VeriMinder: Mitigating Analytical Vulnerabilities in NL2SQL

Shubham Mohole
Cornell University
sam588@cornell.edu

Sainyam Galhotra
Cornell University
sg@cs.cornell.edu

Abstract

Application systems using natural language interfaces to databases (NLIDBs) have democratized data analysis. This positive development has also brought forth an urgent challenge to help users who might use these systems without a background in statistical analysis to formulate bias-free analytical questions. Although significant research has focused on text-to-SQL generation accuracy, addressing cognitive biases in analytical questions remains underexplored. We present VeriMinder,[1], an interactive system for detecting and mitigating such analytical vulnerabilities. Our approach introduces three key innovations: (1) a contextual semantic mapping framework for biases relevant to specific analysis contexts (2) an analytical framework that operationalizes the Hard-to-Vary principle and guides users in systematic data analysis (3) an optimized LLM-powered system that generates high-quality, task-specific prompts using a structured process involving multiple candidates, critic feedback, and self-reflection.

User testing confirms the merits of our approach. In direct user experience evaluation, 82.5% participants reported positively impacting the quality of the analysis. In comparative evaluation, VeriMinder scored significantly higher than alternative approaches, at least 20% better when considered for metrics of the analysis's concreteness, comprehensiveness, and accuracy. Our system, implemented as a web application, is set to help users avoid "wrong question" vulnerability during data analysis. VeriMinder code base with prompts [2] is available as an MIT-licensed open-source software to facilitate further research and adoption within the community.

1 Introduction

Natural Language to SQL (NL2SQL) systems have emerged as a critical technology for democratizing data access, enabling non-technical users to query complex databases without specialized SQL knowledge. However, this positive development is not without significant risks. A technically perfect SQL query derived from a fundamentally flawed analytical question will yield misleading results. Systems like SQLPalm (Sun et al., 2023), SPLASH (Elgohary et al., 2020), and DAIL-SQL (Gao et al., 2023) focus on NL2SQL accuracy but do not consider the analytical quality of the user's original question.

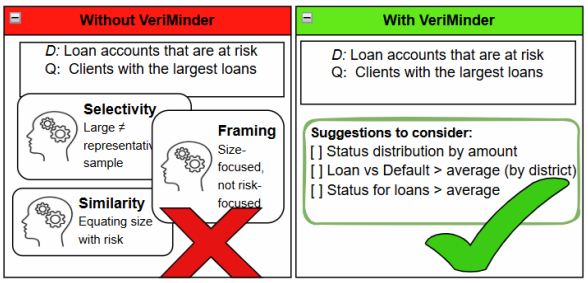

Figure 1: Example from experimental dataset showing VeriMinder mitigating biases via refinement suggestions

Consider this example shown in Figure 1: A financial analyst tasked to identify "loan accounts that are at risk" but asks for "clients with the largest loans." This query exhibits multiple cognitive biases: (1) *Similarity bias* - incorrectly assuming that "largest loans" and "at-risk loans" are similar categories, (2) *Framing bias* - framing the question around loan size rather than risk factors, completely changing what information will be retrieved, and (3) *Selection bias* - focusing only on large loans selects a non-representative subset of potentially risky accounts, as small loans may have higher default rates. While a state-of-the-art NL2SQL system can generate syntactically correct SQL for the original question, it cannot address these analytical blindspots, leaving a critical vulnerability unaddressed.

Research shows cognitive biases significantly impact professional decision-making across fields

[1] https://veriminder.ai
[2] https://reproducibility.link/veriminder

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 448–459

like medicine and laws (Berthet, 2022). The consistent association of these biases, such as anchoring and availability, with detrimental outcomes like health diagnostic inaccuracies underscores the critical need for mitigation systems like VeriMinder. As Peter Drucker said, "The most serious mistakes are not being made due to wrong answers. The truly dangerous thing is asking the wrong question." (Drucker, 1971).

Traditional approaches to mitigating such issues rely on static checklists (Lenders and Calders, 2025) or educational interventions (Thompson et al., 2023), which are challenging to implement consistently. While FISQL (Menon et al., 2025) and SPLASH (Elgohary et al., 2020) offer limited feedback mechanisms, they focus primarily on SQL refinement rather than addressing analytical quality issues (Qu et al., 2024).

To address these challenges, we present VeriMinder, which identifies and mitigates analytical vulnerabilities in NL2SQL workflows. Our interactive web application addresses these vulnerabilities with three innovations: (1) a semantic framework that systematically detects biases and blindspots in analytical questions; (2) a structured analytical process based on the "Hard-to-Vary" principle (Deutsch, 2011); and (3) an optimized LLM-driven refinement interface, integrated with NL2SQL workflows. VeriMinder integrates seamlessly with existing NL2SQL systems through simple configuration, supporting users of such systems with robust analytical question formulation alongside accurate SQL generation. Our evaluation demonstrates that VeriMinder significantly enhances analytical outcomes, outperforming baseline approaches across key analytical metrics.

2 System Architecture

VeriMinder operationalizes Deutsch's **Hard-to-Vary principle** (Deutsch, 2011) through a systematic architecture to identify and mitigate analytical vulnerabilities in user questions (Q), transforming potentially biased queries into robust analytical explanations (E) within a given domain (D) and decision context (C). This principle posits that good explanations are constrained, such that altering their components weakens the explanations or creates inconsistency. Applied to data analytics, a robust explanation E, often operationalized via SQL queries (S), is hard-to-vary if its components necessarily and cohesively address Q in context C, lacking arbitrary elements whose removal wouldn't degrade quality. Easily varied explanations, conversely, allow interchangeable components without specific roles, potentially leading to misleading results from flawed questions (e.g., analyzing broad expense categories instead of particular cost drivers while deciding on governmental cost-cutting measures). VeriMinder enforces this by ensuring the analysis pinpoints specific factors, yielding data-supported, falsifiable explanations that resist variation.

