Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 4, No. 5, 2025

Deep Graph Modeling for Performance Risk Detection in Structured Data Queries

Danyi Gao
Columbia University, New York, USA
dg3224@columbia.edu

Abstract: This paper addresses the performance risk that may arise during the execution of structured query statements. It proposes a graph neural network-based method for SQL query risk classification. The method parses each query into a structured graph representation. A structure-semantics fusion module is used to jointly model operator types and structural dependencies of nodes. This enhances the model's ability to capture query semantics and execution paths. A risk-aware contrastive learning mechanism is also introduced. By constructing positive and negative risk sample pairs, the model improves the clustering and separation of query representations in the discriminative space. This further strengthens its capability in risk identification. Systematic experiments are conducted on several structured query datasets, including JOB, TPC-H, and IMDB. The results show that the proposed method outperforms various existing approaches in terms of accuracy, macro-average F1 score, precision, and recall. It demonstrates clear performance advantages. In addition, ablation studies verify the contribution of each module to overall performance. Transfer experiments also confirm the model's strong generalization ability across different query scenarios. This work provides an efficient and scalable modeling solution for structured query risk analysis. It offers practical value for intelligent optimization and performance assurance in database systems.

Keywords: Structured query, graph neural network, risk classification, contrastive learning

1. Introduction

In modern data-intensive systems, SQL serves as a fundamental component of structured query languages. It is an indispensable interface in enterprise databases, high-performance computing platforms, and data warehouse systems. However, with the explosive growth of data volumes and the increasing complexity of query logic, the performance and resource consumption of SQL queries have become highly unpredictable. A seemingly simple SQL query may lead to severe system bottlenecks due to factors such as suboptimal join orders, missing indexes, or skewed data distribution [1]. These issues may further result in performance degradation or even task failures. Therefore, identifying and managing SQL query risks has become a critical task in database operations and poses higher demands on intelligent optimization.

Traditional SQL query optimization relies heavily on rule-based and cost-based optimizers. These optimizers estimate query costs statically and select an expected optimal execution plan. However, this approach depends on accurate statistics and has limited ability to model complex query structures. It struggles to handle dynamic workloads and diverse query patterns in real-world environments [2]. Moreover, current mainstream optimizers focus on average cost estimates and often ignore potential risk factors, such as long execution time, memory overflows, or excessive I/O consumption. This design, which optimizes for the average but overlooks the extremes, makes systems more prone to instability and unpredictability under risky queries. As a result, it is valuable to study methods for identifying and classifying SQL query risks from structural and semantic perspectives [3].

In recent years, the advancement of deep learning has highlighted the expressive power of Graph Neural Networks (GNNs) in modeling structured data. SQL queries naturally exhibit graph-like characteristics. Their syntactic structures, operator dependencies, and execution plans can all be represented as graph data [4]. This provides both theoretical grounding and practical feasibility for applying GNNs. Compared with traditional sequence-based or tree-based models, GNNs better capture high-order dependencies and global context among operators. This enables more accurate representation of query semantics and structural complexity. Applying GNNs to SQL query risk identification helps overcome existing optimizer limitations and paves the way for intelligent and adaptive database systems [5].

Under this context, encoding the execution plan or syntax of SQL queries into graph structures and classifying their risk types using GNN models allows for early warning of high-risk queries. It also provides guidance for adjusting execution plans, scheduling resources, and rewriting queries. Furthermore, this method offers generalizability and scalability [6]. It can adapt to different database systems and workloads without relying on handcrafted rules. It shows strong potential for generalization. With the adoption of graph-based learning mechanisms, the system can learn risk patterns from accumulated runtime data. This enables it to evolve into a more intelligent and stable component for query optimization [7]. In summary, studying SQL query risk classification based on Graph Neural Networks addresses the urgent need for stability and reliability in modern data systems. It also expands the application of deep learning in the database domain. This research direction combines structured data modeling, risk identification, and machine learning. It is expected to drive the development of intelligent database optimization from both theoretical and practical perspectives. As data scales and query complexity continue to increase, this line of research will provide essential support for ensuring high availability and performance of database systems. It holds significant practical value and broad application prospects.

2. Related work

2.1 Structured data modeling

Structured data modeling has always played a central role in database optimization, data mining, and machine learning. Early methods mainly relied on rule-based or statistics-based modeling [8]. They used handcrafted feature extraction, cost estimation functions, and heuristic strategies to handle structured query tasks. These methods worked in specific scenarios but often lacked the ability to represent complex structural semantics[9]. This limitation becomes clear when dealing with multi-table joins, high-dimensional nested structures, or atypical query logic. Moreover, such models show poor generalization. Static modeling also fails to adapt to changing data distributions and query workloads, becoming a bottleneck in system optimization[10].

