RETRIEVED BUT NOT RANKED: SURFACE-FORM BIAS IN STRUCTURAL RETRIEVAL, FROM MATHEMATICS TO AGENT TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Embedding retrieval is usually validated where surface form and meaning point the same way. We study the case where they are pulled apart on purpose: retrieving items that share structure but not wording, in two unrelated domains under one protocol, competition mathematics (500 queries against 117,088 items) and embodied-agent trajectories (118 queries against 336). In mathematics, strict Hit@1 at the heaviest disguise tier is 0.0% for both production embedders (bootstrap 95% CI [0.0, 0.0]) while the correct item sits in the top 10 nearly always, and in 95.2 to 99.8% of misses the winner is more lexically similar to the query than the answer. In trajectories, all three embedders fall below hypergeometric chance once gold must differ in object and receptacle: retrieval anchors on literal tokens rather than task structure. A lexical reranker control hurts in mathematics and helps in trajectories (closing 26 to 36% of the recoverable gap), so its sign diagnoses whether a benchmark's surface variation is adversarial or incidental. LLM rerankers recover 5 to 63% of the gap in mathematics and 43 to 76% in trajectories; direction replicates across three judges, but effect sizes, tier profiles, and even which judge is the outlier change with domain, and mathematics gains concentrate on well-known competitions (+19.8 points in one of six cells). Finally, in a paired downstream experiment, oracle retrieval was indistinguishable from adversarially bad retrieval (McNemar p = 0.678), because the solver's 69.5% zero-shot accuracy is largely a truncation proxy, with 97 to 100% accuracy on answers that finish within budget.

1 INTRODUCTION

Most retrieval benchmarks reward the easy case. A query about quadratic equations retrieves documents containing the words "quadratic equation," wording and meaning move together, and embedding models trained against such benchmarks inherit that alignment. Structural retrieval breaks the alignment deliberately. Two mathematics problems can call for the same technique while sharing no vocabulary. Two agent trajectories can run the same procedure over completely different objects. Near-identical text can hide a flipped inequality, or a different household task. When surface and structure come apart, which one does embedding retrieval follow?

The question cannot be settled in one domain, because a single-domain answer mixes properties of embedding retrieval with properties of that benchmark's construction. So we run one protocol across two domains that have nothing in common, and this is what turns up the result we consider most useful: a cheap lexical control flips sign depending on how the benchmark was built. No single-domain study could see that, and it changes how "semantic" retrieval evaluations should be read.

Our contributions:

- **A two-domain evaluation of structural retrieval** under one protocol: tiered disguise-requiring gold, exact chance baselines, rank-1 failure taxonomy, bootstrap CIs on every headline number, and each source benchmark's published numbers reproduced as a validation gate.

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.

- **Embedding retrieval anchors on literal surface content in both domains**: zero rank-1 accuracy under heavy disguise in mathematics with the answer retrievable below; below-chance retrieval in trajectories once gold excludes the query's literal object and receptacle tokens.
- **The lexical-control sign flip**: one lexical reranker damages mathematical retrieval and improves trajectory retrieval; a token-class ablation locates the mechanism, and the control's sign diagnoses a benchmark's surface-variation regime.
- **Judge dependence of LLM reranking**: direction replicates across three judges in both domains, while effect size, tier profile, and even which judge is the outlier change with domain, and one judge's gains drop by more than a third across query styles. Neither an effect size nor a judge ranking is portable.
- **A contamination probe at its measured strength**: gains concentrate on well-known competitions, non-null in one of six judge-by-candidate cells, directionally consistent elsewhere.
- **A paired downstream null with its mechanism located**: with a deliberately bad retrieval condition, oracle and adversarial retrieval are indistinguishable, because the 69.5% zero-shot accuracy is largely a truncation proxy with near-zero effective headroom.
- **Eight documented evaluation-integrity incidents**, most involving silent truncation that parsed as valid output.

2 Related work

Structural retrieval benchmarks. MathNet (Alshammari et al., 2026) showed that 27 embedding models fail mathematical-equivalence retrieval (Recall@1 below 5%) while recall stays high, and attributed the failure to superficial overlap. It did not test reranking. The procedural-memory benchmark of Kohar & Krishnan (2025) found the same kind of generalization cliff for trajectory retrieval on ALFWorld (Shridhar et al., 2021), with three limitations we address: a single 384-dimensional encoder (Reimers & Gurevych, 2019), relevance labels from an LLM judge with modest human agreement (Cohen's kappa 0.178), and two-stage retrieval left as future work. We supply that stage on their released data with modern embedders, scored against exhaustive task-type labels instead of judged pools, which also avoids a pool-coverage artifact we document in their protocol.

Retrieve-then-rerank is standard information retrieval (Nogueira & Cho, 2019). We contribute nothing to the architecture; what we contribute is its evaluation under structural relevance instead of topical relevance, with controls built for LLM judges.

LLM-as-judge scales evaluation at the price of importing the judge's training distribution (Zheng et al., 2023). Our contamination analysis makes that concern concrete for reranking: where gains concentrate on items the judge plausibly memorized, claims about reasoning need discounting.

Context that hurts. Irrelevant context degrades LLM problem solving (Shi et al., 2023). MathNet observed embedding-retrieval RAG scoring below zero-shot for three solvers and did not investigate. We measure the link with a paired design and a deliberately bad condition.

3 Shared protocol

Both domains run through one pipeline. We specify it fully here so that every result in Sections 4 through 6 can be reconstructed from this section alone.

3.1 Query and corpus construction

Mathematics. Queries are 500 items from MathNet-Retrieve. Each query is a paraphrase of a source competition problem at one of two disguise tiers, EASY or HARD, written by the benchmark's authors with a Gemini-family model (a provenance fact we return to in Section 4). The corpus is the full 117,088-item MathNet collection, and we rank all of it for every query; nothing is shortlisted before retrieval. Each query has exactly one gold item, its source problem, and the benchmark plants a lexically similar but structurally different near-miss for every query.

Agent trajectories. The corpus is 336 ALFWorld trajectories. Queries are 118 trajectories: the 40 released by the source benchmark plus 78 we drew from ALFWorld's public split to raise statistical power. Each trajectory is a task instruction followed by an action sequence, and ALFWorld's six task types (for example *pick and place*, *heat and place*, *examine in light*) define what counts as a procedure.