2.1 Core Modules and Architecture

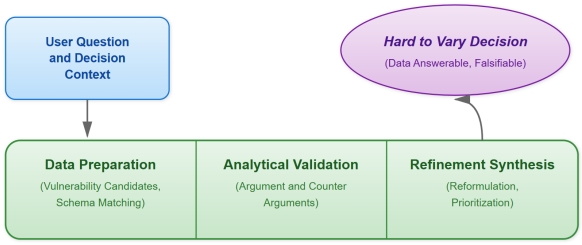

Figure 2: Three-stage framework operationalizing the Hard-to-Vary principle.

The VeriMinder system implements a systematic approach that helps analysts refine vulnerable questions into robust data analysis to operationalize the hard-to-vary principle. As shown in Figure 2, our architecture processes natural language questions through three sequential stages: Data Preparation, Analytical Validation, and Refinement Synthesis.

The system analyzes the question and decision context in the data preparation stage to identify potential analytical vulnerabilities and relevant schema elements. During Analytical Validation, vulnerabilities are detected, and structural analysis is performed using argument components and counter-argument testing to verify their significance. In Refinement Synthesis, the system generates targeted refinement suggestions that help with analysis aligned with a hard-to-vary approach for data-backed explanations for the particular decision context.

VeriMinder implements this framework using a modular service-based architecture (Figure 3) for flexibility, featuring five core services communicating via standardized interfaces: **Auth** (user provisioning/access, future enterprise plugins), **Suggestion** (implements core framework analytics), **NL2SQL** (extends the approach from (Qu et al., 2024) with metadata and dataset-specific distribution information and uses Gemini Flash 2.0

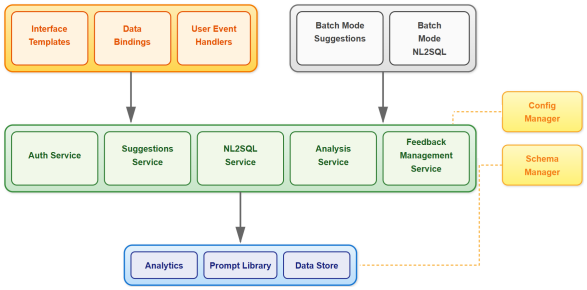

Figure 3: Modular architecture supporting scalability and flexible deployment modes

(Google DeepMind, 2025)), **Analysis** (compares initial vs. refined results for user reflection), and **User Feedback** (collects improvement data). The underlying analytical framework components (detailed in Appendix A.1) comprise 53 categorized cognitive biases (e.g., Memory, Statistical, Framing), data schema patterns (temporal, categorical, numerical detailed in Appendix A.2), the Toulmin model for argument structure evaluation (Toulmin, 1958) (Appendix A.3), and counter-argument frameworks (Greitemeyer, 2023) for questions that help address challenges and refine explanations (Appendix A.4).

For our system implementation, we developed an experimental NL2SQL component based on best practices for LLM-based text-to-SQL generation (Qu et al., 2024; Sun et al., 2023; Gao et al., 2023). VeriMinder is designed to complement existing NL2SQL systems rather than replace them, focusing on the orthogonal problem of analytical question formulation.

2.2 Prompt Formulation Method

VeriMinder offers users for their free-form analytical questions bias-mitigating alternatives through a three-stage workflow (Figure 4). The pipeline is driven by a formally defined hard-to-vary objective but is implemented with practical approximations that respect LLM limits and inference latency.

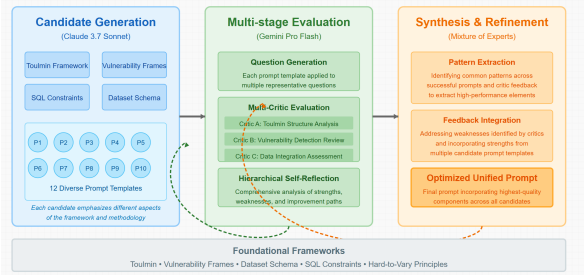

Figure 4: Multi-candidate prompt engineering pipeline with critic feedback and self-reflection.

2.2.1 Information-Theoretic Grounding

The architecture of VeriMinder is guided by a core principle: a robust analytical question should maximize predictive insight about a decision while minimizing its own descriptive complexity, subject to an interactive-latency budget. This section outlines the ideal theoretical framework that motivates our system's design (§2.2.2) and details its translation into a practical, multi-stage LLM pipeline (§2.2.3-2.2.6), concluding with a discussion of its scope and limitations (§2.2.7).

2.2.2 Idealized Theoretical Motivation

We formalize the principle of robust inquiry using the Hard-to-Vary (HV) score, a metric inspired by Deutsch's concept of good explanations (Deutsch, 2011) and the Minimum Description Length (MDL) principle (Rissanen, 1978; Grünwald, 2007). For a set of selected analytical variables, S, and a decision target, T, the HV score is:

$$HV(S) = \frac{I(T;S)}{DL(S)} \tag{1}$$

Here, $I(T;S)$ is mutual information (Cover and Thomas, 2006), and $DL(S)$ is the model's description length. This formulation, which extends normalized information metrics like the Information Gain Ratio (Quinlan, 1993), rewards explanatory density (high information per unit of complexity) and echoes the objective of Information Bottleneck theory (Tishby et al., 2000).

To verify this metric's behavior, we developed a numeric validation suite. As detailed in our code repository, experiments on synthetic Bayesian networks demonstrate the HV score's key properties under idealized conditions. All simulations use an exact mutual information computation and define complexity as the variable set cardinality, i.e., $DL(S) = |S|$. This provides empirical support that the HV score is a sound theoretical target.

