THE TRUTH IS OUT THERE: SEPARATING OWNER DEFINITIONS FROM DATABASE FACTS IN TEXT-TO-SQL

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-SQL benchmarks give agents two kinds of missing information at once. The database settles some of it, such as which column holds a concept or how its values are encoded. The data's owner decided the rest, such as where a threshold lies or which categories count, and no query recovers it. Because benchmark hints supply both together, a score cannot say which one an agent lacked. We separate them with *definition relocation*. For an underspecified question, we write several definitions an analyst could hold, make one of them correct, and move it between the prompt, a glossary table in the database, the schema text, the agent's transcript, and the data owner. Each definition implies a different answer, so the answer shows which one the agent acted on. Where the definition sits matters as much as whether it exists. Open-weight models almost never query the glossary, so a definition stored there is worth nothing to them, while the same rows returned by a query run on their behalf close 35 to 85% of the gap to stating the definition in the prompt. Agents that explore on their own do recover stored definitions, one sentence of instruction makes a hosted model do the same, and how often an agent queries the glossary predicts what a stored definition is worth to it. Documentation of what the database already settles behaves differently: it helps small models and adds nothing measurable for strong ones. A second set of concepts reproduces these results.

1 INTRODUCTION

An agent that answers questions over a database can run queries before it answers: list the distinct values of a column, follow a foreign key, read a lookup table. A question like *how many large wildfires occurred in California in 2007* does not say what "large" means. The usual premise is that an agent with database access can work it out.

Three lines of work address the gap between what a question says and what a query needs. Benchmarks attach the missing information to the question: BIRD supplies expert evidence per item (Li et al., 2023), KaggleDBQA ships column documentation (Lee et al., 2021), and LiveSQLBench supplies a typed knowledge base (BIRD Team, 2025). Interactive benchmarks let the agent recover masked knowledge by asking a user or exploring the environment (Huo et al., 2026), and enterprise evaluations take exploration of the warehouse as the premise (Lei et al., 2025). Systems work induces the missing semantics automatically, from execution feedback or from an agent's own past errors (Lee et al., 2026; Agarwal et al., 2026). All three supply or derive the information in one place, so the effect of *where* it lives has not been isolated.

An agent can in fact work out part of it. Which column records burned area, how the cause codes are spelled, how fires join to states: the data settles all of these, so an agent that queries for them finds them. Where the owner drew the line for "large" is different. A database of fires is equally consistent with 300 acres, 1,000 acres, and 5,000 acres. There is no query that returns the choice, because it lives with whoever asked the question. Benchmark hints typically put both kinds of information in the same sentence and supply them together, so their effects have not been separated.

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.

Why the obvious approach fails. The direct way to separate them is to take questions with human-written hints, label each fact by where its information could have come from, and compare prompts built from one kind or the other. The labeling does not replicate. Five model annotators from five model families and one human annotator, all applying one codebook, agree on the database-versus-owner distinction at Cohen's κ between 0.016 and 0.399 (Appendix J). Two further instrument designs do no better. The distinction itself is consistent where both annotators apply it: when two annotators both label a fact as either database-derivable or an owner choice, they agree on which of the two it is 75 to 97% of the time, depending on the pair. What fails to reproduce is applying it to a single authored sentence, so no claim can rest on it.

Assigning meaning instead of inferring it. We therefore stop inferring what a question meant and decide it. For an underspecified question, we write two or three definitions a competent analyst could hold, verify on the shipped data that they give different answers, and make one of them the assigned definition, independently of the agent and of the data. The assigned definition becomes the ground truth. We then move it while holding the question, the database, the harness, and the tool budget fixed. We consider six positions: *absent*, *stated* in the prompt, *stored* in a glossary table whose rows the schema hides, *conflicting* with a stated definition, *delivered* as schema text, and *served* as the result of a glossary query we execute on the agent's behalf.

This design has two advantages. First, identification: because each definition implies a different answer, a correct answer says which definition the agent acted on rather than only that it was right. Second, a decomposition, since each contrast between positions bounds one step. Absent against stated asks whether the information is *available* at all. Stored against served asks whether the agent *retrieves* it, holding availability fixed. Served against stated asks whether it *uses* what it received, holding retrieval fixed.

Evaluation. We build two definition benchmarks: RD-BIRD, 432 items over BIRD databases, and RD-Kaggle, 675 items over KaggleDBQA databases. We call a model run in one mode a *setting*. Nine settings share one tool-calling harness: Qwen3-4B, 8B, 14B, and 32B, Qwen3-14B in thinking mode, Mistral-Small-24B, GPT-5 mini, GPT-5, and GLM 5.3. Three coding agents run under their own loops. Throughout, the metric is the share of sessions whose executed answer matches the assigned definition. KaggleDBQA's published column documentation, which describes what the data already settles, supplies the other side of the comparison.

Results. Without added instruction, stating a definition is worth 34.8 to 62.2 points of matching accuracy. On RD-Kaggle, storing the same definition in the database leaves every open-weight setting within 0.8 points of its no-definition accuracy. The reason is behavioral: those settings query the glossary in 0 to 2% of sessions, apart from Qwen3-32B at 16%, and they stop after 1 to 2 statements on average, far inside the budget of 12. Getting an agent to look is possible: one sentence of instruction raises Mistral's query rate from 1% to 76%, and coding agents exploring under their own policy on the same items answer correctly in 66.7 to 88.9% of sessions against 31.1 to 34.8% when no definition exists. Running the query for the agent recovers 14.8 to 30.4 of the missing points for open-weight settings and all of them for GPT-5, which rises from 45.9% to 96.3%. The same pattern holds when the agent may ask the data owner instead: open-weight settings ask in 6 to 23% of sessions and act on at most a third of what they are told. On the other side of the boundary, the value of documentation falls from $+10.9$ points $[+4.2, +17.9]$ at Qwen3-4B to $+0.7$ $[-2.5, +4.1]$ at GPT-5, while a stated definition shows no decline across the Qwen family (34.8 to 35.6). The paired difference between the two ends is $+10.1$ points $[+2.6, +16.2]$. We claim only that difference, since the ordering of the settings in between is not reliable (Appendix H).

Contributions.

- An instrument that assigns the meaning of an underspecified question and relocates it, so the answer shows which meaning the agent used.
- Operational contrasts that separate availability, retrieval, and use by moving one fixed definition rather than labeling text.
- Evidence that a stored definition helps only agents that look for it, and that open-weight settings on a fixed loop rarely look, whether the channel is a glossary or a person.

Section 2 states what exploration can and cannot settle and what follows for measurement. Section 3 describes the instrument and the harness. Section 4 reports availability, retrieval, and use. Section 5 compares the channels a deployed system can offer by what each costs and recovers. Section 6 shows the two sides of the boundary on existing benchmarks, including the documentation KaggleDBQA already ships, and Section 7 describes the controls.

2 What Exploration Can and Cannot Settle

Two kinds of missing information. Answering a question over a database requires facts of two kinds. The first kind the data can usually determine: which column holds a concept, how a value is encoded, what format a date is stored in, which key joins two tables. An agent recovers these by querying. Two analysts who query the data will agree on them, because the data decides. The second kind is information that the data leaves open: a threshold, a formula, which categories count, a time window, a tie rule, the representation the answer should take. Here the database is equally consistent with several choices and the analyst who asked has made one. We call such a choice an owner convention, and the sentence that states it an owner definition.

The distinction concerns availability. An owner's choice can be written into the database: a glossary table recording that "large" means 1,000 acres makes it recoverable by query like any other stored fact. For an agent, what matters is therefore whether the information is present somewhere it can reach, whoever originally decided it.

Why guessing is not recovery. When k definitions remain possible and give different answers, an agent with a fixed prior matches the intended one about $1/k$ of the time. Measured accuracy in the absence of any supplied definition therefore reports how often the agent's default happens to coincide with the owner's choice. This is worth reporting, and we do, but it is not evidence that anything was recovered.

Why supplying information is not enough either. An agent handed the right definition still has to write the query, so a setting that gains little from being told something may need the information as much as any other and simply be unable to act on it. Comparing what kinds of information are worth therefore confounds their value with the ability to use them, unless the design separates the two.

What this implies for measurement. Three requirements follow. The intended meaning must be known to the experimenter, or a correct answer cannot be distinguished from a lucky default. The same information must be movable between positions while everything else is held fixed, or availability cannot be separated from retrieval. Finally, the step that turns a received definition into SQL must be measured on its own, or use cannot be separated from either. No existing benchmark meets all three. Ambiguity benchmarks score a question by how many of its admissible readings a model recovers (Saparina & Lapata, 2024; Bhaskar et al., 2023), so no reading is designated as the one the asker intended. Knowledge-supplying benchmarks fix one but always deliver it in the prompt (Li et al., 2023; BIRD Team, 2025), so availability and retrieval never come apart. We therefore build the conditions.

