Confidence Estimation for Text-to-SQL in Large Language Models

Sepideh Entezari Maleki, Mohammadreza Pourreza, Davood Rafiei

University of Alberta
Edmonton, AB, Canada
{sentezar, pourreza, drafiei}@ualberta.ca

Abstract

Confidence estimation for text-to-SQL aims to assess the reliability of model-generated SQL queries without having access to gold answers. We study this problem in the context of large language models (LLMs), where access to model weights and gradients is often constrained. We explore both black-box and white-box confidence estimation strategies, evaluating their effectiveness on cross-domain text-to-SQL benchmarks. Our evaluation highlights the superior performance of consistency-based methods among black-box models and the advantage of SQL-syntax-aware approaches for interpreting LLM logits in white-box settings. Furthermore, we show that execution-based grounding of queries provides a valuable supplementary signal, improving the effectiveness of both approaches.

Introduction

Large Language Models (LLMs) have demonstrated remarkable proficiency in parsing natural language utterances and performing various generation tasks—such as producing text, code, and images—often at human-like level, thereby automating a wide range of complex processes (Touvron et al. 2023; Chowdhery et al. 2022; Brown et al. 2020; Chen et al. 2021). One significant application of LLMs is in translating natural language queries into logic-based SQL statements, enabling non-technical users to interact seamlessly with databases (Rajkumar, Li, and Bahdanau 2022; Gao et al. 2023; Pourreza and Rafiei 2023). Despite their impressive capabilities, LLMs are not yet ready to be deployed in enterprise settings for users who lack the SQL knowledge. A major challenge is ensuring the accuracy and correctness of the generated queries. Many end-users, with limited SQL knowledge, are unable to verify if a generated query is a correct translation of their input.

A key question in this context is if the confidence level of LLMs in generated queries can be estimated, and if those estimates are accurate. Reliable confidence estimates can be quite useful. On the client side, users can evaluate generated queries alongside confidence scores, choosing to discard queries if the confidence falls below a certain threshold. On the server side, the model can abstain from generating a response when confidence is low, reducing the risk of producing incorrect or harmful outputs.

The issue of uncertainty in deep neural networks has been studied under various names, including out-of-distribution detection (Liang, Li, and Srikant 2017), uncertainty estimation(Lakshminarayanan, Pritzel, and Blundell 2017), confidence prediction (Dolezal et al. 2022), and calibration (Guo et al. 2017). However, many standard uncertainty techniques such as Monte Carlo dropout (Gal and Ghahramani 2016; Wang and Manning 2013) and deep ensembles (Lakshminarayanan, Pritzel, and Blundell 2017) are difficult to apply to LLMs in practice, especially for closed-source models with limited access to internals. In the text-to-SQL domain, uncertainty has received limited attention, with only a few studies exploring query abstention(Somov and Tutubalina 2025; Chen et al. 2025) and calibration(Ramachandran and Sarawagi 2024). More broadly, uncertainty in code generation—and text-to-SQL in particular—remains poorly understood. The common fallback of exhaustive testing a generated code, even with recent automation techniques (Liu et al. 2024), is not feasible for end-users.

This paper introduces a benchmark suite for evaluating confidence estimation in text-to-SQL, along with metrics and evaluation protocols. We also propose a novel syntax-aware logit-based model that significantly improves calibration across a variety of query types and LLM families. Our work offers the first systematic comparison of black-box (e.g., output-based) and white-box (e.g., logit-based) approaches in this domain. We evaluate both families of techniques, drawing from strategies used in LLMs more broadly (Geng et al. 2024; Zhang et al. 2023; Wang et al. 2023; Huang et al. 2023a). Our black-box models include verbal and consistency-based approaches, incorporating various prompting strategies such as Chain-of-Thought (COT) reasoning (Wei et al. 2022; Xiong et al. 2023) and ensemble techniques that cluster multiple SQL query generations based on execution outcomes (Gao et al. 2023; Sun et al. 2023; Dong et al. 2023) or feature similarity (Lin, Trivedi, and Sun 2023). In contrast, our white-box approaches leverage logit-based models, where token-level probabilities are aggregated using various compositional schemes to estimate the confidence of model-generated queries.

Our evaluation on Spider and BIRD, two extensive cross-domain datasets, demonstrates that white-box strategies consistently outperform black-box methods. Among the white-box methods, our SQL-syntax-aware estimation emerges as

the most effective, excelling across both long and complex queries in the Bird benchmark and shorter, simpler queries in the Spider benchmark. The choice of aggregation function impacts performance, with average token probability proving to be the most stable, mitigating the impact of low-probability tokens, particularly for our SQL-syntax-aware estimation. Conversely, product aggregation, which treats token probabilities as independent and amplifies the impact of rare tokens, performs better for our schema-aware strategies in simpler query structures. Among black-box methods, the consistency-based approach, leveraging SQL query execution as a signal, is the most reliable, though it incurs latency and cost overhead due to multiple query executions.

We make the following key contributions: (1) We present the first comprehensive benchmark of confidence estimation methods for text-to-SQL, comparing black-box and white-box approaches on Spider and Bird. (2) We introduce SAC, a syntax-aware logit aggregation method that filters out non-informative tokens and reduces expected calibration error (ECE) by up to 16% over existing baselines (3) We propose three white-box models tailored to different stages of SQL generation, offering interpretable and efficient confidence estimates. (4) We adapt multiple black-box methods from other domains to text-to-SQL, enabling meaningful cross-method comparisons. (5) Our broad evaluation across proprietary and open-source LLMs shows that white-box methods—particularly syntax-aware models using average token aggregation—consistently outperform others. Our results highlight the value of incorporating SQL structure into confidence estimation.