2.2.3 Practical Heuristic Proxies

Directly optimizing Eq. 1 is computationally intractable even in structured feature spaces (Nguyen et al., 2014), and becomes exponentially more complex in the open-ended natural language domain where the search space includes all possible question formulations. VeriMinder therefore employs LLM-based *heuristic proxies* guided by the HV formula's intuition. We recognize this is not a formal equivalence; the desirable properties of the HV

score hold exactly only under the formal definition, while our proxies aim to approximate them empirically.

- **LLM Critic Scores for $I(T;S)$:** We use scores from specialized LLM critics as a proxy for information value. The rationale is that high-quality questions (judged on insight, logic, and bias mitigation) are more likely to reduce uncertainty about the decision target. This aligns with Information Foraging Theory (Pirolli and Card, 1995) and the use of LLMs as evaluators (Zheng et al., 2023; Dubois et al., 2024).

- Motivated by evidence that excessive prompt length can degrade LLM reasoning (Jiang et al., 2024), our prompt templates are built around a concise, analytical flow that goes from context analysis to final question selection, designed to produce a minimal set of high-impact questions. We therefore model task complexity through this structured analytical process rather than raw token count.

2.2.4 Stage 1: Ensemble-based Candidate Generation

To explore the analytical space, the system using generates a diverse set of candidates using twelve prompt templates. These templates are themselves the output of an automated meta-level prompt engineering process based on Claude 3.7 Sonnet model (Anthropic, 2025) selected for its intelligence category rank (Artificial Analysis, 2025)), ensuring each targets a distinct analytical angle (e.g., vulnerability detection, schema validation). This ensemble method ensures broad coverage, a technique well-grounded in machine learning for both bagging (Breiman, 1996) and modern LLM prompting (Zhou et al., 2023).

2.2.5 Stage 2: Distributed Critic Evaluation

Generated candidates are evaluated by a panel of three specialized LLM critics (based on the Claude 3.7 Sonnet model). For efficiency, a random subset of **two** critics evaluates each candidate. This implements distributed evaluation analogous to boosting, where a committee of weak learners forms a robust judgment (Schapire, 1990). This aligns with modern methods using self-consistency and multi-agent consensus to improve LLM evaluation (Wang et al., 2023; Li et al., 2024b).

2.2.6 Stage 3: Critic Feedback and Self Reflection

Finally, the system performs a single self-reflection pass that improves prompts using critic feedback. This mirrors self-refinement techniques that improve LLM performance (Madaan et al., 2023; Shinn et al., 2023). At present we execute only one iteration but multiple self-reflection rounds would be a possible natural extension to the current pipeline.

2.2.7 Scope and Limitations

Our approach has three main limitations. First, our production system relies on heuristic search, unlike the exhaustive search in our validation suite. Second, critic scores and our analytical flow stages are pragmatic surrogates, not formal equivalents, for $I(T;S)$ and $DL(S)$. Finally, our current cost model is limited to response structure and does not yet incorporate computational latency.

2.3 Interactive User Interface

VeriMinder's user interface (Figure 5) employs a progressive disclosure pattern for a guided workflow: users provide their questions and context, the system executes the query while analyzing vulnerabilities, suggests refinements for user selection, presents a side-by-side comparison of results, and explains detected issues and fixes. To enhance user experience during intensive computations, server-sent events (SSE) provide streaming updates and educational insights. The system features a pluggable interface and unified abstraction layer to support multiple database types, utilizing SQLite (with the BIRD-DEV benchmark (Li et al., 2023)) for execution and MySQL for tracking application state.

3 Experiments

3.1 Experimental Setup

To comprehensively evaluate VeriMinder, we designed a multi-step assessment framework addressing key research questions: (1) How effective is the VeriMinder solution in improving the analysis using the NL2SQL interface? (2) How does our approach compare with alternative methods for enhancing analytical quality on key accuracy, concreteness, and comprehensiveness metrics (Zhu et al., 2024b)?

The evaluation dataset was derived from the BIRD-DEV benchmark questions. To create realistic decision contexts, we manually crafted the

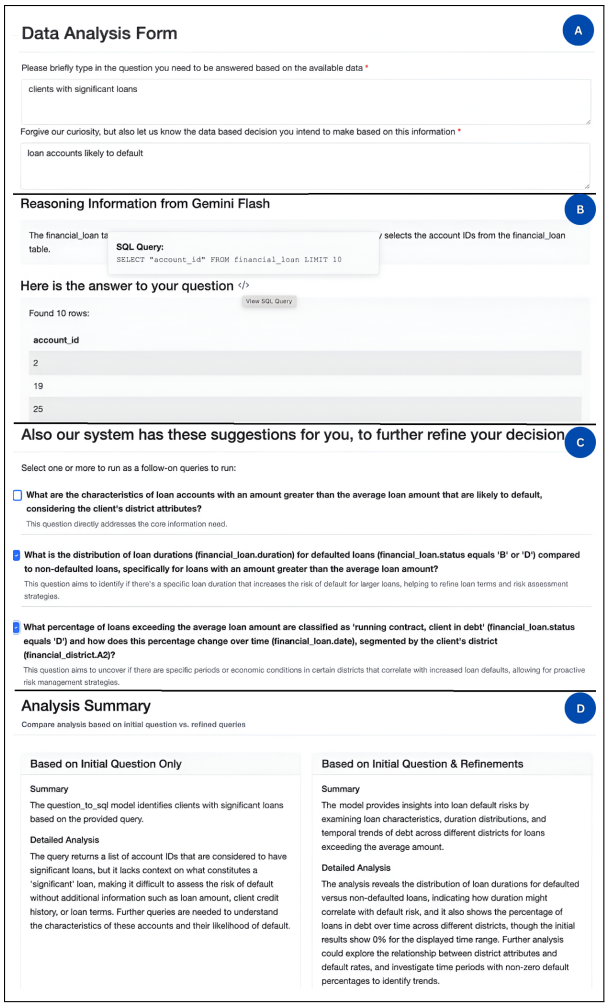

Figure 5: VeriMinder's user interface workflow: (A) Initial Question, (B) Query Results, (C) Refinement Suggestions, (D) Comparative Analysis