3 The Instrument

Concepts and definitions. A concept is a business term that appears in a question without being defined there, together with two or three definitions a competent analyst could hold. For a wildfire database the term is "large wildfire" and the definitions are at least 300 acres, at least 1,000, and at least 5,000. We require that the definitions give different answers on the shipped data, checked by executing each definition's reference query, so that an answer identifies a definition rather than leaving several consistent with it. Questions are phrased so that they use the term without hinting at which definition applies: *how many large wildfires were there in California in 2007*, not *how many fires burned more than 1,000 acres*.

Assignment and positions. Each item is assigned one definition, whose reference query serves as the gold answer. RD-BIRD draws the assigned definition uniformly at random with a recorded

seed. RD-Kaggle instead crosses each of its 50 questions with every one of its definitions, which balances the assignment exactly. In both, the assignment is independent of the agent and of the data. No agent can therefore infer the assigned definition without being told. Six positions vary where the assigned definition can be found:

- **Absent**: neither the prompt nor the database supplies the choice.
- **Stated**: the question arrives with the assigned definition stated.
- **Stored**: a glossary table holds it. The schema shown to the agent lists the table but hides its rows, so the agent must query for them.
- **Conflicting**: the prompt states the assigned definition while the glossary holds a different one.
- **Delivered**: the glossary rows are printed with the schema, so reading them requires no query.
- **Served:** we execute the glossary query ourselves and seed the agent's transcript with the statement and the rows it returned, as though the agent had issued it. The seeded call is not charged to the tool budget.

This last position is what separates retrieval from use. In the stored position a wrong answer can mean the agent never looked, or that it looked and misapplied what it found. In the served position receipt is certain, so a wrong answer can only be the second of these. These are operational contrasts. Serving also shows the agent one use of its tool and leaves it one extra call, and served against stated also changes the form of the definition, from table rows to a sentence, so each contrast bounds a step rather than isolating a mechanism.

Two benchmarks. RD-BIRD uses 9 concepts on BIRD databases with definitions written as SQL fragments, 432 items across four positions. RD-Kaggle uses 10 concepts on 7 of KaggleDBQA's 8 databases with plain-language definitions that name no column and no operator, 50 questions crossed with their definitions into 135 items, run in each position. We call one setting in one position a cell. The first five positions give 675 items, and the served position and the asking channel of Section 5 reuse the same items. The default-matching reference, the accuracy of an agent that always writes a correct query for one fixed admissible definition, is 35.2% for RD-BIRD and 37.0% for RD-Kaggle. Absent accuracy falls below it when an agent's default lies outside the candidate set or its query fails. RD-Kaggle carries the body results because its definitions are prose rather than SQL fragments. We report RD-BIRD in Appendix A.

Harness. One session is a tool-calling loop with two tools: `run_sql`, which executes one read-only SQLite statement and returns at most 20 rows, and `submit_answer`, which ends the session with a query. The budget is 12 tool calls. Open-weight settings run on NVIDIA A40 GPUs through a local server, and the hosted settings, GLM 5.3 included, use their providers' APIs. Open-weight settings sample at temperature 0.2. The GPT-5 models do not accept a temperature setting and use the provider default. The prompt holds the schema with three example rows per table, the question, and whatever the position supplies. The nine settings of Section 1 share this harness. Three coding agents run under their own loops with the `sqlite3` command line and are never pooled with the rest. Appendix B gives their versions and autonomy levels.

Scoring. We execute the submitted query and compare its result set against the reference query of every candidate definition, taking exact set equality first and then, on RD-Kaggle, a documented tolerance for rounding and for extra columns. An answer that matches the assigned definition counts as a match. An answer matching a different definition is recorded as a match to that definition, making the conflicting position readable. Items within a concept share a question template and a database, so confidence intervals resample concepts rather than items (a concept bootstrap).

4 AVAILABLE, RETRIEVED, USED

What is absent is not recovered, at any capability. Table 1 reports the plain-language experiment. With no definition supplied, every setting sits close to its own prior: 4.4% for Qwen3-4B,

13.3% for Mistral, 31.9% for GPT-5, against a default-matching reference of 37.0%. Without added instruction, stating the definition in the prompt raises matching by 34.8 to 62.2 points. The direction of this contrast is fixed by design, since the assigned definition is not a property of the database. It serves as a manipulation check. Its size measures how well each setting turns a stated definition into SQL, which makes it the reference for the contrasts that follow.

Table 1: Plain-language definitions: answers matching the chosen definition (%, 135 items per cell, default-matching reference 37.0%). "Queried glossary" is the share of sessions with glossary queries in the stored position. "+ instruction" asks agents to check for business definitions. Qwen3-14B with thinking, the one shared-harness setting not shown here, is reported on RD-BIRD in Appendix A. The coding agents run under their own loops and are reported in Appendix B.

Setting	Absent	Stated	Stored	Conflicting	Delivered	Queried glossary
Qwen3-4B	4.4	40.0	4.4	42.2	23.7	0%
Qwen3-8B	8.9	43.7	8.1	41.5	24.4	0%
Qwen3-14B	9.6	44.4	9.6	43.7	36.3	2%
Qwen3-32B	10.4	45.2	9.6	47.4	40.0	16%
Mistral-Small-24B	13.3	54.8	12.6	54.1	39.3	1%
GPT-5 mini	28.1	90.4	35.6	90.4	87.4	8%
GPT-5	31.9	92.6	45.9	91.9	94.1	24%
GLM 5.3	36.3	98.5	60.0	96.3	95.6	41%
Qwen3-8B + instruction	8.1	40.7	7.4	41.5	26.7	4%
Qwen3-14B + instruction	10.4	40.7	8.9	41.5	38.5	7%
Mistral + instruction	15.6	55.6	18.5	35.6	42.2	76%
GPT-5 mini + instruction	24.4	92.6	85.2	65.2	93.3	96%
GPT-5 + instruction	31.9	94.1	93.3	60.0	94.8	97%

A stored definition is worth nothing to agents that do not query for it. On RD-Kaggle, for every open-weight setting, the stored cell sits within 0.8 points of the absent cell, and no concept-bootstrap 95% interval for the difference extends beyond -4.0 or $+3.0$. The glossary query rate explains why: Qwen3-4B and Qwen3-8B query the glossary in 0% of stored sessions, Qwen3-14B in 2%, and Mistral in 1%. The only exception is Qwen3-32B, which queries in 16% of sessions and gains nothing (Appendix C explains why). These settings also explore little: 5 to 29% of their stored sessions contain any statement besides the final query, against 62 to 73% for GPT-5 mini and GPT-5. Two of the hosted models, GPT-5 mini and GPT-5, query more often, in 8% and 24% of sessions, and gain from the stored definition: pooled with a replication set of ten further concepts (Appendix D), $+11.8$ $[+5.8, +18.0]$ and $+26.3$ $[+12.2, +41.0]$. On RD-Kaggle's ten concepts alone the gains are 7.4 and 14.0, with intervals that include zero.

Querying is a policy that one sentence changes. One sentence added to the prompt, *before answering, check whether the database defines any business terms used in the question*, moves Mistral's glossary query rate from 1% to 76%. The same sentence moves Qwen3-8B from 0% to 4% and Qwen3-14B from 2% to 7%. A similar sentence in the tool description moves no open-weight setting's accuracy by more than 2.3 points, concept-weighted (Appendix E). The coding agents query without any such instruction. With full transcripts recorded, Codex queries the glossary in 75.6% of stored sessions and answers 99.0% of those correctly, against 27.3% when it does not query. All three reach 66.7 to 88.9% against 31.1 to 34.8% on these items, and the two run on RD-BIRD reach 90.7% and 100%. Looking is necessary but not sufficient. On the shared harness, the instruction makes GPT-5 query in 97% of stored sessions and lifts it from 45.9% to 93.3%, and GPT-5 mini from 35.6% to 85.2% at 96%, as high as the coding agents, while Mistral queries in 76% and ends only 2.9 points above its no-definition accuracy under the same instruction. The coding agents' gains therefore combine looking with a model that can use what it finds. GLM 5.3 separates the two: on our harness, without any instruction, it queries in 41% of stored sessions and gains 22.3 points $[+7.7, +38.0]$, concept-weighted, while under the Claude Code loop the same model gains 35.6.

Serving the rows separates retrieval from use. Table 7 reports the served position alongside the others. For the hosted settings serving closes essentially the whole gap to stated: GPT-5 rises from 45.9% stored to 96.3% served, against 92.6% when the definition is stated, and GPT-5 mini from 35.6% to 88.9% against 90.4% stated. For the open-weight settings both steps fail. Serving adds 14.8 to 30.4 points, and stating the same definition in the prompt adds another 5 to 27. Scaled to each setting's stated accuracy, serving recovers 35 to 85% of the gap between stored and stated, while storing recovers none of the gap between absent and stated (Appendix F). The failure to retrieve is therefore not an artifact of how well these settings write SQL. The replication set, on all eight KaggleDBQA databases, reproduces the stored and served contrasts. Pooled over the 20 concepts and the four open-weight settings, stored minus absent is $+0.3$ points $[-1.5, +2.8]$ and served minus stored $+29.3$ $[+19.1, +40.9]$. The concept-weighted served-minus-stored difference is $+18.7$ points $[+1.3, +41.3]$ at Qwen3-4B, $+20.0$ $[+4.7, +40.7]$ at 8B, $+20.0$ $[+7.7, +32.7]$ at 14B and $+15.7$ $[+3.7, +29.0]$ on Mistral, over 10 concepts.

