V4-Pro: 7/20). The full rubric×judge grid on all models is future work; these cells show both factors move the estimate, with the field structure explaining most of the v1 "fabrication" mass.

Joint pair decomposition (strict core, n=103). For each model we tabulate the four cells n_{11} (flawed handled *and* twin correct), n_{10}, n_{01}, n_{00}, the unjudged remainder (missing coverage), and the conditional P(flawed handled | twin correct). The dominant discordant cell is n_{01} — the twin answered correctly while the flawed side was mishandled — which pools to 68 items concentrated in contradictory (28) and missing-variable (26) items; n_{10} (flagged the defect but answered the twin wrong) is rare. The rightmost column bounds the joint rate if all 103 pairs were judged (missing treated first as all-fail, then all-pass), showing the missing coverage cannot change the ordering materially except for the partial Lite row.

Model	n_{11}	n_{10}	n_{01}	n_{00}	unjudged	P($f \mid t$)	joint range on 103
SWE-2	91	2	6	2	2	93.8%	88–90%
GPT-5.6	78	6	9	3	7	89.7%	76–83%
DeepSeek-V4-Pro	80	4	11	3	5	87.9%	78–83%
DeepSeek-Chat	83	7	7	3	3	92.2%	81–83%
DeepSeek-Flash	85	8	6	3	1	93.4%	83–83%
Gemini-3.1-Flash-Lite	80	3	11	3	6	87.9%	78–83%
Doubao-2.0-Code	82	6	8	3	4	91.1%	80–83%
Doubao-2.0-Code-NT	76	5	9	2	11	89.4%	74–84%
Doubao-2.0-Lite*	26	3	1	0	73	96.3%	25–96%

Table 4: Joint pair decomposition on the 103-pair strict core. n_{ij}: i=1 if the flawed side was handled appropriately, j=1 if the twin was answered correctly; "unjudged" are strict-core pairs lacking a judged response on at least one side. P($f \mid t$) = Pr(flawed handled | twin correct). The joint range treats unjudged pairs first as all-failures then as all-successes; it is a sensitivity bound, not a score.

D DATASET CARD

Following the datasheets methodology (Gebru et al., 2021):

Motivation. ClarPair measures the single-turn decision boundary of general assistants — answer, ask, or decline — a behavior that sits between abstention benchmarks (refusal only) and interactive clarification benchmarks (multi-turn question quality). Created for diagnostic evaluation of clarification policy in LLM-based assistants.

Composition. 162 verified paired requests, English, single-turn. Five defect categories: missing variable (56), ambiguous referent (45), false premise (25), unknowable (20), contradictory (16). Each item carries the raw user request pair and verification flags: 40 twins flagged *live-info* (not answerable by a text-only assistant), 19 flagged *non-minimal edit*; the human item audit adds five *knowledge-defeatable* flawed flags and one *residual-ambiguity* twin flag. No personal data; all requests are synthetic.

Collection process. Pairs drafted by Doubao-2.0-Code from per-category specifications (surface form, length, and domain held constant within a pair); first-pass verification by Doubao-2.0-Lite (three tests, 184 drafted → 162 accepted); strict re-verification by a different-family model (DeepSeek-Chat) under a text-only answerability semantics. Generation and verification prompts released.

Preprocessing. Rejected pairs (22) ship with their rejection reasons. Response labels were produced by LLM judges under two rubric versions — the multi-field behavior contract of §3.3 (primary) and the earlier single-label rubric (sensitivity view) — and are released alongside raw model responses for all nine evaluated configurations.

Recommended uses. Diagnostic evaluation of clarification behavior; regression testing of deployment candidates on the three slices; study of tool-awareness via the live-info slice; reuse of the minimal-pair pipeline for new defect categories. Not intended as a training corpus or as a human-behavior model: items are synthetic and LLM-drafted.

Distribution and maintenance. Released as supplementary material under a permissive license (CC-BY-4.0); includes pairs, flags, prompts, raw responses, both judge label sets, human-audit labels, and the scoring script that regenerates all reported numbers. Live-info items may age as facts change; the release is versioned and the flag file is the source of truth for the scoring partition.

E JUDGE-BOUNDARY CASES (LEGACY V1 RUBRIC)

The cases below concern the *earlier single-label rubric*, kept as a sensitivity view; the v2 field contract resolves this boundary explicitly by separating *conditional content* from *commits answer*. The dominant judge disagreement is the *flag-and-answer* response: the model explicitly names the missing piece, then still supplies conditional content. The rubric counts answering-while-merely-noting-uncertainty as ANSWER, but judges differ on whether an explicit defect-naming followed by substantive conditional content crosses into ANSWER. Two real cases (DeepSeek-Chat responses on flawed items; primary label CLARIFY, re-label ANSWER):

> **Request**: "A professional soccer player scores an average of 0.8 goals per match. How many total goals will they score this season?"
> **Response** (truncated): "It depends on how many matches they play this season. Using the average: Total goals = 0.8 $\times$ number of matches. For example: 10 matches $\rightarrow$ 8 goals … So the answer isn't fixed unless we know the number of matches in the season."

> **Request**: "A landlord charges a $1,200 monthly rent plus a fixed late fee. What total does a late tenant pay for one month?"
> **Response** (truncated): "That question doesn't give enough information to determine the total. The total would be: $1,200 + the fixed late fee. For example: if the late fee is $50, the tenant pays $1,250 …"

Both responses identify the missing variable before giving conditional formulas. The primary judge reads the explicit defect-naming as CLARIFY; the stricter re-labeler reads the substantive conditional content as ANSWER. This is the residual ambiguity that bounds our label accuracy (human audit $\kappa = 0.82$) and is the reason reported clarify rates are conservative.