# Improving Text2Cypher with Confidence-Based Test-Time Strategies

**Rima Dessí[1*], Makbule Gulcin Ozsoy[2*]**
[1*] Higher Colleges of Technology, Sharjah, UAE
[2*] Neo4j, London, UK
rdessi@hct.ac.ae, makbule.ozsoy@neo4j.com

**Abstract**

Advances in Large Language Models (LLMs) have made it possible to convert natural language questions into executable database queries. Text2Cypher focuses on graph databases, converting user questions into queries and providing natural language access to graph-structured data. While significant progress has been made through prompt design, fine-tuning, and iterative refinement, less attention has been given to adaptive test-time strategies that combine multiple generated outputs. In this work, we investigate the impact of confidence-based test-time strategies specifically on the Text2Cypher task by evaluating the model's traces, which are the sequence of tokens generated during the construction of the query. We show that reasoning models generate diverse query candidates but frequently produce syntactic errors and incomplete structures, limiting executability. On the other hand, instruction-tuned models yield more reliable outputs but lack sufficient diversity for effective confidence-based selection. Further, by tuning diversity parameters such as top-p and temperature, we observe consistent improvements in both query accuracy and execution success. Experiments across multiple instruction-tuned models confirm that combining diversity-controlled generation with confidence-aware inference provides a practical, model-agnostic method for improving query generation.

**Keywords:** Natural Language to Query, Text2Cypher, Confidence-Based Inference, Test-Time Strategies

## 1. Introduction

Due to the rapid growth of interconnected data, graph databases such as Neo4j have become essential for managing complex relational data across various applications, such as recommender systems and knowledge graph exploration (Napoli et al., 2025; Senington et al., 2025). Unlike relational databases which organize data in tabular form, graph databases aim to store real-world entities and their relations. These data structures are typically explored by using graph query languages, among which Cypher is widely adopted, particularly in the Neo4j system (Cerjan et al., 2024). This query language enables users to traverse and manipulate such data. Therefore, developing natural language interfaces for these systems has become an increasingly important research direction to over come the limitation of specific technical expertise.

For this objective, Large Language Models (LLMs) have provided a powerful foundation as they have demonstrated exceptional capabilities across a wide range of natural language processing tasks, including question answering, complex reasoning, and code generation (Yang et al., 2026b; Sobo et al., 2025). These advances have enabled natural language interfaces for structured databases, and led to growing interest in tasks focusing on translating natural language questions into executable database queries, such as Text2SQL, Text2SPARQL, and Text2Cypher (Zhu et al., 2025; Sennrich and Ahmadi, 2025; Ozsoy et al., 2025). Specifically, in the domain of graph databases, Text2Cypher bridges the gap between unstructured user inquiries and executable graph queries. This allows users to navigate and retrieve insights from complex, graph-structured data without requiring knowledge of formal query syntax.

Prior work in translating natural language to formal query languages has explored a wide range of strategies to improve structured query generation, including prompt engineering, semantic schema representations, model fine-tuning, and iterative refinement pipelines. (Zhu et al., 2025; Sennrich and Ahmadi, 2025; Ozsoy et al., 2025; Bunkova et al., 2026; Mandilara et al., 2025; Yang et al., 2026a). These methods primarily modify model inputs, representations, or generation procedures. Fewer studies have addressed test-time inference strategies that adaptively control or aggregate output traces, i.e., the sequence of intermediate tokens generated during the construction of the query, to improve output quality (Wei et al., 2022).