To address these issues, deep learning-based methods for structured data modeling have emerged in recent years. They have achieved significant progress, especially in SQL query understanding and execution plan optimization. These approaches encode queries as sequences, trees, or graphs. Neural network models are then used to learn embeddings. This greatly improves the ability to represent complex query structures. Graph-based representations are particularly effective. They naturally express operator dependencies and execution paths in SQL[11]. As a result, they are widely used in tasks such as plan optimization, cardinality estimation, and performance prediction. The rise of Graph Neural Networks has further strengthened this trend. Their strong graph-level and node-level modeling capabilities make them powerful tools for structured data representation. They perform well in many core database tasks[12].

In SQL query modeling research, some studies have converted syntax trees, logical plan graphs, and physical execution graphs into graph data structures suitable for deep learning. Models such as Graph Convolutional Networks and Graph Attention Networks are then applied to extract semantic features. These methods capture long-range dependencies and global structures in queries[13]. They offer a more accurate and robust foundation for task modeling. However, existing work mainly focuses on regression tasks such as cost estimation and plan selection. Discriminative tasks like risk classification and bottleneck diagnosis have received less attention[14]. There remains significant space for research and practical application, especially in how models can precisely represent query structures to support complex classification decisions.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) have become a major advancement in modeling graph-structured data. They have demonstrated strong performance across a range of tasks. Unlike traditional neural networks, GNNs can perform feature propagation and aggregation directly on non-Euclidean structures[15,16]. This allows them to capture high-order relationships between nodes while preserving structural information. The core idea is to iteratively propagate information through the adjacency structure. Each node updates its representation by integrating both local and global context. This capability makes GNNs especially suitable for modeling complex structured objects, such as social networks, knowledge graphs, molecular structures, and SQL query plans or syntax trees in databases[17,18].

In database research, GNNs have been increasingly applied to represent and optimize the internal structure of SQL queries. Some studies model query execution plans as directed graphs. In these graphs, nodes represent operators such as selection, join, and projection. Edges represent the logical order of execution. Encoding such graph structures using GNNs enables the extraction of deep structural features. These features support downstream tasks such as query cost prediction, index recommendation, and query rewriting. These methods overcome the limitations of traditional optimizers that rely on handcrafted rules and static statistics. They use deep learning to automatically extract patterns and learn optimization strategies from real execution data. As a result, they offer improved adaptability and generalization[19].

Although GNNs have achieved promising results in regression tasks, their application in classification tasks such as query risk detection is still in its early stages. Key challenges remain. These include how to design graph representations that better reflect SQL query structures, how to integrate semantic and structural information within GNNs, and how to handle graph heterogeneity and scale variation[20,21]. In addition, domain-specific challenges in databases — such as dynamic graph changes, complex node semantics, and multi-source information fusion—pose further demands on GNN modeling and training strategies. Therefore, applying GNNs to SQL query risk classification not only broadens the range of applications in this domain but also provides a new perspective for exploring the generalization of GNNs in structured tasks[22,23].

3. Method

This study proposes a Graph Neural Network-based method for SQL query risk classification. It builds query structure graphs and introduces graph representation learning to automatically identify potential execution risks in SQL statements. The method first parses SQL queries into syntax graphs or execution plan graphs. It then uses an improved GNN model to encode these structures. Multi-level structural and semantic information is extracted and used to predict the type of risk. To enhance the model's ability to capture query diversity and semantic detail, two key innovations are introduced. The first is the Structure-Semantics Fusion

Module (SSFM). It integrates node operation types with contextual path information during graph encoding. This enables deep fusion of structural and semantic features. The second is the Risk-aware Contrastive Learning mechanism (RCL). It introduces contrastive learning with positive and negative risk samples to improve the discriminative power of query representations. This helps the model better identify high-risk query patterns. Together, these innovations form a deep graph model designed for complex query structures. The model shows strong generalization ability and provides a more expressive framework for SQL query risk classification. The model architecture is shown in Figure 1.

