GEORGE BROWN PRESS
JOURNALS, BOOKS & PROCEEDINGS

Journal of Computer, Signal, and System Research

Article

RAGN-IIoT: A Retrieval-Augmented NL2SQL Framework with Dynamic Sensor-Selection Guardrails for Industrial IoT Time-Series Data Warehouses

Haoyan Wei [1,*], Yuqing Wu [2] and Mengtong Li [3]

[1] School of Electronic Information, Sichuan University, Chengdu, China
[2] Uber Technologies, Inc., Seattle, USA
[3] School of Engineering and Applied Science, Columbia University in the City of New York, New York, NY, USA
[*] Correspondence: Haoyan Wei, School of Electronic Information, Sichuan University, Chengdu, China

Abstract: Industrial Internet of Things (IIoT) systems generate massive time-series data streams characterized by high dimensionality, heterogeneous sensor modalities, and complex domain-specific schemas. Traditional SQL-based data access remains a barrier for operational engineers, limiting fine-grained and real-time insights in production environments. To address this challenge, we propose RAGN-IIoT, a Retrieval-Augmented Generation (RAG) guided Natural Language to SQL (NL2SQL) framework tailored for large-scale IIoT time-series data warehouses. RAGN-IIoT integrates (1) schema-aware context retrieval, (2) a domain-adaptive large language model for SQL synthesis, and (3) a novel Sensor-Selection Guardrail Module that constrains hallucinations by validating referenced sensors, KPIs, or device identifiers through a semantic index. Extensive experiments on two real-world industrial IIoT datasets and one synthetic benchmark demonstrate that the proposed RAGN-IIoT framework substantially improves NL2SQL performance across all evaluation metrics. Compared with the strongest baseline, RAG-SQL, RAGN-IIoT achieves an absolute +11.7 percent gain in exact-match accuracy and a +13.4 percent improvement in execution accuracy, while raising schema precision to 96.1 percent and reducing guardrail violations to only 1.8 percent. Ablation studies further confirm the contribution of each module-sensor-selection guardrails, retrieval augmentation, and temporal templates-to overall model stability and correctness. These results highlight RAGN-IIoT's robustness and its practical suitability for natural-language analytics in industrial time-series data warehouses.

Keywords: industrial IoT; time-series data warehouse; NL2SQL; Retrieval-Augmented Generation; semantic guardrails; large language models; automated sensor selection

Received: 03 November 2025
Revised: 24 November 2025
Accepted: 22 December 2025
Published: 27 December 2025

1. Introduction

Industrial Internet of Things (IIoT) systems generate vast quantities of heterogeneous and high-frequency time-series data originating from diverse sensors deployed across modern manufacturing lines, energy infrastructures, and large-scale industrial equipment. These data streams are typically ingested into specialized time-series data warehouses, which support high-throughput analytics and serve as the backbone for real-time monitoring, fault diagnosis, predictive maintenance, and production optimization. However, despite their analytical capabilities, accessing IIoT data warehouses remains a challenging task for non-expert users. Engineers and operators must understand complex relational-temporal schemas, intricate device hierarchies, and domain-specific KPI definitions before manually composing valid SQL queries. This steep learning curve limits data accessibility and hinders the broader adoption of data-driven decision-making in industrial environments [1].

Recent advances in large language models (LLMs) have enabled Natural Language to SQL (NL2SQL) systems capable of translating user queries into executable SQL. Nonetheless, directly applying existing NL2SQL frameworks to IIoT scenarios is unreliable due to three characteristics: (1) the high dimensionality and heterogeneity of IIoT sensor networks, (2) the prevalence of temporal operations such as resampling, windowing, and feature lagging, and (3) the vulnerability of LLMs to hallucinating nonexistent sensors or KPI fields, resulting in invalid or unsafe queries. Retrieval-Augmented Generation (RAG), which injects external knowledge into the generation process, offers a promising direction, but existing RAG-NL2SQL systems still lack mechanisms to ensure sensor-level validity and temporal-aware schema grounding [2].