3.2 GOLD DEFINITION

Mathematics uses MathNet's expert labels unchanged: the gold item is the source problem. Strict Hit@k counts only the gold; lenient Hit@k also counts the planted near-miss. We report both because the gap between them is the paper's central measurement.

Trajectories. Relevance is defined by task type. We labeled all 336 corpus items exhaustively from ALFWorld's task metadata (rule-derived from the task metadata, then human-audited on stratified 60-item samples of both corpus and queries with zero disagreements), which removes the source benchmark's pooled-relevance problem. The rule classifier is a set of eleven anchored regular expressions over the corpus's 180 unique task-description strings, one template family per task type; every trajectory is tested against all eleven, and any cross-type collision is flagged as ambiguous rather than resolved by rule order. On top of that we apply two disguise requirements of increasing strictness: definition (i) requires a same-type gold to involve a different target object; definition (ii) requires a different object *and* a different receptacle, so that gold and query share neither of the tokens most likely to drive lexical similarity. Because every candidate carries a label, chance is exact and hypergeometric for each query, and we report retrieval against it.

3.3 EMBEDDING RETRIEVAL

We embed each query once and rank it against the full corpus by cosine similarity. In trajectories the corpus text is the task instruction followed by the full numbered state and action sequence, while the query text is the instruction alone; the LLM judges below see exactly these same texts. Mathematics uses two production embedders, gemini-embedding-001 and Qwen3-Embedding-8B (served via DeepInfra). Trajectories add MiniLM-L6-v2, the source benchmark's own encoder, kept as the weak-baseline contrast, and use the Qwen model served on a lab-hosted deployment. The deployment comparison in Section 4 embeds the same 500 texts through both Qwen servings and reruns retrieval under each.

3.4 RERANKING STAGES

Reranking always operates on the same fixed top-10 candidate list from the embedding stage, so every condition is compared on identical inputs.

Lexical control. Before any LLM sees a candidate, a deliberately naive reranker re-sorts the top-10 by a lexical score that averages three signals, each in [0, 1]: Jaccard overlap of lowercased word tokens without stemming, a Levenshtein edit-distance ratio, and a length ratio. In mathematics it compares full problem texts; in trajectories it compares the query instruction with the candidate's task description. The mechanism ablation in Section 5 uses a narrower measure, plain Jaccard over restricted token sets: a closed list of fourteen ALFWorld action verbs for the verb-only variant, and the residual non-verb, non-stopword tokens for the noun-only variant (no part-of-speech tagger is involved). Its job is diagnostic. It can only add surface signal, so its sign tells us whether a benchmark's surface variation is adversarial or incidental.

LLM judges. Three independently trained judges rerank the same top-10: Gemini 3.1 Flash-Lite, GLM-5.2 (fp8, served on a lab endpoint), and Claude Haiku 4.5, all at temperature 0. We use two prompt variants, terse and chain-of-thought (verbatim prompts in Appendix A). Each judge returns a single best candidate, which is placed at rank 1 with the remaining embedding order retained, so reranked Hit@1 is computed on that pick; unparseable outputs fall back to the embedding order and are counted in the truncation audit.

Table 1: Mathematics baseline, strict scoring, n=500. Brackets are bootstrap 95% CIs.

Embedder	Tier	Hit@1	Hit@5	Hit@10
Gemini-emb	Easy	12.2% [9.4, 15.2]	89.8% [87.0, 92.4]	97.6% [96.2, 98.8]
Gemini-emb	Hard	0.0% [0.0, 0.0]	10.0% [7.4, 12.8]	55.4% [51.0, 59.8]
Qwen-emb	Easy	8.6% [6.2, 11.0]	86.8% [83.8, 89.6]	95.2% [93.2, 97.0]
Qwen-emb	Hard	0.0% [0.0, 0.0]	2.8% [1.4, 4.4]	21.0% [17.6, 24.6]

3.5 Metrics and statistics

The primary metric is **share of the recoverable gap closed**: (Hit@1 after reranking minus Hit@1 before) divided by (Hit@10 minus Hit@1 before). A reranker is therefore scored only against what its candidate list made retrievable. All Hit@k values carry bootstrap 95% confidence intervals from resampling queries (10,000 resamples, seed 12345); share-of-gap intervals use the same resample count with seed 54321. Judge comparisons use **paired** bootstrap differences on the same queries (10,000 resamples, seed 42), reported for ten pre-specified pairs (four in mathematics, six in trajectories). Contamination effects use bootstrap intervals on the difference between famous-competition and other-source candidates. The deployment comparison uses McNemar's test on per-query hit flips, and the utility experiment uses McNemar's test on paired per-problem correctness.

3.6 Downstream utility design

A single solver, DeepSeek-v4-flash at a 32,768-token output budget, attempts 210 MathNet problems under three conditions: no context, a deliberately bad retrieval (a lexically similar but structurally unrelated problem), and gold-with-solution. Two independent graders that never include the solver mark the answers, Gemini 3.1 Flash-Lite (primary) and GLM-5.2 (robustness check); each returns an integer score from 0 to 7, binarized as correct at 6 or above, and they agree on 96 to 99% of items (grader prompt in Appendix F). We log finish-reason for every generation so that truncated outputs can be separated from completed ones.

3.7 Validation gates and integrity audits

Before any new measurement, each domain reproduced a published number with the original artifacts: MathNet's Table 4 easy-tier values within one point and the trajectory benchmark's MEDIUM mean average precision within 0.007 (values in Appendix D). Every API response is cached, every reported number is recomputed from frozen aggregates, and finish-reason, per-condition truncation, and spend audits run as standing checks (final cost $17.24 over 4,338 calls); Appendix D.3 lists the eight incidents they surfaced.

4 Domain 1: mathematical problems

Setup. Section 3.1 gives the construction: 500 anchors against 117,088 items, each anchor with three LLM-generated equivalent reformulations (easy, medium, and hard disguise) and roughly three near-miss decoys that keep the surface form and change the mathematics. One design fact matters for everything below: the corpora are byte-identical across tiers and only the gold designation changes, so the tier axis isolates disguise and nothing else.

Baseline. Figure 1 and Table 1 report strict Hit@k. The hard-tier zero is degenerate, in the technical sense: no bootstrap resample of 500 queries ever contains a hit. Lenient Hit@10 is 99 to 100% at both tiers. Retrieval succeeds; ranking fails.