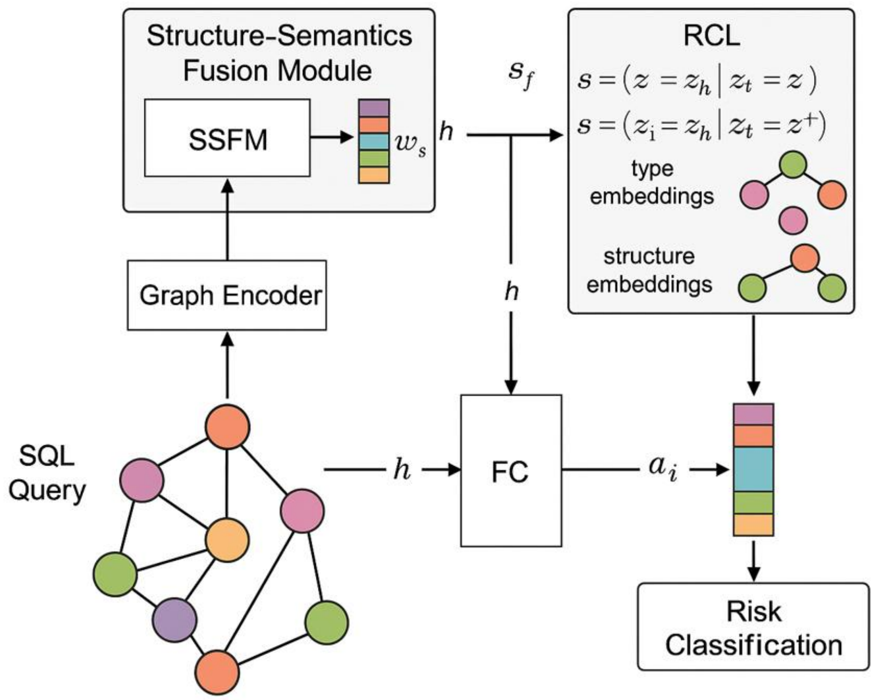

Figure 1. Overall model architecture diagram

3.1 Structure-Semantics Fusion Module

This module aims to jointly model the structural information and semantic features in the SQL query graph to improve the query representation's ability to distinguish risk categories. Considering the structural complexity of SQL queries and the diversity of semantic levels, this paper introduces a structural-semantic fusion mechanism to jointly model the semantic embedding of nodes (such as operator types) and their contextual information in the graph structure. Its module architecture is shown in Figure 2.

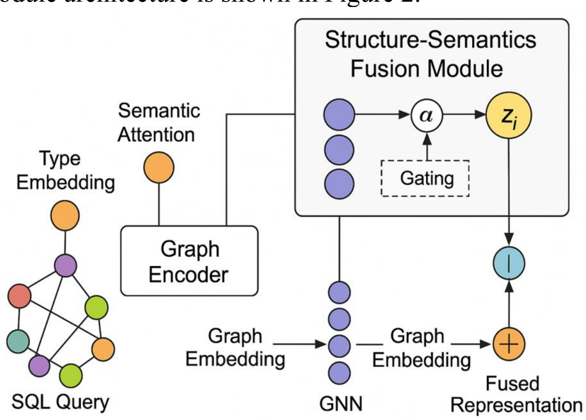

Figure 2. SSFM module architecture

Suppose the input is a query graph $G = (V, E)$, where V represents the set of operator nodes, E represents the set of edges, and the initial embedding corresponding to each node $v_i \in V$ is defined as $x_i \in R^d$.

First, a graph neural network is used to encode the input graph structure and generate the structural representation h_i of each node. The propagation process is as follows:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)}\right)$$

Where $N(i)$ represents the neighbor set of node i, c_{ij} is the normalization coefficient, $W^{(l)}$ is the weight matrix of the lth layer, and σ is the activation function (such as ReLU). At initialization, $h_i^{(0)} = x_i$.

In order to inject semantic awareness, a node-level semantic attention mechanism is introduced after the graph encoding output. We define a semantic embedding e_i for each node, which comes from the embedding table of the operator type. Then the fusion representation is defined as:

$$z_i = a_i \cdot h_i + (1 - a_i) \cdot e_i$$

The fusion weight $\alpha_i \in [0,1]$ is learned through the gating function:

$$\alpha_i = \sigma(w^T[h_i \| e_i] + b)$$

Where $\|$ represents the vector concatenation operation, w, b is the learnable parameter, and σ is the Sigmoid activation function.

After obtaining the fused node representation, a global aggregation function is used to generate a graph-level representation z_G as the structural-semantic fusion representation of the query graph. The aggregation operation is as follows:

$$z_G = READOUT(\{z_i \mid v_i \in V\}) = \frac{1}{|V|}\sum_{i=1}^{|V|} z_i$$

The final output $z_G \in R^d$ is used as the module output for the subsequent risk perception comparative learning module and classifier.