164 decision scenarios following the Case Study Method (Ellet, 2007), ensuring balanced coverage of choice, evaluation, and diagnosis types. Data analytics experts designed these scenarios to represent contexts where analytical vulnerabilities could significantly impact outcomes. We employed TF-IDF vectorization to match each decision with the most semantically relevant question from BIRD-DEV, creating a bipartite relationship. The final decision text was lightly edited for grammar and sentence structure to ensure consistency during the user study without altering the analytical focus of the decision contexts. This methodical approach yielded 164 question-decision pairs, divided into three subsets: 64 pairs (DS1) for human evaluations and 100 pairs (DS2) for automated assessment. An additional smaller subset DS1-T1 of 36 pairs was created from the DS1. All splits were done randomly.

To our knowledge, no direct comparable system focuses on refining user-posed questions and addressing biases and blind spots. So in addition to the **Direct NL2SQL** (standard text-to-SQL generation without analytical enhancements), we evaluated VeriMinder by operationalizing three alternative approaches that the research community has considered for either bias mitigation or holistic analysis: **Decision-Focused Query Generation** (generating questions directly from decision context (Zhang et al., 2025)), **Question Perturbation (PerQS)** (creating variations of the original question (Zhu et al., 2024a)), and **Critic-Agent Feedback (CAF)** (implementing a critic agent providing feedback (Li et al., 2024a)). We use the same LLM (Gemini Flash 2.0) for all baselines as VeriMinder and plan to release them as part of our code release.

A critical aspect of our evaluation methodology was ensuring consistent SQL generation across all compared systems. To isolate the effect of analytical question formulation (our focus) on NL2SQL accuracy, we implemented the same experimental NL2SQL component for all baseline systems and VeriMinder. For our evaluations, we validated that all generated SQL queries executed correctly before assessment, allowing us to focus purely on analytical quality rather than technical SQL correctness.

3.2 User Experience Evaluation

We conducted an interactive user study with the DS1-T1 dataset, recruiting 63 participants from Prolific (Prolific, 2025) with diverse backgrounds. For 30 scenarios, we received submissions from two users each, and for three scenarios, from one user (a total of 63 unique participants). Appendix B1 shows the feedback form presented to participants. The overall effectiveness of our solution in improving analysis quality received 82.5% positive ratings (score of 4 or 5), with Gwet's AC1 of 0.766. Suggestion effectiveness received 74.6% positive ratings, with Gwet's AC1 0.670. Rationale clarity had 66.7% (Gwet's AC1 0.479) and Scenario realism 61.9% (Gwet's AC1 0.457) positive ratings. The reliability scores, particularly for clarity and realism, likely reflect the diverse user base from Prolific. Furthermore, the scenario realism scores may be influenced by the experimental setup, where decision contexts were constrained by matching them to the existing BIRD-DEV dataset questions.

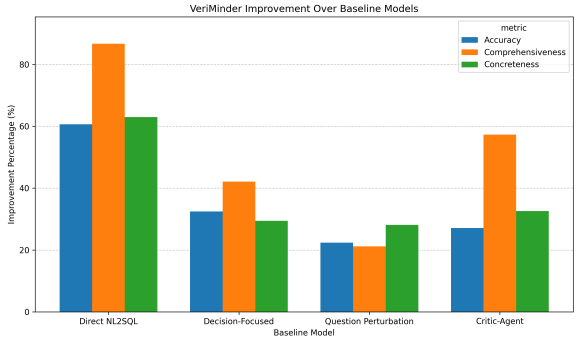

Figure 6: Percentage improvement of VeriMinder over baseline systems on key analytical dimensions

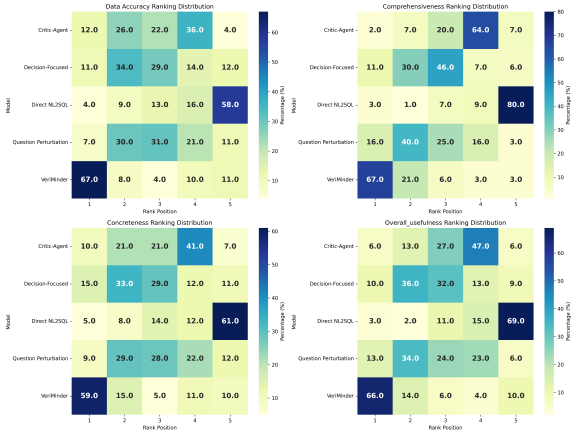

Figure 7: Ranking distribution across analytical dimensions; VeriMinder consistently achieves highest rankings

3.3 Comparative System Evaluation

From the DS1 dataset, we conducted a comparative evaluation of generated analysis questions with one data analyst from each of the two US-based software companies who responded to our request. Appendix B.2 shows the screenshot of the interface these data analyst users used to rate the comparative strength of analysis questions in a decision context. As with the previous test, we only included the successful completions in our analysis (because of an unrelated system outage issue, we failed to get submissions for five entries). For the 59 scenarios, we received submissions from both users. VeriMinder demonstrated strong performance across all dimensions: Accuracy (mean=7.87/10, 95% CI [7.57, 8.18]), Concreteness (mean=7.79/10, 95% CI [7.47, 8.10]), and Comprehensiveness (mean=8.05/10, 95% CI [7.74, 8.36]).

Figure 6 illustrates VeriMinder's percentage improvement over each baseline system. The most substantial improvements were observed against Direct NL2SQL, with gains of 60.4% in Accuracy, 63.2% in Concreteness, and 86.9% in Comprehensiveness. Even against the strongest baseline (Question Perturbation), VeriMinder showed improvements of 22.1% in Accuracy, 28.4% in Concreteness, and 21.2% in Comprehensiveness.