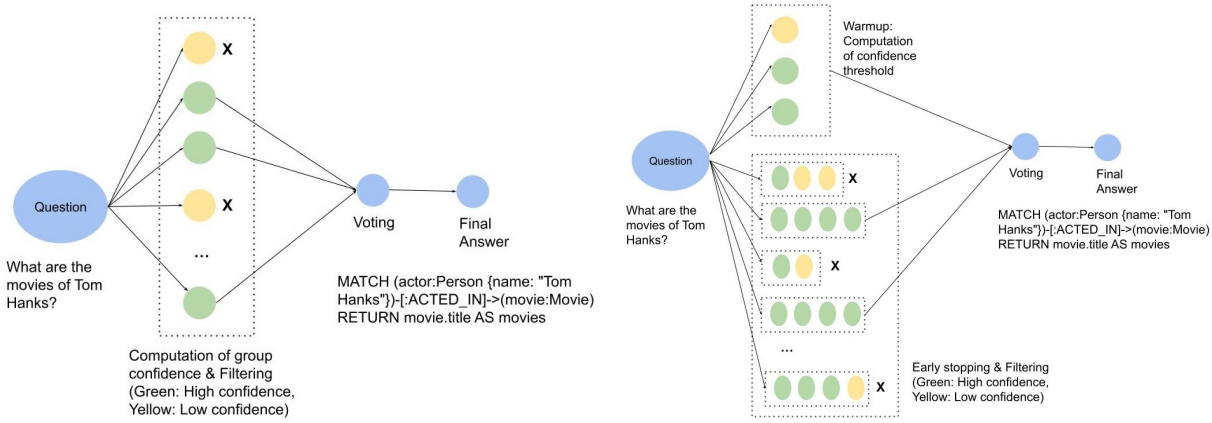
Recent studies have demonstrated that sampling multiple traces and aggregating them, namely self-consistency or parallel thinking techniques (Wang et al., 2023), can improve the output accuracy. Furthermore, a recent work, namely DeepConf (Fu et al., 2025), has shown that confidence-aware mechanism for filtering low-quality traces during or after generation enables more efficient and accurate outputs.

In this work, we investigate the adaptation of confidence-based test-time strategies to the

* Authors contributed equally.



*Proceedings of the Knowledge Graphs and Large Language Models Workshop 2026 (KG-LLM@LREC26)*, pages 45–52
May 16, 2026. 




(a) Offline approach

(b) Online approach

Figure 1: Confidence-aware strategies with offline and online approaches

Text2Cypher task. By leveraging a confidence-aware inference framework (Fu et al., 2025), we explore the impact of monitoring internal model certainty on the reliability of query generation in complex graph database settings (see Figure 1).

Our main contributions in this work are as follows:

- We systematically assess confidence-based test-time strategies (Fu et al., 2025) for Text2Cypher, showing that monitoring model certainty significantly enhances the reliability of generated graph queries.

  ´We demonstrate that while reasoning-focused models provide diverse traces, they struggle with syntactic accuracy in Text2Cypher tasks; conversely, we show that instruction-tuned models offer the structural reliability necessary for executable query generation.

- Instruction-tuned models often generate limited diversity, reducing the framework's effectiveness. By tuning output diversity via parameters like top-p and temperature, we observed that combining instruction-tuned models improved performance up to 0.08 in ROUGE-L score and 0.19 in execution success ratio.

- In order to validate the generality of our approach, we conducted additional experiments using another instruction-tuned model, namely Qwen2.5-7B-Instruct. Experiments confirmed that diversity tuning consistently enhanced Text2Cypher performance.

The paper is organized as follows: Section 2 reviews related work. Section 3 details how confidence-based test-time strategies applied to the Text2Cypher task. Section 4 outlines our experimental setup and presents the evaluation results. Section 5 concludes the paper.

## 2. Related Work

There has been extensive research on mapping natural language to query languages such as SQL and SPARQL (Qin et al., 2022; Katsogiannis-Meimarakis and Koutrika, 2023). This task is formally known as *semantic parsing*, which aims translating natural language into machine-executable logical forms, such as structured database queries. Early approaches primarily relied on rule-based methods and neural encoder–decoder architectures to generate relational database queries (Lin et al., 2020). With the emergence of large language models (LLMs), several studies have demonstrated improved performance in query generation tasks (Gao et al., 2024; Zhu et al., 2025). More recently, research has extended this paradigm to property graph databases, where Text2Cypher (Ozsoy et al., 2025; Mandilara et al., 2025; Yang et al., 2026a) focuses on addressing the distinct structural characteristics of graph data models. However, applying these techniques directly to property graph query languages such as Cypher introduces unique challenges, such as graph schema constraints, path-based logic means, and thus remains largely unexplored.

