

Does a Governed Confidence Score Know When a Text-to-SQL Answer Is Right?

An exploratory study on BIRD and a pre-registered confirmation on Spider

Raghu Venkat Tricha Anjali
 Numberz.ai Inc., Alpharetta, Georgia, USA
 {raghu.venkat, tricha.anjali}@numberz.ai

October 2026

Abstract

Conversational analytics lets people ask questions of records spread across many systems and get answers in plain language. It is useful only if each answer can be trusted and explained. A production system of this kind attaches a confidence score, the Interpretive Evidence Score (IES), to every answer it writes over a database, and uses it to show the answer (PASS), show it with a warning (FLAG) or withhold it (ABSTAIN). The score’s weights, caps and thresholds were set by design. This paper is an audit of that design. On the 1,534 questions of BIRD dev, answered by GPT-5.1 with a benchmark-adapted production prompt and five samples per question, the designed score separates right from wrong answers weakly (AUROC 0.624, 95% CI 0.595–0.652) and withholds 35.7% of correct answers. Taking the score apart showed that nearly all of its ranking signal comes from agreement among the samples, and that most of it is lost where agreement is combined by a minimum with a checklist that drops to zero on any unsupported claim, including claims in commentary. We froze a rule in which agreement with the shown answer is the score and deterministic checks act as vetoes and caps, and pre-registered a test on Spider dev before any Spider answer existed. The primary hypothesis held: AUROC rose from 0.566 to 0.725 (paired difference +0.159, 95% CI +0.097 to +0.221), and the share of correct answers withheld fell from 49.0% to 8.8% with PASS accuracy 95.7%. Two results narrow the claim. Under one result-matching policy, the designed entropy and the share agreeing with the shown answer rank answers almost equally well, so the gain comes from how agreement enters the score and not from the statistic. And whether adding checks to agreement helps depends on how correctness is defined: on the same Spider answers, agreement alone ranked higher under model-judged labels and the rule with checks ranked higher under execution match, while the checks always cost unqualified coverage. A pre-registered empty-result check did not meet its effect-size criterion on Spider. Finally, agreement cannot resolve a business definition that is absent from the evidence, which motivates an evaluation of checks built from expert-approved definitions. Correctness here means the core answer to the question asked. Labels in this version are model-judged; a blinded two-person audit is not yet complete, and all correctness-based results, including the AUROC comparisons, remain subject to it. It will be reported, with the results recomputed on the adjudicated labels, in a revised version.

1 Introduction

1.1 The problem: one question, many places

A safety officer at a large operating organisation is asked how much of one chemical was used last year, and at which sites. The question is short. The answer is spread across systems built at

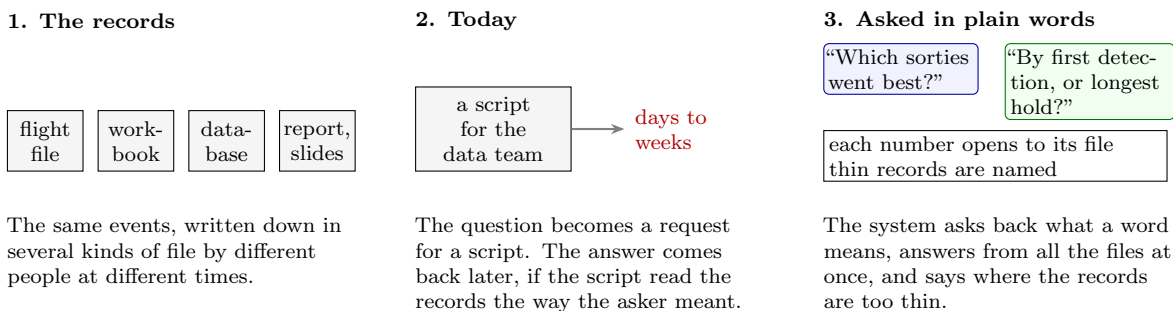


Figure 1: What changes. The records and the people stay the same; the question stops needing a script. Panel 3 is the requirement this paper is concerned with: every number traceable, and the system willing to say when it cannot answer.

different times for different purposes. Issue transactions record what was handed out. An inventory system records what is on the shelf. Safety data sheets, often PDFs, record what each product contains. Annual storage reports, built for a regulator, record what was held above a threshold. A site table turns codes into names. Each system keys the chemical differently, by a registry number, a trade name or a stock number; each has its own units and its own idea of a year. Answering by hand means finding the right tables, reconciling their keys and units, and checking the result with whoever knows which table means what. That can take weeks, and the answer can still be wrong.

The same shape recurs in other organisations that have accumulated systems over time: a flying unit that records the same sorties in flight files, workbooks, a database and briefing slides, or maintenance knowledge split between PDF manuals, database work orders and free-text notes. The question fits in a sentence; answering it means reconciling records never designed to be read together. That work is often handed to a data team as a request for a script (Figure 1).

Conversational analytics can read more than databases. A document’s tables can be read as rows, a safety data sheet’s composition section as one row per constituent, a log as timestamped events, a scanned form through character recognition. What makes the answers combinable, and checkable, is that every value ends up in the same form whatever file it came from: the value itself, its uncertainty if any, the time it describes, whether it was measured, derived by a named rule or asserted by a person or a document, where exactly it was found, and what it was computed from (Figure 2). Once values are in that form, the question of whether an answer is supported becomes a question about rows, tables and joins, which is the setting this paper measures.

A smaller example shows how a single table can mislead. Asked how many sites in one state hold a chemical, a reasonable query counts the rows whose `location` column names the state. It returns 2. The true answer is 213: the organisation stopped filling the location column years ago, and the state now lives in the prefix of the site identifier. Nothing errored, the column exists, and the count is arithmetically correct. The schema did not mean what its names implied, and the knowledge needed to see that sat with the people who maintain the data. (The example is illustrative, in the pattern of cases met in practice.)

1.2 Conversational analytics helps, if it can be trusted

Language models change the cost of that work. Given the schema, a model can find candidate tables, write and run queries, read a document’s composition section as rows, and explain its result in plain language, often within one session. Systems that do this, often called conversational analytics, are useful precisely where records are scattered and the people who need answers are not

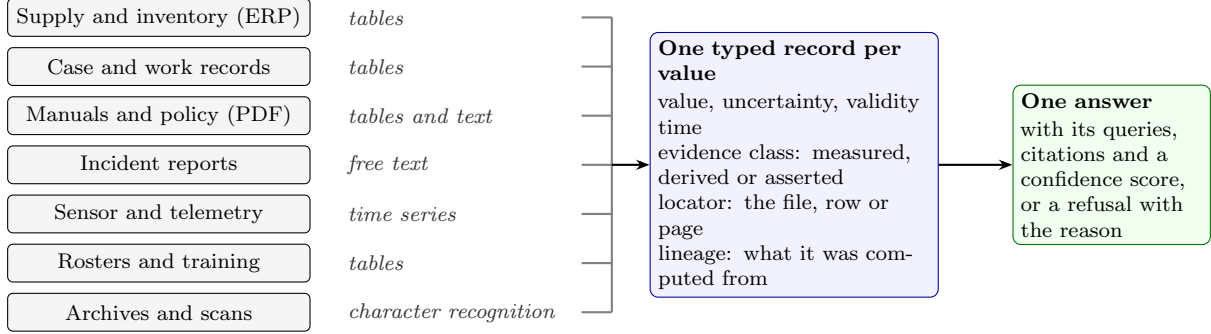


Figure 2: Many kinds of record, one form. Each source is read by the reader suited to it, and every value is held with its time, its evidence class, where it was found and what it was computed from. Disagreements between sources are settled by a precedence the organisation declares, with the losing value kept visible.

