EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

SQL-GRID: A Grounded Retrieval and Interactive Disambiguation Agent for Text-to-SQL over Power Grid Data

Yaoqiang Xu [1]*, Li Li [1], Chen Qian [1] and Jin Zhou [1]

[1]East Branch of State Grid Corporation of China, Shanghai 200120, China

Abstract

Power grid data often contain highly abstract metadata, ambiguous business terms, and complex statistical scopes. These characteristics make general large language models prone to semantic deviation and SQL hallucination in Text-to-SQL tasks. To address this problem, this paper proposes SQL-GRID, a grounded retrieval and interactive disambiguation agent for Text-to-SQL over power grid data. The framework is based on a multi-agent collaboration mechanism. It integrates retrieval, validation, disambiguation, and SQL generation modules to transform natural language intent into accurate SQL statements. To bridge the gap between business semantics and database structures, this paper constructs a Business Knowledge Representation(BKR) based knowledge base, which integrates database structural information, business descriptions, real data samples, and statistical rules. A multi-strategy RAG method is also designed. It combines semantic vector retrieval with keyword matching and provides accurate domain knowledge for SQL generation. To address business ambiguity, this paper further proposes a closed-loop mechanism consisting of pre-generation disambiguation, interactive clarification, and post-generation validation. Through explicit human-machine interaction, the framework guides users to clarify statistical granularity and statistical scope constraints. This process effectively reduces uncertainty during SQL generation. Experimental results show that the proposed method achieves a metadata recall rate of 94% on the power grid dataset. The SQL execution rate increases from 40.2% in the traditional baseline to 84.6%. The semantic matching score reaches 4.85. These results show that the proposed framework significantly improves the accuracy and reliability of self-service queries in complex power grid scenarios.

Keywords: SQL-GRID, Text-to-SQL, Power Grid Data, Grounded Retrieval, Interactive Disambiguation, Business Knowledge Representation(BKR), SQL Hallucination

Received on 16 July 2026, accepted on 25 August 2026, published on 04 September 2026

doi: 10.4108/ew.14071

*Corresponding author. Email: psc01sgcc@139.com

1. Introduction

In recent years, large language models (LLMs) have made rapid progress in natural language understanding, text generation, and complex reasoning [1,2]. Their potential in structured data querying has also become increasingly important. LLMs have shown a strong ability to map natural language intents to executable queries. They have also achieved significant progress in general Text-to-SQL tasks [3–5,6]. This provides a new technical approach for non-technical users to access databases through natural language.

At the same time, power systems are undergoing rapid digital and intelligent transformation. Large amounts of real-time and historical data have been accumulated in dispatch operation, equipment maintenance, power generation management, and raw material procurement. These data provide an important foundation for safe grid operation and

refined management. However, current power grid data analysis methods still mainly rely on professional developers to write SQL statements. They also depend on predefined reports, fixed query templates, or drag-and-drop visualization tools. These methods cannot meet the flexible and changing self-service analysis needs of business users in practical scenarios. Power grid business users usually have strong domain knowledge and business semantic understanding. They can clearly describe business questions and constraints. However, they often lack knowledge of database schemas and SQL programming. This limits the efficiency of data value extraction. Therefore, introducing Text-to-SQL into power grid systems can reduce the technical threshold of data querying and improve query flexibility for business users.

However, power grid data and power grid queries are highly specialized and semantically complex. Existing general Text-to-SQL methods face several challenges in this scenario [7,8,9]. These challenges can be summarized as follows:

(i) **Highly abstract metadata with weak self-descriptiveness:** Table names and field names in power grid databases are highly abstract. Many names use Chinese pinyin abbreviations or mixed Chinese-English naming styles. They cannot directly reflect their business meanings. For example, the table name "DW_H_MON_DC" represents monthly historical data of power plants in grid management. The field "FYGZ" represents the electricity category of "peak, shoulder, valley, and total". Such naming rules greatly increase the difficulty of mapping database structures to business semantics.

(ii) **Complex professional expressions and ambiguous semantic boundaries:** The same business concept may have multiple synonymous expressions, such as "load" and "demand". In contrast, terms with similar surface meanings may have different business semantics. For example, "generator-side power generation" and "grid-connected power generation" both describe generated electricity. However, they differ in statistical boundaries and application scenarios. This may cause semantic mapping errors between natural language queries and underlying data semantics [10,11,12].

(iii) **Complex and inconsistent metric definitions:** The same metric name may correspond to different statistical periods, statistical objects, or spatial scopes in different tables. For example, "grid-connected power generation" may have different calculation methods and business meanings under monthly statistics, daily statistics, single power plant scope, whole-grid scope, dispatching scope, or full-scope statistics. These differences place higher requirements on semantic understanding and SQL construction in Text-to-SQL systems.

(iv) **Widespread implicit calculation logic of metrics:** In power grid business scenarios, many commonly used metrics are not stored as single database fields. Instead, they involve multi-step calculation logic. Such logic is often hidden in business rules. This increases the reasoning burden of SQL construction.

(v) **SQL hallucination under complex table structures:** Power grid data structures are complex.Without accurate metadata constraints, LLMs lack a reliable basis for identifying the correct context. As a result, they may generate SQL statements that are syntactically correct but semantically meaningless. Typical errors include incorrect table associations and invalid query logic. These errors directly affect the credibility and usability of query results [7-9,13].

Overall, these problems often lead to semantic mis-understanding, poor metadata retrieval, and hallucinated schema completion in power grid scenarios. The generated SQL statements may fail to execute or may return irrelevant results. Therefore, the core bottleneck of Text-to-SQL in the power grid domain is not only SQL generation itself. It mainly lies in three aspects: accurate semantic representation of domain metadata, efficient metadata retrieval, and effective multi-level ambiguity resolution.

To address these challenges, this paper proposes SQL-GRID, a grounded retrieval and interactive disambiguation agent for Text-to-SQL over power grid data. SQL-GRID adopts a multi-agent collaboration mechanism [14-17]. It constrains and enhances the SQL generation process through several key stages, including natural language understanding, domain semantic retrieval, ambiguity resolution, and query result verification. The goal is to provide reliable natural-language-driven self-service query capability for power grid business users.

The main contributions of this paper are summarized as follows:

(i) **A business knowledge construction method for power grid data:** To address the problems of highly abstract database naming and implicit business rules in power grid databases, this paper integrates database structural information, business documents, statistical rules, and real data samples to construct a unified Business Knowledge Representation. The proposed representation provides domain knowledge that is both understandable and reasoning-oriented, thereby offering effective support for retrieval augmentation and SQL generation.

(ii) **A high-recall and low-noise multi-strategy retrieval-augmented generation method:** To address the semantic complexity and ambiguous semantic boundaries of power grid data, this paper designs a multi-strategy RAG method that combines semantic vector retrieval and keyword retrieval. This method improves metadata recall while reducing irrelevant retrieval noise. It provides accurate metadata support and concise contextual information for Text-to-SQL generation.

(iii) A semantic-enhanced ambiguity resolution strategy for ambiguous metrics: To address terminology ambiguity, metric definition ambiguity, and statistical scope ambiguity

in power grid business queries, this paper proposes an ambiguity resolution strategy driven by Business Knowledge Representation. The strategy combines ambiguity detection, user-oriented interactive selection, and post-generation SQL validation. It effectively reduces semantic mapping errors and SQL hallucination. It also improves the accuracy and interpretability of generated SQL statements.

Experimental results show that the proposed framework performs well on a power grid business dataset. The average metadata recall rate reaches 94%. Compared with the traditional baseline method, the SQL execution rate increases from 40.2% to 84.6%. The average semantic matching score reaches 4.85 out of 5. These results verify the effectiveness of the proposed method in complex power grid scenarios.

2. Related Work

2.1. Text-to-SQLResearch Progress of LLM-based Text-to-SQL

Large language models (LLMs) have recently attracted significant attention in SQL generation due to their strong text generation capability [1]. LLM-based Text-to-SQL systems have gradually evolved into a modular collaborative paradigm. For example, DIN-SQL[3] and MAC-SQL [4] adopt multi-agent frameworks to decompose tasks into multiple subtasks, thereby improving execution efficiency. MCS-SQL [5], DAIL-SQL [18], and Chase-SQL [6] also follow a workflow that retrieves relevant context, selects schemas, and synthesizes SQL queries.However, these systems generally assume that natural language queries are precisely expressed. When queries contain ambiguity, these methods are prone to hallucination issues, which reduce system reliability [7, 8]. This work also adopts a modular collaborative framework and further introduces ambiguity detection and resolution mechanisms for power grid query scenarios.

2.2. Retrieval-Augmented Generation

The performance of LLMs on knowledge-intensive natural language tasks heavily depends on the accuracy and relevance of the available contextual information [2,19,20]. Retrieval-Augmented Generation (RAG) explicitly introduces external document retrieval modules into the generation process. It is widely regarded as an important technical paradigm for addressing the limitations of LLMs in knowledge coverage and update capability [9].With the development of intelligent agents, RAG systems have gradually evolved into modular and complex intelligent architectures, with retrieval quality becoming a major research focus. Studies [21-23] improve retrieval decisions through reinforcement learning, adaptive retrieval, and self-reflective generation. Deep-RAG [24] provides a flexible end-to-end solution that enables arbitrary models to progressively retrieve information according to their evolving reasoning process. In addition, some methods [25] improve retrieval efficiency by constructing high-quality knowledge representations.Although the context windows of LLMs continue to expand, models may still fail to effectively use relevant information when it is embedded in long and redundant contexts [18]. Therefore, recent RAG research increasingly emphasizes high-quality and concise knowledge representations [26]. This work focuses on structured knowledge representation and efficient retrieval in the power grid domain, aiming to achieve seamless integration between structured information and LLM reasoning.