The same text is worth less as data than as instruction. Served lands within 7 points of delivered, where the identical rows are printed with the schema, for four of the five open-weight settings (Mistral: 27.4 against 39.3, with delivered spanning 23.7 to 40.0), and both sit below stated (40.0 to 54.8). Delivered and stated both place the definition in the prompt and differ only in form, rows against a sentence, while served and delivered share the form and differ only in position. Two further serving controls test whether the remaining gap comes from form (Appendix E). Seeding a lookup that returns no definition changes no open-weight setting's accuracy (every interval includes zero), so the served gain is the information rather than a demonstration of the tool. Serving the definition as one sentence instead of rows adds 6.4 to 17.7 points, concept-weighted. The sentence in the transcript lands within about 5 concept-weighted points of the same sentence in the prompt for three of the four settings, so on RD-Kaggle the remaining gap comes mostly from form rather than from position (Appendix D reports exceptions).

A conflict exists only once the agent retrieves. In the conflicting position every setting that does not query the glossary follows the prompt, which is the only source it read. Once GPT-5 mini queries under the instruction, about a third of its answers follow the glossary instead, and its share matching the stated definition falls from 90.4% to 65.2%. GPT-5 falls from 91.9% to 60.0%, with 36.3% following the glossary. Mistral under the instruction shows the same shift, from 54.1% to 35.6%.

5 The Channels a Deployed System Can Offer

A system that wants an agent to have the owner's definitions has a small set of choices: put them in the prompt, store them where the agent can query, print them with the schema, run the lookup itself, or leave the owner available to answer questions. The preceding section measured the first four. This section measures the last one and compares all five directly.

An agent that will not query will not ask either. We add the asking channel on the absent items, where the definition exists nowhere the agent can reach. The agent may ask one question. Whatever it asks, the owner replies with the definition assigned to that item. No model plays the owner, so the comparison across settings is not confounded by one simulator being more helpful than another. This makes the resulting numbers an upper bound on what asking can buy. Table 8 in Appendix F gives the counts. The same pattern appears in this channel, although the owner is always available. The open-weight settings ask in 6 to 23% of sessions, against 0 to 16% that query the glossary. When they do ask and receive the definition, they mostly fail to act on it: Mistral answers correctly in 1 of the 8 sessions where it asked, Qwen3-8B in 3 of 26, and Qwen3-4B in none of 13. On BIRD Mini-Dev, eight of ten hosted settings, including the newest Claude and GPT models, ask in at most 3% of sessions when offered a similar tool (Appendix G). Only GPT-5 mini and GPT-5 use the channel effectively: GPT-5 mini asks in 35% of sessions and answers 40 of those 47 correctly, close to the 90.4% it reaches when the definition is stated, and GPT-5 asks in 25% and answers 30 of 34. Retrieval and use consistently fail together, whichever channel is offered.

What each channel costs. Serving the glossary costs one statement per session and adds input and output tokens whether or not the question needs the rows: 3 to 2,768 per session on average for the other seven settings, with a median of 463. For GLM 5.3, which otherwise spends tokens searching,

serving saves 1,739. In return, it gains 14.8 to 53.3 points. Instructing the agent to look is cheaper in tokens and much less reliable: it raises Mistral's query rate from 1% to 76% and its accuracy by 5.9 points over the stored condition without the instruction, but does not raise the accuracy of Qwen3-8B or Qwen3-14B at all. Leaving the owner available is cheaper still and less reliable again, since the agent must decide to ask: 21.5 points to GPT-5 mini, 13.3 to GPT-5, and at most 5.9 to any open-weight setting. In the settings we test, storing the definition without an instruction is the weakest option, since it leaves the information unused in five of eight settings. The ordering holds across the capability range we test, which suggests putting owner-supplied definitions in the context and treating a stored glossary as a fallback for an agent that has been told to read it.

A mixture of absent and served predicts the stored cells. The stored position mixes two behaviors. With probability r, the share of sessions that query the glossary, the agent behaves as it does when handed the rows. Otherwise it answers from its default. Predicting $r \cdot \text{served} + (1 - r) \cdot \text{absent}$ matches the observed stored accuracy within 2.9 points on average over the seven cells where more than 5% of sessions query the glossary, the cells where the prediction differs from the absent cell, and within 2.0 over all twelve. The accuracies on the right come from other conditions, while r is measured in the stored sessions themselves, so the prediction tests how retrieval converts into accuracy rather than whether retrieval happens. The instruction case is the sharpest test: it moves GPT-5 mini by 49.6 points and GPT-5 by 47.4, yet the prediction misses each by only 1.1. Two cells miss by more than five points, both in the same direction, with the agent's own query delivering less than ours (Appendix C).

6 The Two Sides of the Boundary

The instrument shows what happens to information the database does not settle. Existing benchmarks supply the other half: information it does.

KaggleDBQA ships column descriptions and code maps written by its authors (Lee et al., 2021), which we take as documentation of what the data settles. Supplying them is worth $+10.9$ points $[+4.2, +17.9]$ to Qwen3-4B and $+0.7$ $[-2.5, +4.1]$ to GPT-5, a paired difference of $+10.1$ $[+2.6, +16.2]$ that survives dropping any one database. Stating a definition, by contrast, is worth $+34.8$ to $+35.6$ across the Qwen family. We claim only the difference between the two ends, and Appendix H gives every setting, the full ordering, and its statistics.

When the metric is invented outright. On LiveSQLBench-Lite, whose business metrics exist only in a supplied knowledge base (BIRD Team, 2025), every setting scores 0.6 to 1.7% without the entries and up to 25.6% with them (Appendix E).

7 Controls

The glossary table is not a giveaway. A table named `glossary` announces what it holds, so a retrieval rate measured against it might be measuring recognition of the name. We built a variant of RD-Kaggle that stores the definitions in a table named `notes` and ran the instruction condition on it. Retrieval is essentially unchanged. Under the instruction, Mistral queries the renamed table in 70% of sessions against 76% with the original name, and answers correctly in 23.7% against 18.5%. Qwen3-14B queries in 9% against 7%, with its accuracy unchanged at 8.9%. The name of the table therefore explains neither why some agents query it nor why others do not.

Further controls. The candidate thresholds are not recoverable from the data, whose values are continuous where the cut-offs sit. The categorical concepts are unrecoverable by construction. The handling of submissions that are not queries, and an audit of BIRD's corrected reference queries, are reported in Appendix E.

Repeatability. Repeating the plain-language experiment for Qwen3-4B, Qwen3-8B, Qwen3-32B, and Mistral changes no cell by more than 2.9 points.

Absent accuracy measures a prior. With no definition supplied, models do not sample among admissible definitions. Each RD-Kaggle question is asked once per definition. Without a definition,

GPT-5 mini and GPT-5 give the same answer to every copy of 50% and 64% of the questions, whereas with the definition stated almost no question gets one fixed answer (0 to 6%). Some of these defaults coincide with KaggleDBQA's own reference queries, which may reflect exposure to that public benchmark (Appendix I). Such exposure would raise absent accuracy and so make the gains we report smaller.

8 Discussion and Limitations

Which source should an agent trust? We score the conflicting position against the prompt, because the prompt is where the asker speaks. A production system might instead trust an approved glossary, or refuse and ask. We measure what agents do when sources disagree. Which policy is right is a separate question.

What the concepts cover. We wrote the 9 and 10 business terms of the two benchmarks ourselves, while a coding agent drafted the replication set's ten (Appendix D). We intentionally gave the glossary tables descriptive names. There are conventions our concepts do not cover: those spread across several documents, those implicit in data patterns, and those depending on who is asking. We choose definitions independently of the visible data. Real questions often carry clues about which definition is intended, so an agent that exploits them would probably do better than ours do here. One concept, *underground release*, has two definitions that are arguably different notions rather than two readings of one term, making it the weakest of RD-Kaggle's ten.

What the measurements can and cannot show. Mentioning a glossary in a query does not establish that the agent actually read or understood the definition. Our comparison allows extra columns and, where a result matches several definitions, takes the first in a fixed order. In practice only 2 of 5,535 RD-Kaggle sessions match more than one definition, and scoring only unique exact matches changes no contrast by more than 2.2 points (Appendix I). A match still shows the outcome a definition implies, not the model's internal reading. On KaggleDBQA the same allowance can hide mistakes in which columns a query returns. We also take KaggleDBQA's documentation as describing what the data settles on its authors' account, without checking each entry's recoverability.

Generalization. Settings differ in family, size, and thinking mode at once, so baseline accuracy does not cleanly isolate any one of them. The BIRD comparison in Appendix J rests on labels whose reliability we measured and found wanting, and on reference queries written alongside the hints. It carries no claim in the body.

9 Related Work

This work sits between three literatures: one enumerates the readings a question admits, one attaches to each question the information it omits, and one builds the missing semantics once and reuses them. Each treats the missing information as a single undifferentiated quantity. We instead make the position of one assigned definition the experimental variable. Table 2 compares the designs.