3.2 Risk-aware Contrastive Learning

In order to enhance the model's ability to discriminate high-risk SQL queries, this study introduces the Risk-aware Contrastive Learning (RCL) mechanism. This module takes the structural-semantic fusion representation as input, and guides the model to learn more aggregated representations of similar risks and discrete representations of heterogeneous risks with the help of positive and negative sample construction and contrast loss functions, thereby improving the clarity of classification boundaries and generalization performance. Its module architecture is shown in Figure 3.

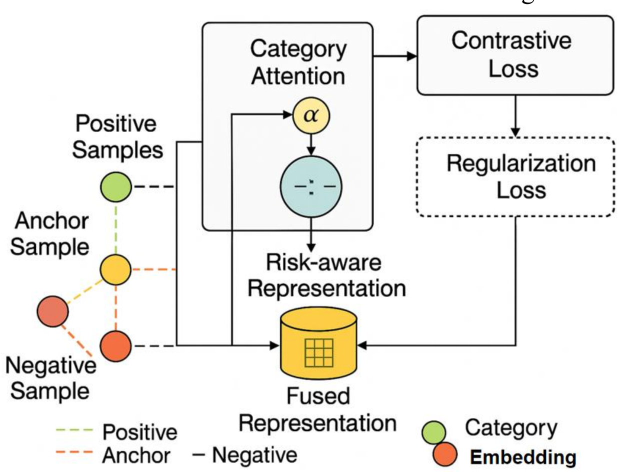

Figure 3. RCL module architecture

Assume that the representation of each SQL query output by the SSFM module is $z \in R^d$. We introduce risk label information into this representation space to construct comparison sample pairs.

Specifically, for a query representation z_i, select another query representation z_j^+ with the same risk label as a positive sample, and select a query representation z_k^- with a different risk label as a negative sample. Through the contrast loss function, we hope that the model will try to bring the positive sample pairs closer and the negative sample pairs farther apart in the representation space. The basic contrast loss can be written as:

$$L_{contrast} = -\log\frac{\exp(sim(z_i, z_j^+)/\tau)}{\exp(sim(z_i, z_j^+)/\tau) + \sum_k \exp(sim(z_i, z_k^-)/\tau)}$$

Where sim represents the cosine similarity function and τ is the temperature coefficient, which is used to control the smoothness of the distribution.

To improve the clarity of the risk label contrast learning, we introduce a category attention mechanism to enhance the perception of risk label related features in a weighted manner. Assuming that each risk category c has a learnable category vector r_c, the matching degree between the current sample and its label is defined as:

$$\alpha_i = \text{softmax}(z_i^T, r_c)$$

This attention weight will be used to enhance the risk sensitivity of the sample representation, and the final output perception representation is:

$$\tilde{z}_i = \alpha_i \cdot z_i + (1-\alpha_i) \cdot r_c$$

We then perform contrastive loss learning based on $\tilde{z}_i$ to further optimize the discriminative structure of the representation space.

In addition, to prevent the collapse of the semantic space between samples, we introduce additional regularization terms to maintain the distribution diversity of representations. The specific form is as follows:

$$L_{reg} = \lambda(T_r(Z^T Z) - \|Z\|_F^2)$$

Where Z is the representation matrix of all samples in the current batch, $\|\cdot\|$ is the Frobenius norm, and λ is the regularization coefficient. Finally, the total loss function of the risk-aware contrastive learning module is defined as:

$$L_{RCL} = L_{contrast} + L_{reg}$$

4. Experimental Results

4.1 Dataset

This study uses the publicly available Join Order Benchmark (JOB) dataset as the main source of experimental data. The dataset is built on the IMDb database and is widely used for query optimization and cost estimation tasks. It features rich structural complexity and strong real-world relevance. The JOB dataset contains 113 complex SQL queries. These queries involve common operations such as multi-table joins, aggregation, and filtering. They can effectively simulate potential risk patterns in practical OLAP scenarios.

The underlying data of the JOB dataset consists of multiple real-world movie-related tables, such as movie_info, cast_info, title, and company_name. The SQL queries involve extensive join operations. The query plan trees are structurally diverse, making them suitable for structure modeling and risk classification tasks. In this study, the original SQL queries are converted into logical query graph structures. These graphs are linked to actual execution plans and performance metrics for GNN training and evaluation.

