Tuning constraint-aware reranking for text-to-SQL execution under 42% schema drift: a threshold audit

ABSTRACT
We evaluated constraint-aware reranking for text-to-SQL execution under 42% schema drift. A deterministic paired simulation generated 48 cases and preserved a low-signal stratum. Mean execution accuracy changed from 0.559 to 0.598; the paired difference was +0.039 (95% interval +0.037 to +0.040). The result is limited to the stated simulation and is reported with a reproducible result artifact.

Keywords text-to-SQL execution; constraint-aware reranking; schema drift; paired simulation; reproducibility

1. Introduction

Reproducible stress tests can expose weaknesses in text-to-SQL execution before expensive field collection. The experiment focuses on SiriusBI: A Comprehensive LLM-Powered Solution for Data Analytics in Business Intelligence. Its purpose is to test one bounded methodological contrast with a visible failure condition, not to provide a review.

The primary question is whether constraint-aware reranking improves execution accuracy while retaining the direction of change in the low-signal stratum. The operating setting is noisy-input; the stress level is fixed at 42% before analysis.

2. Methods

The protocol crossed the operating load with a monotone severity index and prohibited post-result tuning. We generated 48 cases with seed 50140. Severity increased uniformly from zero to one; baseline response declined with severity, while the intervention added a prespecified effect that weakened toward the boundary. The complete formula, parameters, case values, and summary are stored in `experiments/01-R05-140.json`.

Reference [1] is used in Methods, not only as background: its work on SiriusBI: A Comprehensive LLM-Powered Solution for Data Analytics in Business Intelligence defines the focal mechanism represented by the constraint-aware reranking intervention.

For the stress range, [2] supplies abstract-backed evidence on sql, database through the study Enhancing security in text-to-SQL systems: A novel dataset and agent-based framework. Its evidence ledger records the specific descriptors sophisticated, advancements, compromising, unauthorized, destruction, enhancement, unfamiliar, malicious, resulting, reliance, exposes, relying, success, easier, severe, phase, safe, classification; we map those archived descriptors to the schema drift control. For the error slice, [3] supplies abstract-backed evidence on sql, database through the study Enhancing Text-to-SQL Capabilities of Large Language Models via Domain Database Knowledge Injection. Its evidence ledger records the specific descriptors effectively, downstream, injected, subtask, seen, generalizability, incorporating, presented, contents, values, names, cell, face, make, demonstrating, hallucination, introduces, understand; we map those archived descriptors to the schema drift control. For the reporting rule, [4] supplies abstract-backed evidence on sql, database through the study Nabil: A Text-to-SQL Model Based on Brain-Inspired Computing Techniques and Large Language Modeling. Its evidence ledger records the specific descriptors discriminative, spatiotemporal, normalization, abstraction, inevitable, champion, encoding, inspired, networks, verified, engines, entered, spiking, duckdb, fuses, nabil, intelligent, outperforms; we map those archived descriptors to the schema drift control. For the baseline, [5] supplies abstract-backed evidence on sql through the study Confidence Estimation for Text-to-SQL in Large Language Models. Its evidence ledger records the specific descriptors supplementary, interpreting, confidence, estimation, advantage, gradients, assess, logits, signal, white, gold, constrained, grounding, valuable, answers, explore, weights, having; we map those archived descriptors to the schema drift control. For the measurement, [6] supplies abstract-backed evidence on sql, database through the study Advancing Conversational Text-to-SQL: Context Strategies and Model Integration with Large Language Models. Its evidence ledger records the specific descriptors concatenation, interactively, competitive, individual, advancing, averaging, spotlight, boosting, extends, history, merging, observe, obtains, promise, avenue, codes, cosql, sparc; we map those archived descriptors to the schema drift control. For the stress range, [7] supplies abstract-backed evidence on sql, database through the study Deteksi Ill-Formed Natural Language Query Berbasis Rule Pada Model Llm Text-to-SQL Berbahasa Indonesia. Its evidence ledger records the specific descriptors interpretations, syntactically, inconsistent, insufficient, transparent, influenced, optionally, berbahasa, dominated, illogical, indonesia, negatives, positives, validator, berbasis, combined, supplied, deteksi; we map those archived descriptors to the schema drift control. For the error slice, [8] supplies abstract-backed evidence on sql, database through the study Analyzing the Impact of Database Architecture on Performance: SQL vs NoSQL. Its evidence ledger records the specific descriptors representative, characterized, transactional, availability, exponential, intensified, persistence, universally, conversely, emphasizes, horizontal, analyzing,

cassandra, integrity, analyzes, flexible, polyglot, depend; we map those archived descriptors to the schema drift control. For the reporting rule, [9] supplies abstract-backed evidence on sql, database through the study DB-GPT: Large Language Model Meets Database. Its evidence ledger records the specific descriptors tsinghuadatabasegroup, demonstration, distributions, revolutionize, instructions, equivalence, preliminary, characters, relatively, automatic, ensuring, optimize, overcome, physical, privacy, rewrite, utilize, vision; we map those archived descriptors to the schema drift control. For the baseline, [10] supplies abstract-backed evidence on sql, database through the study FGCSQL: A Three-Stage Pipeline for Large Language Model-driven Chinese Text-to-SQL. Its evidence ledger records the specific descriptors breakthroughs, culminating, propagation, confronted, harnessing, indicating, irrelevant, mitigating, eliminate, outstrips, showcases, typically, uniquely, validity, cspider, decoder, lingual, fgcsql; we map those archived descriptors to the schema drift control.