In addition, the existing LLM-based query generation approaches mainly focus on improving the accuracy and the syntactic correctness of generated queries; however, they fail to perform confidence-based selection (Hong et al., 2025). There have been a several studies propose different approaches to quantify the reliability of the generated outputs by the LLMs by exploiting internal confidence signals, self-consistency, and other uncertainty estimation techniques (Fu et al., 2025; Kadavath et al., 2022; Kuhn et al., 2023). These studies, however, mainly utilize reasoning-optimized LLMs and their performance is evaluated on reasoning-centric tasks. In the context of query

generation, their performance has not been sufficiently explored. Especially for the Text2Cypher task, measuring the reliability of generated Cypher queries in real-world settings remains unexplored.

Overall, while substantial research has demonstrated the efficacy of LLMs in generating structured queries especially for Text2SQL and Text2SPARQL, the Text2Cypher domain remains comparatively less explored. Further, the mentioned studies for quantifying the reliability of the generated output focus on reasoning tasks and applying them to the Text2Cypher task remain an open challenge. Therefore, in this study, we bridge this gap by implementing a confidence-based trace filtering mechanism by adapting (Fu et al., 2025) and tailoring it for the Text2Cypher task (see Section 3).

## 3. Confidence-Based Text2Cypher

**Problem Formulation.**Given a natural language question $q$ and a graph database schema $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$, where $\mathcal{V}, \mathcal{E}$, and $\mathcal{P}$ denote the sets of node labels, relationship types, and properties respectively, the Text2Cypher task aims to generate an executable query $c$. Let $\mathcal{G}$ be a graph database instance conforming to schema $\mathcal{S}$.The objective is to define a mapping function $f : (q, \mathcal{S}) \rightarrow c$, where $c$ belongs to the space of all syntactically valid Cypher queries $\mathcal{C}$. To be considered successful, the generated query $c$ must satisfy the following conditions: (i) **Syntactic Validity:** $c$ must adhere to the formal grammar and syntax of the Cypher query language. (ii) **Schema Consistency:** All labels, types, and properties referenced in $c$ must be strictly contained within $\mathcal{S}$. (iii) **Semantic Alignment:** The execution of $c$ over the graph instance $\mathcal{G}$ must return a result set that accurately satisfies the information intent of $q$.

Achieving these conditions for complex queries remains a significant challenge for standard Large Language Models due to structural hallucinations. Therefore, to address these limitations, we leverage a confidence-based inference paradigm by adapting the DeepConf framework (Fu et al., 2025) (see Section 3.1) to the Text2Cypher task. This approach allows to evaluate the model's internal certainty to ensure the final output is both syntactically and semantically accurate. This paradigm acts as a decision layer that aligns stochastic model reasoning with the formal constraints of the Cypher language.

### 3.1. Confidence-Based Inference Paradigm

Deep Think with Confidence (DeepConf) (Fu et al., 2025) improves the efficiency and accuracy of large language model outputs by utilizing confidence-aware test-time reasoning. It uses mechanisms to filter or aggregate low-quality traces either after generation (offline mode) or during generation (online mode) (See Figure 1).

**Offline Mode:** In this mode, multiple candidate reasoning traces are first generated according to a budget (e.g., 5 traces). For each trace, confidence scores are computed using internal metrics, such as group confidence. Low-confidence traces are filtered, optionally, and the remaining traces are aggregated using confidence-weights or voting. This approach emphasizes high-confidence traces without modifying the underlying model or requiring additional training.