Related Works

Confidence prediction and uncertainty estimation have been widely explored in traditional supervised learning (Gawlikowski et al. 2023; Lakshminarayanan, Pritzel, and Blundell 2017; Guo et al. 2017). The shift to LLMs introduces new challenges in confidence estimation, particularly in generative tasks. Black-box methods rely solely on model outputs, using verbalized confidence (Lin, Hilton, and Evans 2022), consistency-based techniques (Manakul, Liusie, and Gales 2023; Xiao et al. 2025), and surrogate models (Shrivastava, Liang, and Kumar 2023) to infer uncertainty without accessing the internal workings of the model.

White-box methods, on the other hand, utilize internal model states, such as logits or activations, to estimate confidence. Logit-based approaches measure uncertainty at the token or sentence levels (Huang et al. 2023b), while semantic-based methods adjust confidence using token relevance and semantic similarity (Stengel-Eskin and Van Durme 2023; Duan et al. 2023). Additionally, probing methods analyze model activations to classify whether an LLM "knows" the answer (Kadavath et al. 2022; Azaria and Mitchell 2023). Ramachandran and Sarawagi (2024) show that simply rescaling a model's full-sequence probability already calibrates text-to-SQL better than self-check prompts.

A related line of work explores query abstention, where the system opts not to produce a query when confidence is low. TriageSQL (Zhang et al. 2020) and TrustSQL (Lee et al. 2024) introduce benchmarks for query abstention but lacked methods for direct confidence scoring. Somov and Tutubalina (2025) treat confidence as a thresholding mechanism over token entropy or sequence likelihood. Chen et al. (2025) presents RTS, a method that abstains during schema linking with conformal guarantees and optional human intervention.

Our work unifies and extends these directions. We systematically compare seven black-box and white-box confidence estimation methods for text-to-SQL, evaluate them on two benchmarks, and introduce a syntax-aware logit aggregation method that ignores low-signal tokens and focuses on schema-linked elements. Unlike prior abstention-based methods, we provide calibrated scalar confidence scores rather than binary decisions.

Methodology

Our approach to confidence prediction encompasses three strategies: verbalized, consistency-based, and logit-based methods, as illustrated in Figure 1. Verbalized confidence prediction entails directly instructing the LLM to articulate its confidence in the generated text, whereas our consistency-based approach generates multiple responses and employs the consistency among these responses as an indicator of confidence. Our logit-based approach leverages the principle that an LLM's token generation is guided by a sequence of probability distributions (logits) over the vocabulary, where the most likely next token is predicted at each step based on the input context.

Verbalized Confidence

Verbalized confidence prediction encompasses a range of approaches where the model is prompted to generate a confidence score alongside its response to a given question. This approach mirrors the way we seek uncertainty assessments from human experts, by asking them to express their confidence in their recommendations. We investigate three variations of verbalized confidence: vanilla verbalized, chain-of-thought, and augmented chain-of-thought. Each variation incrementally increases the information included in the prompts, allowing us to assess how LLMs respond to different levels of contextual detail.Detailed prompts are provided in the Appendix Maleki, Pourreza, and Rafiei (2025).

Vanilla verbalized The most direct approach involves instructing the model to generate a confidence score within a range of 0 to 100 (Xiong et al. 2023; Manakul, Liusie, and Gales 2023). This approach employs an input prompt consisting of three key elements: an instruction to guide the model in generating both a SQL query and a corresponding confidence score, the question presented to the model, and the database schema, complete with three sample rows for each table. Including the database schema and sample rows in the prompt serves the purpose of aiding the model in crafting a SQL query, a practice commonly observed in the literature (Rajkumar, Li, and Bahdanau 2022).

Chain-Of-Thought verbalized The prediction of confidence levels demands a high degree of reasoning ability, involving an assessment of the model's certainty regarding its generated responses. This entails the model not only comprehending the context, represented by the database schema,

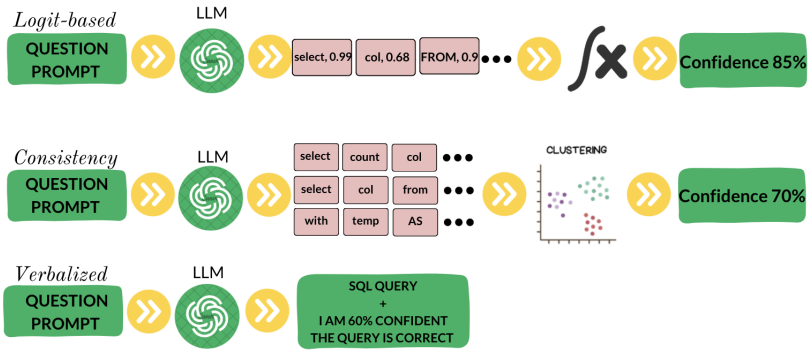

Figure 1: Overview of our methods: verbalized, consistency-based, and logit-based confidence prediction.

and the content of the input question but also engaging in metacognitive reasoning to gauge the accuracy of its output. Building on insights from previous work (Xiong et al. 2023), to facilitate more effective reasoning, we employ a one-shot *chain-of-thought* prompting technique for confidence prediction, as introduced in Kojima et al. (2022); Wei et al. (2022). The input prompt for this method includes instructions, the given question, and the database schema with sample rows, similar to the vanilla confidence prediction. Additionally, we incorporate a reasoning example, commencing with the phrase: "Let's think step by step."