Failure structure. Between 84 and 98% of misses, depending on embedder and tier, are the query's own planted near-miss. That is the trap working as designed, and we can measure how it works: in 95.2 to 99.8% of misses the false positive is more lexically similar to the anchor than gold is, by the same lexical score the control uses. One representative case shows the mechanism. A character-identical problem with a single flipped inequality wins at cosine 0.860, while the true renamed-variable equivalent sits at rank four with cosine 0.821 (Appendix B.1). The lexical reranker control reduces Hit@1 (a 9.1% and 4.8% share of gap lost at easy tier for Gemini-emb and Qwen-emb

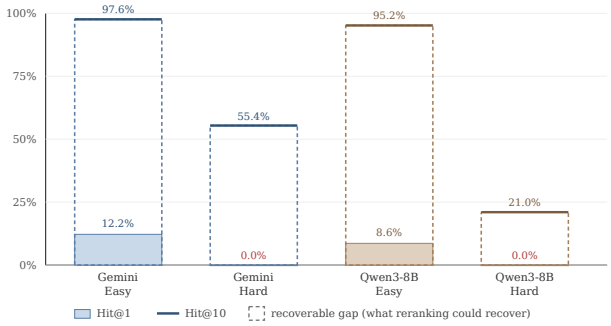

Figure 1: The right answer is retrieved but not ranked first. Strict Hit@1 (bars) against Hit@10 (lines), 500 queries against the full 117,088-item corpus; the dashed span marks the recoverable gap available to any reranker.

Table 2: Share of recoverable gap closed, terse prompt, mathematics. Brackets are bootstrap 95% CIs.

Candidates	Tier	Gemini-j	GLM-j	Haiku-j
Gemini-emb	Easy	20.6% [14.7, 26.3]	10.1% [5.0, 15.0]	58.1% [52.7, 63.3]
Gemini-emb	Hard	44.4% [37.5, 51.3]	10.5% [6.9, 14.4]	5.4% [2.9, 8.3]
Qwen-emb	Easy	35.6% [29.9, 41.1]	12.0% [7.3, 16.6]	63.3% [58.2, 68.1]
Qwen-emb	Hard	41.0% [29.5, 53.3]	18.1% [10.5, 26.7]	6.7% [1.9, 12.4]

respectively; flat at hard tier). Surface similarity is anti-correlated with correctness here by construction, and there is no lexical generation fingerprint left for a reranker to exploit.

LLM reranking. All twelve terse judge-by-configuration cells are positive, and the judges' magnitudes separate beyond sampling noise (Figure 2 and Table 2; paired bootstrap differences between Gemini-j and GLM-j exclude zero in all four configurations). We had hoped a third judge would settle the ordering. It scrambled it instead. Haiku-j is 1.8 to 2.8 times stronger than Gemini-j and more than five times stronger than GLM-j at easy tier (58.1 and 63.3% of gap closed), yet weaker than both at hard tier (5.4 and 6.7%). It is the only judge whose gains degrade under heavier disguise the way naive intuition predicts; Gemini-j's hard-tier gains exceed its easy-tier gains, and GLM-j is flat to mildly rising across tiers. Three judges, three different tier profiles. Chain-of-thought doubles Gemini-j's easy-tier gains (44.7% and 55.4% share closed) while halving its hard-tier gains (22.7% and 26.7%). We expected the easy-tier improvement; the hard-tier drop surprised us. The traces offer one reading. The judge sometimes selects a candidate it itself calls "a direct restatement," which suggests that room to deliberate lets it drift toward the most recognizable sibling. We consider this plausible and have not confirmed it. GLM-j chain-of-thought is excluded entirely: 63.3% of its 2,000 responses truncated mid-reasoning while still parsing (Section 8).

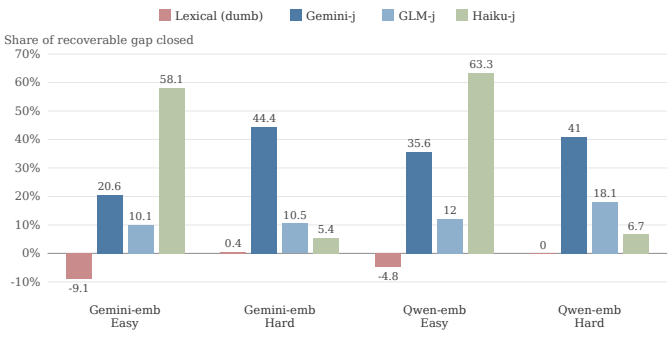

Figure 2: Lexical reranking hurts while LLM reranking helps, mathematics domain. Share of the recoverable gap closed per configuration; all conditions rerank the same top-10 candidates.

Table 3: Strict Hit@1 versus exact chance under three gold definitions, n=118.

Definition	Chance	Qwen-emb	Gemini-emb	MiniLM
(i) different object	15.3%	17.8%	15.3%	9.3%
(ii) different object and receptacle	14.1%	**11.0%**	**8.5%**	**6.8%**
(iii) any object (reference)	16.9%	75.4%	74.6%	50.0%

Contamination. At hard tier, well-known competitions (IMO, USAMO, APMO; n=57) against the pooled rest (n=443) show the gaps in Table E.1 (Appendix E.1). One of six cells is confidently non-null, Gemini-j on Gemini-emb candidates at +19.8 points [+6.7, +33.2]. The other five, including both Haiku-j cells, point the same direction and are individually indistinguishable from zero at this sample size. Fisher tests on Gemini-j's four prompt-by-candidate combinations all reached $p < 0.01$, but we treat the bootstrap intervals as authoritative, so the summary is: contamination established in one cell, directionally consistent elsewhere, unproven as a judge-independent effect, and unmeasured under GLM-j, whose deliberative condition could not be scored. The chain-of-thought traces also show genuine technique-level reasoning, such as correctly identifying an invariant-mod-9 argument. Recognition and reasoning both operate, in proportions this design cannot separate.

Deployment divergence. Two servings of identical Qwen3-Embedding-8B weights agree on five of six retrieval metrics and differ significantly on hard-tier Hit@10 (McNemar exact p = 0.00014), exactly where gold-versus-decoy margins are thinnest; Appendix E.2 reports the comparison.