Ambiguity as coverage. AMBROSIA and AmbiQT construct questions with several valid readings and measure whether a model covers them (Saparina & Lapata, 2024; Bhaskar et al., 2023). Coverage is the right target when no reading is privileged. We assign one reading, which converts coverage into accuracy and lets the same information move between prompt and database.

Supplying knowledge. BIRD attaches expert evidence to each question (Li et al., 2023), KaggleDBQA ships column documentation (Lee et al., 2021), LiveSQLBench supplies a typed knowledge base (BIRD Team, 2025), BIRD-Interact offers both a knowledge base and a simulated user and lets the agent choose between them (Huo et al., 2026), and EntSQL grounds questions in enterprise documents carrying internal metrics and reporting conventions (Liao et al., 2026). This line of work establishes that missing knowledge hurts, but each fact has one fixed home. We instead hold one definition fixed and move it between homes.

Table 2: How the design differs from the closest prior evaluations. "Assigned": the intended meaning is set by the experimenter independently of the data. "Relocated": the same fact is placed in more than one position. "Must retrieve": the agent has to decide to fetch the fact. "Retrieval intervened": retrieval is manipulated rather than read from transcripts. Works: Saparina & Lapata (2024); Bhaskar et al. (2023); Li et al. (2023); Lee et al. (2021); BIRD Team (2025); Huo et al. (2026); Rumiantsau & Fokeev (2026); Agarwal et al. (2026).

Work	Assigned	Relocated	Must retrieve	Retrieval intervened
AMBROSIA, AmbiQT	–	–	–	–
BIRD, KaggleDBQA, LiveSQLBench	–	–	–	–
BIRD-Interact	–	–	✓	–
Semantic-layer study	–	–	–	–
Tk-Boost	–	–	✓	✓
This paper	✓	✓	✓	✓

Knowledge the agent builds for itself. Tk-Boost compares an agent's failed queries with ground-truth SQL to write corrective statements, and ablates their retrieval and application with the statements held fixed (Agarwal et al., 2026). GATE induces groundings from execution feedback alone (Lee et al., 2026), which can reach only what execution settles. We instead hold one definition fixed and vary where it lives, including positions the agent must choose to query. Appendix K relates our design to semantic-layer documents (Rumiantsau & Fokeev, 2026), retrieval-versus-use diagnostics (Es et al., 2024; Liu et al., 2026), counterfactual tasks (Wu et al., 2024), clarification (Zhao et al., 2024), and context conflict.

10 Conclusion

We assigned the meaning of underspecified questions and moved it between the prompt, a glossary table, the schema, the transcript, and the data owner, so that each answer shows which meaning the agent acted on. Stating a definition helps every model. Storing it helps only agents that look for it, which open-weight models on a fixed loop rarely do and coding agents under their own loops usually do. Where definitions live is therefore a design decision worth measuring.

AI Use Statement

The experiments, the analysis scripts, and this manuscript were produced with substantial assistance from a coding agent, under author direction and review. Four specific uses are worth naming. Model annotators label the atomic facts in Appendix J, which also reports the reliability of that labeling. The written justifications accompanying the human annotator's labels were drafted with model assistance. The labels themselves were assigned by the annotator. The authors wrote the concepts, definitions, and questions of the two definition benchmarks. The ten concepts of the replication set in Appendix D were instead drafted by a coding agent under author direction. The audits of benchmark specification quality were model-assisted and are reported as such.

Reproducibility Statement

An anonymized copy of the release accompanies this submission as supplementary material. It contains the two benchmark builders with their concepts, definitions, and questions, the tool-calling harness and the script that runs the coding agents (the local launcher it calls is not included), every session record behind the tables, and the scripts that recompute each table from those records, or, for the mixture table, from the other tables' cells, with the command for each in its README. Section 3 and Appendix A give the scoring rules. Each coding-agent run records the agent's command-line version, in its records or a sidecar file. Sampling in the shared harness is not seeded.

REFERENCES

Shubham Agarwal, Asim Biswal, Sepanta Zeighami, Alvin Cheung, Joseph Gonzalez, and Aditya G. Parameswaran. Arming data agents with tribal knowledge. *arXiv preprint arXiv:2602.13521*, 2026. doi: 10.48550/arXiv.2602.13521. URL https://arxiv.org/abs/2602.13521. 1, 9, 21

Adithya Bhaskar, Tushar Tomar, Ashutosh Sathe, and Sunita Sarawagi. Benchmarking and improving text-to-SQL generation under ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7053–7074. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.436. 3, 8, 9

BIRD Team. LiveSQLBench: A dynamic and contamination-free benchmark for evaluating LLMs on real-world text-to-SQL tasks. Benchmark release, 2025. URL https://livesqlbench.ai/. Base-Lite-SQLite split. 1, 3, 7, 8, 9, 16

Mingwen Dong, Nischal Ashok Kumar, Yiqun Hu, Anuj Chauhan, Chung-Wei Hang, Shuaichen Chang, Lin Pan, Wuwei Lan, Henghui Zhu, Jiarong Jiang, Patrick Ng, and Zhiguo Wang. PRACTIQ: A practical conversational text-to-SQL dataset with ambiguous and unanswerable queries. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 255–273. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.13. 21

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 150–158. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.eacl-demo.16. 9, 21

Avrilia Floratou, Fotis Psallidas, Fuheng Zhao, Shaleen Deep, Gunther Hagleither, Wangda Tan, Joyce Cahoon, Rana Alotaibi, Jordan Henkel, Abhik Singla, Alex van Grootel, Brandon Chow, Kai Deng, Katherine Lin, Marcos Campos, Venkatesh Emani, Vivek Pandit, Victor Shnayder, Wenjing Wang, and Carlo Curino. NL2SQL is a solved problem... not! In *Proceedings of the 14th Conference on Innovative Data Systems Research (CIDR)*, 2024. URL https://www.cidrdb.org/cidr2024/papers/p74-floratou.pdf. 20

Nan Huo, Xiaohan Xu, Jinyang Li, Per Jacobsson, Shipei Lin, Bowen Qin, Binyuan Hui, Xiaolong Li, Ge Qu, Shuzheng Si, Linheng Han, Edward Alexander, Xintong Zhu, Rui Qin, Ruihan Yu, Yiyao Jin, Feige Zhou, Weihao Zhong, Yun Chen, Hongyu Liu, Chenhao Ma, Fatma Ozcan, Yannis Papakonstantinou, and Reynold Cheng. BIRD-INTERACT: Re-imagining text-to-SQL evaluation via lens of dynamic interactions. In *International Conference on Learning Representations*, volume 2026, pp. 44285–44333, 2026. 1, 8, 9

Chia-Hsuan Lee, Oleksandr Polozov, and Matthew Richardson. KaggleDBQA: Realistic evaluation of text-to-SQL parsers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2261–2273. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.176. 1, 7, 8, 9, 17

Morris Lee. Business truth, not SQL accuracy: A rule-gated 7B analytics agent outperforms a direct-prompted 32B baseline. *arXiv preprint arXiv:2608.09254*, 2026. doi: 10.48550/arXiv.2608.09254. URL https://arxiv.org/abs/2608.09254. 20

Youngwon Lee, Jaejin Kim, and Seung-won Hwang. Bootstrapping semantic layer from execution for text-to-SQL. *arXiv preprint arXiv:2606.05634*, 2026. doi: 10.48550/arXiv.2606.05634. URL https://arxiv.org/abs/2606.05634. 1, 9

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. Spider 2.0: Evaluating language models on real-world enterprise text-to-SQL workflows. In *International Conference on Learning Representations*, volume 2025, pp. 28691–28735, 2025. 1, 21

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin Chen-Chuan Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? A BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 42330–42357. Curran Associates, Inc., 2023. doi: 10.52202/075280-1835. Datasets and Benchmarks Track. 1, 3, 8, 9

Chengxi Liao, Tao Xu, Zulong Chen, Chuanfei Xu, Yiyan Wang, Xinyun Wang, Yanlong Zhang, Xiaojun Chen, Zhibo Yang, and Zeyi Wen. EntSQL: A benchmark for grounding text-to-SQL in long-context enterprise knowledge. *arXiv preprint arXiv:2606.03363*, 2026. doi: 10.48550/arXiv.2606.03363. URL https://arxiv.org/abs/2606.03363. 8

Qi Liu, Jiaxin Mao, Fengbin Zhu, and Tat-Seng Chua. Diagnosing search behavior and failure modes in long-horizon search agents. *arXiv preprint arXiv:2608.01913*, 2026. doi: 10.48550/arXiv.2608.01913. URL https://arxiv.org/abs/2608.01913. 9, 21

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.546. 21

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024. doi: 10.52202/079017-0692. Datasets and Benchmarks Track. 21

Michael Rumiantsau and Ivan Fokeev. Semantic layers for reliable LLM-powered data analytics: A paired benchmark of accuracy and hallucination across three frontier models. *arXiv preprint arXiv:2604.25149*, 2026. doi: 10.48550/arXiv.2604.25149. URL https://arxiv.org/abs/2604.25149. 9, 20