2.3. Semantic Ambiguity Resolution

Existing studies have extensively investigated query ambiguity in Text-to-SQL tasks. According to the intervention stage in the processing pipeline, current methods can generally be categorized into three groups: preprocessing, in-processing, and post-processing methods.Preprocessing methods aim to resolve ambiguity before SQL generation. For example, APA [10] fine-tunes LLMs to actively generate clarification requests for ambiguous inputs. User interaction is then utilized to complete ambiguity resolution before the generation stage.In-processing methods focus on identifying and resolving ambiguity during SQL generation. RTS [11] concentrates on the schema linking stage. It utilizes pretrained classifiers on LLM hidden representations and combines consistency prediction techniques to provide reliable probabilistic guarantees for error detection during translation.Post-processing methods mainly refine and optimize initially generated SQL statements. SQLens [13] trains classifiers to predict potential errors and then performs correction and auditing to improve SQL quality. The method proposed in [12] generates multiple interpretation candidates using LLMs and employs pretrained models to complete missing information, thereby producing semantically clearer SQL outputs.This work mainly addresses various ambiguous expressions in power grid business queries. Corresponding ambiguity handling strategies are introduced in all three stages.

2.4. Multi-Agent Collaborative Framework

Multi-agent collaboration aims to utilize collective intelligence to solve complex problems that are difficult for a single agent to handle. In terms of collaboration mechanisms, CAMEL [14] first introduced a role-playing framework for automatic task decomposition. MetaGPT [15] further incorporated Standard Operating Procedures (SOPs) to standardize workflows among agents. Regarding collaboration topology, GPTSwarm [16] proposed an optimizable graph structure to adapt to dynamic task requirements. In addition, studies such as AgentVerse [17] demonstrated that social interaction environments can effectively stimulate emergent capabilities, improve reasoning quality, and suppress hallucinations. Multi-agent systems can achieve collaborative reasoning capabilities beyond those of individual agents. They effectively reduce hallucinations and improve the ability to solve complex

problems. Similarly, this work implements an interactive Text-to-SQL system based on a multi-agent collaborative framework.

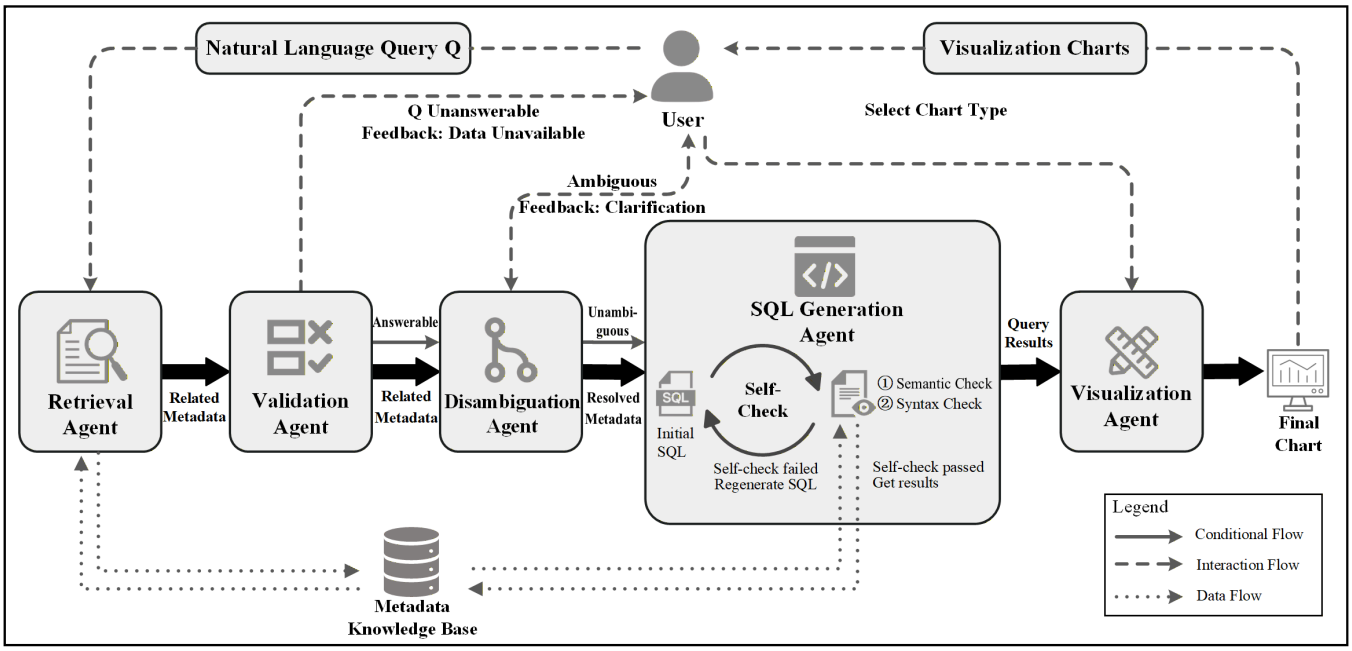

Figure 1. Architecture of SQL-GRID for Text-to-SQL over Power Grid Data

3. Multi-Agent Text-to-SQL Interactive Assistance System for Power Grid Data

To address the limitation that conventional power grid data analysis methods cannot support flexible and self-service analysis needs, this paper proposes SQL-GRID, a Text-to-SQL interactive assistance system based on multi-agent collaboration. The overall architecture is shown in Figure 1. The system consists of five core agents: the Retrieval Agent, Validation Agent, Disambiguation Agent, SQL Generation Agent, and Visualization Agent. These agents work together to achieve end-to-end processing from natural language intent to visualized analytical reports. This section introduces the overall system architecture and the functions of each agent. The key technical details, including the retrieval method and the disambiguation mechanism, are presented in Sections 4 and 5, respectively.

3.1. Framework Design

Framework Workflow

As shown in Figure 1, the agents cooperate along the main process of metadata perception, query feasibility validation, semantic disambiguation, SQL generation and verification, and result visualization. This process forms a complete Text-to-SQL workflow for power grid business queries. Given a database query requirement Q submitted by a power grid business user, the execution process of the multi-agent framework is described as follows:

- **Retrieval and validation stage.** Based on the semantic understanding of Q, the Retrieval Agent retrieves relevant candidate metadata from the external knowledge base. The candidate metadata are then passed to the Validation Agent. The Validation Agent determines whether the current query can be answered by the data source. If the data source cannot support Q, the system informs the user that the current query is not answerable.
- **Ambiguity detection and resolution stage.** If Q is answerable but contains semantic ambiguity, the Disambiguation Agent triggers an interactive clarification mechanism. This mechanism helps the user identify the target metadata. Multi-round interaction continues until the query intent becomes clear and unambiguous.
- **SQL generation and self-verification stage.** The SQL Generation Agent constructs SQL statements based on the ambiguity-free context. It also performs dual self-verification from both semantic and syntactic perspectives. If verification fails, the system automatically performs iterative correction until the SQL statement passes all checks.
- **Visualization stage.** The data results obtained by executing the final SQL statement are rendered into charts by the Visualization Agent.

The whole process supports user interaction. For example, when validation fails, the system promptly informs the user that the current query is not answerable. When ambiguity exists, the system explains the source of ambiguity and provides candidate options for user selection. Users can also choose the preferred visualization style for the final output. This design forms a user-friendly interactive system.

Agent Functions

Retrieval Agent. The Retrieval Agent provides the data foundation for the entire multi-agent framework. It is responsible for accurate metadata retrieval. The input of this agent is the user natural language query Q. The agent adopts a multi-strategy RAG retrieval mechanism, which is described in Section 4, to perform semantic matching over the Semantic-Enhanced Metadata Knowledge Base in the power grid domain. It combines vector-based semantic retrieval and BM25 keyword matching for parallel retrieval. It also applies deduplication fusion and path reachability filtering. This ensures that the retrieved candidate metadata are valid in both semantic relevance and structural availability. This process provides concise and reliable contextual support for the subsequent validation and disambiguation stages.

Validation Agent. The Validation Agent determines whether the current query can be answered by the business database. It receives the candidate knowledge retrieved by the Retrieval Agent together with the user query Q. After performing semantic understanding of Q with the assistance of an LLM, the agent initiates the validation process. The validation process consists of three sequential checks: business scope coverage, data attribute completeness, and business rule satisfaction. If any validation step fails, the system explicitly reports why the query cannot be answered and prompts the user to refine the query, thereby preventing unnecessary downstream processing.

Disambiguation Agent. The Disambiguation Agent detects and resolves ambiguity in queries that are answerable but semantically unclear. Based on semantic-enhanced metadata, the agent compares highly similar candidate metadata from several dimensions, including terminology, business meaning, statistical scope, and temporal granularity. When multiple candidate metadata entries can answer the user query but contain key differences, the system presents the candidate options and their distinguishing features to the user through an interactive interface. For example, for the query power plant generation", both monthly power generation" and "daily power generation" may answer the query, but they differ in temporal granularity. By making ambiguity explicit and asking the user to confirm the intended option, the system effectively reduces the risk of semantic hallucination in subsequent SQL generation.

SQL Generation Agent. The SQL Generation Agent generates SQL statements based on the ambiguity-free query intent and the support of enhanced metadata. The system feeds the semantic parsing result and metadata descriptions into the LLM as contextual input. The LLM then generates an initial SQL statement. The generated SQL then enters the self-verification stage. On the semantic side, the system performs semantic back-translation to verify the consistency between the SQL statement and the original query. On the syntactic side, the system checks syntax correctness and executability in a sandbox environment. If verification fails, iterative regeneration is triggered until a semantically consistent and executable SQL statement is obtained.

Visualization Agent. The Visualization Agent is responsible for visualizing the SQL query results as charts. The agent selects appropriate chart types according to the query semantics and data characteristics, such as bar charts or line charts. For example, trend charts are preferred for time-series metrics to ensure that the results are intuitive and easy to understand.