Augmented chain-of-thought In text-to-SQL tasks, relying solely on the SQL query, database schema, and the given question may be insufficient for accurate confidence prediction. An additional source of valuable information, unique to text-to-SQL, is the execution result of the SQL query, which is unavailable in other areas like question answering. We improve the chain-of-thought confidence prediction method by incorporating the execution results of the generated SQL query. This augmentation provides the model with additional context, enabling more precise confidence scoring. To maintain efficiency and avoid exceeding the model's context window, we constrain the execution result to the first 1000 distinct rows. The prompt for this method builds upon the Chain-of-Thought approach, incorporating execution results directly before the reasoning process.

Self-Check Confidence Inspired by the findings of Kadavath et al. (2022) and Ramachandran and Sarawagi (2024), which show that language models can self-assess the truthfulness of their outputs, we adopt a verbalized self-evaluation strategy with two variants: (T) indicating that the query is correct, and (F) indicating that it is incorrect.

Consistency-based Confidence

Influenced by the self-consistency method, a common approach for improving text-to-SQL generation involves producing multiple output samples and selecting the most consistent answer as the final prediction (Gao et al. 2023; Sun et al. 2023; Dong et al. 2023). In our study, we leverage the consistency between generated SQL queries as a metric for confidence estimation. To produce multiple samples for a given question, we employ the same technique as proposed by the self-consistency method, increasing the temperature to introduce randomness during SQL generation. Raising the temperature redistributes some probability mass from highly likely tokens to less likely ones. However, if the initial probability is already sufficiently high, the token should still have a greater chance of being chosen as the next token. Conversely, when the model lacks confidence in its response, increasing the temperature can make the probability distribution approach a more uniform distribution. As a result, sampling multiple times from the model can yield significantly different answers. Next, we explore various methods for quantifying the consistency between generated samples.

Execution-based consistency SQL query execution on a database instance provides definitive answers to a given question, making execution results a valuable basis for consistency analysis. In our approach, we generate multiple SQL query samples from the LLM, execute them on the database, and cluster these queries based on the extent of exact matches in their retrieved rows. The confidence score for each SQL query is then computed as the proportion of model-generated queries that produce the same execution result, reflecting the model's certainty in its predictions.

Embedding-based consistency While execution-based consistency proves effective in predicting confidence scores, it may not be suitable for large-scale enterprise databases due to the computational cost of executing multiple SQL queries and comparing results. To address this limitation, our embedding-based approach generates contextual embeddings for SQL queries and clusters these embeddings to identify consistencies among the samples. Similar to the execution-based method, the confidence score for a query sample is estimated as the fraction of queries that fall in the same cluster.

Schema-based consistency Several studies have demonstrated the pivotal role of schema links—including table names, column names and mentioned constants—in ensuring the correctness of SQL queries generated by LLMs (Pourreza and Rafiei 2023; Li et al. 2023). In many cases, the accuracy of schema link selection outweighs the specific arrangement of these links with SQL keywords for SQL query correctness. Building on this insight (Pourreza and Rafiei 2023; Cao et al. 2021; Wang et al. 2019; Guo et al. 2019), our schema-based

confidence prediction identifies all schema links from generated SQL query samples and evaluates their consistency by performing an exact match of these schema links across the samples. This method is supported by the observation that multiple correct SQL queries can often be generated for a given question, with the shared schema links serving as a common feature. SQL queries with identical schema links are grouped into clusters, and the confidence score for each query is calculated as the proportion of queries within its cluster.

Logit-based Confidence

Many LLMs, including both open-source and proprietary models like GPT-4o, output token logits that represent their confidence in predicting the next tokens. These logits are influenced by the LLM's inherent understanding of SQL syntax and the input prompt, which includes the SQL schema, query, and relevant examples or context. Our logit-based approach utilizes these logits for confidence prediction. The input prompt for this method mirrors the structure of the vanilla verbal model, but the output includes both the generated tokens and their associated probabilities.

Full-token confidence This method treats all tokens in a generated SQL query—including SQL keywords, table and column names, operators, functions, and formatting elements—equally weighted, aggregating their generation probabilities.

Schema-Linked confidence This method builds on our schema-based approach discussed under consistency-based models, emphasizing the critical importance of accurate schema mapping for query correctness (Pourreza and Rafiei 2023). It exclusively focuses on schema-linked tokens, such as table and column names and condition values, to estimate the model confidence.

SQL-Aware confidence Recognizing that not all components of an SQL query hold equal significance, this method refines confidence estimation by incorporating SQL-specific conventions and distinguishing between critical and non-critical tokens. The goal is to ensure that confidence scores reflect the semantic correctness of queries by prioritizing key tokens such as Schema-Linking, JOIN conditions, and WHERE clauses (Pourreza and Rafiei 2023) while normalizing or ignoring less important elements, such as formatting tokens and optional keywords. This refinement is achieved through a few steps, as discussed in the next section (see the Appendix Maleki, Pourreza, and Rafiei (2025) for more detailed samples).