Statistical analysis confirmed these improvements were significant ($p < 0.001$) with paired t_test across all dimensions and baseline comparisons. Win rates further illustrated VeriMinder's quality, outperforming Direct NL2SQL in 83.9% of Accuracy comparisons, 86.4% of Concreteness comparisons, and 97.5% of Comprehensiveness comparisons. Inter-rater reliability metrics based on the model ranks demonstrated robust agreement in our evaluations, with Gwet's AC1 coefficients of 0.941 for Accuracy, 0.960 for Concreteness, and 0.862 for Comprehensiveness.

3.4 Large-Scale Automated Evaluation

We employed an LLM-based evaluator for dataset DS2 (100 scenarios) (Gemini Flash 2.0). With known limitations of LLM for quantitative scoring (OpenAI et al., 2024; Bubeck et al., 2023) but better performance in verbal analysis and relative ranking (Zheng et al., 2023; Gilardi et al., 2023), our test focused on LLM skills in text comprehension and comparative qualitative assessments. In Appendix B.3, we discuss our approach to the prompt design. For LLM-based evaluation, we first calibrated our automated evaluator (based on Gemini 2.0 Flash) against human judgments on comparative ranking on a subset of 15 examples from DS1, finding a m (Pearson's $r = 0.74, p < 0.001$) that provided us confidence in the automated results.

As Figure 7 shows, VeriMinder consistently achieved the highest first-place rankings: 67.0% for Data Accuracy, 67.0% for Comprehensiveness, 59.0% for Concreteness, and 66.0% for Overall Usefulness. In contrast, Direct NL2SQL received the most last-place rankings across all metrics, highlighting the importance of analytical enhancement beyond raw SQL generation.

3.5 Analysis of Bias Mitigation Effectiveness

The word cloud visualization in Figure 8 highlights VeriMinder's key analytical capabilities as identified through qualitative analysis of LLM response. This visualization was generated through automated content analysis of refinement suggestions across the dataset. As shown in Figure 8,

Figure 8: Key analytical capabilities driving cognitive bias mitigation in VeriMinder

comparative analysis, pattern recognition, and relationship exploration emerge as key capabilities, enabling VeriMinder to mitigate cognitive biases.

3.6 Limitations

Several limitations should be noted. First, deployment in specific domains may require customization of the analytical components. Second, the system's effectiveness depends on the underlying NL2SQL engine quality, implemented here as a simplified service module. We evaluated VeriMinder primarily on BIRD-DEV, which LLMs may have seen during training, raising concerns about information leakage and overestimated SQL success rates on truly unseen databases. The interface is desktop-optimized without accessibility testing. Before general release, critical enhancements include mobile support, accessibility features, multi-query handling, and validation on previously unseen databases to confirm generalization capabilities.

4 Related Work

Our work builds upon research across cognitive bias mitigation, natural language database interfaces, and LLM reasoning techniques in non-ground truth regimes - analytical contexts where there is no single 'correct' answer but varying degrees of analytical quality based on comprehensiveness, accuracy and alignment with decision objectives. Prior work in cognitive bias mitigation has examined biases in data-driven contexts (Kahneman, 2011; Tversky and Kahneman, 1974; Sumita et al., 2024; Ke et al., 2024), but primarily focused on bias awareness rather than active mitigation within analytical workflows. Benchmarks like Spider 2 (Lei et al., 2025) have driven recent advancements in NL2SQL generation (Deng et al., 2025; Wang and Liu, 2025), with LLM-based systems achieving high execution accuracy. However, these systems primarily address technical SQL issues rather than analytical vulnerabilities.

While VeriMinder primarily focuses on analytical question formulation, our evaluation employs a simplified NL2SQL service. This service incorporates metadata and dataset-specific distribution information for SQL generation within our setup, drawing inspiration from recent work on mitigating NL2SQL hallucinations, such as the Task Alignment strategy proposed by (Qu et al., 2024) and LLM based tabular learning tasks enhanced through (Mohole and Galhotra, 2025) columnar statistics for datasets. LLM prompting techniques, including response selection (Zhao et al., 2025), have enhanced reasoning capabilities but might not be suitable for a non-ground truth regime that requires an interactive experience. With our principled approach, inspired by Deutsch's framework (Deutsch, 2011), and a multi-candidate refinement process, we provide a lightweight yet systematic framework for optimizing LLM response for downstream NL2SQL and analysis tasks.

5 Future Work and Conclusion

While VeriMinder currently targets NL2SQL interactions, its analytical core is modality-agnostic, enabling future extensions to Python/pandas code generation for statistical exploration. Building on Self-RAG (Asai et al., 2023), we plan to evolve our self-reflection phase into a multi-head, bias-aware rubric outputting calibrated probabilities for evidence sufficiency, cognitive-bias flags, and statistical validity. These probabilities will both steer an adaptive retriever-generator loop and serve as bias-aware non-conformity scores for Conformal LM (Quach et al., 2024), enabling rejection thresholds that preserve coverage while reducing bias. Our Information-Theoretic Framework extends naturally to this calibration focus—by maximizing $\mathrm{HV}(S)$ over reflection head outputs, Information theory guided pruning could guarantee minimal causal sufficiency while keeping calibration lean.

With VeriMinder, we've presented an end-to-end system for mitigating analytical vulnerabilities in NL queries. By operationalizing the "hard-to-vary" explanations we demonstrated its effectiveness for the NL2SQL use cases. Coupled with SELF-RAG principles and bias-aware Conformal prediction, this research can open avenues for NLIDBs that provide answers not only *probably correct* but also *unbiased and grounded in evidence*.