To enhance model training and classification performance, this study constructs risk labels for JOB queries. The classification is based on dimensions such as execution time and resource consumption, assigning queries to different risk levels. The dataset offers a rich set of structured query graph samples. It also provides high interpretability, making it a suitable benchmark for research on SQL query performance modeling and classification.

4.2 Generate query graph

The query graph generation process aims to convert raw SQL statements into graph-structured data suitable for processing by Graph Neural Networks. This enables more effective capture of structural and semantic information in queries. Specifically, each SQL query is first parsed to extract its Abstract Syntax Tree (AST) or logical query plan. During this process, the logical execution plan is obtained using the optimizer interface provided by the database system. Each operator, such as selection, projection, or join, is identified as a node in the graph. The type, attributes, and position of each node are used as its initial semantic features.

After node construction, edges are created based on operator dependencies or data flow directions. This forms a directed graph or tree structure. Join order, subquery nesting, and filter conditions are represented as specific edge types or path patterns. Edge types carry semantic labels to distinguish between join edges, projection edges, and filter edges. This enhances the graph's ability to represent semantic details. In the end, each SQL query is transformed into a structured query graph. Nodes and edges together describe the execution logic and semantic structure of the query.

To further improve the model's generalization across different query structures, each node is enriched with additional features. These include table names, column counts, and operation types. All features are encoded as input vectors. The entire graph structure is then standardized. This ensures that the GNN can accept variable-length and heterogeneous graph inputs in a unified way. This query graph generation process provides a consistent and high-quality data foundation for downstream structure modeling and risk classification tasks.

4.3 Experimental setup

Based on the constructed query graphs, this study applies a Graph Neural Network model for training and evaluation. The goal is to verify the effectiveness of the proposed method in SQL query risk classification. All experiments are conducted on a server equipped with an NVIDIA A100 GPU, an Intel Xeon processor, and 256GB of RAM. The operating system is Ubuntu 20.04. PyTorch is used as the deep learning framework, and DGL (Deep Graph Library) is used for graph modeling. The input graph includes multi-dimensional features such as node types, operator attributes, and edge types. The model is optimized using a cross-entropy loss function. The Adam optimizer is applied with an initial learning rate of 0.001 and a batch size of 32.

During model training, the JOB dataset is split into training, validation, and test sets in a 7:1:2 ratio. Evaluation metrics include Accuracy, Macro-F1 score, Precision, and Recall. These metrics are used to comprehensively assess the model's discriminative ability and stability in the risk identification task.

4.4 Experimental Results

1) *Comparative experimental results*

First, this paper gives the comparative experimental results with other models. The experimental results are shown in Table 1.

Table 1: Comparative experimental results

Method	ACC	Macro-F1	Precision	Recall
ResNet50[24]	74.3%	71.8%	72.1%	71.4%
LSTM[25]	76.5%	73.9%	75.0%	72.8%
VGG19[26]	72.1%	69.3%	68.7%	70.5%
Transformer[27]	78.6%	75.4%	76.1%	74.9%
Mltr[28]	80.2%	77.3%	78.0%	76.5%
Query2label[29]	82.7%	79.8%	80.5%	79.0%
Ours	86.4%	84.2%	85.0%	83.5%

As shown in the comparative results in Table 1, the proposed method achieves the best performance across all evaluation metrics. It consistently outperforms existing mainstream models. In terms of accuracy (ACC), the proposed model reaches 86.4 percent. This is nearly 4 percentage points higher than Query2label, the second-best model. The result demonstrates that our model has stronger generalization and robustness in identifying SQL query risks.

For the macro-average F1 score, the proposed method achieves 84.2 percent, significantly higher than other models. In particular, it shows substantial improvements over Transformer (75.4 percent) and Mltr (77.3 percent). This indicates that the model can accurately classify both common and minority classes. It remains stable even when handling imbalanced data. This helps reduce classification bias and shows strong adaptability to different risk levels.

The Precision and Recall metrics further support the effectiveness of the method. A Precision of 85.0 percent indicates a low false positive rate in risk prediction. A Recall of

83.5 percent shows that the model can successfully detect most high-risk queries. The method balances accuracy and sensitivity. In real-world applications, this balance is crucial for maintaining the stability of database systems.

Overall, compared with traditional image or sequence-based models such as ResNet50, LSTM, and VGG19, and recent attention-based models like Transformer and Query2label, the proposed method achieves superior performance. By integrating structure-semantic fusion and risk-aware contrastive learning, it enables deep modeling of both SQL query structure and semantic risk signals. This confirms the effectiveness and advancement of the method in structured query risk classification tasks.