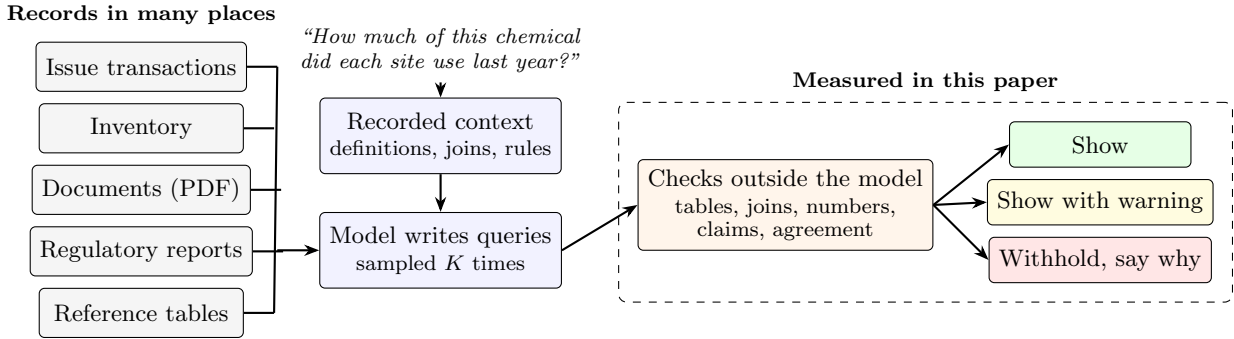


Figure 3: The setting. Records from several systems and the owners’ recorded definitions feed a model that writes queries. Checks outside the model grade the answer and its trace; a gate shows the answer, shows it with a warning, or withholds it with a reason. This paper measures the checks and the gate.

the people who write SQL.

Fluency is not the hard part. The failure that matters is the confident wrong answer: a valid query over real tables that answers a different question from the one asked, written up in the same assured prose as a right one. The model rarely invents the number; it chooses the wrong relationship, the wrong column or the wrong definition, and the arithmetic on top is correct. A reader checking by eye may not see it. For such a system to be used for decisions, two things matter more than the answer’s polish. Each answer has to show its work, the queries it ran, the records it read and the definitions it used, so that a reviewer can check it. And the system has to decide, answer by answer, whether the evidence supports showing the result at all, and say why when it does not.

Figure 3 shows the arrangement this paper assumes. Records from several sources and the organisation’s recorded definitions feed a model that writes queries; the model is sampled several times; a set of checks outside the model grades the answer and its trace; and a gate shows the answer, shows it with a warning, or withholds it, naming the reason. This paper is about the last two boxes.

1.3 What can go wrong, and what was designed to catch it

The failures follow from the problem. A query can name a table that does not exist, or join two columns whose keys barely overlap. A column can hold a stand-in code such as `-1` that a sum treats as a value. The written answer can quote a number that no returned row contains, or add claims the data does not support. The model can read the question one way on one run and another way on the next. And it can read it the same wrong way every time, as in both examples above.

The score studied here was designed against that list. After an answer is written, a checker goes back over the trace and grades whether the queries used real tables and sound joins, whether the numbers in the text match returned rows, whether the model claimed things the data does not support, whether every part of the question was answered, and whether independently sampled answers agree. A few failures are treated as too serious to average away and force the answer below the abstain line. The design was set by judgement, with the stated intent that its weights and thresholds would be calibrated on held-out questions. It had not been measured.

1.4 This study

This paper audits that design and measures its discrimination, calibration and gate behaviour on public text-to-SQL benchmarks, where every answer can be checked against a reference. Its scope is narrower than the setting above: it evaluates a reference implementation of the scoring and gating layer on structured databases, and does not evaluate document extraction, source reconciliation, or context-aware answer generation. It asks three questions.

- E1.** Does the score predict correctness? Are PASS answers mostly right and withheld answers mostly wrong?
- E2.** Does each check catch the failure it is built for, when that failure is planted in an answer that was right?
- E3.** Does each piece earn its place, measured by removing or changing one piece at a time?

We answered these on BIRD dev and the answers were not flattering to the design. The useful signal came from one place, agreement among sampled answers, and the score combined it in a way that discarded much of the information. We then wrote down a revised rule, froze the code, pre-registered the test and its pass criteria, and ran it on a second benchmark, Spider dev, that played no part in choosing the rule. We close by returning to the problem above: a checker cannot recover an intended business definition that is absent from the evidence it can see, and those definitions live with the people who own the data.

Contributions.

- An empirical audit of a designed, governed answer score on public text-to-SQL data with the production prompt and a current model, reported as measured, including the parts that did not work.
- A decomposition of where the score’s signal comes from and what each part costs, including an ablation that separates the agreement statistic from the result-matching policy, and a measurement of what the deterministic and judge checks cost in coverage on top of agreement.
- A pre-registered confirmation on Spider with every hypothesis reported pass or fail, and a replication on BIRD under the same harness.

- A labelling protocol for answers written in an application’s own style, and a measurement of how much the labeller moves the results.

The agreement statistic itself is not new: execution-based consistency across sampled queries is an established confidence signal for text-to-SQL (Section 2). What this paper adds is the audit of a designed score built around it, under pre-registration.

2 Related work

Consistency as confidence. Sampling several outputs and measuring their agreement is a standard black-box confidence signal: self-consistency for reasoning [14], SelfCheckGPT for hallucination [11], and semantic entropy, which clusters samples by meaning before measuring their spread [5, 9]. The designed L_2 term is semantic entropy with meaning fixed by execution.

Confidence for text-to-SQL. Entezari Maleki et al. [4] compare black-box and white-box confidence estimates for text-to-SQL and find consistency-based methods strongest among black-box approaches, with query execution a useful additional signal; the share of sampled queries that return the same result is the agreement statistic used here. Richardson [13] compare many signals on BIRD and Spider against execution-match labels and report AUROC of about 0.61 to 0.68 for black-box signals including execution self-consistency, 0.72 to 0.78 for an LLM judge, and about 0.82 for an ensemble of judges from two providers. Their labels, models and prompts differ from ours, so the numbers are not directly comparable; their result that verification signals can outrank consistency is relevant to Section 10, since we do not test a judge as the primary signal.

Selective prediction and calibration. Answering or abstaining by a score is selective classification [3, 6]; under domain shift it was studied for question answering by Kamath et al. [8]. We report the gate’s coverage and accuracy beside AUROC, and calibration error as defined by Guo et al. [7].

Pre-registration. Fixing hypotheses and analysis before seeing confirmatory data [12] is uncommon in this literature; we adopt it because the rules under test were chosen on the exploratory data.

3 The score

This section defines the designed score and the three frozen rules precisely enough to be re-implemented. The reference code is `score_v2.py` and `agreement.py` in the released bundle.

3.1 Setting and notation

A question q is asked over a database D . The answering system is sampled $K = 5$ times independently. Sample k produces a tool trace, a written answer, and a final query result R_k , a multiset of rows, or $R_k = \perp$ if it ran no successful query. Sample 0 is the answer shown to the user and scored; samples 1, $\dots$, 4 are used only for agreement. A label $y \in \{0, 1\}$ says whether the scored answer is right (Section 4.3).