Token Exclusion Certain tokens—including extra whitespaces, redundant parentheses, and optional keywords such as `INNER` in `INNER JOIN` or `AS` used for aliasing—are excluded from consideration as they do not impact the correctness of an SQL query. This allows the focus to remain on elements critical to query accuracy.

Case Folding and Order Folding SQL syntax is insensitive to the casing of reserved keywords (e.g., `SELECT`, `FROM`, `WHERE`) and the references to table and column names. To account for this, probabilities for case variants (e.g., `select` and `SELECT`) are combined, treating them as equivalent. Similarly, the order of elements within clauses such as `SELECT`, `FROM`, `WHERE`, and `GROUP BY` does not affect query correctness. For instance, `SELECT a, b` and `SELECT b, a` are equivalent. Where applicable and safe, probabilities are adjusted for this flexibility, ensuring the accuracy of confidence estimation.

Synonym Folding Synonymous keywords (e.g., `not equal to`, `!=`, `<>`), symmetric conditions (e.g., `x=y` and `y=x`), and logically equivalent expressions (e.g., `a AND b` and `b AND a`) are treated as interchangeable. Probabilities are adjusted accordingly, where safe, to reflect this equivalence. For a detailed explanation of the probability adjustments for case insensitivity, synonymous keywords, and interchangeable constructs.

Aggregation of token probabilities Our logit-based models aggregate token probabilities into a single confidence score that reflects the model's overall certainty for the generated SQL query. Assuming each token generation is an independent event, the confidence in the query can be calculated as the product of the individual token probabilities. This method, referred to as the *product method*, emphasizes the joint confidence across all tokens and is highly sensitive to any low-probability token, making it particularly effective for short and schema-heavy queries where every token is critical. However, for longer or more complex queries, this sensitivity can result in disproportionately low confidence scores due to the cumulative effect of even minor uncertainties.

As an alternative that closely resembles perplexity [1], the *average method* calculates the mean probability across all tokens. This method provides a normalized view of token probabilities, making it robust for longer or more complex queries. By smoothing the impact of low-probability tokens, the average method ensures a balanced evaluation. However, it may under-penalize critical tokens with very low probabilities, particularly in cases where schema-linked elements or logical conditions are essential to query correctness.

Experiments

Evaluation Setup

Datasets Our evaluation was conducted on the development sets of Spider (Yu et al. 2018) and BIRD (Li et al. 2024), two well-established cross-domain text-to-SQL benchmarks. The Spider dataset consists of 1,034 examples across 20 databases, and the BIRD dataset includes 1,533 question-SQL pairs from 11 databases, covering diverse domains such as healthcare, finance, and education. The BIRD dataset presents more complex SQL query challenges compared to Spider.

LLMs We conduct our experiments using both closed-source and open-source large language models (LLMs), including GPT-3.5-turbo and GPT-4o (closed-source), as well as DeepSeek 6.7B and Qwen2.5 (open-source).

[1] Perplexity is defined as 2^{-m} where m is the mean of log probabilities.

Metrics Execution accuracy (EX) is the gold standard for text-to-SQL, but it requires ground-truth queries. Therefore, in addition to EX, we evaluate each confidence score with two standard proxies (Becker and Soatto 2024; Xiong et al. 2023): **AUC-ROC**, the area under the ROC curve, quantifies how well a score separates correct from incorrect queries (1.0 = perfect, 0.5 = random). **Expected Calibration Error (ECE)** bins predicted probabilities and reports the mean gap between confidence and empirical accuracy (lower is better; 0 means perfect calibration) (see more details in the Appendix Maleki, Pourreza, and Rafiei (2025)).

Baselines. Our black-box baselines are inspired by established hallucination detection techniques for closed-source language models (Manakul, Liusie, and Gales 2023), adapted to the text-to-SQL setting. We adopt the full-token confidence (FTC) framework from Ramachandran and Sarawagi (2024); Among the variants evaluated, the FTC-Product has demonstrated the best performance on both the Spider and BIRD benchmarks, making it a strong baseline for our comparisons.

Models Compared

As summarized in Table 1, our logit-based methods, especially SQL-Aware Confidence (SAC) consistently outperform black-box approaches, including consistency-based, verbalized variants, and Self-check bool, *across both benchmarks and all four LLMs we evaluate* (GPT-3.5, DeepSeek-6.7B, GPT-4o-mini, and Qwen2.5-Coder). SAC delivers up to a 12% improvement in AUC-ROC and reduces ECE by as much as 16% on BIRD relative to the best black-box competitor ($p < 0.05$). BIRD's join-heavy queries expose many schema-linked tokens that SAC leverages, whereas Spider's shorter pattern-style queries provide fewer such cues and are therefore harder to calibrate. By focusing on schema-linked tokens and critical SQL operators, SAC mitigates both overconfidence and underconfidence. The two newer open-source models (GPT-4o-mini and Qwen2.5-Coder) follow the same trend: SAC-Avg remains best on BIRD, while SAC-Prod leads on Spider ($p < 0.05$ in both cases).

The aggregation methods handles uncertainty differently. Product aggregation is highly sensitive to low-confidence tokens, which are more common in longer or more complex queries. This sensitivity can lead to overly pessimistic scores. In contrast, average aggregation dilutes the influence of any single low-confidence token, making it more robust in challenging settings. The SAC method consistently outperforms other models on both datasets because it is robust and focuses only on the most important tokens in the SQL query—namely, schema-related tokens like table names, column names, and values. This syntax-aware focus makes it less sensitive to irrelevant variations and enhances its reliability across different scenarios.