6 Broader Impact Statement

While VeriMinder addresses analytical vulnerabilities, key limitations, and ethical points remain:

Analytical Guidance vs. Guarantee The system offers guidance, not guarantees, enhancing but not replacing user critical thinking. Vulnerability detection may not be exhaustive.

Commercial API Dependencies Reliance on commercial LLMs limits accessibility; future work should explore open-source alternatives.

Cultural and Domain Biases The bias taxonomy is primarily Western-based and may need domain-specific or cultural adaptation.

Potential for Misuse Analytical enhancement tools could be misused; governance frameworks are needed to ensure integrity.

Augmentation vs. Automation VeriMinder augments human analysis, preserving user agency rather than fully automating the process.

We believe addressing analytical vulnerabilities is vital as data access is democratized. VeriMinder is an initial step aiming to inspire further research at the intersection of cognitive science, data analytics, and NLP.

References

Anthropic. 2025. Claude 3.7 sonnet. Accessed June 11, 2025.

Artificial Analysis. 2025. Ai model & api providers analysis. https://artificialanalysis.ai. Accessed: June 10, 2025.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *Preprint*, arXiv:2310.11511.

Vincent Berthet. 2022. The impact of cognitive biases on professionals' decision-making: A review of four occupational areas. *Frontiers in Psychology*, 12:802439.

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24(2):123–140.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *Preprint*, arXiv:2303.12712.

Jean-Paul Caverni, Jean-Marc Fabre, and Michel Gonzalez. 1990. *Cognitive biases*. Advances in psychology. North Holland.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of information theory*, 2nd edition. Wiley-Interscience.

Minghang Deng, Ashwin Ramachandran, Canwen Xu, Lanxiang Hu, Zhewei Yao, Anupam Datta, and Hao Zhang. 2025. Reforce: A text-to-sql agent with self-refinement, format restriction, and column exploration. *Preprint*, arXiv:2502.00675.

David Deutsch. 2011. *The Beginning of Infinity: Explanations that Transform the World*. Penguin UK.

Evanthia Dimara, Steven Franconeri, Catherine Plaisant, Anastasia Bezerianos, and Pierre Dragicevic. 2020. A task-based taxonomy of cognitive biases for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(2):1413–1432.

Peter F. Drucker. 1971. *Men, Ideas, and Politics*. Harper & Row, New York.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Preprint*, arXiv:2305.14387.

Joyce Ehrlinger, Wilson Readinger, and Bora Kim. 2016. Decision-making and cognitive biases. In Howard S. Friedman, editor, *Encyclopedia of mental health*, 2 edition, pages 5–12. Academic Press, Oxford.

Ahmed Elgohary, Saghar Hosseini, and Ahmed Hassan Awadallah. 2020. Speak to your parser: Interactive text-to-SQL with natural language feedback. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2065–2077, Online. Association for Computational Linguistics.

William Ellet. 2007. *The Case Study Handbook: How to Read, Discuss, and Write Persuasively About Cases*. Harvard Business Review Press, Boston, Massachusetts. Accessed: March 26, 2025.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2023. Text-to-sql empowered by large language models: A benchmark evaluation. *Preprint*, arXiv:2308.15363.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30).

Google DeepMind. 2025. Gemini: A family of highly capable multimodal models. Accessed: 2025-03-26.

Tobias Greitemeyer. 2023. Counter explanation and consider the opposite: Do corrective strategies reduce biased assimilation and attitude polarization in the context of the COVID-19 pandemic? *Journal of Applied Social Psychology*, 53(5):306–322.

Peter D. Grünwald. 2007. *The Minimum Description Length Principle*. MIT Press.

Martin Hilbert. 2012. Toward a synthesis of cognitive biases: How noisy information processing can bias human decision making. *Psychology Bulletin*, 138(2):211–237.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *Preprint*, arXiv:2310.06839.

Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

Yuhe Ke, Rui Yang, Sui An Lie, Taylor Xin Yi Lim, Yilin Ning, Irene Li, Hairil Rizal Abdullah, Daniel Shu Wei Ting, and Nan Liu. 2024. Mitigating cognitive biases in clinical decision-making through multi-agent conversations using large language models: Simulation study. *Journal of Medical Internet Research*, 26:e59439.

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. 2025. Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows. *Preprint*, arXiv:2411.07763.

Daphne Lenders and Toon Calders. 2025. Users' needs in interactive bias auditing tools introducing a requirement checklist and evaluating existing tools. *AI Ethics*, 5:341–369.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. 2023. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Preprint*, arXiv:2305.03111.

Michael Y. Li, Vivek Vajipey, Noah D. Goodman, and Emily B. Fox. 2024a. Critical: Critic automation with language models. *Preprint*, arXiv:2411.06590.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024b. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Bangkok, Thailand. Association for Computational Linguistics. ArXiv:2406.11776.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Rakesh R. Menon, Kun Qian, Liqun Chen, Ishika Joshi, Daniel Pandyan, Jordyn Harrison, Shashank Srivastava, and Yunyao Li. 2025. Fisql: Enhancing text-to-sql systems with rich interactive feedback. In *Proceedings of the 2025 International Conference on Extending Database Technology (EDBT)*, pages 1032–1038.

Shubham Mohole and Sainyam Galhotra. 2025. Sifotl: A principled, statistically-informed fidelity-optimization method for tabular learning. KDD'25 (UMC).

Xuan Vinh Nguyen, Jeffrey Chan, Simone Romano, and James Bailey. 2014. Effective global approaches for mutual information based feature selection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*, pages 512–521, New York, NY, USA. Association for Computing Machinery.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Eoin D. O'Sullivan and Susie J. Schofield. 2019. A cognitive forcing tool to mitigate cognitive bias – a randomised control trial. *BMC Medical Education*, 19(12).