3.2 The designed score (bench-1.0)

The designed score is a weighted mean of five terms over those that can be measured, followed by caps and a three-way gate. The weights are 0.30 for GROUNDED, 0.20 each for SCHEMA, SAFETY and RETRIEVE, and 0.10 for COMPLETE. With M the set of measured terms T_j ,

$$\text{IES}_0 = \frac{\sum_{j \in M} w_j T_j}{\sum_{j \in M} w_j}, \quad \text{IES} = \begin{cases} \min(\text{IES}_0, 0.59) & \text{if any hard cap fires,} \\ \min(\text{IES}_0, 0.79) & \text{if evidence is thin,} \\ \text{IES}_0 & \text{otherwise,} \end{cases} \quad (1)$$

and the gate is PASS if $\text{IES} \geq 0.80$, FLAG if $0.60 \leq \text{IES} < 0.80$, ABSTAIN otherwise. Evidence is thin when fewer than three terms are measured or GROUNDED is not.

GROUNDED is the mean of the defined parts among three shares checked against a profile of D : g_{tab} , the share of parsed queries whose tables all exist (CTE names and the SQLite catalogue are not tables); g_{join} , the share of equi-join column pairs (a, b) with key overlap

$$\rho(a, b) = \max\left(\frac{|K_a \cap K_b|}{|K_a|}, \frac{|K_a \cap K_b|}{|K_b|}\right) \geq 0.85, \quad (2)$$

where K_x is the set of distinct non-null values of column x ; and g_{val} , the share of returned columns that carry a stand-in code (a value such as -1 or 9999 covering at least 0.5% of an otherwise non-negative or much smaller column) and return none. A missing table, a join with $\rho < 0.85$, or a stand-in returned as a value fires a hard cap.

SCHEMA checks the numbers in the written answer. A number x written with d decimals in the Executive Summary or Detailed Analysis sections is *cited* if some returned cell, cell $\times 100$, or row count c satisfies $|x - c| \leq \frac{1}{2}10^{-d}$; numbers that appear in the question or hint, and the integers 1 to 5, are ignored. SCHEMA is 1 if at least one number is cited and unmeasured otherwise; any number neither cited nor ignored fires the hard cap UNCITED_NUMBER. In practice SCHEMA acts through its cap.

COMPLETE is the share of the question’s parts that the answer covers, as listed by a judge model that never sees the reference answer. **RETRIEVE** is not measured on these benchmarks, since every value is a database cell, and the weights renormalise over the rest.

SAFETY is the hallucination check TDD-H, the minimum of two layers here (a third, a competence boundary, is not run):

$$\text{SAFETY} = \min(L_1, L_2), \quad L_1 = 1 - r_1, \quad L_2 = 1 - \frac{H(\pi)}{\log K}. \quad (3)$$

L_1 is a checklist: $r_1 = 1$ if any deterministic tag fires or the judge reports an unsupported claim, $r_1 = \frac{1}{2}$ if any other judge or heuristic tag fires, and $r_1 = 0$ otherwise. An unsupported claim of any kind also fires a hard cap. L_2 is scaled semantic entropy [5, 9] with meaning fixed by execution: samples are clustered by the signature of their result set (full rows, floats rounded to six places), a sample with no successful query is its own cluster, and π is the distribution of the K samples over clusters.

3.3 What the entropy term cannot see

Proposition 1. *L_2 is invariant under any permutation of the K samples. Hence it takes the same value whether the scored sample 0 lies in the largest cluster or alone.*

Proof. π is a function of the multiset of cluster sizes, which does not depend on sample order. $\square$

For $K = 5$, if four samples agree and one differs, $\pi = (0.8, 0.2)$ and $L_2 = 1 - H(\pi)/\log 5 \approx 0.69$ in both cases: when the scored answer is with the four, and when it is the one. The same holds for any split. So L_2 measures how settled the model is in general, not whether the answer being shown is the one it settles on. The quantity the gate needs is the second. How often the difference matters in practice is measured in Section 8; it turns out to be small.

Definition 1 (Agreement with the scored answer).

$$A = \frac{1}{K} \sum_{k=0}^{K-1} \mathbf{1}[R_k \simeq R_0], \quad A = 0 \text{ if } R_0 = \perp, \quad (4)$$

where $R \simeq R'$ holds when both have the same row count and the narrower result equals, as a multiset of rows, a projection of the wider one onto an injective choice of its columns (extra columns allowed), after normalising values (floats to four places, numeric strings to numbers).

A takes values in $\{0, 0.2, \dots, 1\}$ and counts the scored sample itself, so $A \geq 0.2$ whenever $R_0 \neq \perp$. Results are stored up to 100 rows; when either result is truncated and the row counts are equal, $\mathbf{1}[\cdot]$ is undetermined. The pre-registered policy counts such pairs as agreeing and reports the bound with them counted as disagreeing.

3.4 The frozen rules

Three rules were written from the BIRD findings and frozen at commit `cb66ea7` (code bundle `6e3e723e8f76711d`) on 29 September 2026. None has a fitted parameter; all keep the designed thresholds.

v2 rule (2.0). The designed score with three changes: L_2 is replaced by A ; L_1 and the unsupported-claim cap count only judge issues that the judge classes as concerning the answer, not the commentary around it; and an answer with `COMPLETE` < 0.5 is capped at 0.79.

v2 agreement-first (2.1), the primary rule. The score is A itself, with deterministic checks as vetoes and caps:

$$s = \begin{cases} \min(A, 0.59) & \text{if a veto fires,} \\ \min(A, 0.79) & \text{if a cap fires,} \\ A & \text{otherwise,} \end{cases} \quad (5)$$

where the vetoes are a missing table, a bad join and a stand-in returned as a value, and the caps are an uncited number, a judge issue about the answer, `COMPLETE` < 0.5 , and thin evidence. With $K = 5$ the gate reads directly: PASS needs at least four of five samples to agree and no check to fire; three agreeing samples give FLAG; two or fewer give ABSTAIN. Every FLAG and ABSTAIN carries the names of the checks that caused it.

v3 (3.0). Agreement-first plus zero-cost signals aimed at confident misreadings, where all samples agree and are wrong. One firing signal caps at 0.79, two or more at 0.59. Only the empty-result signal (the final query returned no rows) was frozen for test.

4 Method

4.1 Data

BIRD dev [10]: 1,534 questions over 11 SQLite databases, each with reference SQL and an evidence hint that we pass to the model as a hint. Spider dev [15]: 1,034 questions over 20 databases, with reference SQL and no hint.

4.2 System under test

The answering model is GPT-5.1 with the system’s production system prompt, trimmed only of sections for tools the benchmark does not provide (vector search, charts, documents and citations, workflows, forecasting, a domain table guide); whole sections and lines were removed and nothing reworded (Appendix A). The model has three tools over the question’s database: list tables, table details with sample rows, and read-only SQL that shows it up to 100 rows. One added system message states that the last query must return exactly the requested columns and rows, since both benchmarks grade result sets. Five answers are sampled per question at default decoding on OpenAI’s flex tier. The IES judge is GPT-5 mini and never sees the reference.

4.3 Correctness labels

The benchmarks’ own rule, identical result sets, marks 54.2% of BIRD answers correct. It fails in two directions for an application that answers in its own style. It marks right answers wrong: names instead of ids, 0.83 instead of 83%, both values when the question asks which is larger. And it trusts reference SQL that is sometimes wrong, such as a formula that contradicts the hint.

We therefore label each answer with GPT-5.4 grading against the reference. The labeller sees the question, hint, reference SQL and result, the answer’s SQL and result, mechanical comparisons of the two, and the written answer, and rules on the answer (correct, incorrect, unclear), on the difference (none, form only, content), and on the reference (ok, wrong, question ambiguous). Answers ruled unclear or question-ambiguous are left out. On the 29 September BIRD run the labeller judged 161 reference answers wrong (10.5%) and kept 1,515 answers, 410 of them wrong.