3.2. Key Technical Challenges

In the complete Text-to-SQL generation process, SQL generation accuracy depends not only on the reasoning ability of LLMs. It also depends heavily on their ability to perceive power grid domain knowledge and resolve complex business semantics. A comprehensive analysis shows that insufficient knowledge perception and amplified semantic ambiguity are two key bottlenecks that affect system performance. Specifically, the system faces the following two technical challenges:

- **Challenge 1: Insufficient metadata perception under semantic-structure disconnection.** During long-term evolution, power grid databases have formed highly abstract and weakly self-descriptive metadata. Problems such as compressed naming and implicit business semantics are common. As a result, structured metadata cannot explicitly express actual business meanings. Without semantic-enhanced representation and effective retrieval mechanisms, the model cannot accurately establish the mapping between query intent and database structures.
- **Challenge 2: SQL hallucination caused by complex business ambiguity.** Power grid business queries often contain multiple forms of semantic ambiguity, such as inconsistent statistical scopes and unclear temporal granularity. Users usually have limited knowledge of database contents. If the system lacks an explicit clarification mechanism, the model may make implicit semantic assumptions. This may lead to SQL statements that are inconsistent with actual business requirements.

To address these challenges, this paper proposes systematic solutions in the following sections:

For Challenge 1, Section 4 proposes a semantic-structure aligned knowledge enhancement and retrieval method for power grid data. By constructing a Business Knowledge Index that integrates business semantics, statistical rules, and real data characteristics, and by combining it with a multi-strategy RAG retrieval mechanism, the proposed method improves the perception of domain knowledge.

For Challenge 2, Section 5 proposes a closed-loop SQL Hallucination suppression method consisting of "pre-

generation disambiguation → interactive clarification →post-generation validation". The method combines semantic enhancement, user interaction, and iterative SQL verification. It continuously detects and corrects ambiguities and execution errors during SQL generation. This ensures the semantic consistency and executability of the final SQL statement.

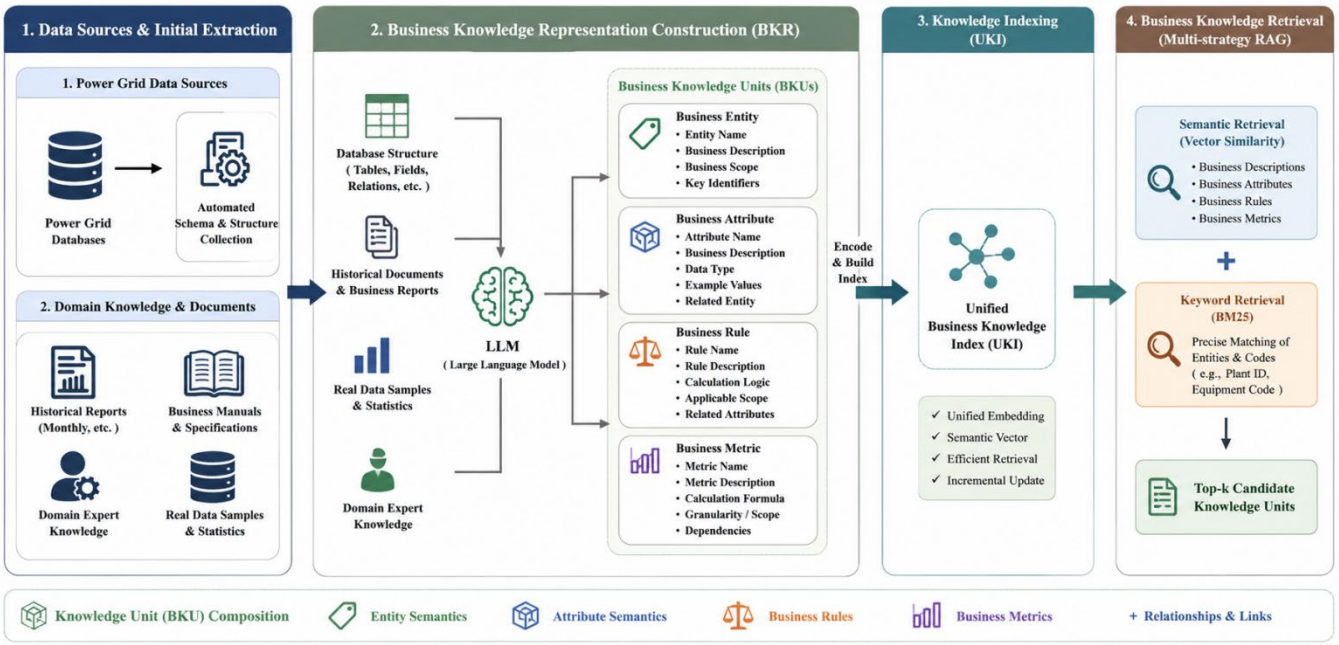

Figure 2. Construction of the Business Knowledge Representation

4. Semantic-Structure Aligned Business Knowledge Perception and Retrieval Method for Power Grid Data

As discussed in Section 3.2, the core bottleneck of Text-to-SQL in power grid scenarios is not the lack of SQL generation capability. Instead, LLMs struggle to consistently perceive the highly abstract database structures and the implicit business semantics embedded in power grid data. When query semantics cannot be accurately aligned with the underlying data structures, the model is prone to generating incorrect field selections, invalid table associations, and other semantic errors.

To address this issue, this paper proposes a Semantic-Structure Aligned Business Knowledge Perception and Retrieval Method for power grid data. Rather than relying directly on raw database schemas, the proposed method focuses on the business semantics and structural constraints of power grid databases. By constructing a Business Knowledge Representation centered on business semantics and combining it with a multi-strategy retrieval mechanism and context refinement strategy, the method provides accurate and reliable domain knowledge for SQL generation.

4.1. Construction of Business Knowledge Representation

Power grid databases are characterized by highly abstract naming conventions, implicit business rules, and complex statistical definitions. To address these challenges, this work constructs a unified Business Knowledge Representation (BKR). Instead of directly relying on database schemas, the proposed representation integrates database structural information, historical business documents, real data samples, and domain expert knowledge to organize business information required for query understanding into a collection of Knowledge Units.

Each Knowledge Unit contains not only structural descriptions of the underlying data but also integrates business semantics, statistical rules, representative example values, and semantic relationships. This representation enables LLMs to understand business concepts directly without depending on the naming conventions of the underlying database objects. The overall construction process is illustrated in Figure 2.

Formally, a Knowledge Unit is defined as Equation (1), where D_i denotes the structural description of the underlying data object,A_i denotes the attribute semantic set, R_i denotes the associated business rule set, E_i denotes representative example values, and L_i denotes semantic links to other Knowledge Units. The complete Business Knowledge Representation is then formulated as Equation (2).

$$KU_i = \{D_i, A_i, R_i, E_i, L_i\} \quad (1)$$

$$BKR = \{KU_1, KU_2, \cdots, KU_n\} \quad (2)$$

Business semantic description generation. The system combines database structural information, attribute composition, relationship constraints, and historical business documents. An LLM is employed to automatically generate business-oriented descriptions and identify the corresponding
business scenario and statistical scope, enabling raw database objects to be associated with explicit business semantics.

Attribute semantic enrichment. The system further incorporates attribute types, real data samples, and business context to generate natural language explanations for important business attributes involved in user queries. Representative example values are also provided to improve the model's understanding of encoded attributes, status attributes, and textual attributes.For an attribute a_{ij} in Knowledge Unit KU_i , its semantic representation is formulated as Equation (3),where t_{ij} is the attribute type,E_{ij} denotes example values, C_i denotes the business context of KU_i , and $\mathrm{f}_{\mathrm{attr}}(\cdot)$ represents the LLM-based attribute semantic enrichment function.

$$A_{ij}^{sem} = f_{attr}(a_{ij}, t_{ij}, E_{ij}, C_i) \quad (3)$$

Business rule enrichment. For business statistical rules that cannot be directly represented by the database, the system constructs business rule descriptions based on historical reports and domain expert knowledge. These rules are further associated with the underlying data structures to support complex metric interpretation and SQL reasoning.Formally, a business rule r_k is represented as Equation (4), where m_k denotes the target metric, $P_k$denotes the calculation procedure, $C_k$denotes applicable conditions, and $G_k$denotes the statistical granularity or scope. The rule-to-data association is defined as Equation (5),where $\phi(\cdot)$ maps a business rule to the required data attributes.

$$r_k = \{m_k, P_k, C_k, G_k\}. \quad (4)$$

$$\phi(r_k) = \{a_1, a_2, \cdots, a_p\}. \quad (5)$$

Finally, each Knowledge Unit is encoded into a unified semantic vector representation and organized within a Unified Knowledge Index, which enables efficient retrieval during subsequent query processing. As shown in Equation (6) and Equation (7), Enc(·) denotes the text embedding function and v_i is the vector representation of KU_i.

$$v_i = Enc(KU_i). \quad (6)$$

$$UKI = \{(KU_i, v_i)\}_{i=1}^{n}. \quad (7)$$

Knowledge quality control. To reduce potential semantic errors introduced during automatic BKR construction, each Knowledge Unit is checked before being incorporated into the Unified Knowledge Index. First, structural consistency is verified by comparing table names, field names, data types, and relationships against the original database schema, thereby preventing nonexistent database objects or invalid structural relationships from being introduced. Second, business-semantic information, including metric definitions, statistical scopes, and calculation rules, is checked against the source business documents, rule descriptions, and available domain knowledge. Knowledge Units containing unresolved conflicts or inconsistent information are marked for further verification before being updated in the knowledge index. This quality-control process reduces the propagation of semantic errors from automatically generated descriptions and improves the reliability of the knowledge used for retrieval and SQL generation.