Irina Saparina and Mirella Lapata. AMBROSIA: A benchmark for parsing ambiguous questions into database queries. In *Advances in Neural Information Processing Systems*, volume 37, pp. 90600–90628. Curran Associates, Inc., 2024. doi: 10.52202/079017-2876. Datasets and Benchmarks Track. 3, 8, 9

Niklas Wretblad, Fredrik Gordh Riseby, Rahul Biswas, Amin Ahmadi, and Oskar Holmström. Understanding the effects of noise in text-to-SQL: An examination of the BIRD-Bench benchmark. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 356–369. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-short.34. 21

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1819–1862. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.102. 9, 21

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*, volume 2024, pp. 35623–35646, 2024. 21

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8541–8565. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.486. 21

Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, Junhong Liu, Dingnan Jin, Hongru Liang, and Tat-Seng Chua. CLAMBER: A benchmark of identifying and clarifying ambiguous information needs in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10746–10766. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.578. 21

Fuheng Zhao, Shaleen Deep, Fotis Psallidas, Avrilia Floratou, Divyakant Agrawal, and Amr El Abbadi. Sphinteract: Resolving ambiguities in NL2SQL through user interaction. *Proceedings of the VLDB Endowment*, 18(4):1145–1158, 2024. doi: 10.14778/3717755.3717772. 9, 21

A RD-BIRD: SQL-Fragment Definitions

Table 3 reports the SQL-fragment version of the instrument, which uses 9 concepts from BIRD databases. Definitions contain SQL fragments, such as `height >= 185` for a tall player. The experiments with plain-language definitions use KaggleDBQA databases and give definitions without column names or SQL operators. The two sets differ in both their concepts and their definition format, so comparing them does not isolate the effect of format. With SQL definitions, stating the definition produces more matching answers than it does with plain-language definitions. The Codex rows use the agent's own tools, and we did not run its stated position. Its counter of glossary mentions also counts table names in the schema, so the table instead reports the `SELECT` statements visible in the stored transcript tail, which gives a lower bound. For the shared harness, glossary queries are SQL statements mentioning `business_definitions`, including failed statements.

How we match answers to definitions. For RD-Kaggle, we first compare the rows returned by the agent's query with the rows returned by each candidate definition's reference query. If no result matches exactly, we allow a tolerance when comparing numbers. For a reference answer consisting of one number, we accept a difference of up to 0.1% of its absolute value, with a minimum tolerance of 0.011. We also accept a match in one cell of a row containing extra columns. For other results, we round floating-point numbers to two decimal places before comparison. The scorer checks candidate definitions in a fixed order and returns the first match.

Without the exploration instruction, stating a convention in the prompt improves the match rate by 53.7 to 76.9 percentage points over providing no definition. Storing the definition in the glossary changes the rate by at most 3.7 points for Qwen and Mistral. These models query the glossary in at most 2% of sessions, except for Qwen3-14B thinking at 12%. The exploration instruction raises Mistral's query rate from 2% to 63% and its match rate with a stored definition from 16.7% to 44.9%. For GPT-5 mini, querying rises from 31% to 99% and matching from 50.9% to 95.4%.

Among sessions in the stored position that queried the glossary, Mistral with the exploration instruction matches the assigned definition in 71 of 135, while GPT-5 mini matches it in 103 of 107.

When the prompt and glossary disagree, adding the exploration instruction lowers the share of answers matching the prompt definition: from 98.1% to 77.8% for GPT-5 mini and from 92.6% to 77.8% for Mistral. Of Codex's 108 answers without a supplied definition, 63 match the definition in BIRD's original evidence.

B Coding Agents Under Their Own Loops

The shared harness fixes the tool loop so that settings differ only in the model. Coding agents do exactly the opposite: each brings its own loop, its own tools, and its own stopping policy. We therefore report them here, in Table 4, and never pool them with the shared-harness settings. Each agent runs the two conditions the comparison needs, no definition and in the database, on the same items.

There are two loops. Codex runs `codex-cli` 0.154.0 with its default model at read-only autonomy, while GLM 5.3 and Claude Opus 5 run under the Claude Code command line, which denies every command at that autonomy level, so they run one level higher. Codex and the other two therefore differ in loop and in permission level at once. This makes them separate observations rather than a controlled comparison of providers. The comparison of GLM 5.3 with Claude Opus 5 is controlled:

Table 3: RD-BIRD, the SQL-fragment version of the instrument: percentage of sessions whose answer matches the definition assigned to that item, on 9 concepts drawn from BIRD databases. Cells hold 108 items, except Mistral with the instruction (216 sessions). Default-matching reference 35.2%. "Queried glossary" is the share of stored sessions that executed a statement naming the glossary table. "+ instruction" adds the sentence asking the agent to check the database for business terms. Unlike Table 1 this table scores by exact set equality against the assigned definition's reference query, without the tolerance described in Section 3. That tolerance would raise the "In the prompt" column by up to 12.1 points and the "Conflicting sources" column by up to 13.0, while leaving the sign and ordering of every contrast unchanged. Dashes mark entries not run. The Codex glossary rate is a lower bound read from the end of each transcript.

Model	Absent	Stated	Stored	Conflicting	Queried glossary
Qwen3-4B	12.0	65.7	15.7	73.1	0%
Qwen3-8B	10.2	85.2	12.0	81.5	2%
Qwen3-14B	16.7	84.3	16.7	85.2	0%
Qwen3-4B thinking	19.4	88.0	20.4	89.8	0%
Qwen3-8B thinking	20.4	96.3	18.5	96.3	0%
Qwen3-14B thinking	25.0	95.4	24.1	94.4	12%
Mistral-Small-24B	14.8	91.7	16.7	92.6	2%
GPT-5 mini	22.2	98.1	50.9	98.1	31%
Qwen3-14B + instruction	21.3	83.3	18.5	83.3	1%
Qwen3-14B thinking + instruction	25.9	98.1	26.9	98.1	13%
Mistral + instruction	16.2	88.9	44.9	77.8	63%
GPT-5 mini + instruction	29.6	98.1	95.4	77.8	99%
Codex, `codex-cli` 0.154.0	30.6	–	90.7	–	$\geq$87%

it holds the loop, the autonomy level, and the items fixed and varies only the model. Between the two, the gain moves from $+35.6$ to $+54.1$ points. All three run hermetically, excluding the operator's own instruction files and skills, against a disposable copy of the database.

Table 4: Coding agents on RD-Kaggle and RD-BIRD: answers matching the assigned definition (%). Each cell is 135 items on RD-Kaggle and 108 on RD-BIRD.

Benchmark	Agent	Absent	Stored	Gain
RD-Kaggle	Codex, `codex-cli` 0.154.0	31.1	77.0	$+45.9$
RD-Kaggle	GLM 5.3, Claude Code 2.1.270	31.1	66.7	$+35.6$
RD-Kaggle	Claude Opus 5, Claude Code 2.1.270	34.8	88.9	$+54.1$
RD-BIRD	Codex, `codex-cli` 0.154.0	30.6	90.7	$+60.2$
RD-BIRD	Claude Opus 5, Claude Code 2.1.270	28.7	100.0	$+71.3$

Two readings follow. First, every agent that explores gains substantially from a definition placed in the database, from $+35.6$ to $+71.3$ points, while no open-weight setting on the shared harness gains anything on RD-Kaggle or more than 3.7 points on RD-BIRD. Even the narrowest of these, the GLM gain, is $+35.6$ with a 95% confidence interval of $[+24.4, +46.7]$.

Second, the baseline is a rate rather than a set. Codex and GLM 5.3 each answer exactly 42 of the 135 RD-Kaggle items without a definition, but they agree on only 32 of them, and 52 are solved by at least one. Without a definition, the agents converge on a common success rate but disagree about which questions they get right. A prior-driven guess predicts this pattern, whereas a retrieved fact would not.

The gain comes from retrieval. In a run of the RD-Kaggle stored cell that records full transcripts, Codex issues a glossary `SELECT` in 102 of 135 sessions (75.6%). It answers 101 of those 102 correctly and 9 of the 33 others (27.3%, near its no-definition cell), so for this agent the stored

definition is worth what it is because the agent retrieves it. The Claude Code loop does not echo its statements in a form we can parse, so we cannot give its query rate.

C THE MIXTURE PREDICTION

This appendix gives the cells behind the prediction reported in Section 5. With probability r, the share of stored sessions that query the glossary, the agent should behave as it does when handed the rows. Otherwise it answers from its default, which the absent position measures. The two accuracies on the right of $r \cdot \text{served} + (1 - r) \cdot \text{absent}$ are measured in other conditions, while r comes from the stored sessions being predicted, and agents may query on items that differ from the rest.

Table 5 gives the cells.