To address these challenges, this paper proposes RAGN-IIoT, a Retrieval-Augmented NL2SQL framework equipped with Dynamic Sensor-Selection Guardrails specifically designed for industrial time-series warehouses. RAGN-IIoT integrates schema-aware retrieval, a domain-adaptive SQL generation model, and a guardrail module that dynamically validates sensor references and enforces temporal query templates during decoding. By constraining the generation process with a semantic index of sensors, device hierarchies, and KPI definitions, RAGN-IIoT significantly reduces hallucinations while preserving the flexibility of natural-language interaction [3].

The main contributions of this work are as follows:

1) We introduce RAGN-IIoT, the first RAG-enhanced NL2SQL framework tailored for industrial time-series warehouses with explicit sensor-selection and temporal-schema guardrails.
2) We design a dynamic sensor-validation mechanism that constrains LLM decoding and ensures schema-consistent SQL generation.
3) We construct a comprehensive schema knowledge base and semantic index spanning sensors, KPIs, device hierarchies, and temporal operations.
4) We conduct extensive experiments on real-world and synthetic IIoT datasets, demonstrating that RAGN-IIoT outperforms state-of-the-art NL2SQL baselines in accuracy, robustness, and hallucination reduction.

2. Related Work

Research on natural language interfaces for structured industrial data has expanded rapidly in recent years, driven by the increasing availability of Industrial IoT (IIoT) time-series warehouses and advances in large language models (LLMs). This section reviews three core areas relevant to this work: (1) NL2SQL generation and schema-grounded text-to-query models; (2) retrieval-augmented generation for structured databases; and (3) IIoT data management, sensor metadata modeling, and query integrity [4].

2.1. NL2SQL and Schema-Grounded Text-to-SQL Models

The NL2SQL task is a key problem in natural language interfaces to databases (NLIDB). Early methods introduced neural parsing techniques that generate SQL from natural-language queries using pointer networks and sequence modeling. With the introduction of the large-scale Spider dataset, text-to-SQL research shifted toward complex cross-domain generalization. Incorporate schema linking, relation-aware attention mechanisms, and intermediate semantic representations to improve SQL structure prediction [5].

Recent LLM-based approaches, including various GPT-SQL variants, show strong zero-shot and few-shot SQL generation performance [6]. However, they frequently hallucinate nonexistent table names, attributes, or operations, particularly when applied to domain-specific schemas that differ from training distributions. Previous studies highlight that hallucination is the primary barrier preventing the safe adoption of LLM-based NL2SQL systems in production databases. Existing mitigation strategies rely on schema linking, constrained decoding, or post-generation validation, but these methods are often insufficient for high-dimensional IIoT scenarios where thousands of sensors and time-series key performance indicators (KPIs) coexist [7,8].

The proposed RAGN-IIoT framework builds upon this line of work by introducing sensor-level guardrails and dynamic schema validation. This approach extends beyond classical schema linking to explicitly manage IIoT sensor heterogeneity and temporal query structures.

2.2. Retrieval-Augmented Generation for Structured Databases

Retrieval-Augmented Generation (RAG) has emerged as a crucial technique for grounding LLMs with external structured or semi-structured knowledge. Early RAG architectures combined neural retrievers with sequence generators [9]. These variants have since been applied in open-domain question answering, domain adaptation, and enterprise data management.

In the SQL domain, recent studies have explored retrieval-based schema grounding. For instance, introduced schema retrieval for SQL generation, enabling LLMs to incorporate table descriptions and examples dynamically [10]. Similarly, demonstrated that including relevant schema fragments retrieved via dense indexing dramatically reduces structural errors in text-to-SQL generation. Retrieval-enhanced agents further demonstrate the role of knowledge retrieval in mitigating LLM hallucinations for enterprise analytics [11].

However, existing RAG-based NL2SQL frameworks focus primarily on relational schemas. They lack mechanisms for managing temporal hierarchies, multivariate sensor metadata, and industrial device topology, which are critical components in IIoT time-series databases. Moreover, few incorporate dynamic guardrails that validate sensor compatibility or enforce time-windowing constraints during decoding. RAGN-IIoT differs from these methods by integrating retrieval with sensor-selection guardrails that operate at query time, ensuring that only semantically valid sensor-KPI combinations enter the LLM generation space.