Peter Pirolli and Stuart Card. 1995. Information foraging in information access environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 51–58, USA. ACM Press/Addison-Wesley Publishing Co.

Prolific. 2025. Prolific. Web-based platform. Accessed: 2025-03-26.

Ge Qu, Jinyang Li, Bowen Li, Bowen Qin, Nan Huo, Chenhao Ma, and Reynold Cheng. 2024. Before generation, align it! a novel and effective strategy for mitigating hallucinations in text-to-sql generation. *Preprint*, arXiv:2405.15307.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. 2024. Conformal language modeling. *Preprint*, arXiv:2306.10193.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471.

Robert E. Schapire. 1990. The strength of weak learnability. *Machine learning*, 5(2):197–227.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Michael Soprano, Kevin Roitero, David La Barbera, Davide Ceolin, Damiano Spina, Gianluca Demartini, and Stefano Mizzaro. 2024. Cognitive biases in fact-checking and their countermeasures: A review. *Information Processing & Management*, 61(3):103672.

Yasuaki Sumita, Koh Takeuchi, and Hisashi Kashima. 2024. Cognitive biases in large language models: A survey and mitigation experiments. *Preprint*, arXiv:2412.00323.

Ruoxi Sun, Sercan Ö. Arik, Alex Muzio, Lesly Miculicich, Satya Gundabathula, Pengcheng Yin, Hanjun Dai, Hootan Nakhost, Rajarishi Sinha, Zifeng Wang, and Tomas Pfister. 2023. Sql-palm: Improved large language model adaptation for text-to-sql (extended). *arXiv preprint arXiv:2306.00739*.

John Thompson, Helena Bujalka, Stephen McKeever, Adrienne Lipscomb, Sonya Moore, Nicole Hill, Sharon Kinney, Kwang Meng Cham, Joanne Martin, Patrick Bowers, and Marie Gerdtz. 2023. Educational strategies in the health professions to mitigate cognitive and implicit bias impact on decision making: a scoping review. *BMC Medical Education*, 23(455).

Naftali Tishby, Fernando C. Pereira, and William Bialek. 2000. The information bottleneck method. *Preprint*, arXiv:physics/0004057.

Stephen E. Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Yihan Wang and Peiyu Liu. 2025. Linkalign: Scalable schema linking for real-world large-scale multi-database text-to-sql. *Preprint*, arXiv:2503.18596.

Yueheng Zhang, Xiaoyuan Liu, Yiyou Sun, Atheer Alharbi, Hend Alzahrani, Basel Alomair, and Dawn Song. 2025. Can llms design good questions based on context? *Preprint*, arXiv:2501.03491.

Eric Zhao, Pranjal Awasthi, and Sreenivas Gollapudi. 2025. Sample, scrutinize and scale: Effective inference-time search by scaling verification. *Preprint*, arXiv:2502.01839.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. *Preprint*, arXiv:2211.01910.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, and Xing Xie. 2024a. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, LAMPS '24, page 57–68, New York, NY, USA. Association for Computing Machinery.

Zining Zhu, Haoming Jiang, Jingfeng Yang, Sreyashi Nag, Chao Zhang, Jie Huang, Yifan Gao, Frank Rudzicz, and Bing Yin. 2024b. Situated natural language explanations. *Preprint*, arXiv:2308.14115.

Appendix A Analytical Framework Components

Our framework integrates four complementary analytical perspectives via an optimized LLM prompt to identify and mitigate vulnerabilities (biases, data mismatches, logical flaws, framing issues) in natural language queries before SQL generation.

A.1 Cognitive Biases Framework

Incorporates 53 cognitive biases relevant to data analysis (Soprano et al., 2024; Dimara et al., 2020; Hilbert, 2012; Caverni et al., 1990; Ehrlinger et al., 2016), mapping NL query patterns to potential reasoning pitfalls. Categories include:

1. Memory Biases (8): Hindsight, Imaginability, Recall, Search, Similarity, Testimony, False Memory, Availability.

2. Statistical Biases (9): Base Rate Neglect, Chance, Conjunction, Correlation, Disjunction, Sample Size Neglect, Subset Bias, Gambler's Fallacy, Probability Neglect.

3. Confidence Biases (8): Completeness Illusion, Illusion of Control, Confirmation Bias, Desire Bias, Overconfidence, Redundancy Illusion, Dunning-Kruger Effect, Bias Blind Spot.

4. Methodological Biases (12): Data Quality Neglect, Multiple Testing Fallacy, Selection Bias, Method Fixation, Tool Overconfidence, Selectivity, Success/Self-Serving Bias, Test Inability, Anchoring, Conservatism, Reference Dependence, Regression to Mean.

5. Framing & Contextual Biases (16): Framing Effect, Linear Assumption, Mode Influence, Order Effect, Scale Distortion, Primacy Effect, Recency Effect, Granularity Illusion, Attenuation Bias, Complexity Avoidance, Escalation of Commitment, Habit, Inconsistency, Rule Adherence, Fundamental Attribution Error, Bandwagon Effect.

A.2 Data Schema Patterns

Examines NL query alignment with data types. Key NL2SQL considerations: **Temporal:** Handling date/time formats (e.g., 'DATEPART'), consistent aggregation.

1. **Categorical:** Resolving ambiguity (e.g., 'LA' vs 'Los Angeles'), implicit hierarchies.
2. **Numerical:** Interpreting average/median correctly (e.g., 'AVG'), handling outliers.
3. **Relationship:** Inferring 'JOIN' paths, verifying functional dependencies (e.g., city $\rightarrow$ zip).
4. **Data Quality:** Assessing missing data ('NULL', 'COALESCE'), inconsistencies (e.g., negative counts).
5. **Transformation:** Needs for normalization (per capita), discretization ('CASE WHEN'), aggregation ('GROUP BY').