Table 5: Predicting the stored cells of RD-Kaggle from the absent and served cells and the stored sessions' query rate r. r is the share of stored sessions containing a glossary query. Absent is the no-definition cell of Table 1 and Served the served cell of Table 7. Predicted is $r \cdot \text{served} + (1 - r) \cdot$ absent, and Observed is the stored cell. For Qwen3-4B, Qwen3-8B, Qwen3-14B and Mistral, r and Observed come from the stored arm run alongside served (Table 7). For the other rows they come from Table 1. Rows marked "+ instruction" add the sentence asking the agent to check the database for business terms. Mean absolute error is 2.0 points over the twelve cells, and 2.9 over the seven with $r > 5\%$.

Setting	r	Absent	Served	Predicted	Observed	Error
Qwen3-4B	0%	4.4	24.4	4.4	3.7	-0.7
Qwen3-8B	1%	8.9	31.1	9.1	9.6	$+0.5$
Qwen3-14B	2%	9.6	31.1	10.0	9.6	-0.4
Qwen3-32B	16%	10.4	40.0	15.0	9.6	-5.4
Mistral-Small-24B	0%	13.3	27.4	13.3	12.6	-0.7
GPT-5 mini	8%	28.1	88.9	33.0	35.6	$+2.6$
Qwen3-8B + instruction	4%	8.1	31.1	9.0	7.4	-1.6
Qwen3-14B + instruction	7%	10.4	31.1	11.8	8.9	-2.9
Mistral + instruction	76%	15.6	27.4	24.6	18.5	-6.1
GPT-5 mini + instruction	96%	24.4	88.9	86.3	85.2	-1.1
GPT-5 + instruction	97%	31.9	96.3	94.4	93.3	-1.1
GLM 5.3	41%	36.3	97.0	61.0	60.0	-1.0

The two cells that miss by more than five points, Qwen3-32B and Mistral under the instruction, miss in the same direction: the prediction assumes that a session querying the glossary behaves like a served session, but a query the agent writes itself returns less than the one we write for it. In these two settings the served position therefore overstates what the agent's own retrieval delivers. Agents may also query on items that differ from the rest, which the mixture does not model.

D A REPLICATION SET OF TEN NEW CONCEPTS

Ten concepts can be chosen, knowingly or not, so that lookup fails. We therefore built a second set of ten concepts on all eight KaggleDBQA databases, none sharing a term with RD-Kaggle: new reactor, old-school release, popular release, property crime, long-burning fire, veteran player, high-scoring match, upset win, heavily federally funded district, and imported sample. Unlike RD-Kaggle, these concepts were drafted by a coding agent under author direction, with the same builder, the same rules for definitions, and the same check that every definition gives a different answer on the data. The set has 50 questions crossed with their definitions, 140 items per position. Each candidate concept was screened with a trial run, and three whose reference answers depended on how the data was formatted rather than on the definition were replaced (capacities stored as text, empty date strings that compare as early, and a revenue column in undocumented thousands of dollars). One question asks for a national count that equals its definition's threshold, which gives the answer away once the definition is stated. Dropping it changes no contrast by more than 1.6 points, so Table 6 keeps it.

Table 6: The replication set: answers matching the assigned definition (%, 140 items per cell), scored as in Table 1. "Queried" is the share of in-the-database sessions with a glossary query.

Setting	Absent	Stated	Stored	Conflicting	Delivered	Served	Queried
Qwen3-4B	5.0	51.4	6.4	47.1	27.9	27.1	2%
Qwen3-8B	3.6	45.0	2.1	47.9	28.6	38.6	0%
Qwen3-32B	6.4	60.0	6.4	59.3	49.3	61.4	0%
Mistral-Small-24B	6.4	65.7	8.6	65.7	59.3	55.0	0%
GPT-5 mini	16.4	90.0	34.3	87.9	82.1	88.6	26%
GPT-5	11.4	87.1	52.1	85.0	90.0	86.4	52%

The stored and served contrasts reproduce. Resampling the ten concepts, the concept-weighted stored minus absent is $+2.7$ $[-2.7, +11.3]$ for Qwen3-4B, -1.3 $[-5.0, +2.0]$ for Qwen3-8B, 0.0 for Qwen3-32B, and $+3.0$ $[-1.3, +9.7]$ for Mistral, while served minus stored is $+19.3$ $[+2.7, +38.0]$, $+35.7$ $[+20.0, +53.0]$, $+53.3$ $[+34.7, +71.3]$, and $+43.3$ $[+24.7, +61.3]$. The hosted models query more often on this set, and their gains from storage now exclude zero: $+17.3$ $[+8.0, +27.0]$ for GPT-5 mini at a 26% query rate and $+38.7$ $[+17.3, +60.7]$ for GPT-5 at 52%, and serving adds $+34.0$ $[+11.7, +58.7]$ for GPT-5. Concept-weighted, stated minus absent ranges from $+40.3$ to $+72.7$. Absent cells differ between the sets (GPT-5 reaches 31.9% on the main set and 11.4% here), as expected when absent accuracy measures how often a model's defaults match the assigned definitions, and Qwen3-32B, which queries in 16% of stored sessions on the main set, queries in none here. Pooling both sets, a logistic model with concept-clustered standard errors over the four open-weight settings gives $+0.1$ points for stored against absent ($p = 0.92$) and $+31.1$ for served against stored ($p = 2.4 \times 10^{-7}$). These weight sessions, while the bootstrap estimates above weight concepts, which is why they differ slightly. The form effect of Section 4 does not replicate uniformly: Qwen3-32B's served cell (61.4) exceeds its stated cell (60.0), and Mistral's rows shown with the schema (59.3) beat its served rows (55.0), as they do on the main set (39.3 against 27.4).

E Further Controls

This appendix gives the controls summarized in Section 7.

What serving adds. Serving changes three things at once: the agent receives the definition, sees one successful use of its tool, and reads the definition as table rows. Two controls on the 135 stored items separate them. In the first, the seeded lookup returns glossary rows that do not include the concept, so the agent sees the tool used but receives no definition. In the second, the seeded lookup returns the assigned definition as the same sentence the stated position uses.

Setting	Stored	Seeded, no definition	Served as rows	Served as sentence	Stated
Qwen3-4B	3.7	5.2	24.4	36.3	40.0
Qwen3-8B	9.6	8.9	31.1	37.3	43.7
Qwen3-32B	9.6	12.6	40.0	48.9	45.2
Mistral-Small-24B	12.6	11.1	27.4	48.1	54.8

Resampling concepts and weighting them equally, the uninformative lookup minus stored is $+1.7$ $[-1.0, +4.7]$, -0.7 $[-4.0, +3.0]$, $+3.3$ $[-1.7, +8.0]$, and -0.7 $[-8.0, +6.3]$. The sentence minus the rows is $+12.0$ $[+2.7, +21.0]$, $+6.4$ $[-4.3, +20.7]$, $+10.0$ $[+1.3, +19.0]$, and $+17.7$ $[+5.7, +30.0]$. Stated minus the served sentence is $+3.0$ $[-1.3, +7.7]$, $+4.9$ $[-11.4, +16.3]$, -4.0 $[-10.7, +2.0]$, and $+8.7$ $[+0.3, +17.3]$. Six Mistral sessions and one Qwen3-8B session in the sentence arm did not finish and are excluded.

The hint in the tool description. A deployed system might describe hidden tables in the tool definition rather than in the prompt. We added one sentence to the description of `run_sql`, *some tables, such as a glossary, define business terms used in questions; their rows are hidden from the schema and must be queried*, and ran the 135 stored items with it. Qwen3-4B and Qwen3-8B still never query, and reach 3.7% and 7.4% against 4.4% and 8.1% without the sentence. Qwen3-32B

queries in 32% of sessions against 16% and reaches 11.1% against 9.6%, and Mistral queries in 23% against 1% and reaches 14.8% against 12.6%. Every concept-bootstrap interval for the difference includes zero, the widest being $[-2.7, +6.7]$.

LiveSQLBench-Lite. LiveSQLBench-Lite tasks depend on business metrics defined only in a supplied knowledge base (BIRD Team, 2025). Without those entries every setting scores between 0.6 and 1.7%. Supplying the entries a task needs raises accuracy to 6.8% at Qwen3-4B and 25.6% at Mistral. The rise with capability is a floor effect, so we claim only the weaker statement supported by these results: the information is unavailable, and supplying it helps. Supplying the whole knowledge base instead of the entries a task needs costs up to 8.9 points.

SQLite build. Execution runs under a step budget, so a slow query is recorded as an error on a slower SQLite build: three GPT-5 sessions differ between 3.49.1 and 3.53.4, moving one cell of Table 1 by 0.7 points.

The thresholds are not inferable from the data. If the data were cleanly partitioned around a candidate threshold, an agent could infer it without being told. They are not. The wildfire data hold 1,880,465 fires whose deciles run 0.1, 0.1, 0.2, 0.5, 1.0, 1.0, 2.2, 5.0 and 17.0 acres, with the largest gap anywhere in the middle 90% of the distribution measuring 0.1 acres. Salaries in the high-earner concept and state scores in the underperforming-state concept are similarly continuous where their candidate cut-offs sit. The categorical concepts are unrecoverable by construction: the data lists the crime categories but does not say which subset counts as violent.

Answer format. The harness rejects a submission that is not a query and asks for a resubmission.