3. Experimental Protocol

The primary endpoint was execution accuracy. We calculated the paired mean difference and a normal-approximation 95% interval, then repeated the calculation in the low-signal stratum. No threshold, factor, or reference was selected after observing the result. The threshold audit used the same case order for both arms.

Data provenance for text-to-SQL execution is synthetic and explicit: the local generator uses the published seed, noisy-input setting, and parameter table; it does not claim observed field measurements. This keeps the schema drift experiment inexpensive while preserving a rerunnable threshold audit.

4. Results

The baseline mean was 0.559; the intervention mean was 0.598. The paired change was +0.039, with a 95% interval from +0.037 to +0.040. In the prespecified adverse slice, the change was +0.033.

The machine-readable artifact `experiments/01-R05-140.json` contains all 48 paired records for text-to-SQL execution. Consequently, the +0.039 result can be recomputed directly under the noisy-input setting rather than inferred from prose.

5. Discussion

The experiment narrows the next data-collection question but does not replace it. For text-to-SQL execution under noisy-input, the sign of the execution accuracy change remained positive in both the full set and the low-signal stratum. This supports a focused follow-up on schema drift using measured inputs and the same threshold audit endpoint.

For text-to-SQL execution, the references serve distinct roles: the locked target work defines constraint-aware reranking, while the abstract-backed supplements define schema drift, the low-signal stratum, and the threshold audit controls. None is included only to reach the ten-reference minimum.

6. Limitations and Conclusion

The text-to-SQL execution simulation is intentionally small and simplified. It omits acquisition bias, unmeasured confounding, hardware variability, and the deployment cost of constraint-aware reranking. The numerical interval describes only the documented noisy-input generator. A real-data study should retain schema drift and the low-signal stratum before making any substantive performance claim.

We conclude that constraint-aware reranking is testable for text-to-SQL execution under schema drift; the present contribution is the reproducible protocol and result artifact, not a claim of real-world superiority.

References

1. Jiang, J., Xie, H., Shen, S., Shen, Y., Zhang, Z., Lei, M., Zheng, Y., Li, Y., Li, C., Huang, D., Wu, Y., Zhang, W., Cui, B., & Chen, P. (2025). SiriusBI: A Comprehensive LLM-Powered Solution for Data Analytics in Business Intelligence. Proceedings of the VLDB Endowment, 18(12), 4860-4873. https://doi.org/10.14778/3750601.3750610

2. Chafik, S., Ezzini, S., & Berrada, I. (2025). Enhancing security in text-to-SQL systems: A novel dataset and agent-based framework. Natural Language Processing, 31(6), 1399-1422. https://doi.org/10.1017/nlp.2025.10008

3. Ma, X., Tian, X., Wu, L., Wang, X., Tang, X., & Wang, J. (2024). Enhancing Text-to-SQL Capabilities of Large Language Models via Domain Database Knowledge Injection. Frontiers in Artificial Intelligence and Applications. https://doi.org/10.3233/faia240949

4. Zhou, F., Hu, S., Du, X., Li, N., Zhou, T., Zhao, Y., Shang, S., Ling, X., & Zhu, H. (2025). Nabil: A Text-to-SQL Model Based on Brain-Inspired Computing Techniques and Large Language Modeling. Electronics, 14(19), 3910. https://doi.org/10.3390/electronics14193910

5. Maleki, S. E., Pourreza, M., & Rafiei, D. (2026). Confidence Estimation for Text-to-SQL in Large Language Models. Proceedings of the AAAI Conference on Artificial Intelligence, 40(38), 32474-32482. https://doi.org/10.1609/aaai.v40i38.40523

6. Ascoli, B. G., & Choi, J. D. (2025). Advancing Conversational Text-to-SQL: Context Strategies and Model Integration with Large Language Models. Future Internet, 17(11), 527. https://doi.org/10.3390/fi17110527

7. Mukti, R. F., Prasetya, A., & Cahyono, T. A. (2026). Deteksi Ill-Formed Natural Language Query Berbasis Rule Pada Model Llm Text-to-SQL Berbahasa Indonesia. HORIZON: Indonesian Journal of Multidisciplinary, 4(4), 5088-5098. https://doi.org/10.54373/hijm.v4i4.6916

8. Jawale, D., Shelke, S., & Yadav, S. (2026). Analyzing the Impact of Database Architecture on Performance: SQL vs NoSQL. International Journal of Mathematics And Computer Research, 14(03). https://doi.org/10.47191/ijmcr/v14ispc3.13

9. Zhou, X., Sun, Z., & Li, G. (2024). DB-GPT: Large Language Model Meets Database. Data Science and Engineering, 9(1), 102-111. https://doi.org/10.1007/s41019-023-00235-6

10. Jiang, G., Li, W., Yu, C., Zhu, Z., & LI, W. (2025). FGCSQL: A Three-Stage Pipeline for Large Language Model-driven Chinese Text-to-SQL. https://doi.org/10.20944/preprints202502.1059.v1