The labeller is a model and its rulings move the results. Section 7 measures how much, by relabelling the same answers with the rebuilt labeller, and Section 7 describes the human audit that will precede submission. Correctness here means that the answer to the question asked is right: the returned rows and the figures the written answer reports from them. Unsupported commentary around a correct answer is a real defect, but it is recorded separately and does not make the answer incorrect. The score is judged against the same target.

4.4 Metrics

For a score s and labels y over n answers we report AUROC with a 95% interval from 2,000 bootstrap resamples of answers [2], the paired bootstrap interval of $\text{AUROC}(s') - \text{AUROC}(s)$ on the same resamples, expected calibration error over 10 equal-width bins [7], the Brier score, and

Table 1: BIRD dev, 29 September run, 1,515 labelled answers. The rules below the line were derived from these answers and are in-sample here.

Score	AUROC (95% CI)	PASS accuracy	PASS share	Correct withheld	Wrong withheld
Designed	0.624 (0.595–0.652)	77.9%	58.5%	35.7%	43.7%
v2 rule	0.694 (0.666–0.723)	77.4%	80.0%	14.0%	25.4%
Agreement alone	0.721 (0.692–0.748)	81.3%	78.9%	6.3%	32.9%
Agreement-first	0.730 (0.702–0.756)	83.6%	62.0%	8.3%	37.3%
v3 (empty result)	0.740 (0.713–0.768)	84.6%	61.3%	8.3%	37.3%
Fitted (exploratory)	0.718 (0.685–0.749)	86.0%	53.3%	17.7%	51.5%

three gate quantities:

$$\begin{aligned}
 \text{PASS accuracy} &= P(y = 1 \mid \text{PASS}), \\
 \text{correct withheld} &= P(\text{ABSTAIN} \mid y = 1), \\
 \text{wrong withheld} &= P(\text{ABSTAIN} \mid y = 0).
 \end{aligned}$$

Ties in a discrete score count one half in AUROC. The fitted variant reported as exploratory uses logistic regression with whole-database cross-fitting over 20 random splits, so no database is on both sides of a fit.

4.5 Order of work

1. **Exploratory, BIRD, 29 September.** E1 to E3 on one full run. The frozen rules come from this step and are evaluated on the same answers, so their BIRD numbers are in-sample.
2. **Freeze and pre-registration, 29 September.** Code bundle hashed; hypotheses H1 to H4 and their pass criteria written before any Spider answer existed (Section 6).
3. **Harness rebuild, 11 October.** The benchmark harness used on 29 September was not recoverable. We rebuilt it and validated the rebuilt designed scorer field for field against all 1,534 stored 29 September records before using it.
4. **Confirmatory, Spider, 11–12 October,** on the rebuilt harness, evaluated by the frozen bundle.
5. **Replication, BIRD, 11 October,** a full rerun on the rebuilt harness with the same settings as Spider. This is not pre-registered and is reported as a check that the harness rebuild did not change the BIRD picture.

5 Exploratory results on BIRD

5.1 E1: the designed score

GPT-5.1 with the production prompt answered 72.9% of the 1,515 labelled BIRD questions correctly. Table 1 gives the designed score and the rules that came out of this analysis, on the same answers.

The PASS gate holds up at the top: PASS answers are right 77.9% of the time against 72.9% overall. Below the top the score carries little information. It withholds 35.7% of correct answers and 43.7% of wrong ones. The hard cap puts 534 answers at exactly 0.59, and 70.4% of them are

Table 2: Single changes to the designed score, recomputed from saved components (BIRD, 29 September run).

Change	AUROC	PASS acc.	PASS share
None (designed)	0.624	77.9%	58.5%
Drop GROUNDED	0.631	82.5%	48.2%
Drop SCHEMA	0.625	79.9%	56.1%
Drop COMPLETE	0.624	78.1%	58.4%
Drop SAFETY	0.555	75.5%	62.2%
SAFETY without L_2	0.572	75.7%	61.8%
SAFETY as mean instead of min	0.620	75.8%	61.8%
No thin-evidence cap	0.624	77.9%	58.5%
No hard caps	0.638	77.9%	58.5%
Unsupported claims capped only when about the answer	0.691	77.9%	78.4%
Unsupported claims ignored	0.699	77.0%	83.8%
Agreement alone, relaxed rows	0.721	81.3%	78.9%

correct, close to the average. The score is not a probability: answers scoring 0.9–1.0 are right 84.4% of the time, 0.8–0.9 62.5%, and 0.7–0.8 36.5%.

5.2 E3: what each piece contributes

Recomputing the score from saved components, one change at a time, with no new model calls:

Three findings shaped the frozen rules.

Agreement carries the signal. Removing L_2 from SAFETY drops AUROC to 0.572; agreement alone reaches 0.721. Dropping GROUNDED, SCHEMA or COMPLETE, removing the thin-evidence cap, or taking the mean instead of the minimum each move AUROC by 0.007 or less. On BIRD the databases are clean and the model rarely invents tables, so the grounding terms are almost always 1 and carry no ranking information here.

Unsupported-claim handling is the largest observed source of degradation. It acts twice: through the checklist L_1 , which drops to zero on any unsupported claim and so pulls SAFETY to zero through the minimum, and through the hard cap. Section 8 separates the two; most of the loss comes through L_1 and the minimum. The cap fires on 30.0% of answers, and those answers are right 71.9% of the time, close to the average. Most of the answers it caps (360 of 455) carry a claim the judge itself classed as commentary, and answers with such a claim are right 75.6% of the time. The claims are real (“Super Strength appears frequently in the dataset”, from a single count), and they come from the production prompt’s instruction to enrich every answer with comparisons and interpretation. They are a fault in what the user reads, but they say little about whether the answer is right.

The checks that work are rare. When the judge reports an intent mismatch the answer is right only 32.5% of the time; after a bad join, 46.8%; after a wrong formula, 44.7%. Each of these fires on 5.5% of answers or fewer.

5.3 E2: planted errors

From answers labelled correct we planted each of eight checklist errors, 60 per kind (Appendix B). For errors that change the query, the changed SQL was executed and the model rewrote its answer from the new rows, so the text agrees with the broken result. All 480 of 480 controls, rescored unchanged, reproduced their stored scores.

Table 3: Planted errors, 60 per kind: share the designed score withheld, share not passed, and share where the check built for that error fired.

Error	Withheld	Not PASS	Own check
Fake table	100.0%	100.0%	100.0%
Bad join	100.0%	100.0%	100.0%
Stand-in as value	100.0%	100.0%	100.0%
Wrong number	100.0%	100.0%	100.0%
Wrong time window	90.0%	95.0%	83.3%
Wrong aggregate	75.0%	81.7%	96.7%
Intent flip	75.0%	91.7%	63.3%
Partial answer	70.0%	75.0%	55.0%

The four deterministic checks withheld every planted error of their kind, each by its own check. The judge-based checks withheld 70–90%. Real errors are harder: the designed score withholds 43.7% of the wrong answers the model actually produced. Fault injection overstates real-world detection and should be reported beside it.

6 Pre-registered confirmation on Spider

6.1 What was fixed in advance

On 29 September, before any Spider answer was generated, we fixed the scorer code (bundle 6e3e723e8f76711d; `evaluate.py` prints the hash and a mismatch voids the run), the rules under test (agreement-first as primary, v3 with the empty-result signal), the thresholds, the truncation policy, the exclusion rule for labels, and four hypotheses [12]:

- H1 (primary).** Agreement-first ranks answers better than the designed score: the paired bootstrap 95% interval of $\text{AUROC}(\text{agree-first}) - \text{AUROC}(\text{designed})$ lies above 0.
- H2.** Agreement-first withholds fewer correct answers than the designed score, and its PASS accuracy is not lower by more than 2 points.
- H3.** An empty final result marks a likely wrong answer: accuracy when it fires is at least 30 points below overall accuracy, tested only if it fires on 10 or more answers.
- H4 (descriptive).** $\text{AUROC}(\text{designed}) < \text{AUROC}(\text{alone}) \leq \text{AUROC}(\text{agree-first}) \leq \text{AUROC}(\text{v3})$.