Specification quality. On BIRD the expert-corrected reference query is sometimes unreachable from the original question and hint. An audit of 100 questions finds 46 whose reference query the question and hint do not support. This lowers the attainable accuracy of every condition alike. We report contrasts with and without the audited questions in Appendix J.

F Positions of the Same Definition

Table 7 gives the per-setting numbers behind Section 4.

G Frontier Models on BIRD Mini-Dev

We ran ten hosted settings on BIRD Mini-Dev under three conditions: *explore* (the question alone, with database access), *ask* (explore plus a tool to ask the user one question), and *hint* (the question with BIRD's expert evidence). In the ask condition a simulated user (GPT-4.1) who knows the intended question and the evidence answers only the point the agent raises. Every cell uses the same 497 questions, the harness's fixed set (Mini-Dev's 500 without three it excludes), on the shared harness of Section 3. BIRD's evidence is not decomposed into database facts and owner definitions (Appendix J explains why), so this is a comparison on the benchmark's own terms, not a further position of the relocation instrument. Table 9 gives the cells.

The expert evidence helps every setting, by 4.3 points for Claude Opus 5.5, 10.4 for Claude Opus 5, and 12.9 to 30.3 for the others. Eight of the ten settings ask in at most 3% of sessions, while GPT-5 mini asks in 18% and GPT-5 in 9%. Offering the tool changes accuracy even where the agent does not use it: restricted to the questions on which the agent did not ask, the ask condition differs from explore by $+10.8$ points for GPT-5 mini and $+7.0$ for GPT-5 (exact paired test, $p < 0.001$), by $+4.1$ to $+4.7$ for the Claude settings, and by -2.4 and -3.7 for GPT-5.5 and GPT-5.6 Sol. The ask cells therefore do not measure the value of asking, and we report only the ask rates as a finding. As in Section 5, a channel the agent must choose to use is rarely used.

H The Two Sides of the Boundary in Detail

Figure 1 plots the two series of Section 6 for the Qwen family.

Table 7: Where the same definition arrives, and what each position is worth. Sessions matching the assigned definition on the 135 stored items of RD-Kaggle, default-matching reference 37.0%, scored as in Table 1. *Stored* means the definition sits in a glossary table the agent may query, *served* that we run the lookup and place the rows in its transcript, and *stated* the same sentence in the prompt. For Qwen3-4B, Qwen3-8B, Qwen3-14B, and Mistral, stored comes from the same run as served, so two cells differ slightly from Table 1 (3.7 against 4.4, and 9.6 against 8.1). No open-weight setting below 32B queries the glossary unprompted in more than 2.2% of stored sessions. Qwen3-32B queries in 16%, and the hosted GPT-5 and GLM 5.3 in 24% and 41%. Every served session contains a query by construction. The two share columns scale each position to the setting's stated accuracy: stored share is (stored − absent)/(stated − absent), and served share is (served − stored)/(stated − stored). Concept-bootstrap intervals for the served share exclude zero for every setting, and for the stored share include zero for every setting except GLM 5.3, whose stored gain excludes zero ($+22.3$ $[+7.7, +38.0]$, concept-weighted).

Setting	Absent	Stored	Served	Stated	Stored share	Served share
Qwen3-4B	4.4	3.7	24.4	40.0	-2%	57%
Qwen3-8B	8.9	9.6	31.1	43.7	$+2\%$	63%
Qwen3-14B	9.6	9.6	31.1	44.4	0%	62%
Qwen3-32B	10.4	9.6	40.0	45.2	-2%	85%
Mistral-Small-24B	13.3	12.6	27.4	54.8	-2%	35%
GPT-5 mini	28.1	35.6	88.9	90.4	$+12\%$	97%
GPT-5	31.9	45.9	96.3	92.6	$+23\%$	108%
GLM 5.3	36.3	60.0	97.0	98.5	$+38\%$	96%

Table 8: The asking channel on the 135 absent items of RD-Kaggle. The owner always replies with the assigned definition. "Correct having asked" counts sessions that asked and then answered with the assigned definition. On the sessions where the agent did not ask, accuracy matches the no-definition cell for every setting (differences -5.0 to $+0.8$ points, none significant under an exact paired test), so offering the channel does not by itself change accuracy.

Setting	Absent	Asked	Answer matched	Correct having asked
Qwen3-4B	4.4	10%	4.4	0 of 13
Qwen3-8B	8.9	19%	10.4	3 of 26
Qwen3-32B	10.4	23%	16.3	10 of 31
Mistral-Small-24B	13.3	6%	14.8	1 of 8
GPT-5 mini	28.1	35%	49.6	40 of 47
GPT-5	31.9	25%	45.2	30 of 34

Documentation of what the data settles. KaggleDBQA ships column descriptions and code maps written by its authors (Lee et al., 2021). Supplying them is worth $+10.9$ points at Qwen3-4B (32.6% without hints), $+9.7$ at Qwen3-8B, $+7.1$ at Mistral, $+6.4$ at Qwen3-32B, $+5.2$ at Qwen3-14B with thinking, $+4.9$ at GPT-5 mini, $+2.6$ at Qwen3-14B (56.9% without hints), and $+0.7$ at GPT-5, which is indistinguishable from zero ($[-2.5, +4.1]$, resampling the 8 databases) and becomes -1.1 under strict scoring. The intervals at the two ends do not overlap, while Qwen3-14B's $+2.6$ also includes zero.

The paired difference between the two ends, computed from unrounded gains, is $+10.1$ points $[+2.6, +16.2]$, stays between $+7.5$ and $+11.6$ when any one database is dropped, and survives dividing each gain by the headroom above the no-hint accuracy. The ordering in between is not reliable. The rank correlation between no-hint accuracy and gain is -0.71 over the eight settings ($p = 0.058$, exact permutation), Qwen3-32B answers fewer questions without hints than Qwen3-14B yet gains more, and Qwen3-14B with thinking gains less than two weaker settings. We therefore claim only that documentation is worth less to the strongest model than to the weakest.

Table 9: BIRD Mini-Dev accuracy (%) on the same 497 questions for every cell. The 95% intervals are about ± 4 points. Ask rate is the share of ask sessions in which the agent asked.

Model	Explore	Ask	Hint	Ask rate (%)
GPT-4.1 mini	37.8	38.4	55.9	1
GPT-4.1	38.0	38.0	50.9	3
GPT-5 mini	34.4	49.1	57.1	18
GPT-5	42.9	52.3	73.2	9
GPT-5.5	45.7	43.3	59.0	0
GPT-5.6 Sol	40.2	37.0	55.7	2
GPT-6 Luna	40.8	40.4	54.9	0
Claude Sonnet 5	47.5	52.9	63.8	2
Claude Opus 5	62.8	67.0	73.2	1
Claude Opus 5.5	73.6	78.3	77.9	1

Against a definition the data does not settle. Inside one model family, on the same harness, the two kinds of information behave differently as the model improves (Figure 1). Stating a definition is worth $+35.6$ points at Qwen3-4B and $+34.8$ at 8B, 32B, and 14B, while documentation is worth $+10.9$, $+9.7$, $+6.4$, and $+2.6$. Within the family, the documentation gain shrinks from $+10.9$ to $+2.6$ while the stated gain stays flat. The 4B-to-14B difference in documentation gain is not significant on its own ($+8.2$, $[-0.9, +15.9]$). No single concept carries the stated gains: dropping any one leaves them between $+24.1$ and $+38.1$. The two series come from different items and scoring rules. A stated gain also sits between a floor set by the prior and a ceiling set by SQL-writing ability, so we read the comparison as descriptive rather than as a scaling law. Neither series depends on labeling any text, since we assign the meaning and the documentation is the benchmark's own.

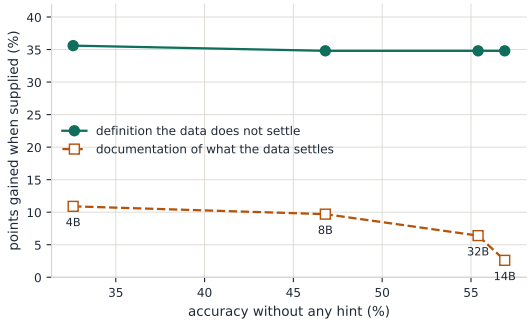


Figure 1: The two sides of the boundary, inside one model family. Each point is a Qwen3 setting, placed by how well it does with no hint at all. Filled markers: what a definition the data does not settle is worth, from the plain-language experiment (stated minus absent). Open markers: what documentation of something the data does settle is worth, on KaggleDBQA (documentation minus exploration, relaxed scoring). The documentation gain is smaller at Qwen3-14B than at Qwen3-4B, while the definition's gain is not, but the 4B-to-14B difference in documentation gain is not significant on its own, and the two series come from different items and scoring rules. Neither series depends on labeling any text: the definitions are assigned by us, and the documentation is the benchmark's own.

I Does the Answer Track the Assigned Definition?