5 Domain 2: agent trajectories

Setup. The released benchmark of Kohar & Krishnan (2025) supplies the 336 AgentInstruct ALFWorld trajectories and 40 of the queries; Section 3.1 describes the corpus and Section 3.2 the task-type gold. Three details belong here. The 78 added queries come from ALFWorld's public valid_unseen split, every unique goal string available at a lightweight public source; the planned 150 was unreachable without a heavy simulation dependency, and we report the shortfall rather than patch it. The rule classifier's task-type labels were human-verified on stratified 60-item samples of both corpus and queries, with zero disagreements in both audits. And three benchmark corrections are documented in the repository: one released query relabeled on an unambiguous verb conflict ("chill" implies cooling, not heating); the release's silently defaulting task-type field bypassed in favor of our own mapping; and the paper's 78-trajectory expert corpus, which is absent from the release, so all results use the AgentInstruct corpus.

Tiers. STRICT requires the same task type and a different target object (the disguise requirement). A harsher variant (ii) additionally requires a different receptacle. LENIENT accepts any same-task-type trajectory. Chance is computed per query via the exact hypergeometric tail, since gold-set sizes vary (mean strict set 51.2 of 336).

Retrieval anchors on literal tokens. Pooled strict Hit@1 sits at 17.8% [11.0, 25.4] (Qwen-emb), 15.3% [9.3, 22.0] (Gemini-emb), and 9.3% [4.2, 14.4] (MiniLM) against chance of 15.3%. The production models are at or barely above chance with CIs straddling the chance line; the weak encoder is below it. On its own that is ambiguous, and the definition-robustness check in Table 4 resolves it. Under definition (ii), the cleanest test of generalizing past the query's literal object and container tokens, all three embedders fall below chance (Figure 3). A model indifferent to structure would score at chance. Scoring systematically below it means the ranking is actively steered away from cross-object, cross-receptacle structural matches, and the thing doing the steering is token anchoring. The mechanism differs from the mathematics case (below chance here, zero-but-retrievable there); the direction of the bias is the same. Rankings follow the query's literal tokens.

The two query subsets behave differently (the original 40 are substantially harder than the added 78; Appendix E.3), a benchmark-construction effect confounded with phrasing style.

The lexical control flips sign. We ran this control expecting it to hurt, as it had in mathematics. It helped, and by enough to make it the trajectory domain's statistically firmest result: share of strict gap closed is +25.9% [11.3, 41.2] (Qwen-emb), +36.4% [23.4, 50.0] (Gemini-emb), and +32.1% [18.5, 47.1] (MiniLM), all CIs excluding zero. A verb-only variant, run on the original-40 subset, reproduces most of the effect there, while a noun-only variant contributes little (0 to 12.5% share

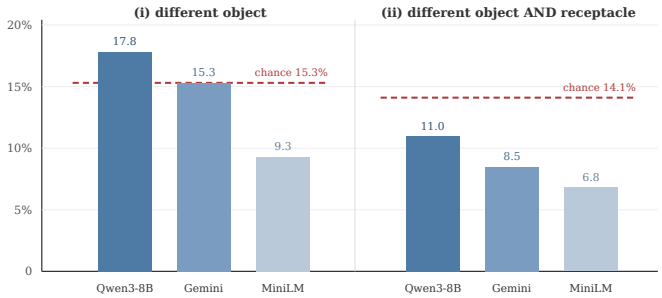

Figure 3: Trajectory retrieval anchors on literal tokens. Strict Hit@1 for the three embedders against exact hypergeometric chance (dashed line), n=118, under gold definition (i), different object, and definition (ii), different object and receptacle. Under (ii) all three embedders fall below chance: sub-random, not indifferent.

Table 4: Share of recoverable gap closed, terse prompt, trajectories, n=118. Brackets are bootstrap 95% CIs.

Candidates	Gemini-j	GLM-j	Haiku-j
Qwen-emb	42.6% [27.3, 59.3]	**68.5%** [50.9, 87.5]	46.3% [30.2, 63.0]
Gemini-emb	45.5% [31.9, 59.7]	**75.8%** [60.7, 91.4]	43.9% [30.4, 57.9]
MiniLM	58.9% [42.1, 76.8]	**75.0%** [56.9, 93.2]	48.2% [32.2, 64.8]

closed). A within-domain phrasing slice (verb-first "heat some X" versus adjective-first "put a hot X", task types 3 to 5) shows the lexical reranker helping in every cell for both phrasings. The apparent contradiction between domains dissolves in the construction of the two benchmarks. MathNet's equivalents are adversarially paraphrased to suppress lexical overlap while its decoys preserve it, so lexical signal points exactly the wrong way. ALFWorld's surface variation is incidental, so verb and receptacle overlap is a real correlate of task type. The control costs nothing to compute, and its sign indicates which regime a benchmark occupies before any expensive evaluation runs.

LLM reranking, and a judge reversal. At n=118, terse judges close substantial strict gap for every embedder, and the judge ranking inverts relative to mathematics (Table 5). We had expected Gemini-j to lead here as it does there. Instead GLM-j, which trails Gemini-j by a factor of 2 to 4 in mathematics, leads by 1.3 to 1.7 times; paired bootstrap differences (GLM-j minus Gemini-j and GLM-j minus Haiku-j, same queries and candidate lists, 10,000 resamples) exclude zero for every embedder, including the one pairing where the marginal intervals overlap. The third judge settles which side is anomalous. Haiku-j lands within a few points of Gemini-j on the two production embedders and well inside its intervals on all three (43.9 to 48.2% of gap closed), so two independent judges converge on this domain's magnitude while GLM-j alone diverges upward. Each domain therefore has a different outlier judge: Haiku-j in mathematics, GLM-j here. Provenance sharpens the picture. On the two production embedders' candidate sets, Gemini-j's gains drop steeply on the templated new-78 queries (36.6 to 40.0%, versus 61.5 to 62.5% on the human-paraphrased original 40), while its MiniLM cell is stable (56.3 versus 60.0) and GLM-j holds steady everywhere (68.3 to 76.0%). Judge performance is sensitive to query style within a domain, and to domain itself. The judge-level failure taxonomy is consistent across all nine judge-by-embedder cells: siblings dominate (72 to 88% of strict misses, near-misses 8 to 20%), and the third judge replicates the pattern (per-cell tables in Appendix C). The judges have essentially solved the procedural confusion and fail almost exclusively on the disguise requirement, a categorically more benign error than the embedders' structural confusions. (Truncation audits: 1.8% capped or unparsed in the two-judge run, isolated to GLM-j's known terse-output profile; 0.9% in the Haiku-j run; both within threshold.)