4.2. Retrieval Context Refinement and Scale Control for SQL Generation

To simultaneously support the diverse semantic expressions in power grid business queries and the precise matching of entity identifiers, this work designs a dual-path retrieval mechanism that combines semantic vector retrieval with BM25 keyword retrieval. A weighted fusion strategy together with path consistency constraints is further introduced to improve the quality of retrieved candidate knowledge.

Given a user query Q which is defined as Equation (8), the system first employs an LLM to extract key semantic elements, including business entities, metric names, and filtering conditions, and then constructs a semantic vector representation of the query. During the retrieval stage, two retrieval paths are executed in parallel. On the one hand, semantic vector retrieval is performed by computing the cosine similarity between the query vector and different types of business knowledge, including business semantic descriptions, business attributes, and business rules, to retrieve the most relevant candidate Knowledge Units. For semantic vector retrieval, the relevance score between Q and KU_i is computed as Equation (9).On the other hand, BM25 keyword retrieval is employed to accurately match domain-specific entities, such as power plant IDs and equipment codes.For keyword retrieval, the BM25 score is computed as Equation (10). The retrieval results from the two paths are combined using a weighted fusion strategy, as shown in Equation (11). Here, α denotes the weighting parameter. In this work, α is set to 0.65 to achieve a good balance between semantic retrieval capability and entity matching precision. The fusion weight α and the number of retained candidates K were determined using the development set and were kept fixed during evaluation on the test set. We compared several candidate configurations by jointly considering metadata recall, exact entity matching, and retrieval-context size. The configuration α=0.65 and K=8 provided the best overall balance between semantic retrieval and keyword-based exact matching and was therefore adopted in all subsequent experiments. The test set was not used for parameter selection.

$$q = Enc(Q). \quad (8)$$

$$s_i^{vec} = cos(q, v_i) = \frac{q \cdot v_i}{\|q\|\|v_i\|}. \quad (9)$$

$$s_i^{bm25} = BM25(Q, KU_i). \quad (10)$$

$$s_i^{final} = \alpha \cdot s_i^{vec} + (1 - \alpha) \cdot s_i^{bm25}. \quad (11)$$

When conflicts occur between the two retrieval paths, the system applies a conflict resolution rule. For example, if the same Knowledge Unit is retrieved by both methods but the

ranking difference is greater than three positions, the vector retrieval result is prioritized because vector retrieval is better at capturing implicit semantics. The final candidate set is formulated as Equation (12). This operation removes duplicate Knowledge Units and retains only the most relevant candidates for SQL generation.

$$C_k = TopK(\{KU_i\}_{i=1}^{n}, s_i^{final}). \quad (12)$$

After deduplication based on metadata IDs, the system generates a concise candidate set. This mechanism effectively suppresses retrieval noise while maintaining high recall.

4.3. Business-Semantics-Oriented Knowledge Base Retrieval

Excessive retrieval information may cause contextual redundancy and interfere with the reasoning attention of LLMs. To avoid this problem, this work introduces a strict context refinement strategy during the generation stage. During context construction, the system only preserves the final Top-K relevant metadata. Intermediate retrieval information, such as similarity scores and tool invocation traces, is excluded from the main context. This design prevents the model from performing unnecessary secondary reasoning over the retrieval process itself during SQL generation. It also reduces attention distraction.

Through the above strategy, the system significantly reduces token consumption while preserving sufficient semantic information. As a result, the model can focus more effectively on critical business logic. This improves the generation stability and result accuracy of the multi-agent Text-to-SQL system in power grid scenarios.

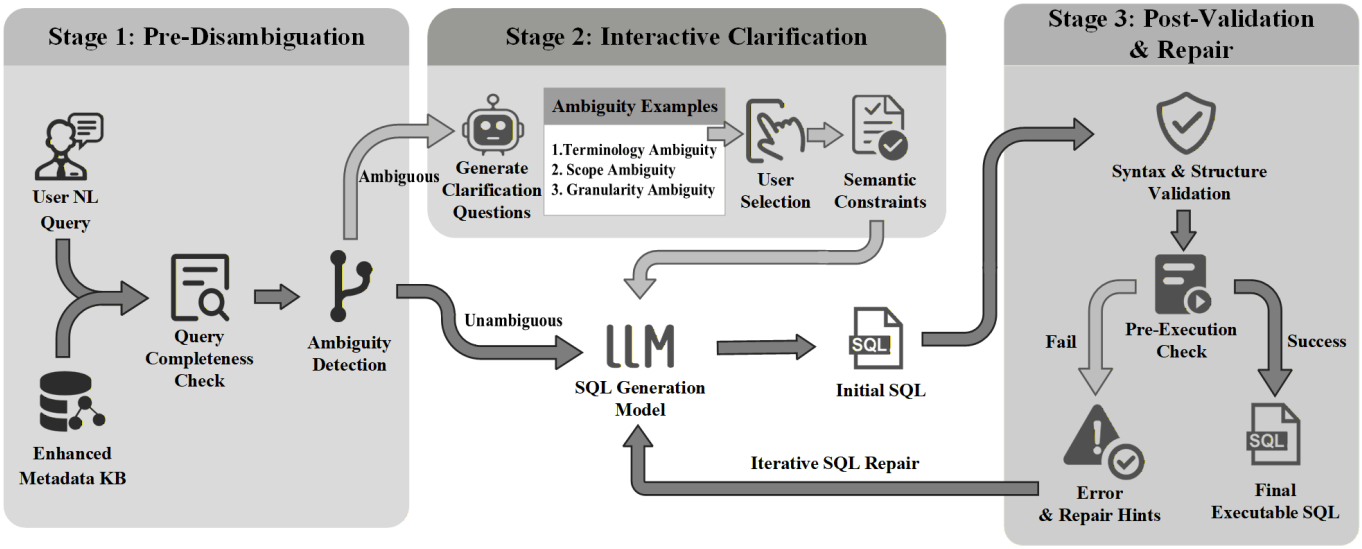

Figure 3. Multi-Stage SQL Hallucination Suppression Framework

5. Semantic-Enhanced Ambiguity Resolution Method

After constructing the Semantic-Enhanced Metadata Knowledge Base, the LLM obtains sufficient domain knowledge support for power grid scenarios. However, in real business queries, enhanced metadata retrieval alone is still insufficient to fully avoid SQL generation errors. On the one hand, user queries often contain imprecise descriptions or implicit constraints, which may introduce semantic ambiguity. On the other hand, SQL is a highly structured language. It has strict requirements for syntax and execution conditions. If these problems are not properly handled, the model may generate SQL statements that are semantically inconsistent or non-executable. This leads to the problem of SQL hallucination.

To address this problem, this work introduces a multi-stage ambiguity resolution and hallucination suppression mechanism throughout the Text-to-SQL pipeline. The mechanism covers the stages before, during, and after SQL generation. It combines pre-generation ambiguity detection, interactive semantic clarification, and post-generation validation and repair. In this way, it forms a closed-loop SQL quality control process.

5.1. Definition and Classification of Semantic Ambiguity in Power Grid Queries

This work defines “semantic ambiguity” as the existence of multiple reasonable database mapping paths under a given user query and the current metadata knowledge base. In power grid scenarios, a query may correspond to multiple reasonable interpretations under different business contexts or statistical scopes. To address this problem, this work

identifies three major types of semantic ambiguity. Their definitions and detection rules are described below.

Terminology ambiguity. If a business term in the query matches multiple semantically similar fields in field-level metadata, the query is identified as containing terminology ambiguity. For example, the term "power generation" may correspond to generator-side power generation, grid-connected power generation, or total power generation in the same table. This type of ambiguity is mainly detected by checking whether the semantic similarity among candidate fields exceeds a predefined threshold.

Statistical scope ambiguity. If candidate metadata correspond to multiple tables with similar semantics but different statistical scopes, the query is identified as containing statistical scope ambiguity. For example, DDKJ_H_DAY_DW" and DW_H_DAY_DW" represent daily grid electricity data under the dispatching scope and the whole-grid scope, respectively. Both tables are highly relevant to the query "daily grid electricity". This type of ambiguity is mainly detected by checking whether candidate tables share the same business topic but differ in statistical scope.

Granularity ambiguity. If the query does not explicitly specify temporal or spatial granularity, while the candidate metadata contain multiple granularity levels, the query is identified as containing granularity ambiguity. For example, the query calculate power generation" may correspond to the monthly power generation table DW_H_MON_DC" or the daily power generation table "DW_H_DAY_DC".

If these ambiguities are not identified before SQL generation, the model may make unstable probabilistic guesses. This can lead to hallucinated SQL generation. Therefore, an explicit ambiguity resolution process is necessary.

Ambiguity detection criteria. To further clarify the triggering conditions of semantic ambiguity, the system combines the number of candidate mappings, semantic relevance, and differences in business attributes. Terminology ambiguity is triggered when a query term can be mapped to at least two candidate fields whose semantic relevance exceeds a predefined threshold τ. Statistical-scope ambiguity is triggered when two or more candidate tables belong to the same business topic but differ in their statistical-scope attributes. Granularity ambiguity is triggered when the user query does not explicitly specify temporal or spatial granularity, while two or more valid candidate mappings with different granularities are retrieved. In our implementation, τ is set to 0.85 based on the development set and is kept fixed throughout all experiments.

5.2. Multi-Stage Semantic Resolution and SQL Hallucination Suppression Mechanism

To address semantic uncertainty commonly found in power grid business queries, this work proposes a multi-stage ambiguity resolution and hallucination suppression mechanism. The mechanism consists of "pre-generation semantic disambiguation → interactive semantic clarification →post-generation syntax validation and repair". The overall workflow is illustrated in Figure 3.