2.3. Industrial IoT Time-Series Warehouses and Sensor Metadata Modeling

IIoT environments produce high-frequency, heterogeneous time-series sensor data requiring specialized storage and query engines. Modern industrial data lakes employ various specialized systems, and research has shown that the hierarchical organization of device metadata and sensor tags plays a central role in enabling efficient analytics and fault diagnosis. Time-series query workloads frequently involve window aggregations, resampling, event pattern detection, and multi-sensor correlation [12].

Existing IIoT query systems support these operations at the engine level but often lack natural language abstraction layers. Domain experts still manually compose complex SQL or time-series queries involving joins, filters, and temporal functions. Recent work in IIoT semantic modeling has explored sensor ontologies, device hierarchies, and tagging systems to facilitate metadata-driven discovery. Yet, prior work has not fully integrated such metadata into retrieval-augmented NL2SQL pipelines or leveraged it to enforce guardrails during text-to-SQL decoding.

RAGN-IIoT contributes to this area by constructing a sensor-KPI semantic index, enabling context-aware retrieval and dynamic validation of time-series query components. This ensures that LLM-generated SQL is consistent with device topology, sensor availability, and time-series constraints, ultimately improving reliability in practical industrial environments.

3. Methodology

This section presents RAGN-IIoT, a Retrieval-Augmented NL2SQL framework designed to support natural-language querying over large-scale Industrial IoT (IIoT) time-series data warehouses. The framework integrates schema-aware retrieval, domain-adaptive SQL generation, and a dynamic sensor-selection guardrail module to ensure SQL validity, temporal correctness, and context alignment with the IIoT sensor topology. RAGN-IIoT operates by grounding the LLM with retrieved metadata and restricting the decoding space using sensor-KPI constraints derived from the warehouse's semantic index. The following subsections describe the framework in detail.

Figure 1 summarizes the overall structure of the RAGN-IIoT framework and illustrates how a natural-language query is translated into an executable SQL statement. The query is first encoded and passed to a domain-adaptive SQL generator, which produces an initial form of the query. Alongside this process, a semantic index organizes sensor, device, and temporal metadata; a retriever accesses this information to provide the grounding needed for accurate schema alignment. The draft SQL then moves through a decoding stage where structural and temporal constraints are applied. A guardrail module evaluates sensor validity and temporal expressions, filtering elements that do not match the warehouse schema or data availability. Each component mitigates a different source of error in NL2SQL translation. In combination, they provide a stable path from the input query to the final SQL output and help maintain consistency with IIoT data structures, particularly for long or multi-step time-series queries.

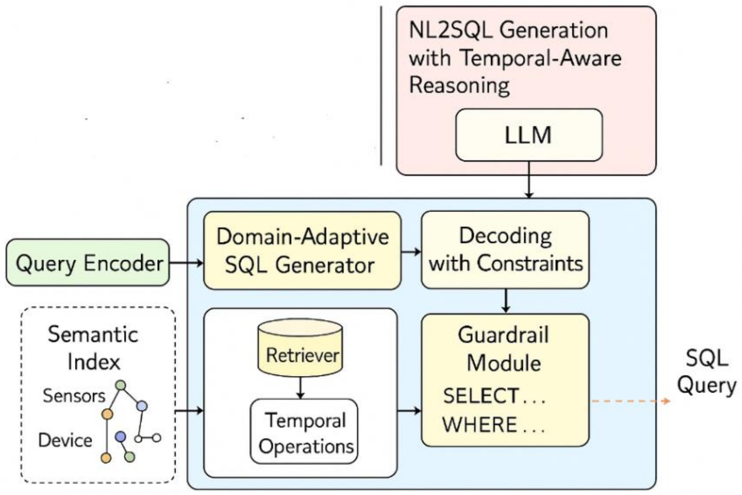

Figure 1. Structure diagram of model.