Execution-based consistency stands out among black-box models, achieving the highest AUC-ROC and lowest ECE scores, highlighting its effectiveness in leveraging external feedback through query execution when model logits are unavailable. On the Spider dataset, it occasionally matches or even outperforms certain logit-based methods, demonstrating its ability to capture correctness in simpler query scenarios reliant on execution feedback. In contrast, verbalized approaches often exhibit poor calibration, with ECE exceeding 50% in some cases, indicating significant overconfidence. The narrower performance gap between verbalized methods and logit-based methods on Spider can be attributed to Spider's simpler query structures, which improves the effectiveness of verbalized confidence estimation.

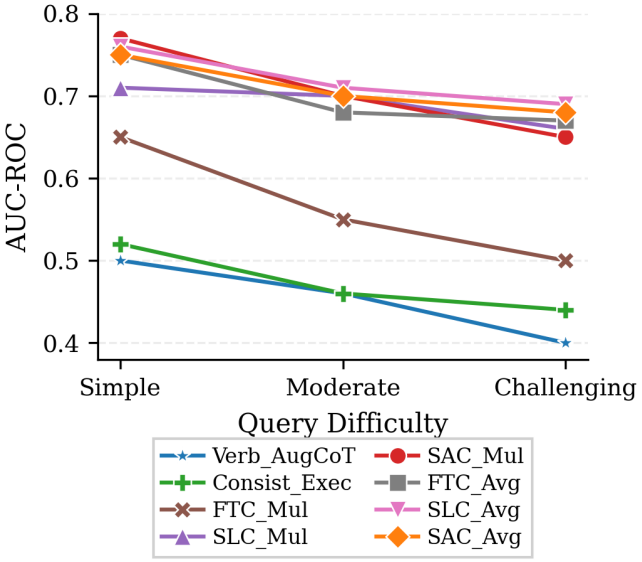

Figure 2: Performance across these predefined difficulty levels in the BIRD dataset and GPT-3.5 model.

Performance Varying Query Difficulty

Figures 2 depict model performance across various query difficulty levels for the Bird dataset. Challenging queries typically feature heavier schema-linked, multi-table joins, aggregations, and nested clauses. As the SQL queries become more complex, the models exhibit reduced confidence in token generation, resulting in lower confidence scores for the generated queries, as expected.

The figure shows a clear trend of declining model performance as query difficulty progresses from simple to challenging. This decline is most pronounced for the execution-based Consistency, Augmented-COT Verbalized, and FTC-Mul methods, where the models show a steep drop in performance. In contrast, the SAC models exhibit a more gradual decline in performance, indicating a better handling of complex SQL structures due to their focus on crucial SQL tokens and schema elements. Unlike FTC-Mul, which is adversely affected by the larger number of tokens in complex queries, SAC models manage to maintain relatively higher mean scores across all levels of difficulty, underscoring their robustness in face of complexity. These trends are consistent with observations from the Spider dataset, as detailed in the extended version Maleki, Pourreza, and Rafiei (2025). However, it is noted that models utilizing multiple aggregation techniques and the execution-based Consistency show relatively better performance stability on the Spider dataset, indicating their effectiveness in managing simpler queries.

Dataset	Approach	Method	GPT-3.5		DeepSeek		GPT-4o-mini		Qwen2.5	
			AUC↑	ECE↓	AUC	ECE	AUC	ECE	AUC	ECE
Spider	Logit-based	FTC – Average	63.07	22.10	58.44	24.41	68.35	20.23	61.81	23.36
		FTC – Product	68.10	19.38	63.04	20.00	77.23	17.15	70.54	19.56
		SLC – Average	62.98	23.70	53.08	29.07	65.70	23.19	63.77	25.82
		SLC – Product	65.38	22.13	61.23	22.67	69.93	20.94	67.04	20.70
		SAC – Average	65.01	23.08	58.12	27.90	68.89	21.75	65.11	21.91
		SAC – Product	**71.66***	**16.98**	**65.26***	**19.80**	**79.87***	**14.11**	**74.89***	**15.08**
	Black-box	Consistency(Exec)	65.91	19.13	63.25	19.98	71.82	17.21	68.32	19.04
		Consistency (Embed)	58.42	24.34	60.62	23.07	65.31	21.52	63.74	22.30
		Consistency (Schema)	59.97	23.04	61.65	22.34	67.32	20.11	65.76	20.78
		Verbalized (Vanilla)	55.45	25.98	56.04	25.29	56.14	25.42	54.91	26.01
		Verbalized (COT)	58.14	24.93	60.57	23.11	57.20	24.47	55.39	25.11
		Verbalized (Aug COT)	58.70	23.54	60.94	22.97	61.05	22.78	59.32	23.29
		Self-Check Bool	55.81	25.16	58.24	25.61	57.25	25.11	56.81	25.01
Bird	Logit-based	FTC – Average	74.88	19.37	74.02	20.65	75.18	19.04	76.58	19.01
		FTC – Product	72.36	28.21	67.77	27.34	72.43	24.04	72.31	25.11
		SLC – Average	77.07	19.18	76.66	20.23	78.54	18.20	78.08	20.21
		SLC – Product	73.71	23.90	69.88	23.53	77.73	20.25	74.03	22.96
		SAC – Average	**79.06***	**12.15**	**77.94***	**11.06**	**83.06***	**10.03**	**79.94***	**11.03**
		SAC – Product	73.00	22.88	71.19	21.11	81.36	18.23	75.27	19.19
	Black-box	Consistency (Exec)	67.36	27.14	66.48	28.34	76.34	20.74	71.18	26.11
		Consistency (Embed)	61.21	31.12	60.96	34.31	73.82	24.75	67.23	28.89
		Consistency (Schema)	64.47	29.98	63.16	30.51	74.14	23.32	70.08	27.12
		Verbalized (Vanilla)	55.67	58.61	56.23	54.22	58.43	52.02	56.14	53.22
		Verbalized (COT)	57.12	53.11	58.84	50.49	59.48	48.77	57.03	50.26
		Verbalized (Aug COT)	57.73	52.29	59.21	46.31	63.16	29.99	60.78	33.23
		Self-Check Bool	55.81	48.11	56.69	44.42	59.51	42.52	56.46	43.16