Algorithm 1 Multi-stage Semantic Disambiguation Process

Require: User query Q, Retrieved Knowledge Units $\mathcal{C}$
Ensure: Clarified query Q^*
Extract semantic intents and constraints from Q:

$$I \leftarrow \text{LLM\_Extract}(Q)$$

Match extracted intents I with candidate Knowledge Units $\mathcal{C}$
if only one valid semantic mapping exists **then**
    $Q^* \leftarrow Q$
else
    Identify ambiguous semantic dimensions
    Generate clarification options $\mathcal{O}$
    Request user selection u from $\mathcal{O}$
    Update query constraints with u
    $Q^* \leftarrow \text{Update}(Q, u)$
end if
return Q^*

Pre-generation stage: semantic disambiguation based on enhanced metadata

The system first takes the user query and the retrieved highly relevant Knowledge Units as input. As described in Algorithm 1, the LLM extracts semantic intents and constraints from the query and performs semantic completeness validation. The model compares the business concepts in the query with the descriptions of candidate Knowledge Units and detects one-to-many mapping conflicts, such as a query term matching multiple statistical scopes at the same time.

Generation stage: interactive semantic clarification and explicit constraint specification

As described in Algorithm 1, When semantic ambiguity is detected, the system automatically generates targeted clarification questions according to the ambiguity type and the business descriptions in the enhanced metadata. The corresponding metadata information is also presented to the user. Taking statistical scope ambiguity as an example, the system asks the user to confirm the data source: A. DDKJ_H_DAY_DW" (daily grid electricity data under the dispatching scope); B. DW_H_DAY_DW" (daily grid electricity data under the whole-grid scope). This interaction guides the user to explicitly specify the desired statistical scope. The user's selection is then transformed into an explicit semantic constraint. It is combined with the original query as the input condition for SQL generation.

Post-generation stage: executability-oriented syntax validation and iterative repair

Even after pre-generation ambiguity resolution and interactive clarification, the generated SQL may still fail due to syntax errors, invalid structural associations, or inconsistent filtering conditions. To further suppress SQL hallucination, this work introduces a closed-loop validation and repair mechanism. The complete SQL hallucination suppression process is summarized in Algorithm 2. The algorithm evaluates generated SQL from both execution and semantic consistency perspectives and iteratively repairs invalid SQL statements based on validation feedback. For example, the actual format of a date field may be 2025-11-04", while the generated SQL incorrectly uses 2025.11.04". To address this problem, the system introduces an executability-oriented validation and repair mechanism after SQL generation. First, based on the confirmed semantic constraints, the system checks the syntax and structural validity of the generated SQL. The checked items include table-field relationships, the legality of aggregation function usage, and the completeness of multi-table join conditions. Then, the system performs pre-execution verification to test SQL executability. If validation fails, the system feeds the error information and repair instructions back to the generation model. It then triggers targeted local regeneration. After a limited number of validation and repair iterations, the system outputs a final SQL statement that is semantically consistent and stably executable. As illustrated in Algorithm 2, the proposed mechanism forms a closed-loop process from SQL generation, execution validation, error analysis, and iterative repair.

Algorithm 2 SQL Hallucination Suppression and Validation Process

Require: Clarified query Q^*, Retrieved Knowledge Units $\mathcal{C}$
Ensure: Valid SQL statement S

Generate initial SQL statement:

$$S^{(0)} \leftarrow \text{GenerateSQL}(Q^*, \mathcal{C})$$

$t \leftarrow 0$
while maximum repair iterations not reached **do**
  Execute SQL $S^{(t)}$ in the database
  if SQL execution succeeds **then**
    Perform semantic consistency validation between $S^{(t)}$ and Q^*
    if semantic consistency is satisfied **then**
      return $S^{(t)}$
    end if
  end if
  Analyze execution errors or semantic conflicts:

$$E^{(t)} \leftarrow \text{AnalyzeError}(S^{(t)})$$

  Repair SQL based on validation feedback:

$$S^{(t+1)} \leftarrow \text{Repair}(S^{(t)}, E^{(t)}, Q^*)$$

  $t \leftarrow t + 1$
end while
return final SQL statement $S^{(t)}$

6. Experimental Results and Analysis

To verify the effectiveness of the proposed SQL-GRID system for Text-to-SQL over power grid data in complex power grid business scenarios, this paper conducts experimental analysis from several aspects, including overall performance, module contribution, retrieval quality, interactive disambiguation effectiveness, and system efficiency. The experiments are designed to evaluate whether semantic-enhanced metadata can alleviate the semantic-structure gap in power grid data, whether multi-strategy retrieval can improve metadata recall quality, whether Interactive Disambiguation can reduce semantic deviation in SQL generation, and whether the complete

system can maintain an acceptable interaction cost while improving accuracy.

6.1. Experimental Settings and Evaluation Metrics

Dataset construction. To realistically reflect the semantic understanding challenges in power grid business scenarios, this paper constructs an evaluation dataset containing 500 representative natural language queries. The dataset covers five core business domains: meteorological risk monitoring, fuel market, equipment maintenance, real-time dispatching, and operation analysis. Each domain contains 100 queries. To ensure that the evaluation samples reflect the complexity of power grid Text-to-SQL tasks, each query contains at least one of the following characteristics: unclear business terminology or multiple synonymous expressions; unspecified statistical scope or temporal granularity; implicit metric calculation; multi-table association or complex filtering conditions. For each sample, domain experts manually annotate the corresponding standard SQL statement, related tables and fields, ambiguity type, and expected query result. These annotations are used as Ground Truth for experimental evaluation. The composition of the dataset is shown in Table 1.

Table 1. Composition of the Evaluation Dataset

Business Domain	Number of Queries	Main Ambiguity Type
Meteorological Risk Monitoring	100	Time range and regional range
Fuel Market	100	Market source and statistical scope
Equipment Maintenance	100	Equipment type and code mapping
Real-Time Dispatching	100	Dispatching scope and time window
Operation Analysis	100	Metric definition and daily/monthly granularity

Experimental environment. All experiments are conducted under the same software and hardware environment. The base model used in this work is qwen3-coder:30b-a3b-q8_0. Only the metadata enhancement method, retrieval strategy, Interactive Disambiguation module, and post-generation validation module are configured differently across experiments. This setting ensures a fair comparison.

Comparison methods. This paper sets the following comparison methods:

(i) **Schema-only:** Only the raw database schema and the user query are provided to the LLM. No semantic enhancement, retrieval augmentation, or Interactive Disambiguation is used.

(ii) **General RAG:** A knowledge base is constructed based on the raw schema. Vector retrieval is used to retrieve relevant tables and fields. However, semantic-enhanced information such as business descriptions, example values, metric definitions, and calculation logic is not included.

(iii) **Semantic-Enhanced RAG:** Enhanced information, including business descriptions, real data samples, statistical rules, and calculation knowledge, is introduced. Retrieval is then performed based on this enhanced information.

(iv) **Semantic Enhancement+Multi-Strategy Retrieval:** Based on semantic-enhanced metadata, both BM25 keyword retrieval and semantic vector retrieval are used for multi-path recall and fusion ranking.

(v) **Complete System:** Based on semantic-enhanced metadata and multi-strategy retrieval, the complete system further introduces pre-generation ambiguity detection, interactive semantic clarification, and post-generation SQL validation and repair.

Evaluation metrics. This paper uses the following metrics to evaluate system performance:

(i) **SQL execution rate:** The proportion of generated SQL statements that can be executed in the database without errors.

(ii) **Average semantic matching score:** This metric evaluates the consistency between the generated SQL and the original user query intent. The score is evaluated by domain experts with the assistance of LLMs. The maximum score is 5. The semantic matching score was independently assessed by two domain evaluators with experience in power-grid business analysis. For each query, the evaluators compared the original user request, the ground-truth SQL, and the generated SQL according to the criteria in Table 2. The LLM was used only to assist in interpreting the business semantics of SQL statements and did not directly determine the final score. When the difference between evaluator scores exceeded 1, the sample was re-examined and the final score was determined after discussion. This procedure was adopted to reduce subjectivity and improve scoring consistency.The detailed scoring criteria are shown in Table 2.

(iii) **Metadata recall rate:** This metric measures the proportion of manually annotated relevant tables or fields that are hit in the Top-K candidate metadata returned by retrieval. It includes table_recall, column_recall, and avg_recall.

(iv) **Average repair rounds:** This metric measures the average number of iterations required by the validation and repair mechanism after SQL generation.

(v) **Average interaction rounds:** This metric measures the average number of clarification rounds initiated by the system to resolve ambiguity.

(vi) **Average response time:** This metric measures the average time required by the system from receiving a query to outputting the SQL statement and result explanation. The actual thinking time of the user is not included.

(vii) **Average token consumption:** This metric measures the average number of tokens consumed in one query process, including prompts, retrieval context, clarification content, and model outputs.

Table 2. Semantic Matching Scoring Criteria

Semantic Matching Score	Scoring Criteria
5	The generated SQL fully matches the business semantics. The tables, fields, filtering conditions, and statistical scopes are all correct.
4	The generated SQL generally matches the business semantics, with only minor deviations that do not affect the core business intent of the query.
3	The generated SQL partially matches the business semantics, but contains obvious deviations in business understanding.
2	The generated SQL only matches partial keywords and fails to correctly represent the business intent.
1	The generated SQL is largely unrelated to the semantics of the user query.
0	No valid SQL statement is generated.

6.2. Overall Performance Comparison and Module Contribution Analysis

This experiment compares the overall performance of different system configurations. It also analyzes the contribution of each core module by gradually introducing different modules. Specifically, this paper compares the Schema-only method using only the raw database schema, General RAG with vector retrieval, Semantic-Enhanced RAG with semantic-enhanced metadata, Semantic Enhancement+ Multi-Strategy Retrieval with both BM25 and vector retrieval, and the Complete System with Interactive Disambiguation and post-generation validation. The experimental results are shown in Table 3. Figure 4 shows the dual-axis comparison of the SQL execution rate and the average semantic matching score.