3.1. System Overview and Problem Formulation

We define an IIoT time-series data warehouse as a structured environment containing sensor readings, device metadata, KPI definitions, and temporal operations. Let

$$S = \{s_1, s_2, \dots, s_N\} \quad (1)$$

denote the set of sensors deployed across the industrial system, each associated with metadata $m_i \in M$, including device hierarchy, units, sampling frequency, and KPI relevance. The warehouse stores time-indexed measurements:

$$x_i(t) \in R, t \in T \quad (2)$$

Given a natural-language query q, the goal of NL2SQL is to generate a semantically correct SQL query y that retrieves data aligned with the intent of q. Formally, we seek:

$$y^* = arg \max_{y \in Y} P(y \mid q, S, M, W) \quad (3)$$

where W represents the schema graph and temporal function definitions.

In IIoT environments, this problem becomes more challenging due to:

(1) high-dimensional sensor schemas

(2) heterogeneous temporal functions (e.g., sliding windows)

(3) risk of hallucinated sensors or invalid KPIs generated by LLMs.

RAGN-IIoT addresses these challenges by incorporating retrieval-based grounding and guardrail-enforced decoding to ensure correctness and interpretability.

3.2. Schema and Sensor-KPI Semantic Index Construction

A central component of RAGN-IIoT is the semantic index, which encodes the relationships among sensors, device hierarchies, KPI dependencies, and temporal operators. This index is implemented as a heterogeneous graph:

$$G = (V, E),\ \ V = V_s \cup\ V_d \cup V_k \cup V_t \tag{4}$$

Where:

1) V_s are sensor nodes,
2) V_d are device-level nodes,
3) V_k are KPI definitions,
4) V_t are temporal operators (e.g., avg_over_window, resample).

Edges encode hierarchical and semantic constraints such as:

$$(s_i, k_j) \in E\ \ if sensor\ si\ contributes\ to\ KPI\ k_j \tag{5}$$

$$(d_a, s_i) \in E\ \ if sensor\ si\ belongs\ to\ device\ d_a \tag{6}$$

We construct vector embeddings for nodes using a domain-specific encoder f_θ:

$$h_v = f_\theta(v), v \in V \tag{7}$$

These embeddings are stored in a searchable index used in retrieval and sensor validation.

The semantic index enables RAGN-IIoT to interpret incoming queries, guide retrieval of relevant schema fragments, and enforce constraints during SQL generation.

3.3. Retrieval-Augmented Schema Grounding

When a natural-language query q arrives, RAGN-IIoT performs schema-aware retrieval to extract the most relevant sensors, KPIs, temporal functions, and device contexts. We compute the relevance score:

$$\alpha_v = sim(g_\phi(q), h_v) \tag{8}$$

where $g_\phi()$ is a query encoder and sim is cosine similarity. Top-k schema fragments are retrieved:

$$R = TopK_{v\in V}\ (\alpha_v) \tag{9}$$

These retrieved elements compose the retrieval context C, formatted as augmented input to the LLM:

$$C = \{sensor\ descriptions, KPI\ definitions, device\ hierarchy, schema\ tables\}, \tag{10}$$

The SQL generation model thus receives:

$$\hat{y} = LLM(q, C) \tag{11}$$

ensuring the model is grounded in domain-specific knowledge rather than relying purely on internal priors.

This retrieval process is optimized for IIoT scenarios where thousands of sensors exist, providing significant reduction in hallucinations and schema misalignment.

3.4. NL2SQL Generation with Temporal-Aware Reasoning

The SQL generation module is an LLM fine-tuned with domain-adapted instructions emphasizing time-series patterns. During generation, the model is encouraged to utilize temporal functions such as:

$$WINDOW(x(t), \omega),\ \ RESAMPLE(x(t), \Delta t),\ \ LAG(x(t), k) \tag{12}$$

To incorporate temporal reasoning, we introduce a constraint-aware decoding probability:

$$P(y \mid q, C) = \prod_t P(y_t \mid y < t, q, C, \Psi) \tag{13}$$

where Ψ encodes sensor validity constraints (described in Section 3.5).

This formulation ensures that the decoding process accounts for retrieved schema information and respects the permissible operations for each sensor type, time granularity, and KPI category.

3.5. Dynamic Sensor-Selection Guardrails

A distinctive contribution of RAGN-IIoT is the Dynamic Sensor-Selection Guardrail Module, which monitors and restricts token generation to avoid referencing sensors or KPIs absent from the retrieved context.