6 Downstream: does retrieval quality reach the solver?

MathNet reported gains of up to 12 points from expert retrieval and, without investigating, observed embedding-retrieval RAG scoring below zero-shot for three solvers. We measure the link with the condition their design lacked, deliberately bad retrieval, using the setup in Section 3.6.

Setup details that matter for reading the results: 210 mathematics queries; three conditions (no retrieval, adversarially reranked top-1, and the gold equivalent with its solution); solver DeepSeek-v4-flash at a 32,768-token budget; two independent graders at 96.2 to 98.6% per-condition binary agreement, with the solver's developer excluded from grading. Truncation is 30.5 to 31.4%, uniform across conditions. The condition-correlated truncation that invalidated an earlier pilot (Section 8) is absent here, though the residual rate never resolved across three cap increases (80%, then 50%, then 31%) and is carried as a stated caveat on every accuracy below.

The result is a paired null, and the paired analysis sharpens it. Accuracy is flat (67 to 70% in every condition, both graders). McNemar on none versus gold: 13 queries where context hurt against 10 where it helped ($p = 0.678$). None versus dumb: 11 against 8 ($p = 0.648$). Both graders concur. On the only subset where context could have acted, the 64 of 210 queries that failed zero-shot, gold recovers 10 (15.6%) while breaking 13 of the 146 already solved, a net of minus 3. Deliberately bad retrieval nets the identical minus 3. For this solver, oracle retrieval was indistinguishable from adversarial retrieval.

A complete-answers-only check then locates the mechanism. Restricting to the 127 of 210 queries whose answers finished within budget in all three conditions, accuracy is 97.6 to 100% in every condition under both graders, with zero to two discordant pairs per comparison (McNemar p between 0.50 and 1.00). In the no-context condition, completion and correctness nearly coincide under Grader A: the 146 finished answers and the 146 correct answers are both size 146 but overlap on 144, with two truncated answers credited for stating the right value before the cutoff and two finished derivations simply wrong; 206 of 210 queries align in total, and the equality of the two counts is a numerical coincidence rather than a set identity. The headline 69.5% zero-shot accuracy is therefore closer to a proxy for whether the derivation fits the budget than to a measure of solving ability. The null holds for a specific reason: this solver has essentially no headroom on problems it can finish, so there was almost nothing for oracle context to improve. We report both framings because they are different claims. One says retrieval did not move accuracy; the other says there was almost nothing for it to move. A composition check in Appendix E.4 shows the ceiling does not reduce to recognition of famous problems. This is one solver in one domain and we claim nothing wider. It remains a measured counterweight to the assumption that retrieval quality transfers downstream, with headroom now identified as the binding constraint rather than a conjectured moderator.

7 Cross-domain synthesis

1. **Surface over structure, whenever the two are separable.** Mathematics shows 0% strict rank-1 under heavy disguise with the answer retrievable below; trajectories show below-chance retrieval for all three embedders once gold excludes both the literal object and receptacle. The mechanisms differ. In both cases these embedding models rank by literal surface content.
2. **The lexical control's sign diagnoses the benchmark rather than the retriever.** Adversarial surface variation makes lexical signal anti-informative, incidental variation makes it informative, and a single-benchmark claim about "semantic" retrieval that omits this axis conflates the two regimes. The control costs nothing to run.
3. **LLM reranking recovers substantial gap in both regimes; direction is the only portable finding.** Across three judges, effect sizes span more than eightfold within a single configuration, tier profiles disagree (one judge improves under heavier disguise, one is flat, one degrades), each domain has a different outlier judge (paired bootstrap differences excluding zero in every configuration), and one judge's gains drop by more than a third across query styles within a domain. Any single reranking effect size is a property of the judge, prompt, domain, and query style jointly. Where provenance varies, memorization measurably contributes.
4. **Stages fail differently.** Embedder errors are structural confusions; judge errors are disguise failures over the correct structure (siblings, 72 to 88% of judge misses in every trajectory cell). Pipeline evaluation should ask what kind of error survives each stage, and only then how many.
5. **The retrieval-to-solver link cannot be assumed.** Under a paired design with a bad-retrieval control, one competent solver extracted nothing from oracle context. A complete-answers-only analysis attributes this to near-zero effective headroom rather than to context

being ignored, which is itself a caution about reading zero-shot accuracy under token budgets.

8 Evaluation-integrity findings

We record eight evaluation-integrity incidents, most involving silent truncation that parsed as valid output; in every case we excluded the affected numbers rather than attempting repairs. Appendix D.3 gives the full list and the standing audit practices that caught them, which are the reason the other numbers in this paper can be trusted.

9 Limitations

Mathematics uses a single fixed seed; bootstrap CIs quantify within-sample uncertainty only. Three judges is a small sample of judges. Magnitudes vary by more than eightfold among them, each domain has a different outlier, and tier profiles disagree, so the observed spread is best read as a lower bound on judge variability; one further candidate proved structurally infeasible (Section 8). The trajectory domain has one dataset family, one 336-item corpus, and 118 queries (short of the 150 target for stated availability reasons). Provenance and phrasing style are confounded in the old-versus-new subset contrast, and one judge's sensitivity to that contrast means judge results should not be assumed stable across query styles. Task-type labels are rule-inferred and human-verified on samples rather than authored. Contamination attribution is correlational; the well-known subset may differ in ways beyond training exposure. MathNet's equivalents and decoys were generated and partly verified by Gemini-3-flash, a same-family model as Gemini-j; family-level affinity to Gemini-generated paraphrases is a possible unmeasured contributor to Gemini-j's mathematics advantage and to the one significant contamination cell, which pairs a Gemini judge with Gemini-embedding candidates. The downstream null is one solver whose effective headroom proved near zero once truncation was controlled for; the 31% solver-truncation rate is carried as a caveat and, per Section 6, is itself the binding constraint. Mathematics variants and decoys are LLM-generated; lexical controls constrain generation-process confounds without eliminating them.