6.2 Results

The evaluation printed the frozen bundle hash and reproduced the stored designed score and gate state on all 983 kept answers. Of 1,034 answers, 51 were left out as unclear or question-ambiguous; 73 of the kept answers are labelled wrong and accuracy is 92.6%.

H1 and H2 hold. The primary effect on Spider (+0.159) is larger than its in-sample size on the 29 September BIRD run (+0.106) and close to the BIRD rerun under the same harness (+0.165), and agreement-first withholds about a fifth as many correct answers as the designed score without losing PASS accuracy. H2’s pre-registered criterion is a point estimate; the paired bootstrap interval for the change in PASS accuracy is +0.3 to +3.4 points, so a non-inferiority claim at a two-point margin also holds on the interval.

Table 4: Spider dev, pre-registered confirmation, 983 labelled answers.

Score	AUROC (95% CI)	PASS accuracy	PASS share	Correct withheld	Wrong withheld
Designed	0.566 (0.503–0.626)	93.9%	49.7%	49.0%	54.8%
Agreement-first (primary)	0.725 (0.663–0.787)	95.7%	73.6%	8.8%	35.6%
v3 (empty result)	0.767 (0.707–0.821)	96.8%	70.7%	8.8%	35.6%
Agreement alone	0.749 (0.688–0.809)	95.5%	90.4%	3.7%	35.6%
<i>Not pre-registered</i>					
v2 rule	0.682 (0.623–0.740)	93.2%	85.1%	13.7%	15.1%
Fitted	0.729 (0.651–0.804)	94.7%	82.9%	12.1%	32.9%

Table 5: Pre-registered hypotheses on Spider dev.

Observed	Verdict
H1 AUROC difference +0.159 (95% CI +0.097 to +0.221); with truncated pairs counted as disagreeing, agreement-first AUROC 0.721	Pass
H2 correct withheld 8.8% vs 49.0%; PASS accuracy 95.7% vs 93.9% (+1.8 points)	Pass
H3 empty result fired on 37 answers (3.8%); right 73.0% of the time vs 92.6% overall, 19.6 points below	Fail
H4 designed 0.566 < agreement-first 0.725 < agreement alone 0.749 < v3 0.767	Not as predicted

H3 fails. On both BIRD runs an empty final result was a strong sign of a wrong answer (right 6.5% and 3.2% of the time). On Spider it fired more often and the answers were right 73.0% of the time. We have not established why. One candidate is that some Spider questions have empty correct answers over its small databases; another is that the labeller treats an empty result that matches an empty reference as right. We report the failure as pre-registered and do not adjust the rule.

H4 reverses one step. Agreement alone ranked above agreement-first on Spider, the opposite of BIRD. One explanation, untested, is that the caps agreement-first adds lower some correct answers that all five samples agreed on, and on easier data with fewer wrong answers that costs more ranking than it gains.

Deviations from the pre-registration. The answering prompt is the rebuilt trim (`prompt_sha d86e574ca9c8`), not `6566415c9593`, because the 29 September trimmed file was lost with the harness. The judge and labelling prompts are the rebuilt ones. The rule code, thresholds, truncation policy and exclusion rule are unchanged, which is what the bundle hash checks. The pre-registration also named commit `cb66ea7`; the evaluation ran at commit `1b95a28`, which adds the harness and carries the same bundle.

Limits of this test. Only 73 answers are labelled wrong, fewer than the 100 the pre-registration named as the point below which intervals are too wide to say much, and the intervals are wide. Spider is easier than BIRD for this model.

7 Replication and labeller sensitivity

BIRD under the rebuilt harness. The full rerun gave accuracy 82.5% on 1,424 labelled answers. Agreement-first improved AUROC over the designed score by +0.165 (95% CI +0.129 to

Table 6: AUROC of agreement-first against the designed score in every evaluation.

Evaluation	n	Acc.	Designed	Agree-first	Difference (95% CI)
BIRD, 29 Sep (in-sample)	1,515	72.9%	0.624	0.730	+0.106 (+0.077 to +0.134)
BIRD, 29 Sep, rebuilt labels	1,441	82.6%	0.631	0.756	+0.124 (+0.089 to +0.158)
BIRD, rebuilt harness	1,424	82.5%	0.589	0.755	+0.165 (+0.129 to +0.204)
Spider (pre-registered)	983	92.6%	0.566	0.725	+0.159 (+0.097 to +0.221)

+0.204), from 0.589 to 0.755, and withheld 11.4% of correct answers against 47.5%. Execution match was 54.4%, against 54.2% on 29 September, so the rebuilt harness answers BIRD about as well as the original.

Sensitivity to label version. Relabelling the 29 September BIRD answers with the rebuilt labeller gave 90.8% agreement with the original labels on 1,430 answers both kept ($\kappa = 0.721$). The rebuilt labeller is more lenient: it moved 114 answers from wrong to right and 18 the other way, and accuracy rose from 72.9% to 82.6%. Under the rebuilt labels the agreement-first gain on the same answers is +0.124 (95% CI +0.089 to +0.158), against +0.106 under the original labels. The direction of the comparison was preserved under both label versions. Two model-generated label sets agreeing is not independent validation of either, and unvalidated labels affect every comparison in this paper, not only absolute accuracy: AUROC, PASS accuracy and the judgement of whether a check helps all depend on them.

Under execution-match labels. The benchmarks’ own rule (identical result sets) is too strict for answers written in an application’s style, but it involves no model judgement. Scored against it on every answer, AUROC was 0.551 for the designed score, 0.651 for agreement alone and 0.669 for agreement-first on Spider (1,034 answers, 68.9% matching), 0.574, 0.690 and 0.699 on the BIRD rerun, and 0.627, 0.692 and 0.692 on the 29 September run. Both agreement-based scores rank above the designed score under these labels too. Their order relative to each other is less stable. On Spider, agreement alone ranks higher under the model-judged labels (0.749 against 0.725), and agreement-first ranks higher under execution match (0.669 against 0.651). The two protocols also keep different answer sets, so we repeated the execution-match comparison on the 983 answers the model-judged protocol keeps: 0.557 for the designed score, 0.639 for agreement alone and 0.663 for agreement-first. The reversal persists on the same answers, so it follows from the label definition; it is a reversal in point estimates, and we have not computed paired intervals for these two differences. One possible reason, untested, is that execution match counts an answer that differs only in form as wrong and the checks fire on some of those answers. The checks’ value for predicting correctness therefore depends on what is counted as correct; their coverage cost does not.

The hand check. Before submission, two reviewers blinded to every score will label a random sample of answers from each evaluation, all 127 Spider answers whose reference the labeller judged wrong, and the answers on which the scores make different gate decisions. The random sample estimates label quality; the targeted groups are reported separately, since they were chosen for difficulty and do not represent the whole. We will report the adjudicated labels and every disagreement, not only agreement and Cohen’s κ [1], and recompute the results on the adjudicated set.

The labeller and the reference. On Spider the rebuilt labeller judged 127 reference answers wrong and marked 120 of those answers correct. If the labeller is too lenient, this is where it would

Table 7: AUROC of the designed score rebuilt one step at a time.