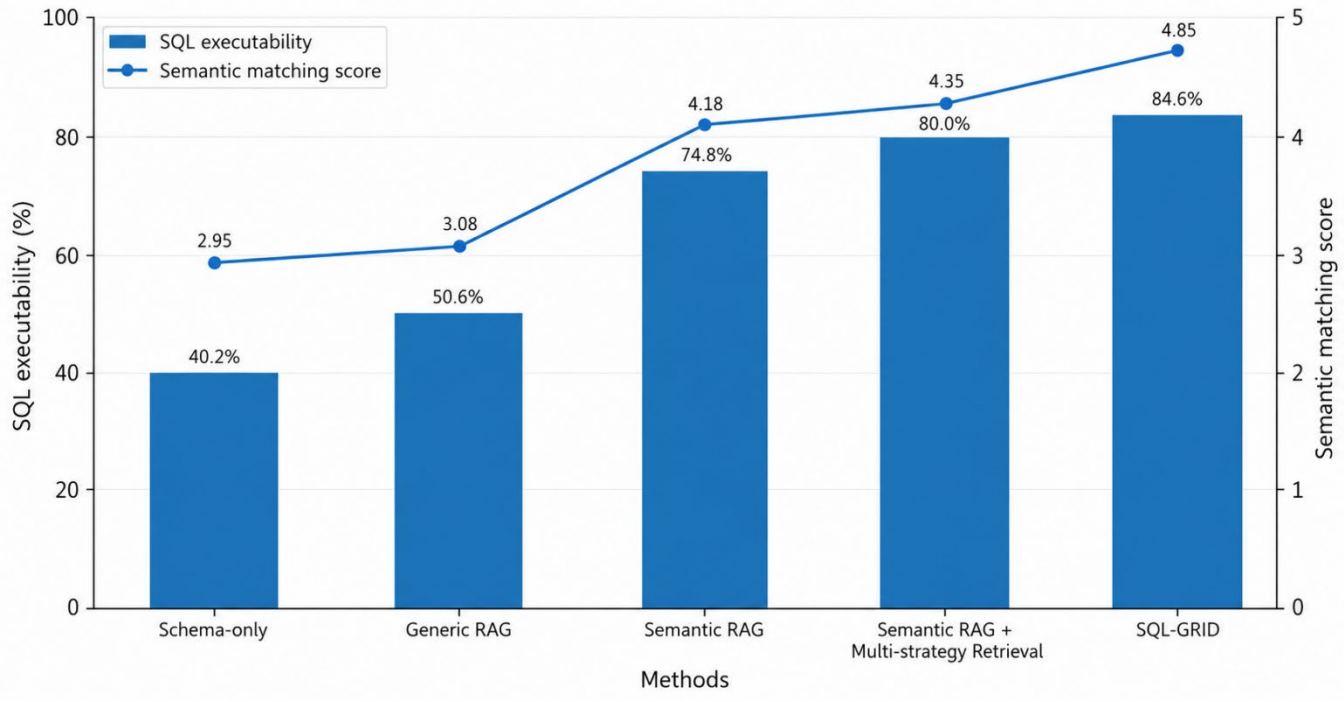

Figure 4. Comparison of SQL Executability and Semantic Matching Score among Different Methods

Table 3. Overall Performance Comparison of Different Methods

Method	Meta.	Retr.	Dis.	Val.	Exec.	Score	Repair	Time/s	Tokens
Schema-only	No	No	No	No	40.2%	2.95	3.66	99.4	24052
General RAG	No	Vector	No	No	50.6%	3.08	3.41	111.7	18260
Semantic-Enhanced RAG	Yes	Vector	No	No	74.8%	4.18	4.56	112.9	20874
SE + Multi-Strategy Retrieval	Yes	BM25+Vector	No	No	80.4%	4.35	4.62	113.5	21420
SQL-GRID	Yes	BM25+Vector	Yes	Yes	84.6%	4.85	3.46	120.2	19761

As shown in Table 3, the Schema-only method only relies on the raw database schema. It is difficult for this method to understand highly abstract table names and field names in power grid databases. Therefore, its SQL execution rate is only 40.2%, and its average semantic matching score is 2.95. General RAG introduces part of the structured context through external retrieval. It improves the SQL execution rate to 50.6%. However, the retrieved content lacks business semantic explanations. It provides limited support for field meanings, statistical scopes, and implicit metric calculations. Therefore, the average semantic matching score only increases to 3.08.

After introducing semantic-enhanced metadata, the system expands raw database objects into enriched business knowledge that incorporates business semantic descriptions, representative example values, and statistical rules. As a result, the SQL execution rate increases to 74.8%, and the average semantic matching score increases to 4.18. This result shows that the key bottleneck of power grid. Text-to-SQL is not the lack of SQL generation capability of LLMs. Instead, the key bottleneck lies in the stable perception of domain metadata semantics.

Furthermore, after combining BM25 and vector retrieval based on semantic enhancement, the system can support both exact matching of entity identifiers and semantic matching of natural language business expressions. The SQL execution rate increases to 80.4%, and the average semantic matching score increases to 4.35. This shows that Multi-Strategy Retrieval has strong complementary advantages in power grid scenarios. It can further improve metadata recall quality and downstream SQL generation performance.

Finally, based on semantic enhancement and Multi-Strategy Retrieval, the Complete System introduces Interactive Disambiguation and post-generation validation. The SQL execution rate further increases to 84.6%, and the average semantic matching score increases to 4.85. This is the best performance among all methods. The results show that Interactive Disambiguation can effectively reduce semantic deviation caused by statistical scope ambiguity, temporal granularity ambiguity, and terminology ambiguity. The post-generation validation mechanism can further filter non-executable or inconsistent SQL statements.

For the auxiliary metrics, the Complete System has an average response time of 120.2s and an average token consumption of 19761. These values change compared with the previous methods, but the performance improvement is significant. Overall, the stepwise comparison in Table 3 clearly demonstrates the contribution of each module. These results also verify the effectiveness of the formalized pipeline from Knowledge Unit construction, dual-path retrieval, interactive disambiguation, to post-generation validation. Semantic enhancement mainly improves domain understanding. Multi-Strategy Retrieval mainly improves metadata recall quality. Interactive Disambiguation and post-generation validation further improve the accuracy and robustness of the final SQL statements.

The increased response time of the complete system mainly results from the additional multi-stage processing required for reliable SQL generation, including multi-strategy knowledge retrieval, ambiguity detection, interactive clarification when necessary, and post-generation execution validation and iterative repair. In particular, validation failures may introduce additional LLM inference and database execution overhead due to targeted SQL regeneration. Therefore, the current implementation prioritizes semantic correctness and reliability for complex business queries rather than minimizing response latency. Future work will investigate parallel execution of independent agent operations, caching of frequently retrieved knowledge, more compact retrieval contexts, and lightweight validation models to further reduce end-to-end latency.

6.3. Recall Capability Analysis of the Multi-Strategy Retrieval Mechanism

This experiment evaluates the performance of three retrieval strategies, namely BM25, Vector, and BM25+Vector, in the power grid metadata retrieval task. The manually annotated relevant tables and fields are used as the Ground Truth. Precision@K, Top-1 hit rate, and context token count are used as evaluation metrics. The experimental results are shown in Table 4.

Table 4. Impact of Retrieval Strategies on Metadata Recall Quality and Retrieval Efficiency

Method	table_recall	column_recall	avg_recall	Precision @K	Top-1 Hit Rate	Context Tokens
BM25	0.88	0.63	0.76	0.83	0.67	1326
Vector	0.92	0.77	0.85	0.78	0.76	1564
BM25+Vector	**1.00**	**0.88**	**0.94**	**0.86**	**0.83**	**1840**

As shown in Table 4, BM25 achieves high Precision@K when keywords or entity identifiers are clearly expressed. However, it has limited coverage for synonymous natural language expressions and implicit business semantics. Its column_recall is only 0.63. Vector retrieval can better capture semantic associations, such as the relationship between electricity during "valley periods" and "FYGZ". Therefore, its column_recall increases to 0.77. However, Vector retrieval is slightly weaker in distinguishing similar entity identifiers and fields with close statistical scopes.After combining BM25 and Vector retrieval, the system uses both the precision of keyword matching and the semantic generalization ability of vector retrieval. The table_recall reaches 1.00, the column_recall increases to 0.88, and the average recall rate reaches 0.94. Although the fusion strategy slightly increases the number of candidate metadata items and the context token count, the increase remains within an acceptable range. Therefore, the proposed retrieval strategy can meet the real-time requirements of interactive query scenarios.

6.4. Effectiveness Analysis of the Interactive Disambiguation Module

To verify the effect of the Interactive Disambiguation module on complex business queries, the experiment shown in Table 5 is conducted. Without the ambiguity resolution module, the system can generate executable SQL statements based on semantic-enhanced metadata. However, when facing ambiguity in terminology, statistical scope, and temporal granularity, the model still tends to make implicit selections according to retrieval scores. After introducing Interactive Disambiguation, the system can identify potential one-to-many mapping relationships before SQL generation. It can also transform user selections into explicit constraints through clarification questions. As a result, the average semantic matching score increases from 4.35 to 4.85, and the SQL execution rate increases by 2.2%.

Table 5. Impact of the Disambiguation Module on SQL Semantic Alignment and SQL Executability

Method	Average Semantic Matching Score	SQL Execution Rate
Without Disambiguation Module	4.35	80.4%
With Disambiguation Module	**4.85**	**82.6%**

To further distinguish the respective contributions of Interactive Disambiguation and post-generation validation, the results in Tables 3 and 5 are jointly analyzed. Starting from the Semantic Enhancement + Multi-Strategy Retrieval configuration in Table 3, the SQL execution rate is 80.4%, and the semantic matching score is 4.35. After introducing Interactive Disambiguation, as shown in Table 5, the SQL execution rate increases to 82.6%, while the semantic matching score improves substantially to 4.85. This indicates that Interactive Disambiguation primarily improves semantic alignment by explicitly resolving ambiguity in terminology, statistical scope, and granularity before SQL generation. Based on this configuration, the complete SQL-GRID system further incorporates post-generation validation and repair, increasing the SQL execution rate from 82.6% to 84.6%, while the semantic matching score remains at 4.85. These results suggest that post-generation validation primarily improves SQL executability and robustness by detecting and repairing syntactic, structural, and execution-related errors after the semantic intent has been clarified.