10 What this paper does not claim

That embedding retrieval is generally poor: lenient and easy-tier performance are strong in both domains. That lexical reranking is generally good or bad: its sign depends on benchmark construction, which is exactly what makes it informative. That LLM reranking solves structural retrieval: the majority of hard-tier gap remains unclosed. That memorization explains reranking gains: both mechanisms demonstrably operate. That downstream irrelevance generalizes beyond the measured solver. That the two domains share a mechanism: below-chance and zero-but-retrievable are different failure shapes, and the common claim is behavioral rather than mechanistic.

11 Conclusion

Across two unlike domains, embedding retrieval tracks literal surface content over underlying structure whenever the two are separable, to the point of ranking below random selection when gold requires generalizing past the query's literal tokens. A reranking stage recovers much of what ranking loses. Nothing about its magnitude transfers across judges, prompts, domains, or query styles, and part of the recovery is attributable to memorization where provenance varies. The most useful instrument in the study also turned out to be the cheapest: a lexical reranker whose sign reveals whether a benchmark's surface variation is adversarial or incidental. We suggest structural-retrieval evaluations report that control routinely, along with exact chance baselines and per-condition truncation audits.

Reproducibility Statement

Seed 42 in mathematics; all 118 available trajectory queries used, so no query sampling. All raw results, per-query cached judge and grader responses, spend logs, correction banners,

and both benchmark-correction patches are in the repository; `results/FINAL_NUMBERS.md` is the authoritative flat digest of every number cited here with its source file, and `results/RESULTS_SUMMARY.md` maps each section to its script and raw output. Total experiment cost was $17.24 across 4,338 recorded API calls. Code, results files, and reproduction materials are included in the supplementary material, and will be released publicly upon de-anonymization.

AI USE STATEMENT

AI assistants (Claude Fable 5, Anthropic, including via Claude Code) were used throughout this work under the first author's direction: to draft and edit text, to implement analysis and evaluation code, to generate figures, to propose and give feedback on experimental controls and statistical design, and to assist in interpreting results. All research questions, design decisions, and claims are the authors' own. Every reported number was recomputed from cached raw responses and frozen aggregates and independently verified, and every reference was checked against its arXiv record. The authors take full responsibility for the content.

REFERENCES

S. Alshammari, K. Wen, A. Zainal, M. Hamilton, N. Safaei, S. Albarakati, W. T. Freeman, and A. Torralba. MathNet: A Global Multimodal Benchmark for Mathematical Reasoning and Retrieval. In *ICLR 2026*, 2026. arXiv:2604.18584.

I. Kohar and A. Krishnan. A Benchmark for Procedural Memory Retrieval in Language Agents. *arXiv preprint arXiv:2511.21730*, 2025.

R. Nogueira and K. Cho. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.

N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP-IJCNLP 2019*, 2019. arXiv:1908.10084.

F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. Chi, N. Schärli, and D. Zhou. Large Language Models Can Be Easily Distracted by Irrelevant Context. In *ICML 2023*, 2023. arXiv:2302.00093.

M. Shridhar, X. Yuan, M.-A. Côté, Y. Bisk, A. Trischler, and M. Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *ICLR 2021*, 2021. arXiv:2010.03768.

L. Zheng, W.-L. Chiang, Y. Sheng, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *NeurIPS 2023*, 2023. arXiv:2306.05685.

A VERBATIM JUDGE PROMPTS

All three judges (Gemini-j, GLM-j, Haiku-j) use identical prompt wording within each domain; the third judge's script imports the same template object the other two use rather than reimplementing it, so there is one terse prompt per domain rather than three. The text below is the exact string sent to the API. {`anchor`} and {`candidates`} are substituted with the anchor problem or task text and the ten numbered candidate texts before sending.

A.1 MATHEMATICS DOMAIN, TERSE PROMPT (GEMINI-J, GLM-J, HAIKU-J)

```
You are given a math competition problem ("ANCHOR") and 10 candidate
    problems, labeled 1 through 10. Exactly one candidate relies on the
    same underlying mathematical technique or method as the anchor --
    the same core idea you would actually use to solve it -- even though
    it may look completely different on the surface.
```

```
IGNORE surface-level similarity when deciding: shared variable names,
    shared wording, shared language, shared story framing (e.g. both
    about chessboards, or both in the same language), or shared numbers
    do NOT mean same technique. A candidate can look nearly identical in
    phrasing to the anchor and still use a completely different
    technique, and a candidate can look nothing like the anchor on the
    surface and still use the exact same technique.

Focus only on: what mathematical concept, theorem, or method would you
    actually use to solve each problem. Which candidate needs the same
    one as the anchor?

ANCHOR:
{anchor}

CANDIDATES:
{candidates}

Respond with ONLY the candidate number (1-10). No explanation, no other
    text.
```

A.2 Mathematics domain, chain-of-thought prompt (Gemini-J and GLM-J only)

Same task framing as Appendix A.1, with a step-by-step reasoning instruction and a structured final-answer marker.

```
You are given a math competition problem ("ANCHOR") and 10 candidate
    problems, labeled 1 through 10. Exactly one candidate relies on the
    same underlying mathematical technique or method as the anchor --
    the same core idea you would actually use to solve it -- even though
    it may look completely different on the surface.

IGNORE surface-level similarity when deciding: shared variable names,
    shared wording, shared language, shared story framing (e.g. both
    about chessboards, or both in the same language), or shared numbers
    do NOT mean same technique. A candidate can look nearly identical in
    phrasing to the anchor and still use a completely different
    technique, and a candidate can look nothing like the anchor on the
    surface and still use the exact same technique.

Focus only on: what mathematical concept, theorem, or method would you
    actually use to solve each problem. Which candidate needs the same
    one as the anchor?

ANCHOR:
{anchor}

CANDIDATES:
{candidates}

Think step by step. First state the core technique needed to solve the
    ANCHOR. Then, for each candidate in turn, briefly note the technique
    it actually needs and whether it matches the anchor's. After going
    through all 10, conclude.

End your response with your final answer on its own line, in exactly
    this format:
FINAL ANSWER: <candidate number>
```

A.3 MATHEMATICS DOMAIN, CONCISE CHAIN-OF-THOUGHT VARIANT (DIAGNOSTIC ONLY)