**Online Mode:** Online mode evaluates confidence during token generation and applies early stopping to reduce computation. In this mode, first an offline warm-up stage is executed to estimate a confidence threshold. During generation, traces whose confidence falls below this threshold are terminated early. The remaining traces are aggregated to produce final outputs, improving accuracy while reducing the number of tokens generated.

### 3.2. Task-Specific Adaptation and Refinement

In this work, we adapt DeepConf (Fu et al., 2025) for the Text2Cypher task. Our contributions include: (i) preprocessing traces to remove superficial differences, (ii) increasing output diversity, including parameter tuning, random seed patching, and prompt diversification, and (iii) applying and comparing base, offline, and online inference modes.

**Trace Preprocessing and Cleaning:** Many traces for the Text2Cypher task differ only in formatting, such as whitespace or quotation marks (single vs. double quotes). For instance, the traces `MATCH (n:Movie) WHERE n.title = "Inception"` and `MATCH (n:Movie) WHERE n.title = 'Inception'` differ only in quotation marks, but if not processed, they are treated as distinct outputs. To reduce redundancy and improve aggregation, we post-process the generated traces by collapsing such variations. This ensures that confidence-based filtering focuses on meaningful differences between outputs rather than minor formatting changes.

**Diversity Tuning:** Instruction-tuned models often generate outputs with limited variability, which can reduce the effectiveness of confidence-based aggregation. In order to increase diversity, we adjust parameters such as top-p, top-k, and temperature, and also patch the DeepConf implementation to use random seeds for each trace instead of sequential seeds. This produces more varied traces for aggregation. Additionally, small variations are added to the standard Text2Cypher prompts, en-

| Diversity Level | Parameters |
|---|---|
| Light | temperature=0.2, top_p=0.95, top_k=20 |
| Moderate | temperature=0.9, top_p=0.99, top_k=60 |
| High | temperature=1.2, top_p=0.999, top_k=80 |

Table 1: Diversity parameter settings used for instruction-tuned models in the Text2Cypher task

couraging alternative query structures, variable names, or formatting. Each trace uses a different diversification hint in a round-robin manner. Furthermore, we experiment with three levels of diversity (light, moderate, and high) to study their impact on DeepConf performance (see Table 1).

**Inference Modes:** We evaluate models using three inference modes: (i) Base: the model as is, without DeepConf; (ii) Offline: DeepConf applied in offline mode; and (iii) Online: DeepConf applied in online mode. This setup allows us to explore the effect of confidence-aware strategies on the quality of generated Cypher queries.

## 4. Experiments

This section describes our experimental setup and presents evaluation results.

### 4.1. Experimental Setup

**Data:** We conducted our experiments using the publicly available Text2Cypher dataset (Ozsoy et al., 2025), which consists of cleaned instances drawn from multiple sources and databases. Due to limited computational resources, we focused on a subset of the dataset, specifically previously identified hard examples (Ozsoy, 2025). In their analysis, hard examples were defined based on prior analysis of model performance across data sources and databases. Authors showed that samples associated with three demonstration databases, namely "recommendations", "companies" and "neoflix", are more challenging compared to other instances. By selecting these challenging instances, our experimental subset is composed of 789 test samples, which allows us to efficiently evaluate model performance on difficult cases.

**Models:** All experiments were performed on the test set using foundational models. For the analysis, we primarily used a range of models including reasoning and instruction-tuned variants, namely 'deepseek-ai/DeepSeek-R1-Distill-Qwen-7B', 'google/gemma-2-9b-it' and 'Qwen/Qwen2.5-7B-Instruct' models. Evaluations were conducted using vLLM (Kwon et al., 2023), and an additional post-processing step was applied to the generated outputs to remove unwanted text, such as redundant 'cypher:' prefixes, ensuring clean query outputs for evaluation.