A.3 Toulmin Argument Structure

Evaluates the implicit argument in the NL query/SQL based on Toulmin's model (Toulmin, 1958):

1. **Claim Clarity/Relevance:** Does SQL capture NL assertion and align with context? ('SELECT', 'WHERE').
2. **Evidence Sufficiency/Validity:** Enough reliable data retrieved? ('COUNT', 'LEFT JOIN'). Trustworthy sources?
3. **Warrant Validity/Applicability:** Is NL-to-SQL logic sound? Respects constraints? (CTEs, domain checks).
4. **Backing:** Logic supported by standard practices/definitions?.
5. **Qualifier Precision/Scope:** Acknowledges limits (confidence, scope 'WHERE', rounding)?.
6. **Rebuttal Considerations:** Alternative queries, interpretations ('JOIN' confounders), exceptions ('EXCLUDE')?.

A.4 Counter-Argument Frameworks

Systematically challenges the NL query/formulation for analytical rigor:

1. **Conclusion Rebutters:** Scope limitation needed? Alternative queries yield different conclusions?
2. **Premise Rebutters:** Relies on inaccurate/incomplete ('IS NULL')/non-representative data? Metric appropriate?
3. **Argument Undercutters:** Hidden assumptions questionable? Alternative explanations (confounders via 'JOIN')?
4. **Framing Challenges:** Right question for the problem? Neglects perspectives/temporal frames? Aggregation level suitable?
5. **Implementation Challenges:** Feasibility issues or unintended consequences suggested by data?

Appendix B Experimental Setup Details

B.1 Interactive User Study Questionnaire

We designed an intuitive questionnaire to assess user experience with VeriMinder across four key dimensions: scenario realism, suggestion effectiveness, rationale clarity, and impact on analysis. Users rated each dimension on a 5-point Likert scale. Figure 9 shows the feedback form used in our interactive study.

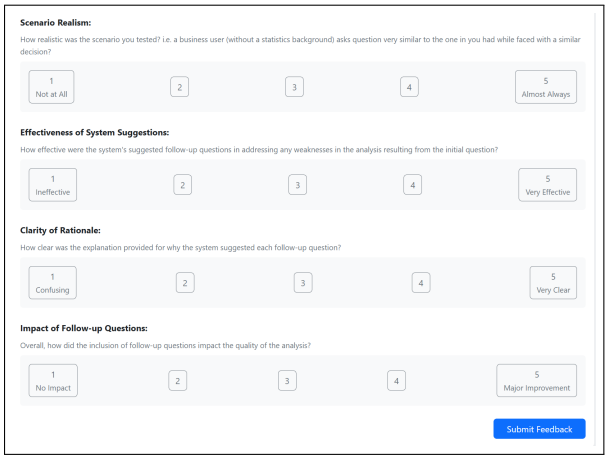

Figure 9: Interactive user study feedback interface

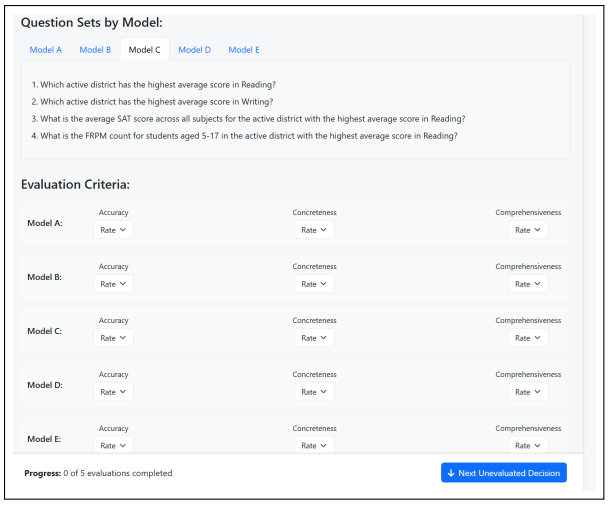

Figure 10: Comparative evaluation interface for assessing analytical quality across methods

B.2 Comparative System Evaluation

The comparative evaluation required participants to rate all five systems (VeriMinder, Direct NL2SQL, Decision-Focused Query Generation, Question Perturbation, and Critic-Agent Feedback - with names anonymized during the testing) on three analytical dimensions: accuracy, concreteness, and comprehensiveness. Participants rated each dimension on a 10-point scale for each system, allowing for direct comparison. Figure 10 shows the evaluation interface.

B.3 Automated Evaluation Procedure

1. **Goal:** To assess the analytical quality of query sets generated by VeriMinder and four baseline systems against the large-scale dataset (100 pairs).
2. **Methodology:** Employed an LLM evaluator (Gemini Flash 2.0) (Google DeepMind, 2025) using a structured prompt that included:
   (a) The decision context and original NL question.
   (b) Database schema snippets and relevant evidence context.
   (c) The complete set of successfully executed SQL query results generated by *each* of the five systems (VeriMinder, Direct NL2SQL, Decision-Focused, PerQS, CAF) for the given decision scenario. Our choice of LLM was primarily driven by the response time (Artificial Analysis, 2025) and streaming support dictated by our user interface requirements.
3. **Evaluation Task:** The LLM was instructed to:
   (a) Holistically evaluate each system's *entire set* of queries and results in the decision context.
   (b) Assess each system based on **Data Accuracy - Fidelity of Fetched Results to NL Question Intent**, **Comprehensiveness**, **Concreteness**, and **Overall Usefulness** in the context of the decision goal.
   (c) Apply the **SLOW** framework (**S**ure, **L**ook, **O**pposite, **W**orst) (O'Sullivan and Schofield, 2019) to identify uncertainties, missing information, alternative interpretations, and potential problematic conclusions for each system's output and the combined analysis.
4. **Output:** The process yielded structured evaluations for each system and a comparative assessment, including relative rankings across the specified analytical dimensions.