This paper further analyzes the processing effect of Interactive Disambiguation according to different ambiguity types. The ambiguities in user queries are divided into terminology ambiguity, statistical scope ambiguity, and granularity ambiguity. The semantic matching scores before and after disambiguation and the ambiguity detection accuracy are compared. The experimental results are shown in Table 6. Interactive Disambiguation improves all three types of ambiguity. Among them, granularity ambiguity achieves the highest semantic matching score after disambiguation, reaching 4.90. This indicates that for

granularity-related questions, such as whether the query should be calculated by day or by month, the system can help users quickly clarify the query target through simple clarification options. Statistical scope ambiguity achieves the highest detection accuracy, reaching 0.94. This shows that the statistical scope, business topic, and definition descriptions in semantic-enhanced metadata can effectively support the system in identifying key differences among candidate tables. The detection accuracy of terminology ambiguity is relatively lower. The main reason is that some business terms have highly similar semantic boundaries. For example, concepts such as power generation", grid-connected electricity", and "generator-side power generation" require specific business contexts for accurate distinction.

Table 6. Disambiguation Performance under Different Ambiguity Types

Ambiguity Type	Samples	Detection Acc.	Score Before	Score After
Terminology Ambiguity	140	0.89	4.21	4.78
Statistical Scope Ambiguity	170	0.94	4.29	4.86
Granularity Ambiguity	190	0.93	4.51	4.90

6.5. Case Studies

To more intuitively demonstrate the workflow of the proposed SQL-GRID system in real power grid business scenarios, this section presents two representative case studies. These cases show the ability of the system to handle complex semantic ambiguity and generate highly accurate SQL statements. The two cases include an operation analysis query with temporal granularity ambiguity and a fuel market query with statistical scope ambiguity.

The two cases presented in this section are selected from the 500-query evaluation dataset described in Section 6.1. They represent two typical ambiguity scenarios, namely statistical-scope ambiguity and temporal-granularity ambiguity. The case studies are provided to illustrate the end-to-end workflow of SQL-GRID, and their results are already included in the overall quantitative evaluation reported above.

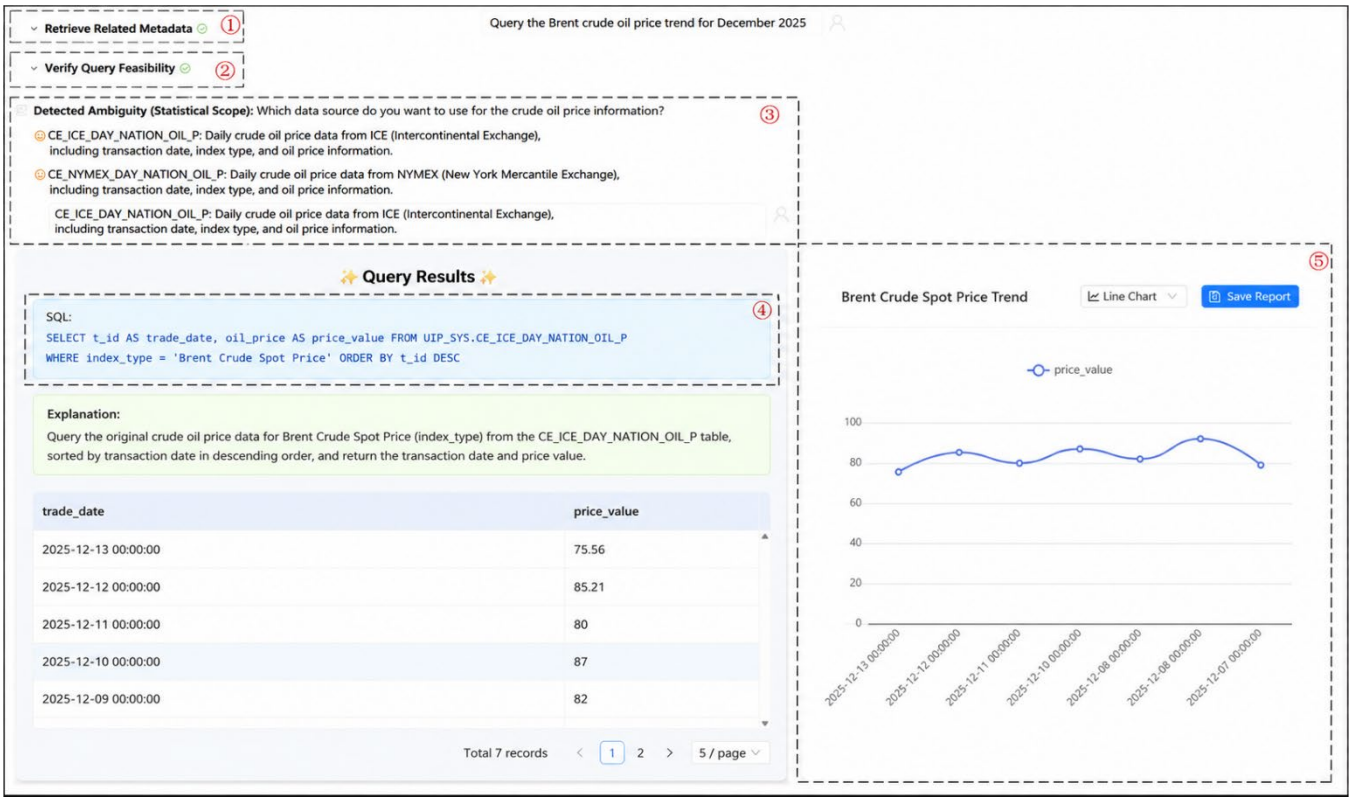

Figure 5. Demonstration of Case Study 1

Case 1: Statistical Scope Ambiguity Resolution in the Fuel Market Scenario

When the user inputs the query "Retrieve the Brent crude oil price trend from the exchange in December 2025", the system starts the execution process.

First, the Retrieval Agent retrieves two tables related to crude oil price: "CE_ICE_DAY_NATION_OIL_P" (Intercontinental Exchange ICE data) and "CE_NYMEX_DAY_NATION_OIL_P" (New York Mercantile Exchange NYMEX data). Second, the Validation Agent confirms that the current query can be answered by the database. Third, the Disambiguation Agent detects that "crude oil price" corresponds to multiple market sources in the database, while the user has not specified the statistical scope. Therefore, the interactive clarification mechanism is triggered. As shown in Figure 5, the ambiguous candidate tables are presented to the user for selection. The user selects "CE_ICE_DAY_NATION_OIL_P" (Intercontinental Exchange ICE data). Fourth, based on the clarified intent, the SQL Generation Agent identifies the Intercontinental Exchange as the target data source and constructs an accurate SQL query. Finally, the Visualization Agent selects an appropriate chart type to visually present the query results.

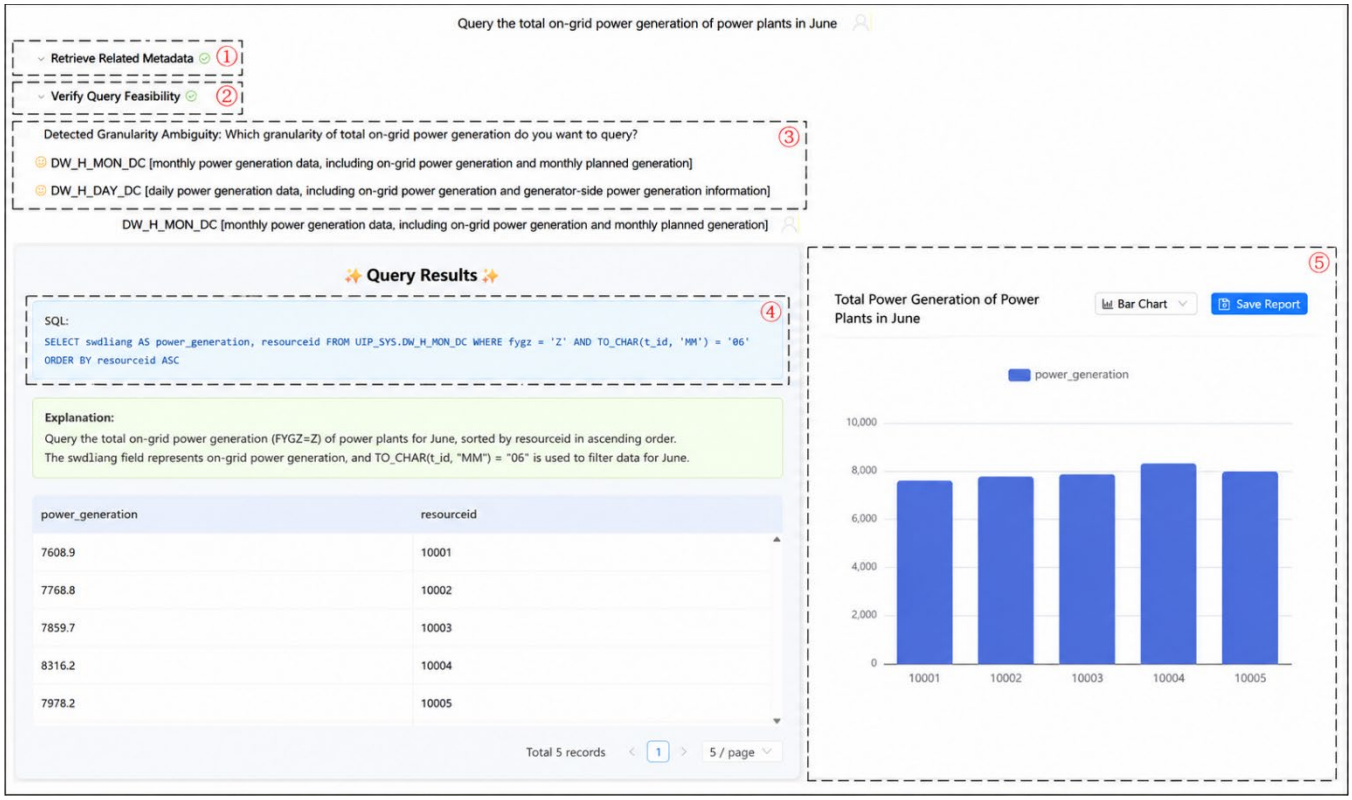

Figure 6. Demonstration of Case Study 2

Case 2: Temporal Granularity Ambiguity Resolution in the Operation Analysis Scenario.