Table 1: Performance on Spider and Bird dev sets. FTC = Full Token Confidence, SLC = Schema-Linked Confidence, SAC = SQL-Aware Confidence. * indicates statistically significant improvement over the best baseline ($p < 0.05$).

Performance Varying Query Length

Figure 3 illustrates how the Black-box and White-box approaches perform as average query lengths vary within the Bird benchmark. We group queries into Short (0–15 tokens), Medium (16–25), and Long (26+), as shown in the Figure For Spider, the majority of queries, 5,046, were categorized as Short, indicating a prevalence of less complex or shorter queries within this dataset. Additionally, 2,749 queries were classified as Medium, and 1,844 as Long. For Bird we have a significant prevalence of longer queries, with 5,966 falling into the Long category, compared to 651 in the Medium and 597 in the Short categories.

The length of a query often correlates with its complexity, and we observe a general trend of diminishing performance as query length increases. This is particularly noticeable in the FTC model, where longer queries exacerbate the impact of low-probability tokens, leading to significant score reductions. The SAC model, in contrast, demonstrates superior performance in handling longer queries.

The choice of aggregation method distinctly affects performance: multiplication aggregations tend to magnify penalties associated with longer queries, adversely impacting the FTC model. In contrast, average aggregation smooths out these effects and ensures more stable performance across different query lengths. This highlights that aggregation functions which normalize based on query length tend to yield better results, especially in the context of longer and more complex queries. A contrasting scenario is observed in the Spider dataset, as depicted in the extended version (Maleki, Pourreza, and Rafiei 2025). Here, multiplication aggregation functions appear more effective, largely because the queries are generally shorter compared to those in the Bird dataset.

Performance Varying Schema Link Size

Figure 4 visualizes the impact of schema complexity on models performance within the Bird benchmark. We bucket queries by schema link count into Low (0–5), Moderate (6–9), and High (10+). For the Spider dataset, the majority of queries (4979) were categorized as Low, indicating a prevalence of simpler or shorter queries, followed by 4074 Moderate and 605 as High complexity queries. In contrast, the Bird dataset presents a significant prevalence of schema-heavy queries, with 4947 falling into the High category, compared to 1172 in Moderate and 1105 in the Low category. Schema heaviness often correlates with increased query complexity as queries with more schema links—such as a greater number of tables, columns, and values—tend to exhibit reduced confidence scores.

In environments characterized by schema-heavy queries, the SLC and SAC models consistently outperform other methods. These models prioritize schema-linked tokens, which play a pivotal role in enhancing the accuracy of confidence estimation for TexttoSQL tasks. By focusing primarily on schema links, even to the exclusion of other factors, these

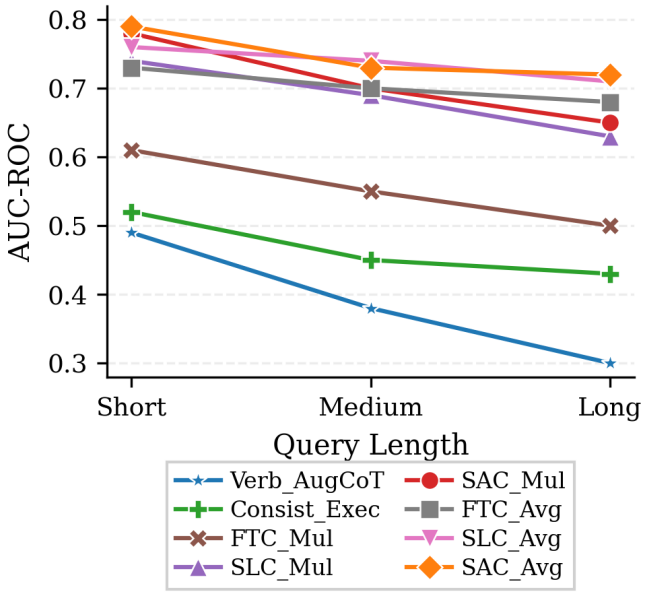

Figure 3: The distribution of query lengths and model performance (GPT-3.5) on the BIRD dataset.