**Evaluation Metrics:** We employed two evaluation procedures to measure model performance: *(i) Translation-Based (Lexical) Evaluation:* This method compares generated Cypher queries with ground-truth queries at the textual level. *(ii) Execution-Based Evaluation:* This method executes both the generated and ground-truth Cypher queries on the target databases and compares their outputs stored as string and sorted lexicographically. In addition, it reports execution statistics, including syntax and runtime error ratios.

In order to compute these evaluation metrics, we used the Hugging Face Evaluate library (HuggingFace, 2024). Execution error statistics were collected separately during query execution. We report the ROUGE-L score and Error statistics as the primary evaluation metrics.

### 4.2. Evaluation Results

We evaluate the applicability of DeepConf (Fu et al., 2025) to the Text2Cypher task using four complementary analyses: (i) comparing reasoning-focused and instruction-tuned models, (ii) examining instruction-tuned models with diversity tuning, (iii) experimenting with an additional instruction-tuned models to assess generality and (iv) analyzing computational costs.

**Comparison of Reasoning and Instruction–Tuned Model Results** DeepConf (Fu et al., 2025) was originally proposed to utilize reasoning traces and apply confidence-aware filtering based on local confidence estimates to improve generation accuracy. Following this approach, we first evaluated a reasoning model, DeepSeek-R1-Distill-Qwen-7B, using both the offline and online variants of DeepConf for Cypher query generation. As shown in Table 2 and confirmed through manual inspection, the reasoning model produces highly noisy outputs for the Text2Cypher task, with a high rate of syntax and runtime errors

https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B

https://huggingface.co/google/gemma-2-9b-it

https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

| Model type | Inference Mode | ROUGE-L | Execution Success Ratio |
|---|---|---|---|
| Reasoning model | Base | 0.4025 | 0.1267 |
| | Online | 0.4919 | 0.1622 |
| | Offline | 0.4888 | 0.1445 |
| Instruction-tuned model | Base | 0.7004 | 0.8200 |
| | Online | 0.7060 | 0.8276 |
| | Offline | 0.7095 | 0.8416 |

Table 2: Comparison of reasoning (DeepSeek-R1-Distill-Qwen-7B) and instruction-tuned (Gemma2-9B-it) models on Text2Cypher, showing ROUGE-L scores for lexical evaluation and execution success rates for execution-based evaluation.

| Model | Diversity Level | Inference Mode | ROUGE-L (lexical) | ROUGE-L (exec) | Execution Success Ratio |
|---|---|---|---|---|---|
| Gemma-2-9b-it | Light | Base | 0.7004 | 0.2343 | 0.8200 |
| | | Online | $0.7060^{+0.56}$ | $0.2399^{+0.56}$ | $0.8276^{+0.76}$ |
| | | Offline | $0.7095^{+0.91}$ | $0.2427^{+0.84}$ | $0.8416^{+2.16}$ |
| Gemma-2-9b-it | Moderate | Base | 0.6817 | 0.1982 | 0.7769 |
| | | Online | $0.7162^{+3.45}$ | $0.2411^{+4.29}$ | $0.8530^{+7.61}$ |
| | | Offline | $0.7081^{+2.64}$ | $0.2375^{+3.93}$ | $0.8162^{+3.93}$ |
| Gemma-2-9b-it | High | Base | 0.6286 | 0.1680 | 0.6388 |
| | | Online | $0.7099^{+8.13}$ | $0.2396^{+7.16}$ | $0.8365^{+19.77}$ |
| | | Offline | $0.6948^{+6.62}$ | $0.2224^{+5.44}$ | $0.7503^{+11.15}$ |
| Qwen2.5-7B-Instruct | Moderate | Base | 0.6898 | 0.1917 | 0.7098 |
| | | Online | $0.7125^{+2.27}$ | $0.2189^{+2.72}$ | $0.7833^{+7.35}$ |
| | | Offline | $0.7202^{+3.04}$ | $0.2391^{+4.74}$ | $0.7896^{+7.98}$ |

Table 3: Comparison of instruction-tuned models (Gemma-2-9b-it and Qwen2.5-7B-Instruct) across different diversity levels. Performance improvements over the base model are indicated as superscripts, scaled to percentage points.