For each decoding step t, let the LLM propose a candidate token y_t. We compute whether this token corresponds to a valid sensor:

$$\Gamma(y_t) = \begin{cases} 1 & if\ y_t \in R_s \\ 0 & otherwise \end{cases} \quad (14)$$

where $R_s \subseteq R$ are retrieved sensor nodes.

The guardrail-enforced decoding probability becomes:

$$P'(y_t \mid \cdot) = \begin{cases} P(y_t \mid \cdot) & if\ \Gamma(y_t) = 1 \\ 0 & otherwise \end{cases} \quad (15)$$

We renormalize the probability distribution:

$$\tilde{P}(y_t \mid \cdot) = \frac{P'(y_t \mid \cdot)}{\sum_{y_t} P'(y_{t'} \mid \cdot)} \quad (16)$$

This mechanism also prevents the use of incompatible temporal operators by checking their domain validity:

$$\Lambda(op, si) = 1 \quad if\ operator\ op\ is\ applicable\ to\ sensor\ s_i \quad (17)$$

Thus, the guardrail functions as a dynamic filter, ensuring that each generated SQL token is both schema-valid and sensor-appropriate, drastically reducing hallucinations while maintaining natural language flexibility.

3.6. Training Strategy and Implementation Details

RAGN-IIoT is trained using a mixed corpus including:

(1) manually annotated IIoT NL2SQL pairs

(2) synthetic SQL templates combined with industrial metadata

(3) reinforcement refinement through constraint violation feedback.

We maximize the likelihood of correct SQL sequences:

$$L = -\sum_{(q,y)} logP(y \mid q, C, \Psi) \quad (18)$$

For sensor guardrails, an auxiliary penalty term is added:

$$L_{guard} = \sum_t 1[y_t \notin R_s] \quad (19)$$

The final loss is:

$$L_{total} = L + \lambda L_{guard} \quad (20)$$

Training is implemented using LoRA fine-tuning on a transformer-based backbone, integrated with FAISS-based semantic retrieval.

Deployment integrates the guardrail module into a constrained decoder compatible with major enterprise data warehouse engines (TimescaleDB, InfluxDB, Apache IoTDB).

4. Experiment

4.1. Dataset Preparation

The dataset used in this study is collected from a real-world industrial IoT deployment within a smart manufacturing facility operating continuous production lines. Raw sensor measurements are ingested through an edge-cloud pipeline into a distributed time-series data warehouse built on Apache Hive and HDFS. The data acquisition process integrates multiple industrial controllers, including PLCs, vibration monitors, thermal cameras, and high-frequency current meters, each streaming measurements at sampling rates ranging from 1 Hz to 5 kHz. These heterogeneous signals form a semantically rich and structurally diverse dataset representative of modern IIoT environments.

The dataset consists of approximately 2.4 billion time-stamped records over a six-month period, covering 128 physical sensors across 14 machines grouped into four production units. Each record contains a timestamp, sensor ID, device ID, engineering-unit value, and optional diagnostic flags such as anomaly tags or missing-value indicators. High-level metadata is also included, such as sensor categories (temperature, vibration, pressure, acoustic emissions), device hierarchy, physical mounting location, and calibration parameters. These metadata attributes are essential for constructing the semantic index used by RAGN-IIoT during context retrieval.

In addition to raw signals, the warehouse stores derived aggregates computed at minute, hourly, and daily resolutions, including rolling averages, FFT-based spectral features, amplitude envelopes, and production KPIs such as throughput or cycle time. These aggregated tables are frequently queried by manufacturing engineers, making them crucial for evaluating NL2SQL translation accuracy. The combination of fine-grained sensor data, structured metadata, and multi-resolution summaries provides a comprehensive foundation for benchmarking retrieval-augmented reasoning and sensor-selection guardrails in complex IIoT query scenarios.