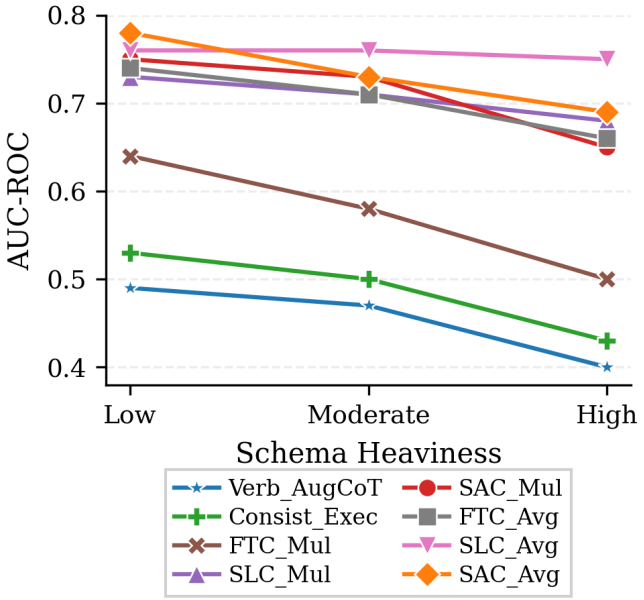

Figure 4: The distribution of schema heaviness and model (GPT-3.5) performance in the BIRD dataset.

models provide a robust measure of model confidence. This approach is particularly effective in the Bird dataset, where schema complexity is high. A similar pattern is observed in the Spider dataset, confirming the utility of this approach across different levels of schema heaviness, as shown in the extended version (Maleki, Pourreza, and Rafiei 2025).

The choice of aggregation strategy, average versus product, affects performance depending on query characteristics, such as complexity and length. As shown in Figure 2 for BIRD, SAC-Mul performs best on simple queries, but its advantage narrows as queries grow more complex, with SAC-Avg eventually outperforming it. A similar pattern is seen for Spider, though the effect is less pronounced, likely due to Spider's generally shorter and simpler queries. These trends are further supported by analyses of varying query length and schema heaviness. See the extended version (Maleki, Pourreza, and Rafiei 2025).

Impact of Execution Feedback

Incorporating execution feedback into our SQL query models significantly enhances performance, particularly with SAC methods which exhibit the highest AUC scores across the Spider and Bird datasets. Notably, execution grounding substantially improves results on the more complex Bird dataset, achieving an AUC of 79.06 and an ECE of 12.15 for GPT-3.5. Detailed results and methodological comparisons are provided in the Appendix Maleki, Pourreza, and Rafiei (2025).

Impact of SQL-Specific Features

Table 2 shows that SQL-specific features are important for SAC-Avg on BIRD. Removing non-critical tokens causes the largest drop, while case/order, synonym, and equivalent-expression folding provide smaller but consistent gains. Excluding non-critical tokens led to the largest performance drop (-5.7% AUC, +4.2% ECE), showing its central role in accurate confidence estimation. Case and order folding also proved important, reducing AUC by 3.8% and 1.2% when removed.

Feature Category	**AUC**	**ECE↓**
SAC-Average	79.06	12.15
w/o Token Exclusion	−5.7%	+4.2%
w/o Case Folding	−3.8%	+1.3%
w/o Order Folding	−1.2%	+1.1%
w/o Synonym Folding	−0.5%	+0.3%
w/o Equiv. Expr.	−0.7%	+0.4%

Table 2: Impact of ablating SQL-specific features on SAC-Avg (BIRD).

Conclusions and Future Work

This work presents an exploration of confidence estimation for text-to-SQL, addressing a critical gap in the application of LLMs for natural language to SQL translation. Our study demonstrates the superiority of white-box methods, particularly the SQL-Aware model, which excels across diverse queries due to SQL-specific adjustments and robust aggregation strategies like average aggregation. While black-box methods, such as consistency-based approaches, offer simplicity, they are constrained by latency and cost. Future work could explore hybrid methods combining the strengths of white-box and black-box approaches, alternative aggregation strategies for complex queries, and evaluations on real-world databases.

Acknowledgments

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada.

References

Azaria, A.; and Mitchell, T. 2023. The internal state of an LLM knows when it's lying. *arXiv preprint arXiv:2304.13734*.

Becker, E.; and Soatto, S. 2024. Cycles of Thought: Measuring LLM Confidence through Stable Explanations. *arXiv preprint arXiv:2406.03441*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cao, R.; Chen, L.; Chen, Z.; Zhao, Y.; Zhu, S.; and Yu, K. 2021. LGESQL: line graph enhanced text-to-SQL model with mixed local and non-local relations. *arXiv preprint arXiv:2106.01093*.

Chen, K.; Chen, Y.; Koudas, N.; and Yu, X. 2025. Reliable Text-to-SQL with Adaptive Abstention. *Proceedings of the ACM on Management of Data*, 3(1): 1–30.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. d. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Dolezal, J. M.; Srisuwananukorn, A.; Karpeyev, D.; Ramesh, S.; Kochanny, S.; Cody, B.; Mansfield, A. S.; Rakshit, S.; Bansal, R.; Bois, M. C.; et al. 2022. Uncertainty-informed deep learning models enable high-confidence predictions for digital histopathology. *Nature communications*, 13(1): 6572.

Dong, X.; Zhang, C.; Ge, Y.; Mao, Y.; Gao, Y.; Lin, J.; Lou, D.; et al. 2023. C3: Zero-shot Text-to-SQL with ChatGPT. *arXiv preprint arXiv:2307.07306*.