2) Ablation Experiment Results

Furthermore, this paper gives the ablation experiment results, and the experimental results are shown in Table 2.

Table 2: Ablation Experiment Results

Method	ACC	Macro-F1	Precision	Recall
Baseline	80.1%	76.8%	77.5%	75.4%
+SSFM	83.3%	80.2%	81.0%	79.5%
+RCL	82.4%	79.1%	80.2%	77.8%
Ours	86.4%	84.2%	85.0%	83.5%

As shown in the ablation results in Table 2, both core modules proposed in this study—the Structure-Semantics Fusion Module (SSFM) and the Risk-aware Contrastive Learning module (RCL)—have a significant positive impact on model performance. Compared with the baseline model, introducing either module leads to notable improvements. This indicates that each mechanism enhances the model's representation and classification abilities from different perspectives.

Specifically, after adding the SSFM module, the model's accuracy improves from 80.1 percent to 83.3 percent. The Macro-F1 score increases from 76.8 percent to 80.2 percent. This is the largest performance gain among all single-module settings. It shows that the structure-semantic fusion mechanism effectively captures both semantic and structural details in the query graph. It enhances the model's ability to represent complex query risks and improves stability in multi-class classification tasks.

In comparison, the model with the RCL module also shows clear improvements. The accuracy reaches 82.4 percent, and the Macro-F1 score rises to 79.1 percent. Although slightly lower than SSFM, the simultaneous increase in Precision and Recall indicates that RCL focuses more on reinforcing the decision boundary for risk detection. It strengthens the model's discriminative power and reduces both false negatives and false positives.

When both modules are used together, forming the complete version of the proposed method (Ours), all evaluation metrics reach their highest values. Accuracy rises to 86.4 percent, and the Macro-F1 score reaches 84.2 percent. These results confirm that SSFM and RCL provide complementary strengths. Their combined effect significantly improves overall performance and validates the effectiveness and soundness of the proposed design.

3) Loss function changes with epoch

This paper also gives a graph of the loss function changing with epoch, and the experimental results are shown in Figure 4.

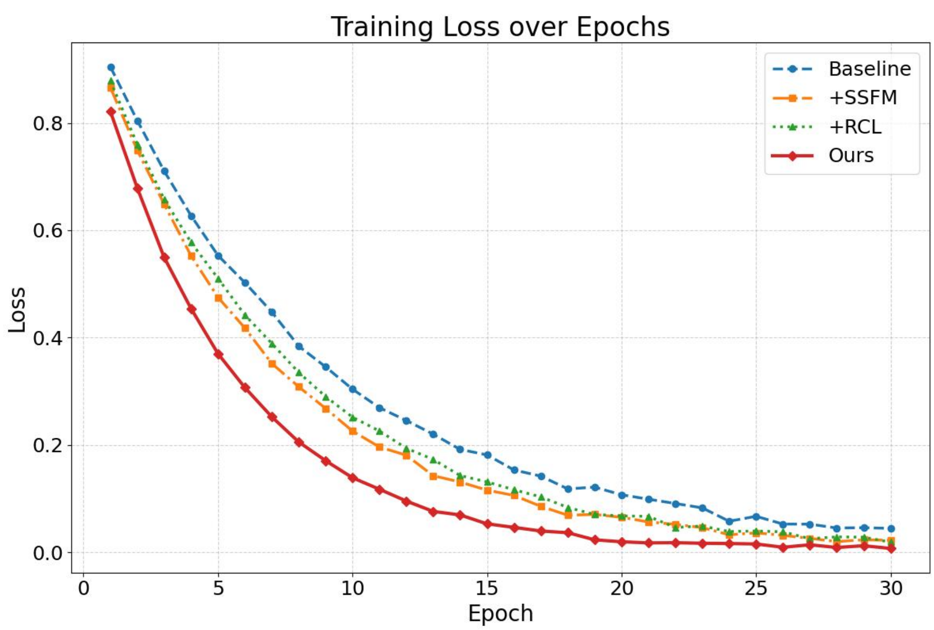

Figure 4. Training Loss over Epoch

As shown in Figure 4, the loss functions of the four models exhibit a generally stable downward trend during training. This indicates that the models gradually converge as their parameters are optimized. The baseline model shows a slower decrease in loss. Its loss remains relatively high during the first ten epochs, suggesting limitations in feature extraction and representation ability.