(up to 88%) as well as incomplete query structures. Typical failures include malformed queries such as `MATCH (n:Movie) WHERE n.revenue > n.budget`, which is missing a `RETURN` clause, and `MATCH (m:Movie {bornIn: 'France'}), (p:Person {bornIn: 'France'}) WHERE m.born = p.born IN m)-[:ACTED_IN]->p RETURN m ORDER BY m.born LIMIT 3`, which contains syntactial and semantics errors.

Prior work on Text2Cypher (Ozsoy et al., 2025; Mandilara et al., 2025; Yang et al., 2026a) mainly adopts instruction-tuned models for structured query generation. Following this direction, we use an instruction-tuned model, Gemma-2-9b-it, and compare it with a reasoning model under the DeepConf framework in terms of syntactic validity and overall performance. As shown in Table 2, the instruction-tuned model produces Cypher queries with substantially fewer syntax and runtime errors (below 18%). However, the performance gains from DeepConf are more limited compared to the reasoning model. While DeepConf improves the ROUGE-L score of the reasoning model from 0.4025 to 0.4888, the instruction-tuned model only increases from 0.7004 to 0.7095.

This difference suggests that DeepConf is particularly effective at filtering the low-quality or malformed outputs of reasoning models. In contrast, instruction-tuned models already generate relatively clean outputs, leaving less room for improvement through confidence-aware aggregation. These observations motivate exploring diversity tuning for instruction-tuned models to further enhance the effectiveness of confidence-aware test-time strategies.

**Instruction-Tuned Models with Diversity Tuning** We explored the effect of output diversity on the performance of DeepConf for the Text2Cypher task. Table 3 shows the performance results of the instruction-tuned model, namely Gemma-2-9b-it, at three diversity levels (See Section 3).

| Model | Inference Mode | Avg. Tokens per Instance | Avg. Time per Instance (s) |
|---|---|---|---|
| Gemma-2-9b-it | Base | 48.57 | 2.207 |
| | Online | 2207.49 | 23.337 |
| | Offline | 1286.33 | 15.124 |
| Qwen2.5-7B-Instruct | Base | 50.28 | 1.720 |
| | Online | 2057.69 | 15.883 |
| | Offline | 1826.26 | 11.469 |

Table 4: Comparison of computational cost for moderately diverse instruction-tuned models (Gemma-2-9B-it and Qwen2.5-7B-Instruct) across different inference modes: base, online, and offline.

| User Question | Inference Mode | Generated query |
|---|---|---|
| List the movies released in the year the user "Omar Huffman" was born. | Base | MATCH (u:User name:"Omar Huffman")<-[:RATED]-(m:Movie) RETURN m |
| | Online | MATCH (u:User name:"Omar Huffman")<-[:RATED]-(m:Movie) WHERE m.released = u.born RETURN m |
| Find the movies that have been released in the same year that a user rated a movie. | Base | MATCH (m:Movie)-[:RATED]->(u:User)<-[:RATED]-(m2:Movie) WHERE m.released = m2.released RETURN m |
| | Offline | MATCH (u:User)-[:RATED]->(m1:Movie) MATCH (m2:Movie) WHERE m1.year = m2.year RETURN m2 |

Table 5: Illustrative examples of improved Text2Cypher outputs using confidence-based inference.

We observe that increasing diversity enhances the impact of DeepConf. For example, with high diversity, the ROUGE-L score improves from 0.6286 (Base) to 0.7099 (Online), an absolute increase of 0.0813. Similarly, the execution success ratio increases from 0.6388 to 0.8365, an absolute improvement of 0.1977. In contrast, for lower diversity settings, improvements are smaller. For example, with light diversity, ROUGE-L increases from 0.7004 to 0.7095 and the execution success ratio from 0.8200 to 0.8416.