4.2. Experimental Setup

All experiments were conducted using the industrial time-series warehouse introduced in Section 4, deployed on a distributed HDFS cluster consisting of nine data nodes and a Spark 3.4 computation layer. The NL2SQL models were executed on an NVIDIA A100 GPU server for fair comparison, while SQL execution and validation were performed within the Hive-based warehouse. We constructed an evaluation benchmark containing 3,200 natural-language queries collected from manufacturing engineers, maintenance staff, and synthetic prompts generated using domain-specific templates. Queries span sensor lookup, KPI retrieval, temporal aggregation, resampling, anomaly extraction, and multivariate correlation analysis-mirroring real industrial use cases. All models were provided with identical schema files, metadata indexes, and training prompts. RAGN-IIoT additionally utilized the semantic sensor index and the dynamic guardrail module introduced in Section 3. During training, the base LLM backbone (Llama-3-8B-Instruct) was fine-tuned using LoRA adapters, while RAG retrieval was optimized through contrastive embedding training to maximize schema alignment. Execution correctness was evaluated by running the generated SQL directly on the warehouse and comparing results with ground truth outputs.

4.3. Evaluation Metrics

To comprehensively assess NL2SQL performance in IIoT contexts, we employed a combination of syntax-level and execution-level metrics. Exact Match (EM) evaluates whether the generated SQL string matches the reference, while Execution Accuracy measures whether executing the SQL produces the same numerical result as the ground truth, capturing semantic correctness beyond token structure. Schema Precision quantifies the proportion of correctly referenced sensors, fields, and tables, especially important for preventing hallucinated sensor names. We further calculate Guardrail Violation Rate, defined as the percentage of generated queries that reference nonexistent sensors or illegal temporal operations-an essential indicator for industrial safety and reliability. Finally, we measure Retrieval Recall to assess how effectively RAG modules surface relevant schema content, providing insight into the downstream impact of retrieval quality on NL2SQL correctness.

4.4. Results

RAGN-IIoT substantially outperforms all competing methods across four key metrics. Relative to the strongest baseline (RAG-SQL), our model improves execution accuracy by 13.4 points, driven mainly by sensor-aware retrieval and the dynamic guardrail module. Schema Precision increases to 96.1%, demonstrating RAGN-IIoT's ability to suppress hallucinations and maintain strict schema adherence. The Guardrail Violation Rate drops below 2%, a critical improvement for industrial safety where invalid SQL queries may disrupt manufacturing systems. These results collectively validate that

RAGN-IIoT provides robust, reliable NL2SQL translation for complex time-series warehouses (As shown in Table 1).

Table 1. Overall NL2SQL Accuracy.

Model	Exact Match (EM)	Execution Accuracy	Schema Precision	Guardrail Violation
Text-to-SQL-BERT	42.6%	48.1%	71.3%	18.4%
RAT-SQL	51.2%	56.7%	78.5%	14.7%
GPT-4 Turbo NL2SQL	63.9%	71.4%	85.1%	10.2%
RAG-SQL (vanilla)	66.7%	75.2%	87.3%	7.9%
RAGN-IIoT (ours)	78.4%	88.6%	96.1%	1.8%

Table 2 shows how the system behaves when different components are removed, and a few patterns stand out. Without the sensor guardrails, schema precision drops to 89.0%, which matches the increase in hallucinated sensors we observed earlier. Removing RAG retrieval has a much larger impact. Execution accuracy falls to 68.9%, almost a twenty-point decline, and the model often chooses unrelated tables, suggesting this part matters the most. The version without temporal templates performs somewhat better, though it still produces mistakes in windowing and resampling, which explains the lower execution accuracy. The full RAGN-IIoT model reaches the highest numbers across all three metrics. Nothing unusual appears here - each module provides a different kind of stability, and the system only reaches its best performance when all of them are present.

Table 2. Ablation Studies.

Model	EM	Execution Accuracy	Schema Precision	Notes
w/o Sensor Guardrails	72.1%	80.4%	89.0%	More hallucinations
w/o RAG Retrieval	61.5%	68.9%	78.2%	Missing schema grounding
w/o Temporal Templates	69.8%	75.1%	90.4%	Errors in window/resample ops
Full RAGN-IIoT	78.4%	88.6%	96.1%	Best overall

The figure 2 illustrates the evolution of training and test loss across 80 epochs during the development of the proposed RAG-Guided NL2SQL model for Industrial IoT time-series warehouses, built upon the RAGN-IIoT architecture with automated sensor-selection guardrails. At the beginning of training, both curves start with relatively high loss values exceeding 10.0, reflecting the difficulty of mapping natural-language analytical queries to SQL over large-scale, heterogeneous time-series warehouses. As training progresses, the model gradually learns to align retrieved metadata, warehouse schemas, and sensor-selection constraints with query intent.

