Calibrating constraint-aware reranking for text-to-SQL execution under 37% schema drift: a error-stratified estimate

ABSTRACT
We evaluated constraint-aware reranking for text-to-SQL execution under 37% schema drift. A deterministic paired simulation generated 56 cases and preserved a rare-condition slice. Mean execution accuracy changed from 0.499 to 0.539; the paired difference was +0.040 (95% interval +0.038 to +0.043). 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

Performance averages can obscure whether schema drift changes the reliability of text-to-SQL execution. The experiment focuses on SiriusDeliver: Automating Data Warehouse Delivery at Tencent. 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 rare-condition slice. The operating setting is sparse-observation; the stress level is fixed at 37% before analysis.

2. Methods

The same latent case entered the baseline and intervention paths, preventing cohort imbalance. We generated 56 cases with seed 50041. 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-R05-041.json`.

Reference [1] is used in Methods, not only as background: its work on SiriusDeliver: Automating Data Warehouse Delivery at Tencent defines the focal mechanism represented by the constraint-aware reranking intervention.

For the stress range, [2] supplies abstract-backed evidence on sql through the study SQL/PGQ data model and graph schema. Its evidence ledger records the specific descriptors describing, documents, property, neo4j, mapping, graph, three, schema; 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 Enhanced Financial Text-to-SQL Generation via Fine-Grained SQL Refinement. Its evidence ledger records the specific descriptors substantially, degradation, incorporate, refinement, alignment, diversity, necessity, agnostic, captures, dialects, implicit, reflects, dialect, grained, jointly, largely, adapt, logic; 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 Combining Text-to-SQL and Large Language Models for Maintenance Decision Support. Its evidence ledger records the specific descriptors deterministically, communication, inappropriate, establishing, transparency, contributes, suggestions, artificial, documented, historical, verifiable, vocabulary, personnel, replicate, traceable, ablation, bridging, combines; we map those archived descriptors to the schema drift control. For the baseline, [5] 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 measurement, [6] supplies abstract-backed evidence on sql, database through the study A COMPARATIVE STUDY OF LLM - POWERED DATABASE INTERFACES VERSUS TRADITIONAL SQL SYSTEMS FOR INVENTORY MANAGEMENT. Its evidence ledger records the specific descriptors deterministic, unprecedented, prioritizing, emonstrate, guaranteed, llmpowered, sqlalchemy, identical, inventory, penalties, averaged, backends, degraded, parallel, mission, slower, versus, incur; 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 Pengembangan Model Migrasi Database Relational ke NoSQL Memanfaatkan Metadata SQL. Its evidence ledger records the specific descriptors dikembangkan, memanfaatkan, transformasi, berdasarkan, penyimpanan, terstruktur, diperlukan, diterapkan, eksperimen, mengajukan, pendekatan, perbendaan, dilakukan, kecepatan, kelemahan, melakuakn, menyimpan, merupakan; 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 A comparative evaluation of large language models for detecting SQL injection vulnerabilities in web applications. Its evidence ledger records the specific descriptors vulnerabilities, confidential, intractable, potentially, circumvent, considered, manipulate, popularity, codellama, detecting, prominent, technique, emerging, payloads, produced, stronger, boolean, created; 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 Improving Text-to-Sql Conversion for Low-Resource Languages Using Large Language Models. Its evidence ledger records the specific descriptors configurations, augmentation, introducing, translating, contribute, emirozturk, previously, exceeding, languages, publicly, scripts, turkish, before, llama2, llama3, tt2sql, widely, total; we map those archived descriptors to the schema drift control. For the baseline, [10] supplies abstract-backed evidence on database through the study ChIP-GPT: a managed large language model for robust data extraction from biomedical database records. Its evidence ledger records the specific descriptors immunoprecipitation, comprehensively, identification, fundamentally, characterize, emphasizing, iteratively, customized, predefined, seamlessly, adaptable, chromatin, hindering, amassing, centered, medicine, reliably, archive; 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 rare-condition slice. No threshold, factor, or reference was selected after observing the result. The error-stratified estimate 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, sparse-observation setting, and parameter table; it does not claim observed field measurements. This keeps the schema drift experiment inexpensive while preserving a rerunnable error-stratified estimate.

4. Results

The baseline mean was 0.499; the intervention mean was 0.539. The paired change was +0.040, with a 95% interval from +0.038 to +0.043. In the prespecified adverse slice, the change was +0.034.

The machine-readable artifact `experiments/01-R05-041.json` contains all 56 paired records for text-to-SQL execution. Consequently, the +0.040 result can be recomputed directly under the sparse-observation setting rather than inferred from prose.

5. Discussion

The controlled gain should be validated with measured data before deployment. For text-to-SQL execution under sparse-observation, the sign of the execution accuracy change remained positive in both the full set and the rare-condition slice. This supports a focused follow-up on schema drift using measured inputs and the same error-stratified estimate 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 rare-condition slice, and the error-stratified estimate 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 sparse-observation generator. A real-data study should retain schema drift and the rare-condition slice 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. Xie, H., Zhou, X., Yang, J., Shen, S., Wang, Z., Zheng, Y., Xu, T., Shi, Y., Zong, Z., Li, Y., Chen, P., Jiang, J., He, D., Yan, X., & Jiang, J. (2026). SiriusDeliver: Automating Data Warehouse Delivery at Tencent. arXiv. https://doi.org/10.48550/arXiv.2608.09185

2. Furniss, P., & Green, A. (2023). SQL/PGQ data model and graph schema. https://doi.org/10.54285/ldbc.qzsk3559

3. Wang, Y., Chen, Y., Chen, R., Shu, H., Liu, P., & Xu, W. (2026). Enhanced Financial Text-to-SQL Generation via Fine-Grained SQL Refinement. https://doi.org/10.2139/ssrn.6502095

4. Beckmann, S., Wiesner, K., Tebruegge, C., & Grum, M. (2026). Combining Text-to-SQL and Large Language Models for Maintenance Decision Support. https://doi.org/10.2139/ssrn.7103027

5. 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

6. Guo, M., & Sun, Y. (2026). A COMPARATIVE STUDY OF LLM - POWERED DATABASE INTERFACES VERSUS TRADITIONAL SQL SYSTEMS FOR INVENTORY MANAGEMENT. NLP & Text Mining, 11-21. https://doi.org/10.5121/csit.2026.160602

7. Utomo, M. N. Y. (2020). Pengembangan Model Migrasi Database Relational ke NoSQL Memanfaatkan Metadata SQL. Jurnal Teknologi Elekterika, 4(2), 1. https://doi.org/10.31963/elekterika.v4i2.2212

8. Hairan, B., & Şahman, M. A. (2026). A comparative evaluation of large language models for detecting SQL injection vulnerabilities in web applications. PeerJ Computer Science, 12, e4015. https://doi.org/10.7717/peerj-cs.4015

9. Öztürk, E. (2025). Improving Text-to-Sql Conversion for Low-Resource Languages Using Large Language Models. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 14(1), 163-178. https://doi.org/10.17798/bitlisfen.1561298

10. Cinquin, O. (2024). ChIP-GPT: a managed large language model for robust data extraction from biomedical database records. Briefings in Bioinformatics, 25(2), bbad535. https://doi.org/10.1093/bib/bbad535