When the user inputs the query "Retrieve the total grid-connected power generation of each power plant in June", the system starts the execution process.

First, based on keywords such as power plant" and power generation", the Retrieval Agent retrieves two highly relevant tables: "DW_H_MON_DC" (monthly power plant data table) and "DW_H_DAY_DC" (daily power plant data table). Second, the Validation Agent confirms that both tables contain power generation fields and that the user query can be answered by the current database. Third, the Disambiguation Agent detects temporal granularity ambiguity between the daily and monthly dimensions in the candidate metadata, while the user has not explicitly specified the required temporal granularity. Therefore, the interactive clarification mechanism is triggered. As shown in Figure 6, the ambiguous candidate tables are presented to the user for selection. The user selects "DW_H_MON_DC" (monthly data). Fourth, based on the clarified intent, the SQL Generation Agent identifies the monthly table and constructs an accurate SQL query. Finally,

the Visualization Agent selects an appropriate chart type to visually present the query results.

The above case studies demonstrate that the proposed framework can not only understand professional business terms but also reduce uncertainty at the source through the "pre-generation disambiguation → interactive clarification" mechanism. This ensures that the generated SQL statements have clear logic and accurate semantics.

7. Conclusion

To address the challenges of self-service data querying and the semantic-structure gap in the digital transformation of power grid systems, this paper proposes SQL-GRID, a grounded retrieval and interactive disambiguation agent for Text-to-SQL over power grid data. SQL-GRID constructs a Semantic-Enhanced Metadata Knowledge Base by integrating business semantics, database schema information, and metadata descriptions. It also combines Multi-Strategy Retrieval with a closed-loop Interactive Disambiguation strategy. This strategy consists of pre-generation ambiguity detection, interactive clarification, and post-generation SQL validation. The proposed system effectively alleviates SQL hallucination when LLMs process highly abstract database names, complex statistical scopes, and ambiguous business terms in power grid data scenarios.

Experimental results show that SQL-GRID significantly improves metadata recall, SQL execution rate, and semantic matching performance compared with traditional baseline methods. The average metadata recall rate reaches 94%. The SQL execution rate increases from 40.2% to 84.6%. The average semantic matching score reaches 4.85 out of 5. These results verify the effectiveness, robustness, and practical value of SQL-GRID in complex power grid data scenarios.

Nevertheless, SQL-GRID still relies on manual expertise during the construction of semantic-enhanced metadata. The experimental dataset also remains limited in scale. Future work will further explore automatic metadata evolution mechanisms, methods for reducing interaction costs, and large-scale validation in real business environments. These studies will help evaluate the generalization capability and engineering applicability of the proposed system.

Acknowledgements.

This research was funded by the Science and Technology Project of the East China Branch of State Grid Corporation of China, grant number 52992425001D.

References

[1] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.

[2] Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems 33, Online, 6–12 December 2020; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 1877–1901.

[3] Pourreza, M.; Rafiei, D. DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction. In Proceedings of the Advances in Neural Information Processing Systems 36, New Orleans, LA, USA, 10–16 December 2023; Curran Associates, Inc.: Red Hook, NY, USA, 2023; pp. 36339–36348.

[4] Wang, B.; Ren, C.; Yang, J.; Liang, X.; Bai, J.; Chai, L.; Yan, Z.; Zhang, Q.-W.; Yin, D.; Sun, X.; et al. MAC-SQL: A Multi-Agent Collaborative Framework for Text-to-SQL. In Proceedings of the 31st International Conference on Computational Linguistics, Abu Dhabi, United Arab Emirates, 19–24 January 2025; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 540–557.

[5] Lee, D.; Park, C.; Kim, J.; Park, H. MCS-SQL: Leveraging Multiple Prompts and Multiple-Choice Selection for Text-to-SQL Generation. In Proceedings of the 31st International Conference on Computational Linguistics, Abu Dhabi, United Arab Emirates, 19–24 January 2025; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 337–353.

[6] Pourreza, M.; Li, H.; Sun, R.; Chung, Y.; Talaei, S.; Kakkar, G.T.; Gan, Y.; Saberi, A.; Özcan, F.; Arık, S.Ö. CHASE-SQL: Multi-Path Reasoning and Preference Optimized Candidate Selection in Text-to-SQL. In Proceedings of the Thirteenth International Conference on Learning Representations, Singapore, 24–28 April 2025.

[7] Farquhar, S.; Kossen, J.; Kuhn, L.; Gal, Y. Detecting Hallucinations in Large Language Models Using Semantic Entropy. Nature 2024, 630, 625–630.

[8] Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.; Madotto, A.; Fung, P. Survey of Hallucination in Natural Language Generation. ACM Comput. Surv. 2023, 55, 248:1–248:38.

[9] Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ACM Trans. Inf. Syst. 2025, 43, 42:1–42:55.

[10] Kim, H.J.; Kim, Y.; Park, C.; Kim, J.; Park, C.; Yoo, K.M.; Lee, S.-G.; Kim, T. Aligning Language Models to Explicitly Handle Ambiguity. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 1989–2007.

[11] Chen, K.; Chen, Y.; Koudas, N.; Yu, X. Reliable Text-to-SQL with Adaptive Abstention. Proc. ACM Manag. Data 2025, 3, Article 69, 1–30.

[12] Saparina, I.; Lapata, M. Disambiguate First, Parse Later: Generating Interpretations for Ambiguity Resolution in Semantic Parsing. In Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria, July 2025; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 16825–16839.

[13] Gong, Y.; Lei, C.; Qin, X.; Vaidya, K.; Narayanaswamy, B.; Kraska, T. SQLens: An End-to-End Framework for Error Detection and Correction in Text-to-SQL. In

Proceedings of the Advances in Neural Information Processing Systems 38, 2025.
[14] Li, G.; Hammoud, H.A.A.K.; Itani, H.; Khizbullin, D.; Ghanem, B. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In Proceedings of the Advances in Neural Information Processing Systems 36, New Orleans, LA, USA, 10–16 December 2023; Curran Associates, Inc.: Red Hook, NY, USA, 2023; pp. 51991–52008.
[15] Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S.K.S.; Lin, Z.; et al. MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
[16] Zhuge, M.; Wang, W.; Kirsch, L.; Faccio, F.; Khizbullin, D.; Schmidhuber, J. GPTSwarm: Language Agents as Optimizable Graphs. In Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024; Proceedings of Machine Learning Research; PMLR, 2024; Volume 235, pp. 62743–62767.
[17] Chen, W.; Su, Y.; Zuo, J.; Yang, C.; Yuan, C.; Chan, C.-M.; Yu, H.; Lu, Y.; Hung, Y.-H.; Qian, C.; et al. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
[18] Gao, D.; Wang, H.; Li, Y.; Sun, X.; Qian, Y.; Ding, B.; Zhou, J. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. Proc. VLDB Endow. 2024, 17, 1132–1145.
[19] Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-T.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the Advances in Neural Information Processing Systems 33, Online, 6–12 December 2020; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 9459–9474.
[20] Jiang, P.; Xiao, C.; Jiang, M.; Bhatia, P.; Kass-Hout, T.; Sun, J.; Han, J. Reasoning-Enhanced Healthcare Predictions with Knowledge Graph Community Retrieval. In Proceedings of the Thirteenth International Conference on Learning Representations, Singapore, 24–28 April 2025.
[21] Jin, B.; Zeng, H.; Yue, Z.; Yoon, J.; Arık, S.Ö.; Wang, D.; Zamani, H.; Han, J. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. In Proceedings of the Conference on Language Modeling, Montreal, QC, Canada, 7–10 October 2025.
[22] Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
[23] Gao, J.; Li, L.; Ji, K.; Li, W.; Lian, Y.; Fu, Y.; Dai, B. SmartRAG: Jointly Learn RAG-Related Tasks from the Environment Feedback. In Proceedings of the Thirteenth International Conference on Learning Representations, Singapore, 24–28 April 2025.
[24] Guan, X.; Zeng, J.; Meng, F.; Xin, C.; Lu, Y.; Lin, H.; Han, X.; Sun, L.; Zhou, J. DeepRAG: Thinking to Retrieve Step by Step for Large Language Models. In Proceedings of the Fourteenth International Conference on Learning Representations, Rio de Janeiro, Brazil, 23–27 April 2026.
[25] Jiang, P.; Cao, L.; Xiao, C.; Bhatia, P.; Sun, J.; Han, J. KG-FIT: Knowledge Graph Fine-Tuning upon Open-World Knowledge. In Proceedings of the Advances in Neural Information Processing Systems 37, Vancouver, BC, Canada, December 2024; Curran Associates, Inc.: Red Hook, NY, USA, 2024; pp. 136220–136258.
[26] Liu, N.F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P. Lost in the Middle: How Language Models Use Long Contexts. Trans. Assoc. Comput. Linguist. 2024, 12, 157–173.