Duan, J.; Cheng, H.; Wang, S.; Wang, C.; Zavalny, A.; Xu, R.; Kailkhura, B.; and Xu, K. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR.

Gao, D.; Wang, H.; Li, Y.; Sun, X.; Qian, Y.; Ding, B.; and Zhou, J. 2023. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. *arXiv preprint arXiv:2308.15363*.

Gawlikowski, J.; Tassi, C. R. N.; Ali, M.; Lee, J.; Humt, M.; Feng, J.; Kruspe, A.; Triebel, R.; Jung, P.; Roscher, R.; et al. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 1–77.

Geng, J.; Cai, F.; Wang, Y.; Koeppl, H.; Nakov, P.; and Gurevych, I. 2024. A Survey of Confidence Estimation and Calibration in Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 6577–6595.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International conference on machine learning*, 1321–1330. PMLR.

Guo, J.; Zhan, Z.; Gao, Y.; Xiao, Y.; Lou, J.-G.; Liu, T.; and Zhang, D. 2019. Towards complex text-to-sql in cross-domain database with intermediate representation. *arXiv preprint arXiv:1905.08205*.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. 2023a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Huang, Y.; Song, J.; Wang, Z.; Chen, H.; and Ma, L. 2023b. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Lee, G.; Chay, W.; Cho, S.; and Choi, E. 2024. Trustsql: A reliability benchmark for text-to-sql models with diverse unanswerable questions. *arXiv preprint arXiv:2403.15879*.

Li, J.; Hui, B.; Qu, G.; Li, B.; Yang, J.; Li, B.; Wang, B.; Qin, B.; Cao, R.; Geng, R.; et al. 2023. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *arXiv preprint arXiv:2305.03111*.

Li, J.; Hui, B.; Qu, G.; Yang, J.; Li, B.; Li, B.; Wang, B.; Qin, B.; Geng, R.; Huo, N.; et al. 2024. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36.

Liang, S.; Li, Y.; and Srikant, R. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Lin, S.; Hilton, J.; and Evans, O. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Lin, Z.; Trivedi, S.; and Sun, J. 2023. Generating with Confidence: Uncertainty Quantification for Black-box Large Language Models. *arXiv preprint arXiv:2305.19187*.

Liu, J.; Xia, C. S.; Wang, Y.; and Zhang, L. 2024. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36.

Maleki, S. E.; Pourreza, M.; and Rafiei, D. 2025. Confidence estimation for text-to-sql in large language models. *arXiv preprint arXiv:2508.14056*.

Manakul, P.; Liusie, A.; and Gales, M. J. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Pourreza, M.; and Rafiei, D. 2023. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *arXiv preprint arXiv:2304.11015*.

Rajkumar, N.; Li, R.; and Bahdanau, D. 2022. Evaluating the text-to-sql capabilities of large language models. *arXiv preprint arXiv:2204.00498*.

Ramachandran, A.; and Sarawagi, S. 2024. Text-to-SQL Calibration: No Need to Ask–Just Rescale Model Probabilities. *arXiv preprint arXiv:2411.16742*.

Shrivastava, V.; Liang, P.; and Kumar, A. 2023. Llamas Know What GPTs Don't Show: Surrogate Models for Confidence Estimation. *arXiv preprint arXiv:2311.08877*.

Somov, O.; and Tutubalina, E. 2025. Confidence estimation for error detection in text-to-sql systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 25137–25145.

Stengel-Eskin, E.; and Van Durme, B. 2023. Calibrated interpretation: Confidence estimation in semantic parsing. *Transactions of the Association for Computational Linguistics*, 11: 1213–1231.

Sun, R.; Arik, S. O.; Nakhost, H.; Dai, H.; Sinha, R.; Yin, P.; and Pfister, T. 2023. SQL-PaLM: Improved Large Language ModelAdaptation for Text-to-SQL. *arXiv preprint arXiv:2306.00739*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, B.; Shin, R.; Liu, X.; Polozov, O.; and Richardson, M. 2019. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. *arXiv preprint arXiv:1911.04942*.

Wang, C.; Liu, X.; Yue, Y.; Tang, X.; Zhang, T.; Jiayang, C.; Yao, Y.; Gao, W.; Hu, X.; Qi, Z.; et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Wang, S.; and Manning, C. 2013. Fast dropout training. In *international conference on machine learning*, 118–126. PMLR.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Xiao, Q.; Bhattacharjya, D.; Ganesan, B.; Marinescu, R.; Mirylenka, K.; Pham, N. H.; Glass, M.; and Lee, J. 2025. The Consistency Hypothesis in Uncertainty Quantification for Large Language Models. *arXiv preprint arXiv:2506.21849*.

Xiong, M.; Hu, Z.; Lu, X.; Li, Y.; Fu, J.; He, J.; and Hooi, B. 2023. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. *arXiv preprint arXiv:2306.13063*.

Yu, T.; Zhang, R.; Yang, K.; Yasunaga, M.; Wang, D.; Li, Z.; Ma, J.; Li, I.; Yao, Q.; Roman, S.; et al. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Zhang, Y.; Dong, X.; Chang, S.; Yu, T.; Shi, P.; and Zhang, R. 2020. Did you ask a good question? a cross-domain question intention classification benchmark for text-to-sql. *arXiv preprint arXiv:2010.12634*.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; et al. 2023. Siren's song in the AI ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.