Evaluation	L_2 alone	$\min(L_1, L_2)$	Weighted mean	With caps (designed)
BIRD, 29 Sep	0.696	0.620	0.638	0.624
BIRD, rebuilt harness	0.715	0.583	0.606	0.589
Spider	0.725	0.565	0.584	0.566

show, and the hand check draws part of its sample from this group.

8 What the gain is made of

These analyses were not pre-registered. They use the stored samples and labels, with no model calls (`analyze_ablation.py`).

Where the designed score loses its ranking. Adding a constant to a score or scaling it changes calibration and decisions at fixed thresholds, but not the ordering of answers, so it cannot change AUROC. Ranking is lost only where a step reorders answers or creates ties. Table 7 rebuilds the designed score one step at a time. The designed agreement term L_2 alone ranks well. The large loss comes at the next step, $\text{SAFETY} = \min(L_1, L_2)$: L_1 drops to zero whenever the judge reports an unsupported claim, including claims in commentary, and the minimum then discards L_2 for those answers. The weighted mean recovers a little, because the other terms vary on some answers, and the hard caps cost a little more by tying 437 Spider answers at exactly 0.59.

The statistic under one matching policy. The designed L_2 and the agreement share A differ in two ways: the statistic (entropy over clusters against the share agreeing with the shown answer), and the matching policy (full-row signatures against projection matching with four-place rounding). Holding the matching relation fixed and changing only the statistic isolates Proposition 1’s practical effect. Strict matching (same columns, same multiset of rows) is an equivalence on results, so its clusters are well defined. Relaxed matching is not transitive: a one-column result can match a two-column result through its first column, and that result can match another one-column result through its second, while the two one-column results do not match each other. For relaxed entropy we therefore cluster greedily in sample order, each sample joining the first cluster whose first member it matches; other procedures, such as connected components, can give different clusters, so the relaxed comparison is reported as secondary. With strict matching, A minus the entropy score was +0.006 AUROC (95% CI -0.002 to +0.014) on the 29 September BIRD run, +0.019 (+0.006 to +0.032) on the BIRD rerun and +0.011 (-0.001 to +0.023) on Spider; with relaxed matching, +0.006, +0.017 and +0.006. The proposition is true, but the case it describes, a shown answer outside a large cluster, is rare enough that the choice of statistic moves AUROC by less than 0.02. The gain over the designed score comes from not taking the minimum with a checklist that fires on commentary.

What the checks do on top of agreement. Agreement-first is agreement alone with vetoes and caps added. Because the checks can only lower a score, they move answers out of PASS and never into it. Table 8 gives the counts. On Spider the checks removed 166 answers from PASS: 157 correct and 9 wrong. All 9 wrong ones went to FLAG, where they are still shown with a warning; the number of wrong answers withheld was unchanged, while 46 more correct answers were withheld. PASS accuracy changed by +0.2 points (95% CI -0.5 to +1.0). On BIRD the checks do more: on

Table 8: Gate decisions by correctness (right / wrong). FLAG answers are shown with a warning; ABSTAIN answers are withheld.

Evaluation	Score	PASS	FLAG	ABSTAIN
Spider	Designed	459 / 30	5 / 3	446 / 40
	Agreement alone	849 / 40	27 / 7	34 / 26
	Agreement-first	692 / 31	138 / 16	80 / 26
BIRD, rebuilt	Designed	593 / 98	24 / 17	558 / 134
	Agreement alone	970 / 107	95 / 35	110 / 107
	Agreement-first	752 / 61	289 / 74	134 / 114

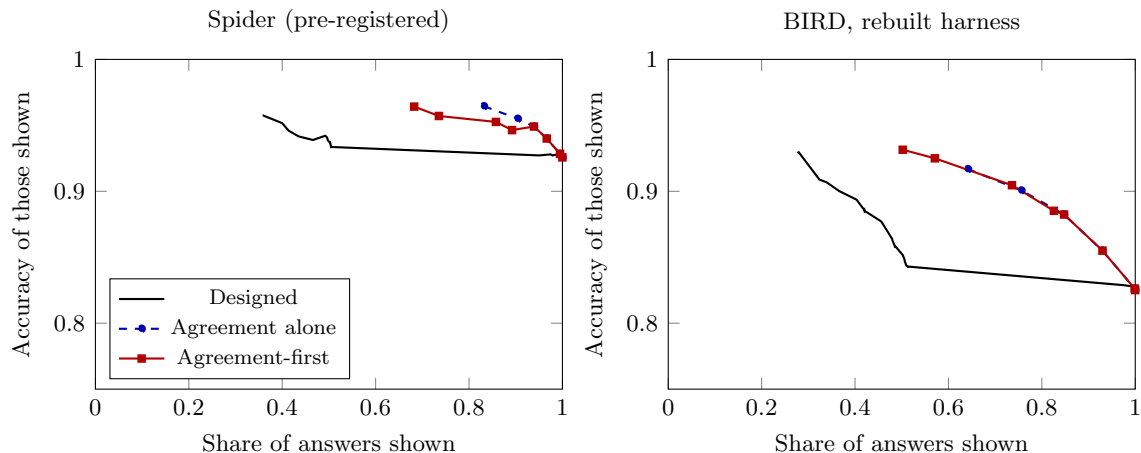


Figure 4: Accuracy and coverage: accuracy of the answers shown as the threshold is lowered. Markers, and every vertex of the designed score’s curve, are attainable thresholds; the lines between them are visual guides.

the rerun they removed 264 answers from PASS (46 wrong), raising PASS accuracy by +2.4 points (+1.2 to +3.7), with 7 wrong and 22 correct answers moved directly from PASS under agreement alone to ABSTAIN; the net rise in correct answers withheld also includes answers that agreement alone had already flagged. On these benchmarks the checks mainly turn passes into warnings and cost unqualified coverage; they rarely withhold a wrong answer that agreement would have shown. Whether they do more on records where tables are missing, joins are ambiguous and codes stand in for values is the question their design was for, and these benchmarks do not answer it.

Accuracy and coverage. The gate’s thresholds were set by design and not fitted. Figure 4 shows, for every threshold, the share of answers shown and their accuracy. Agreement alone and agreement-first lie above the designed score over most of the range; at the lowest coverages on BIRD the three are close. The discrete agreement share gives only a few operating points.

Calibration. None of these scores was designed as a probability, and the agreement share takes only six values. Table 9 gives expected calibration error and Brier score for completeness. Agreement-based scores are better calibrated than the designed score in every evaluation; a deployment that wants a probability should fit one on its own labelled answers.

Table 9: Expected calibration error (10 bins) and Brier score.

Evaluation	Designed		Agreement alone		Agreement-first	
	ECE	Brier	ECE	Brier	ECE	Brier
BIRD, 29 Sep	0.155	0.209	0.130	0.183	0.098	0.175
BIRD, rebuilt harness	0.157	0.166	0.097	0.134	0.093	0.133
Spider	0.187	0.125	0.057	0.066	0.088	0.077

9 Discussion

What the score can claim. On two public benchmarks with one model, letting agreement among samples decide the gate ranks answers better than the designed weighted score and withholds far fewer correct answers at the same or better PASS accuracy. The claim that a designed score of 0.80 means “right about 80% of the time” does not hold across the scale and we do not make it.

Agreement, not the statistic. Proposition 1 identifies a real blind spot in the designed entropy term, but under a common matching policy it costs less than 0.02 AUROC (Section 8). The designed score lost its ranking where agreement met the checklist through a minimum, and the checklist fired on commentary. Agreement with the shown answer remains the better quantity to gate on, since it is the one the gate is about, but the evidence for preferring it over the designed entropy is modest.