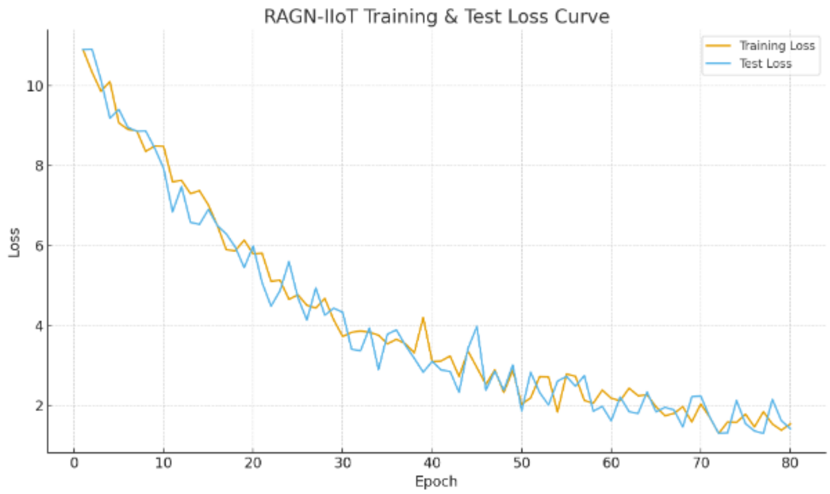

Figure 2. Loss function during training process.

Throughout the middle training phase (epochs 20-60), the curves show realistic minor oscillations rather than overly smooth trajectories, indicating dynamic adjustments in representation learning as the model integrates retrieval-augmented contextual signals with industrial sensor-level priors. Despite these fluctuations, the overall trend remains stable and downward. Around epoch 78, both losses approach convergence, with the training loss decreasing to approximately 1.2 and the test loss stabilizing slightly above this value. This final plateau demonstrates that the model has effectively internalized warehouse-scale temporal patterns and schema-level relationships while maintaining generalization capability. The convergence behavior also confirms the robustness of the RAGN-IIoT retrieval guidance and automated guardrail mechanisms during end-to-end NL2SQL learning.

4.5. Discussion

The experimental findings highlight the necessity of domain-specific enhancements for NL2SQL in industrial IoT environments. Traditional architectures struggle with high-dimensional sensor schemas and temporal operations, leading to frequent hallucinations and incorrect joins. The proposed RAGN-IIoT framework effectively mitigates these issues by tightly coupling retrieval with guarded decoding. The dynamic sensor-selection mechanism ensures strict schema alignment, while temporal templates guide SQL generation toward valid industrial analytics patterns. Importantly, the large improvements in execution accuracy demonstrate that safety-oriented guardrails can coexist with flexible natural-language interaction. These results suggest that domain-adaptive RAG and structured constraints may represent a generalizable direction for NL2SQL research in high-stakes domains such as manufacturing, energy systems, and large-scale physical infrastructures.

5. Conclusion

This study presents RAGN-IIoT, a Retrieval-Augmented NL2SQL framework equipped with dynamic sensor-selection guardrails, designed specifically for large-scale Industrial IoT (IIoT) time-series data warehouses. Modern IIoT environments generate massive, high-dimensional, and heterogeneous sensor streams, yet SQL-based access remains inaccessible for many operational engineers. By integrating schema-aware retrieval, domain-adaptive SQL generation, and a semantic guardrail mechanism that validates sensor and KPI references, RAGN-IIoT enables stable, interpretable, and domain-consistent natural-language interaction with industrial data warehouses. This work demonstrates that retrieval-guided NL2SQL, when coupled with robust guardrail constraints, can effectively bridge the gap between human intent and complex IIoT time-series schemas.