These results show that higher diversity allows DeepConf to more effectively filter or aggregate outputs. Additionally, the online variant of DeepConf tends to produce larger gains than the offline variant in higher diversity settings, highlighting the benefit of adaptive confidence-based selection.

Table 5 illustrates that confidence-based inference improves Text2Cypher outputs. In the examples, the approach corrects semantic errors present in the base queries. For example, adding the appropriate "WHERE" clauses to filter by the birth year or matching movies rated in the same year. In some cases, it also fixes minor syntactic issues, such as relationship directions, resulting in queries that are both syntactically valid and semantically correct.

**Generality Across Instruction-Tuned Models** We validate the generality of our findings by testing an additional instruction-tuned model, namely Qwen2.5-7B-Instruct. This allows us to check whether the observed benefits hold across different model architectures. Using the moderately diverse setup, we report the performance of both Gemma-2-9b-it and Qwen2.5-7B-Instruct models with moderate level of diversity in Table 3.

The results show that applying DeepConf consistently improves Text2Cypher performance for multiple instruction-tuned models. For Qwen2.5-7B-Instruct, ROUGE-L increases from 0.6898 (Base) to 0.7125 (Online) and to 0.7202 (Offline), and the execution success ratio rises from 0.7098 (Base) to 0.7833 (Online) and to 0.7896 (Offline). These trends are similar to those observed for Gemma-2-9b-it, with online DeepConf often providing slightly larger gains than offline. Overall, this confirms that confidence-aware test-time strategies are effective across multiple instruction-tuned architectures.

**Computational Cost Analysis** While applying DeepConf improves Text2Cypher performance, it also increases computational cost. Table 4 reports the average token usage and inference time per instance for each approach.

The offline and online modes generate multiple traces for confidence-based aggregation, resulting in higher token usage and longer inference times compared to the base setup. In our experiments,

the number of traces differs across modes based on the input parameters we provided (Base: 1 trace, Offline: 30 traces, Online: 45 traces including 15 warm-up traces). As a result, online mode has the highest computational cost, offline mode is lower, and the base setup is the most efficient. Overall, these results show the trade-off between higher computational cost and the improvements in query quality from confidence-aware test-time strategies.

## 5. Conclusion

Text2Cypher translates natural language questions into executable graph database queries. In this work, we explored DeepConf (Fu et al., 2025), a confidence-aware test-time inference framework, for this task. We initially applied DeepConf to a reasoning model, which produces diverse Cypher outputs but often suffers from syntax errors and incomplete queries, making execution on a graph database difficult. To overcome these issues, we shifted to instruction-tuned models, which generate more reliable outputs, but their default diversity is limited. By systematically adjusting output diversity through parameters such as top-p and temperature, and applying confidence-based strategies in both online and offline modes, we observed consistent improvements in Text2Cypher performance. To validate the generality of this approach, we tested an additional instruction-tuned model and observed similar gains, showing that confidence-aware test-time strategies can enhance structured query generation across multiple models.

Generating multiple traces, however, increases the computational cost, and the limited diversity of instruction-tuned models may reduce the benefits of confidence-based aggregation. Future work could incorporate syntax checks or formal grammar rules to reduce errors and enable early stopping. Incorporating validation-based rewards, inspired by reinforcement learning, could help models prioritize higher-quality outputs. These strategies have the potential to make confidence-aware inference both faster and more accurate for Text2Cypher.

## Declaration on Generative AI Usage

During the preparation of this work, the author(s) used Generative AI tools in order to: 'Improve writing style', 'Paraphrase and reword', 'Code debugging and fixes'. After using these tool(s) or service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

## 6. References


Olga Bunkova, Lorenzo Di Fruscia, Sophia Rupprecht, Artur M Schweidtmann, Marcel JT Reinders, and Jana M Weber. 2026. Grounding large language models in reaction knowledge graphs for synthesis retrieval. *arXiv preprint arXiv:2601.16038*.