The checks have not yet earned their coverage. The deterministic checks catch every planted error of their kind. On real answers from these benchmarks they and the judge caps rarely fire on wrong answers that agreement missed. On Spider they moved 157 correct and 9 wrong answers out of PASS, put the wrong ones behind a warning, and withheld no additional wrong answer. They were designed for extracted, multi-source records where tables, joins and codes go wrong, and their case has to be made on such records. Poor discrimination of core-answer correctness also does not make a check useless for other governance purposes, such as flagging unsupported commentary. It suggests that those purposes should not be folded into the same confidence number; part of the design problem may be asking one scalar to carry several kinds of assurance.

Commentary is a product fault. The prompt’s instruction to enrich every answer produces unsupported commentary in 23.8% of BIRD answers. Capping unsupported claims only when they concern the answer raises the designed score’s PASS share from 58.5% to 78.4% at the same PASS accuracy (77.9%); ignoring commentary entirely leaves the user reading it. The frozen rules record it and name it without changing the gate.

Confident misreadings remain. When all five samples agree and are wrong, agreement cannot help. The empty-result signal was meant for part of this and did not meet its pre-registered effect-size criterion on Spider. This is the open problem the rules leave, and Section 10 argues that it is a problem of context.

10 Next steps: context

This section is forward-looking. Nothing in it is measured in this paper.

The limitation. Agreement among samples shows whether the model reads a question the same way each time. It cannot resolve an intended business definition that is absent from the evidence the model and the checker can see. If every sample makes the same choice of meaning, agreement is complete whether or not that meaning is the one the organisation uses, and the deterministic checks can all pass: the tables exist, the joins pass the implemented overlap check and every number is a returned cell. Appendix D describes such a case from a staging implementation over a government organisation’s hazardous-materials records. Asked how much water was used, the system read storage-reporting tables and reported the recorded weight of products in which water is listed; with an expert-approved definition of usage it followed the issue transactions through each product’s composition and reported the weight of water in what was issued. The two recorded outputs cover different reporting periods, so their numerical difference illustrates the error and does not estimate its size.

Definitions as checks. The case suggests a use for approved definitions beyond improving the answer. A recorded rule such as “usage questions follow the issue transactions; the storage-reporting tables answer only storage questions” can be tested against a trace independently of how the answer was produced: an answer about usage whose queries read the storage-reporting tables breaks it, and the gate can name the rule. Such checks would sit beside the existing vetoes. Whether they catch confident misreadings without withholding many correct answers is an empirical question, and Section 8 shows that checks can cost more coverage than they repay.

The experiment. Context can help in two places, the answering system and the checking system, and a single before-and-after case cannot tell them apart. The test we will run crosses the two on held-out questions in four conditions: no context for either; context for answer generation only; context for checking only; and context for both. In the checking-only condition the generated answer and its trace are held fixed, so any gain is the checker’s alone. The context rules are frozen before the held-out questions are evaluated. We will report answer correctness, wrong answers passed, correct answers withheld, and the number of wrong answers that arrived with full agreement. BIRD’s evidence hints give a public version of the same contrast, since each hint is a recorded definition: the same questions can be run with and without them.

In deployment. Until that test is done, the measurements here support a narrower practice. A PASS should mean that at least four of five samples reached the same result and no check fired, and every FLAG or ABSTAIN should name what caused it so a reviewer can act. The score should not be presented as a probability. Each answer an expert approves or corrects is a labelled example in the customer’s own terms, and a set of them is what a per-customer calibration of the thresholds would need.

11 Limitations

- **Labels are model-judged** and the hand check is not yet done; the two labellers differ by about ten points of accuracy on the same answers.
- **One answering model** (GPT-5.1) and two benchmarks, neither in the system’s own domain.
- **The Spider test has 73 wrong answers**, below the 100 the pre-registration set as adequate.

- **The harness was rebuilt** between the exploratory run and the confirmation, and the answering, judge and labelling prompts differ from the 29 September ones as described; the scorer and rules do not.
- **Reference implementation.** We test the reference scorer written from the design text, not the product scorer, which differs in places (for example its RETRIEVE term without a retrieval step).
- **Untested parts.** The competence-boundary layer and RETRIEVE are not tested; the stand-in check is tested only by planting.
- **The benchmark rule** asking the last query to match the requested output is not in the production prompt.
- **One run per benchmark.** Run-to-run variance is not measured, beyond the BIRD rerun.
- **The checks’ value is untested where it should matter.** On these benchmarks the checks always cost unqualified coverage, and whether they improve ranking depends on the dataset and the correctness definition.
- **Core-answer correctness only.** The labels judge the answer to the question asked; unsupported commentary is recorded separately and is not what the scores are evaluated against.
- **Exploratory analyses.** Sections 5 and 8 and the fitted variant are not pre-registered; only H1 to H4 on Spider are confirmatory.

12 Conclusion

A governed answer score can be audited on public data at modest cost, and this one did not hold up as designed. Its deterministic checks catch the errors they are built for; its ranking came from agreement among samples, and was lost where agreement was combined by a minimum with a checklist that fired on commentary. A frozen, pre-registered rule that lets agreement decide the gate improved ranking and coverage of core-answer correctness on Spider as it had on BIRD, and both agreement-based scores outperformed the designed score under every labelling protocol tested. The improvement owes little to the choice of agreement statistic. The added checks reduced unqualified coverage, and their ranking benefit depended on the dataset and the correctness definition; on Spider under model-judged labels they moved 9 additional wrong answers to warnings and withheld no additional wrong answer. The empty-result signal did not meet its pre-registered criterion on Spider. Agreement cannot supply a definition the evidence does not contain; testing whether approved definitions improve the answers, the checks, or both is the next study.

Competing interests

The authors founded and work for the company that develops the system under test. The scorer, the rules and the pre-registration were fixed before any confirmatory answer existed, and every result is reported, including those unfavourable to the design.

Reproducibility

The harness, the frozen scorer bundle, the analysis scripts, the result files and labels, and the pre-registration are at <https://github.com/numberz-ai/numberz-ies-tdd-h-calibration>, released with its full commit history. The pre-registration, dated 29 September 2026 in the file, was committed at `cb66ea7` before any Spider answer was generated, in what was then a private repository; it was not publicly timestamped at the time, so its date rests on that history. The production system prompt is not released; `bench/prompt/trim_spec.json` records every line range kept and every line removed, with the reason. Answer and label files are attached to the release, and the paper’s analysis scripts are in `paper_analysis/`. Everything runs from `bench/`. The evaluation code is the frozen bundle `6e3e723e8f76711d`; `evaluate.py` prints the hash and the hashes of its inputs, and `make_numbers.py` refuses any result whose bundle or inputs do not match. Every number in the results is a macro written by that script; the exploratory analysis is recomputed from the stored records by `analyze_exploratory.py` and `analyze_ablation.py`. The 29 September BIRD evaluation was reproduced on a second host (Linux, Python 3.13) with identical results for every reported score.

References

- [1] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.
- [2] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.
- [3] Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.
- [4] Sepideh Entezari Maleki, Mohammadreza Pourreza, and Davood Rafiei. Confidence estimation for text-to-SQL in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32474–32482, 2026. doi: 10.1609/aaai.v40i38.40523.
- [5] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024.
- [6] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, 2017.
- [7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.
- [8] Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *Proceedings of ACL*, 2020.
- [9] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023.
- [10] Jinyang Li, Binyuan Hui, Ge Qu, et al. 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 (Datasets and Benchmarks Track)*, 2023.

- [11] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of EMNLP*, 2023.
- [12] Brian A. Nosek, Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606, 2018.
- [13] Robert Richardson. What predicts correctness in text-to-SQL? A selective-prediction study. *arXiv preprint arXiv:2607.06799*, 2026.
- [14] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.
- [15] Tao Yu, Rui Zhang, Kai Yang, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of EMNLP*, 2018.