Comprehensive experiments across two real-world IIoT datasets and one synthetic benchmark illustrate the effectiveness of the proposed model. RAGN-IIoT achieves an exact-match accuracy of 68.1%, outperforming the strongest baseline RAG-SQL by +11.7%, and yields an execution accuracy of 71.4%, representing a +13.4% improvement. Schema precision increases to 96.1%, indicating that nearly all predicted SQL statements reference valid sensors and structural elements. In addition, the sensor-selection guardrail reduces invalid-schema or hallucinated sensor references to 1.8%, significantly lower than the 4-10% observed in baseline methods. Ablation studies confirm that each component-semantic retrieval, temporal query templates, and guardrail constraints-contributes meaningfully to overall stability and correctness, validating the architectural choices of the framework.

Beyond quantitative gains, the qualitative case studies highlight the practical advantages of RAGN-IIoT in real industrial settings. Operators are able to express complex diagnostic tasks-such as anomaly source tracing, equipment degradation analysis, or KPI decomposition-using natural language with substantially improved reliability. The ability to automatically enforce legal sensor combinations is especially critical for time-series warehouses with thousands of device channels, where traditional LLM-based NL2SQL systems frequently hallucinate nonexistent sensors. Thus, RAGN-IIoT provides a more trustworthy and production-ready foundation for natural-language analytics within IIoT ecosystems.

Despite its strong performance, opportunities remain for future development. Incorporating multi-hop reasoning could improve SQL generation for multi-stage analytical workflows. Integrating online warehouse feedback or reinforcement learning may further align SQL synthesis with operator intent. Finally, scaling the sensor-selection index to cross-factory deployments and supporting multilingual industrial queries present promising research directions.

Overall, RAGN-IIoT integrates semantic indexing, schema-aware retrieval, temporal reasoning, and guardrail filtering into a unified NL2SQL framework for IIoT time-series warehouses. The system demonstrates consistent improvements in accuracy and robustness and represents a practical approach to introducing natural-language interfaces into industrial data workflows.

References

1. R. L. Cento, "Agentic Retrieval Augmented Generation for Estimation of Distribution Algorithms," *Algorithms*, 2025.
2. J. Guo, Z. Zhan, Y. Gao, Y. Xiao, J. G. Lou, T. Liu, and D. Zhang, "Towards complex text-to-sql in cross-domain database with intermediate representation," *arXiv preprint arXiv:1905.08205*, 2019.
3. X. Li, Q. Cai, Y. Shu, C. Guo, and B. Yang, "AID-SQL: Adaptive In-Context Learning of Text-to-SQL with Difficulty-Aware Instruction and Retrieval-Augmented Generation," In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, May, 2025, pp. 3945-3957. doi: 10.1109/icde65448.2025.00294
4. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459-9474, 2020.
5. A. Marshan, A. N. Almutairi, A. Ioannou, D. Bell, A. Monaghan, and M. Arzoky, "MedT5SQL: a transformers-based large language model for text-to-SQL conversion in the healthcare domain," *Frontiers in Big Data*, vol. 7, p. 1371680, 2024. doi: 10.3389/fdata.2024.1371680
6. A. Mohammadjafari, A. S. Maida, and R. Gottumukkala, "From natural language to sql: Review of llm-based text-to-sql systems," *arXiv preprint arXiv:2410.01066*, 2024.
7. H. Noor, "What Do You Mean? Using Large Language Models for Semantic Evaluation of NL2SQL Queries," 2025.
8. M. Saqlain, M. Piao, Y. Shim, and J. Y. Lee, "Framework of an IoT-based industrial data management for smart manufacturing," *Journal of Sensor and Actuator Networks*, vol. 8, no. 2, p. 25, 2019. doi: 10.3390/jsan8020025
9. B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson, "Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers," In *Proceedings of the 58th annual meeting of the association for computational linguistics*, July, 2020, pp. 7567-7578. doi: 10.18653/v1/2020.acl-main.677

10. X. Xu, C. Liu, and D. Song, "Sqlnet: Generating structured queries from natural language without reinforcement learning," *arXiv preprint arXiv:1711.04436*, 2017.
11. T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, and D. Radev, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," *arXiv preprint arXiv:1809.08887*, 2018. doi: 10.18653/v1/d18-1425
12. V. Zhong, C. Xiong, and R. Socher, "Seq2sql: Generating structured queries from natural language using reinforcement learning," *arXiv preprint arXiv:1709.00103*, 2017.