Maja Cerjan, Kornelije Rabuzin, and Martina Sestak. 2024. Implementing domains in neo4j. *Int. J. Intell. Inf. Database Syst.*, 16(3):258–285.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2024. Text-to-sql empowered by large language models: A benchmark evaluation. *Proc. VLDB Endow.*

Zijin Hong, Zheng Yuan, Qinggang Zhang, Hao Chen, Junnan Dong, Feiran Huang, and Xiao Huang. 2025. Next-Generation Database Interfaces: A Survey of LLM-Based Text-to-SQL . *IEEE Transactions on Knowledge & Data Engineering*.

HuggingFace. 2024. Huggingface evaluate. https://huggingface.co/evaluate-metric.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

George Katsogiannis-Meimarakis and Georgia Koutrika. 2023. A survey on deep learning approaches for text-to-sql. *VLDB J.*, 32(4):905–936.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *CoRR*.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2020. Bridging textual and tabular data for cross-domain text-to-sql semantic parsing. In *EMNLP 2020*.

Ioanna Mandilara, Christina Maria Androna, Eleni Fotopoulou, Anastasios Zafeiropoulos, and Symeon Papavassiliou. 2025. Decoding the mystery: How can llms turn text into cypher in complex knowledge graphs? *IEEE Access*.

Rosario Napoli, Antonio Celesti, Massimo Villari, and Maria Fazio. 2025. Unlocking advanced graph machine learning insights through knowledge completion on neo4j graph database. In *IEEE Symposium on Computers and Communications, ISCC 2025*.

Makbule Gulcin Ozsoy. 2025. Text2cypher: Data pruning using hard example selection. *arXiv preprint arXiv:2505.05122*.

Makbule Gulcin Ozsoy, Leila Messallem, Jon Besga, and Gianandrea Minneci. 2025. Text2cypher: Bridging natural language and graph databases. In *COLING 2025*.

Bowen Qin, Binyuan Hui, Lihan Wang, Min Yang, Jinyang Li, Binhua Li, Ruiying Geng, Rongyu Cao, Jian Sun, Luo Si, et al. 2022. A survey on text-to-sql parsing: Concepts, methods, and future directions. *arXiv preprint arXiv:2208.13629*.

Richard Senington, Amos H. C. Ng, Ludwig Mittermeier, and Sunith Bandaru. 2025. Graph databases for group decision making in industry: A comprehensive literature review. *IEEE Access*, 13:148533–148546.

Kilian Sennrich and Sina Ahmadi. 2025. Conversational lexicography: Querying lexicographic data on knowledge graphs with sparql through natural language. In *Proceedings of the 5th Conference on Language, Data and Knowledge*, pages 289–300.

Andrei Sobo, Awes Mubarak, Almas Baimagambetov, and Nikolaos Polatidis. 2025. Evaluating llms for code generation in HRI: A comparative study of chatgpt, gemini, and claude. *Appl. Artif. Intell.*, 39(1).

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Chao Yang, Changyi Li, Xiaodu Hu, Hao Yu, and Jinzhi Lu. 2026a. Enhancing knowledge graph interactions: A comprehensive text-to-cypher pipeline with large language models. *Inf. Process. Manag.*, 63(1):104280.

Yahan Yang, Soham Dan, Dan Roth, and Insup Lee. 2026b. On calibration of multilingual question answering llms. *Trans. Mach. Learn. Res.*, 2026.

Gao Yu Zhu, Wei Shao, Xichou Zhu, Lei Yu, Jiafeng Guo, and Xueqi Cheng. 2025. Text2sql: Pure fine-tuning and pure knowledge distillation. In *NAACL 2025*.