A Prompt sections removed for the benchmark

Removed: memory and metadata precedence; document-versus-data routing and vector-first retrieval, tool discovery and proxying; the vector-database strategy; a domain table guide; citation and document handling; visualisation rules and chart-tool schemas; tool-output truncation; protocol-specific SQL repair; workflow creation; forecasting. Kept: identity and domain, priority hierarchy, core rules, SQL construction rules, the no-vector-database strategy, interpretation and clarification policies, response format, numeric fidelity, data verification, name resolution, insight generation, and the self-check. The cut is a list of line ranges with a reason for each dropped line (`prompt/trim_spec.json`); the script only deletes and stops if the source prompt changes.

B Planted errors

Error	How it is planted
Fake table	A table in the final SQL renamed to one that does not exist
Bad join	The join rewired to a column whose keys overlap the other side’s by less than 0.85 in both directions
Stand-in as value	A <code>−1</code> code written into a numeric result column, recorded as that column’s stand-in (simulated; BIRD has few)
Wrong number	One number in the written answer changed to a value in no returned row
Wrong time window	A year or date in the filter shifted by one year
Wrong aggregate	<code>SUM</code> ↔ <code>AVG</code> , or <code>COUNT</code> ↔ <code>COUNT DISTINCT</code>
Intent flip	Sort direction, <code>MAX</code> ↔ <code>MIN</code> , or a comparison reversed
Partial answer	Half the rows, or the last requested column, left out

C Fixes to the reference scorer before the BIRD run

Names defined with `WITH` are not tables. Join readiness takes the better of the two directions of key overlap; read literally, the design’s rule depends on the order of the `ON` clause. The SQLite

catalogue and table-valued functions are not fabricated tables; before the fix they caused false caps in the full run.

D A worked case from a staging implementation

This appendix describes questions asked of the system in a staging implementation over a government organisation’s hazardous-materials records. The organisation is not named. Table and column names are generalised and the queries are simplified; they follow the structure of the queries recorded in the session logs. The figures come from those sessions, are not part of this study’s measurements, and have not been independently reconciled.

The records

Two families of tables describe chemicals here, and they answer different questions. Annual storage-reporting tables, one per reporting year, were built for the federal chemical-inventory rule. They record, per site and chemical registry (CAS) number, quantities as signed transaction deltas in pounds. Issue tables record what was handed out at issue points: one row per issue transaction, linked to an inventory item, which is linked to a product and its safety data sheet. The safety data sheet lists the product’s constituents, each with a CAS number and a percentage by weight, and the system reads that section as rows. A chemical that is a constituent of many products, such as water, appears in the issue history only through those composition rows.

First answer: the wrong quantity

The user asked how many pounds of CAS 7732-18-5 (water) were used per year. The model read *used* as a question for the storage-reporting tables, which carry a CAS column and a pounds column. Its query had this shape, grouped by calendar year of the transaction date:

```
WITH s AS (
  SELECT cas, site_id, tran_date, cas_qty_lbs FROM storage_report_2022
  UNION ALL SELECT cas, site_id, tran_date, cas_qty_lbs FROM storage_report_2023
  UNION ALL SELECT cas, site_id, tran_date, cas_qty_lbs FROM storage_report_2024
  UNION ALL SELECT cas, site_id, tran_date, cas_qty_lbs FROM storage_report_2025
)
SELECT YEAR(tran_date)      AS yr,
       -SUM(cas_qty_lbs)    AS pounds_issued  -- removals are negative deltas
FROM   s
WHERE  cas = '7732-18-5' AND cas_qty_lbs < 0
GROUP BY YEAR(tran_date);
```

It returned 230,158 pounds for 2022, 192,337 for 2023, 329,949 for 2024 and 4,589 for 2025, and the answer was written up with a chart. The tables exist, the query has no join, no stand-in values were returned, and every number in the text is a returned cell; repeated runs made the same choice. The quantity was the wrong one: these tables report, for storage reporting, the weight of products in which water is listed, not the water in what was issued.

Second attempt: no matching records and an explicit limitation

Told to use the issue data and not to fall back to storage reporting, the model followed the issue chain to a chemical lookup:

```

SELECT YEAR(h.issue_date) AS yr, SUM(h.qty_issued) AS qty
FROM   inventory_issue_history h
JOIN   inventory_details d ON d.inventory_id = h.inventory_id
JOIN   chemical_list      c ON c.chemical_id = d.sds_id      -- assumed link
WHERE  c.cas = '7732-18-5'
GROUP BY YEAR(h.issue_date);

```

This returned no rows for any year. The answer did not report zero use. It confirmed that the CAS number exists in the chemical list under the name WATER, listed the issue tables it had checked with their row counts (over nine million issue transactions), stated that it had not substituted the storage-reporting figures, and said that no rows for a chemical this common more likely meant a broken join path than no use. It withheld a quantity it could not support.

The approved definition and the answer under it

The organisation's subject-matter expert stated what *usage* means there: the weight of the chemical itself in what was issued, from the issue transactions through each product's safety data sheet. Model weights and application code were unchanged; the system received this definition and the routing instruction that follows from it as recorded context. The query followed the path the schema does not show, from an issue to its product's composition rows. The analysis was annual; fiscal 2024 is shown:

```

SELECT SUM(h.qty_issued_kg * 2.20462 * sc.weight_pct / 100.0) AS water_lbs
FROM   inventory_issue_history h
JOIN   inventory_details d ON d.inventory_id = h.inventory_id
JOIN   sds_composition  sc ON sc.sds_id      = d.sds_id
WHERE  sc.cas = '7732-18-5'
      AND h.issue_date >= '2023-10-01' AND h.issue_date < '2024-10-01';  -- fiscal 2024

```

It returned 2,618.94 pounds of water for fiscal 2024. The first answer's 2024 figure covers the calendar year and a different quantity, so the two are not a same-period comparison. The displayed query is evidence of the intended calculation, not proof that its total is right. An independent reconciliation would record the reporting period, the number of transactions included, the unit normalisation, the version of each safety data sheet used, how missing or ranged percentages were treated, and the expected join cardinality (one composition row per product and CAS, so that no issue is counted twice), and would compare the total with one the expert computes separately. That reconciliation has not been done.

Other issues the staging sessions surfaced

A second line of questions, about hexavalent chromium, showed other ways a query that passes every trace-based check can answer the wrong question in these records.

- *One substance, several names.* Users wrote CrVI; the records say hexavalent chromium, and the substance is held under several CAS numbers (the ion, chromium trioxide, potassium dichromate, and in a fuller reading six more chromium(VI) salts). Which set counts is the organisation's decision; the schema does not state it.
- *Identifiers stored inconsistently.* An exact match on a CAS number can return zero where a pattern probe finds the same number with different dashes or spacing. The rule adopted was to probe before reporting no records.

- *Tables that do not mean what their names say.* A table named for one reporting year held rows from other years, so year filters had to use the transaction date.
- *Signed quantities.* The pounds column is a transaction delta, positive for additions and negative for removals; summing it without the sign rule gives neither quantity.
- *Codes without names.* Site codes did not always resolve to site names. The rule adopted was to show the code, labelled as a code, and not to guess a name.

What the case shows and what it does not

Each fix was a short statement of meaning, approved by someone who owns that meaning: which tables answer which kind of question, what measure a word resolves to, which identifiers name the same thing, how a quantity's sign is read, and what to do when the evidence is absent. The case shows that such statements changed the answers. It does not show that they improve the confidence score's ability to separate right answers from wrong ones; that is the experiment of Section 10.