We re-executed every RD-Kaggle session and compared its result with every candidate definition rather than stopping at the first match. In the plain condition, 2 of 5,535 sessions match two or more definitions, so the fixed matching order can misattribute at most one session. Matches that need

the numeric tolerance are 0 to 4.4% of a cell. Counting only unique exact matches to the assigned definition leaves every contrast in Table 1 within 2.2 points of the reported value. The largest change is Qwen3-14B's stated gain, $+34.8$ against $+32.6$, so the Qwen family's stated gains span $+32.6$ to $+35.6$ under strict scoring.

Each question appears once per definition in every position, which lets us ask whether an agent's answer follows the assignment. With the definition stated, every copy of a question matches its own assignment for 20 to 36% of questions on the open-weight settings, 78% on GPT-5 mini, 82% on GPT-5. Stored, the same share is 0% for every open-weight setting, 4% for GPT-5 mini, and 18% for GPT-5, but 38 to 82% for the coding agents. Without a definition, the share of questions answered identically whatever the assignment is 10 to 22% for the open-weight settings, 50% for GPT-5 mini, and 64% for GPT-5. Stronger agents hold a fixed default when no definition is available and switch to the assigned one when it is stated or retrieved. On seven of the ten concepts, every GPT-5 mini answer that matches an admissible definition matches the same one. On three of them, that default coincides with the definition used by KaggleDBQA's own reference queries, which may reflect exposure to that public benchmark rather than a prior formed on the question.

J Supplying Facts by Kind on BIRD

We ran the same contrast on hints written by BIRD's authors, by splitting each hint into atomic facts, labeling each fact by where its information could have come from, and building prompts that contain only one kind. We report it here rather than in the body because the labeling is not reliable enough to carry a claim.

The labeling does not replicate. Five model annotators from five model families applied one codebook to the atomic facts, covering 1,519, 1,519, 1,519, 1,518 and 1,399 of them, and one human annotator labeled a 248-fact sample. On the distinction between database-derivable facts and owner choices, pairwise Cohen's kappa ranges from 0.016 to 0.399. The best pair reaches 0.399. The human annotator against the annotator whose labels this experiment uses reaches 0.290. Two further instruments do no better: asking who decided the content gives 0.203 between two annotators, and asking whether a query can recover it gives 0.256. When two annotators both label a fact as either database-derivable or an owner choice, they agree on which of the two it is 75 to 97% of the time, depending on the pair. The distinction is therefore consistent where annotators apply it, but applying it to a single authored sentence is not reproducible. This is the reason the experiments in the body assign meaning by construction instead of inferring it.

What the experiment shows, read as provisional. No claim in the body rests on this subsection. We report it because the reliability failure above is itself a result, and because the one subset that independent annotators agree on shows how little of the estimate survives. BIRD hints contain both facts recoverable from the database and choices the owner must supply. We compare how much each type helps models at different baseline accuracies. We split 1,355 hint clauses into 1,519 individual facts and label where each fact comes from: the question, the database, external documentation, or an owner choice, with an additional category for unclear cases. These results remain preliminary: the conditions were built from one model annotator's labels, and the human check reported above does not clear the reliability gate.

We give the agent database facts alone, owner choices alone, or both sets together, and compare each result with a run without hints. The combined hint contains exactly the facts in the two single-type hints. To check whether any extra hint helps, we also supply an unrelated hint of similar length from another question on the same database. Database facts occur in 338 questions, owner choices in 131, and both in 73.

On the 73 questions with both kinds of fact, adding owner choices to database facts improves accuracy by 17.1 points $[9.9, 24.4]$. This estimate combines eight model settings and two runs in a linear mixed model with random intercepts for questions and databases. Owner choices also outperform unrelated hints by 21.0 points $[14.3, 27.7]$. We report every subset we examined, including the one that lowers the estimate, because a series of robustness checks that all move an estimate upward is itself a reason for suspicion.

Questions kept	Beyond database facts	Against unrelated facts
All eligible (73 carry both kinds)	$+17.1$ $[9.9, 24.4]$	$+21.0$ $[14.3, 27.7]$
Same owner labels in both passes (108)	$+18.9$ $[8.7, 29.2]$	$+22.0$ $[14.7, 29.3]$
Audited-unreachable dropped (46 removed)	$+23.4$ $[16.8, 30.0]$	$+26.9$ $[21.0, 33.0]$[b]
Annotators assign the same buckets	$\mathbf{+12.8}$ $[\mathbf{-8.3}, \mathbf{34.0}]$	–

[b]A question bootstrap with 2,000 resamples. The mixed model does not converge on that subset, so no Wald interval exists for it. The first three rows move the estimate up. The fourth is the one that matters: restricting to questions where independent annotators would build the same conditions leaves 3 questions carrying both kinds of fact out of 73, and the contrast is no longer distinguishable from zero. This is the sensitivity analysis a reader should weigh, and the reason this experiment sits in an appendix rather than in the body.

J.1 Labeling Protocol

This subsection describes how the facts above were labeled, and what the reliability measurement at the head of this appendix was measuring.

For the BIRD experiment in Appendix J, we split 1,355 hint clauses into 1,519 individual facts. Each fact specifies one mapping, encoding, formula, threshold, category, time window, or way to join tables. Annotators inspect each fact alongside the question and database, then apply the first matching category:

- **Stated in the question.** The question states or directly implies the fact, and the annotator cites the relevant phrase.
- **Recoverable from the database.** A query can establish the fact, such as which column holds a concept, how tables join, how values are encoded, or what a lookup-table entry means, and the annotator supplies a supporting query. The existence of the required columns alone is insufficient.
- **External documentation.** The database stores a code or unit without explaining it, and the fact supplies its meaning.
- **Owner convention.** The question and data allow at least two interpretations that give different answers. The annotator supplies both interpretations.
- **Unclear.** The fact is garbled or the available evidence does not support a decision.

The human check used a 248-fact sample, labeled by one annotator and stratified by database and by the category assigned by the model annotators. We measure agreement with Cohen's kappa. The protocol required a kappa of at least 0.6 on the database-recoverable against owner-convention distinction. The observed 0.290 falls below that threshold, which is why the BIRD experiment is reported as provisional rather than built from adjudicated labels.

Codex labeled all 1,519 facts twice with database access, agreeing on 91% with a kappa score of 0.87. Of 204 facts labeled as owner conventions in either pass, 157 received that label in both. We excluded another model annotator after manual inspection found that it treated column existence as sufficient evidence that a fact was recoverable from the database, including for question restatements and formula choices. We use the first Codex pass, noting that consistent labels from one model need not be correct.

K Extended Related Work

This appendix extends Section 9 with the lines of work the main text only names.

Semantic layers. Practitioners argue that enterprise text-to-SQL needs owner-defined business terms (Floratou et al., 2024), and WarehouseReliabilityBench scores analytics agents on questions with two valid business definitions, treating a semantic layer as part of the system rather than as a variable (Lee, 2026). Supplying business definitions as a semantic-layer document improves accuracy on a paired benchmark of 99 questions across three frontier models (Rumiantsau & Fokeev,

2026). With the document supplied, all three land between 67.7 and 68.7%. The document therefore explains essentially all of the significant variance, while model choice within that tier explains none, although per-model gains differ from $+17.2$ to $+23.2$ points. Two things separate that setting from ours. First, they vary the presence of the semantics, always in the prompt, across three models within five points of each other, whereas we vary its position inside one family whose accuracy without hints spans 32.6% to 56.9%. Second, what falls in our measurements is documentation of what the data already settles, which they do not vary.

Retrieval versus use. The split between retrieving evidence and using it appears in RAG and search-agent diagnostics (Es et al., 2024; Ru et al., 2024; Liu et al., 2026). Tk-Boost ablates retrieval and application of fixed corrective statements for text-to-SQL (Agarwal et al., 2026). The RAG diagnostics read the split from transcripts or from a retriever. Tk-Boost intervenes on retrieval and application of statements it wrote from ground-truth SQL. We intervene on the position of a definition we assign, including a position the agent must decide to query.

Counterfactual tasks and clarification. Assigning the meaning follows the counterfactual-task logic of Wu et al. (2024), which separates a model's default from the rule it applies, here applied to where the meaning lives rather than what it is. Clarification systems for text-to-SQL ask the user to resolve ambiguity (Zhao et al., 2024; Dong et al., 2025), and language models often fail to ask when a request is ambiguous (Zhang et al., 2024). Our asking channel instead fixes the reply to the assigned definition, so it measures whether the agent asks and uses the reply rather than the quality of its question.

Retrieval, conflict, and benchmark noise. Retrieval helps least where a model already knows the answer (Mallen et al., 2023). The database-settled half of our boundary reproduces that result inside a database. The owner half has no counterpart, since no parametric knowledge can supply a threshold an owner chose. The conflicting position is an inter-context conflict in the taxonomy of Xu et al. (2024), adjacent to the context-memory conflicts of Xie et al. (2024). In our setting, the conflict exists only once the agent retrieves. Errors in benchmark references affect measured accuracy (Wretblad et al., 2024), which our audit reports as a limitation. Enterprise evaluations take exploring the warehouse as the premise (Lei et al., 2025), a premise we test by moving one definition in and out of the database.