This variant was tested on a 30-query sample after the prompt in Appendix A.2 caused 63.3% of GLM-j's full chain-of-thought run to truncate mid-reasoning (Section 8, incident 1). It never replaced that prompt in a full run and is not one of the headline judge configurations; it is included because it is part of the truncation story.

```
You are given a math competition problem ("ANCHOR") and 10 candidate
    problems, labeled 1 through 10. Exactly one candidate relies on the
    same underlying mathematical technique or method as the anchor --
    the same core idea you would actually use to solve it -- even though
    it may look completely different on the surface.

IGNORE surface-level similarity when deciding: shared variable names,
    shared wording, shared language, shared story framing (e.g. both
    about chessboards, or both in the same language), or shared numbers
    do NOT mean same technique. A candidate can look nearly identical in
    phrasing to the anchor and still use a completely different
    technique, and a candidate can look nothing like the anchor on the
    surface and still use the exact same technique.

Focus only on: what mathematical concept, theorem, or method would you
    actually use to solve each problem. Which candidate needs the same
    one as the anchor?

ANCHOR:
{anchor}

CANDIDATES:
{candidates}

Think step by step, but be CONCISE -- this is a triage judgment, not a
    full solution write-up. State the core technique needed for the
    ANCHOR in one sentence. Then for each candidate, in one short phrase
    each, name its technique and say match or no match -- do not
    re-derive or fully solve any candidate. If you are torn between two
    candidates, pick the better one directly rather than re-litigating
    the comparison at length.

End your response with your final answer on its own line, in exactly
    this format:
FINAL ANSWER: <candidate number>
```

A.4 TRAJECTORY DOMAIN, TERSE PROMPT (GEMINI-J, GLM-J, HAIKU-J)

No chain-of-thought variant exists for this domain; chain-of-thought was run only in mathematics.

```
You are given an agent task instruction ("ANCHOR") and 10 candidate task
    trajectories, labeled 1 through 10. Exactly one candidate follows
    the same underlying PROCEDURE as the anchor -- the same
    transformation type (e.g. simple placement, clean-then-place,
    heat-then-place, cool-then-place, examine-with-light, two-object
    placement) -- even though it may involve a completely different
    object.

IGNORE which specific object is involved when deciding: matching object
    names, matching receptacles, or similar surface phrasing do NOT mean
    same procedure. A candidate can involve the exact same object as the
    anchor and still follow a different procedure, and a candidate can
    involve a completely different object and still follow the exact
    same procedure.
```

```
Focus only on: what sequence of actions / transformation type would you
    need to perform for each. Which candidate needs the same one as the
    anchor?

ANCHOR:
{anchor}

CANDIDATES:
{candidates}

Respond with ONLY the candidate number (1-10). No explanation, no other
    text.
```

B WORKED EXAMPLES

B.1 MATHEMATICS: THE FLIPPED-INEQUALITY NEAR-MISS

This is the representative case cited in Section 4. Query `sau_2012_bb6608`:

> Determine all positive integers n such that the inequality
>
> $$\sqrt{x-1} + \sqrt{x-2} + \cdots + \sqrt{x-n} < x$$
>
> holds for every real number $x \geq n$.

Gold target, the easy-tier equivalent reformulation:

> Find every natural number $m \geq 1$ such that for any real number $x \geq m$, the inequality
>
> $$\sum_{k=1}^{m} \sqrt{x-k} < x$$
>
> is satisfied.

Table B.1 lists the top ten candidates by embedding cosine similarity. The rank-1 item is character-identical to the query with the inequality reversed; the gold item sits at rank four.

Table B.1: Top-10 retrieval for query `sau_2012_bb6608`, embedding cosine similarity.

Rank	Cosine	Candidate	Note
1	0.8599	`sau_2012_bb6608::nm::0`	Wrong top-1: identical text, $<$ flipped to $>$
2	0.8396	`sau_2012_bb6608::nm::2`	Sum replaced with product
3	0.8311	`sau_2012_bb6608::eq::medium`	Sibling reformulation
4	0.8208	`sau_2012_bb6608::eq::easy`	**Gold**, ranked fourth
5	0.8157	`sau_2012_bb6608::nm::1`	Square roots changed to cube roots
6 to 10	0.75 to 0.79	various other-base items	Unrelated problems, lower similarity

B.2 TRAJECTORIES: A SIBLING MISS

Query `easy_9` (EASY tier, task type *pick and place simple*, target object `remotecontrol`, goal receptacle `sofa`):

> Place a remote control on the sofa

GLM-j's chosen candidate, `alfworld_286`, is a sibling miss:

> put a remotecontrol in armchair.

Same object, same task type, different receptacle: a literal same-object duplicate. This is the sibling failure mode exactly. The judge picked the trajectory that shares the query's literal object instead of one requiring the same procedure over a different object, which is what strict scoring rewards. A representative strict-gold trajectory for this query, `alfworld_118`, is:

> put some newspaper on sofa.

Different object, same task type, same receptacle as the query: the structural match the judge did not pick.

C PER-CELL FAILURE TAXONOMIES

C.1 MATHEMATICS: BASELINE RETRIEVAL, DOMINANT MISS CATEGORIES

Table C.1 gives the raw counts behind the failure-structure statement in Section 4: the number of strict misses whose rank-1 item is the query's own planted near-miss, and the number whose rank-1 item is a sibling equivalent reformulation.

Table C.1: Mathematics baseline, rank-1 miss categories by embedder and tier (counts, n=500).

Embedder	Tier	Own near-miss	Sibling variant
Gemini-emb	Easy	420	13
Gemini-emb	Hard	420	74
Qwen-emb	Easy	446	7
Qwen-emb	Hard	446	50

C.2 TRAJECTORIES: LLM-RERANKER MISS TAXONOMY, ALL NINE CELLS

Table C.2 gives the per-cell breakdown behind the sibling-dominance statement in Section 5. Percentages are of strict misses within each judge-by-embedder cell.

Table C.2: Trajectory reranker strict-miss taxonomy, n=118 queries per cell.