After introducing the Structure-Semantics Fusion Module (+SSFM) and the Risk-aware Contrastive Learning Module (+RCL), the models show a faster loss reduction at the early stages of training. The overall convergence speed is significantly higher than that of the baseline. This indicates that the enhancements brought by both modules in model structure and optimization objective improve feature learning efficiency. As a result, the models can capture discriminative information more quickly and converge faster.

Notably, the complete model (Ours) maintains the lowest loss throughout the training process. It reaches a stable point around the 15th epoch. This shows that the model not only converges faster but also achieves a better final loss value. It suggests strong synergy between SSFM and RCL, which together enhance the model's overall capacity in risk classification tasks.

Overall, the comparison of training loss curves clearly demonstrates the proposed method's advantages in optimization efficiency and training stability. By introducing structure modeling and discriminative learning mechanisms, the model improves its representation power and more easily reaches a low-bias parameter space. This lays a solid foundation for better classification performance in later stages.

4) Comparison of the robustness of the model under different training set sizes

Furthermore, this paper also presents the robustness comparison of the model under different training set sizes, as shown in Figure 5. This analysis serves to illustrate the model's ability to maintain stable performance as the amount of training data varies, which is critical for assessing its practical applicability in real-world environments where data availability may be limited or inconsistent. Understanding how the model adapts to different data scales provides valuable insight into its learning efficiency, scalability, and reliability. It also reflects the importance of designing models that can generalize well even when exposed to limited training samples, ensuring consistent risk classification across diverse deployment scenarios.

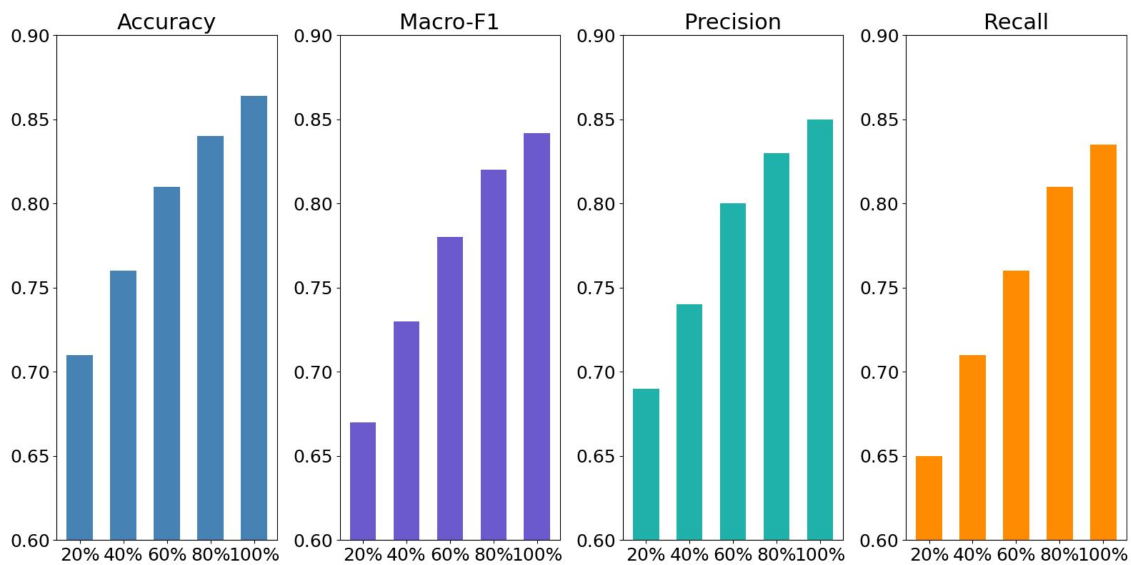

Figure 5. Robustness of the model under different training set sizes

As shown in Figure 5, the evaluation metrics of the model improve steadily under different training set sizes. This indicates strong adaptability and robustness to varying data volumes. As the training set size increases from 20 percent to 100 percent, accuracy improves from around 71 percent to 86.4 percent. This shows that the model's classification ability significantly improves with more samples and can more accurately identify query risk types.

The Macro-F1 score also increases consistently with more training data, reaching nearly 84 percent. This suggests that the model performs well not only on major classes but also maintains stable performance on minority classes under imbalanced distributions. The rising trend of this metric reflects the model's strong generalization ability and robustness in handling complex structured queries across different risk levels.