Embedder	Judge	Misses	Sibling	Near-miss	Other	Unparsed
Qwen-emb	Gemini-j	74	85.1%	13.5%	1.4%	0.0%
Qwen-emb	GLM-j	60	88.3%	11.7%	0.0%	0.0%
Qwen-emb	Haiku-j	72	84.7%	11.1%	2.8%	1.4%
Gemini-emb	Gemini-j	70	85.7%	14.3%	0.0%	0.0%
Gemini-emb	GLM-j	50	80.0%	14.0%	2.0%	4.0%
Gemini-emb	Haiku-j	71	85.9%	12.7%	1.4%	0.0%
MiniLM	Gemini-j	74	74.3%	20.3%	5.4%	0.0%
MiniLM	GLM-j	65	72.3%	7.7%	9.2%	10.8%
MiniLM	Haiku-j	80	75.0%	17.5%	5.0%	2.5%

D VALIDATION GATES AND INTEGRITY AUDITS

D.1 VALIDATION GATES

Before any new measurement, each domain reproduces a published number with the original artifacts. MathNet's Table 4 easy-tier values are recovered within one point (12.2, 89.8, and 97.6% for Hit@1, Hit@5, and Hit@10 against the published 11.36, 90.68, and 96.93%), and the trajectory benchmark's MEDIUM mean average precision is recovered within 0.007 (0.753 against the published 0.746) using the source paper's own encoder. A deviation beyond either tolerance would have halted the study.

D.2 Integrity audits

Every API response is cached and every reported number is recomputed from frozen aggregates. Standing audits include finish-reason checks on all generations, per-condition truncation rates, and spend reconciliation (final cost $17.24 over 4,338 calls). Section 8 documents the eight incidents these audits surfaced.

D.3 Evaluation-integrity findings

We record eight incidents. Most were caught by routine audits that existed only because of earlier incidents, and in every case we excluded the affected numbers rather than attempting repairs. (1) GLM-j chain-of-thought judging truncated at its token cap in 63.3% of 2,000 responses while parsing successfully; a parsed response is not a concluded one, and the condition was excluded entirely. (2) A six-condition solver pilot truncated 49.6% of answers, correlated with experimental condition (50 to 53% with context versus 38.6% without), which invalidated it outright; the oracle condition's accuracy on complete answers was 82.9% against a reported 63.6%. (3) A hidden-thinking-token quirk in one judge API silently burned output budgets twice before being fixed at the configuration level. (4) A spend-tracking discrepancy against the maintainer's own dashboard triggered a full stop and a call-by-call reconstruction from caches, closing the gap without new spend and producing code-enforced spend tracking. (5) Successive solver-cap increases reduced truncation from 80% to 50% to 31% without resolving it, which suggests truncation tracks problem difficulty at least as much as budget. (6) A third judge initially appeared infeasible: DeepSeek-v4-flash as a reranker judge was still mid-reasoning with empty output at 24,000 tokens and 227 seconds on a single query, matching GLM-j's chain-of-thought behavior across a second provider, while a non-reasoning judge (Haiku-j) later completed the full 2,354-call run with 0.1 to 0.9% truncation. What governs visible deliberation appears to be task framing rather than raw capability. (7) A shared-infrastructure solver run degraded fourfold under sustained load and was abandoned rather than waited out. (8) Bootstrap resampling tightened this project's own earlier contamination claim from "positive in most cells" to "confidently non-null in one." Three further problems were caught before submission against frozen aggregates or raw caches: a suffix-matching scoring bug in the CI pass, a silent dependency drift that broke an embedder, and a set-equality claim built on two counts that coincided at 146 without being the same set. Standing practice: finish-reason audits after every generation run; exact-tokenizer verification of suspicious length distributions; per-condition truncation reporting wherever a budget cap exists; recomputation verified against frozen aggregates.

E Additional results moved from the main text

E.1 Contamination table

Table E.1: Contamination gap (well-known minus rest), hard tier, terse prompt.

Judge	Candidates	Gap (pts)	95% CI
Gemini-j	Gemini-emb	+19.8	**[+6.7, +33.2]**
Gemini-j	Qwen-emb	+8.1	[-1.4, +18.6]
GLM-j	Gemini-emb	+1.4	[-5.0, +8.8]
GLM-j	Qwen-emb	+5.6	[-1.2, +13.9]
Haiku-j	Gemini-emb	+4.5	[-1.6, +12.0]
Haiku-j	Qwen-emb	+2.4	[-1.6, +7.9]

E.2 Deployment divergence

Deployment divergence. Two servings of identical Qwen3-Embedding-8B weights (mean pairwise cosine 0.9947 over 500 identical texts, never above 0.999, both unit-normalized) agree on five of six retrieval metrics. They differ significantly on the sixth, hard-tier Hit@10: 17 discordant queries in one direction against 1 in the other (McNemar exact p = 0.00014), and the difference lands exactly where gold-versus-decoy margins are thinnest. The practical lesson is that two deployments of the same model cannot be treated as interchangeable when ranking margins are thin.

E.3 QUERY-SUBSET SPLIT

We report the two query subsets separately because they behave differently. The original coverage-balanced 40 queries are substantially harder (strict Hit@1 10.0% versus 21.8% for Qwen-emb; 5.0% versus 20.5% for Gemini-emb) than the raw validation-split 78. This is a benchmark-construction effect confounded with phrasing style, and we flag it as such.

E.4 FAME-COMPOSITION CHECK FOR THE DOWNSTREAM CEILING

A composition check addresses the natural follow-up, whether the ceiling reflects recognition of famous problems. 30 of the 210 queries come from well-known competitions, and near-ceiling accuracy on completable problems is fame-uniform (100.0% versus 99.1% under Grader A on the complete subset), so the ceiling does not reduce to fame. The well-known subset's higher raw accuracy in the full sample (80.0% versus 67.8%) tracks its lower truncation rate (26.7% versus 31.1%), which is what the completion-proxy reading predicts. Group counts are small and these figures are reported descriptively, without significance testing.

F UTILITY GRADER PROMPT

A single grading prompt is built once per query and condition and passed to both graders (Grader A, Gemini 3.1 Flash-Lite; Grader B, GLM-5.2). {problem}, {reference}, and {candidate} are substituted with the query text, the concatenated reference solution(s), and the solver's candidate solution before sending.

```
You are grading a candidate solution to a mathematics olympiad problem
    against a reference solution.

PROBLEM:
{problem}

REFERENCE SOLUTION:
{reference}

CANDIDATE SOLUTION:
{candidate}

Score the candidate solution's correctness on a scale from 0 to 7, where
    7 means fully correct (or containing only minor errors that don't
    affect the core reasoning) and 0 means completely incorrect or no
    meaningful progress. Judge whether the candidate's mathematical
    reasoning and final conclusion are consistent with the reference,
    not writing style or presentation. The reference may be in a
    different language than the candidate -- grade the mathematical
    content, not the language.

Respond with ONLY the integer score (0-7). No explanation, no other text.
```