Precision and Recall both grow in parallel, with a small gap between them. This shows that the model achieves high detection capability while maintaining a low false positive rate. The improvements become more evident when the training set exceeds 60 percent. This indicates that the risk-aware contrastive learning mechanism enhances feature representation and boundary modeling during the mid-to-late training stages.

Overall, the experiment confirms the proposed method's stability and performance scalability as data volume increases. The model does not show signs of overfitting. Instead, it continues to improve with larger datasets. This demonstrates good scalability and suggests its suitability for deployment in real database systems for risk classification of large-scale and complex queries.

5) The impact of query graph structure complexity on model discrimination ability

This paper also explores the impact of query graph structure complexity on the model's discriminative ability, as shown in Figure 6. By analyzing how the model responds to increasing levels of structural complexity, the study highlights the importance of understanding the relationship between query representation and classification performance. This aspect of the work emphasizes that structural features play a critical role in risk identification tasks, and that capturing variations in complexity is essential for building models with strong adaptability and generalization. It also suggests that effective risk classification in real-world scenarios requires models to be sensitive to diverse and often intricate structural patterns inherent in SQL queries.

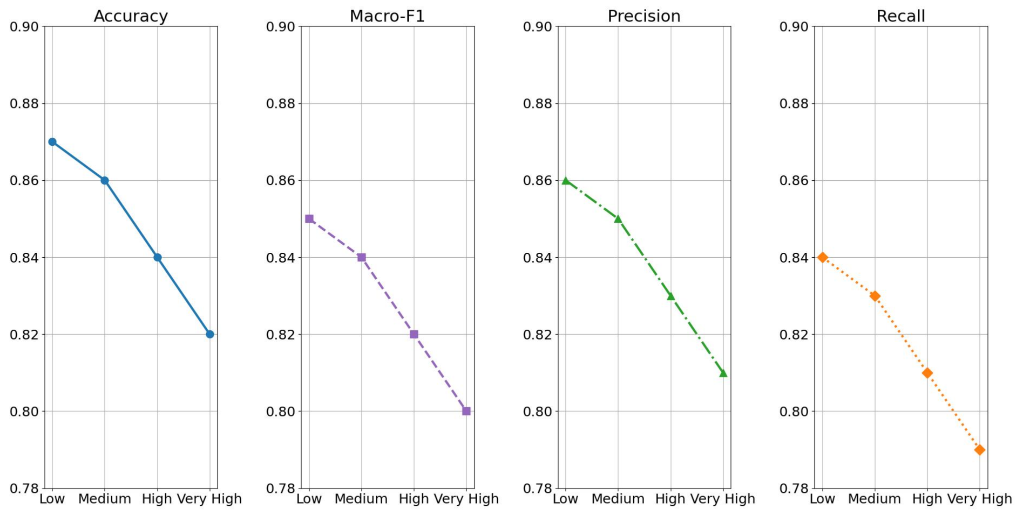

Figure 6. The impact of query graph structure complexity on model discrimination ability

As shown in Figure 6, the model's performance on four major evaluation metrics shows an overall declining trend as the query graph complexity increases from Low to Very High. This indicates that the complexity of query structures has a certain impact on the model's classification capability. The performance drop becomes more noticeable at the High and Very High levels, suggesting greater modeling challenges when handling highly complex structures.

The accuracy curve shows that classification accuracy decreases from around 87 percent to 82 percent as structural complexity increases. However, the performance remains relatively high. This suggests that the model can effectively learn risk classification features for simple queries. For deeper structures with more JOIN operations or nested clauses, the model's ability to capture local features may be limited, affecting final predictions.

The decline in Macro-F1 and Precision further reveals the model's sensitivity to class balance and precision control. As structure becomes more complex, the model's ability to detect minority class risks weakens. This may result from diluted feature signals or increased redundancy in the graph, which reduces the effectiveness of feature aggregation. Even so, Precision remains relatively high across all levels, indicating stable performance in identifying low-risk queries.

The Recall curve shows a more significant drop, indicating a higher chance of missing high-risk queries in complex structures. This suggests that the model becomes less effective at detecting high-risk cases as complexity grows. Overall, the proposed method maintains a certain level of robustness when handling complex queries. However, the results also highlight the need to further improve structure-sensitive modeling to enhance classification and generalization performance under highly complex scenarios.

6) Generalization performance evaluation of model migration to other structured query datasets

Finally, this paper also gives the generalization performance evaluation of the model migration to other structured query datasets, and the experimental results are shown in Figure 7.

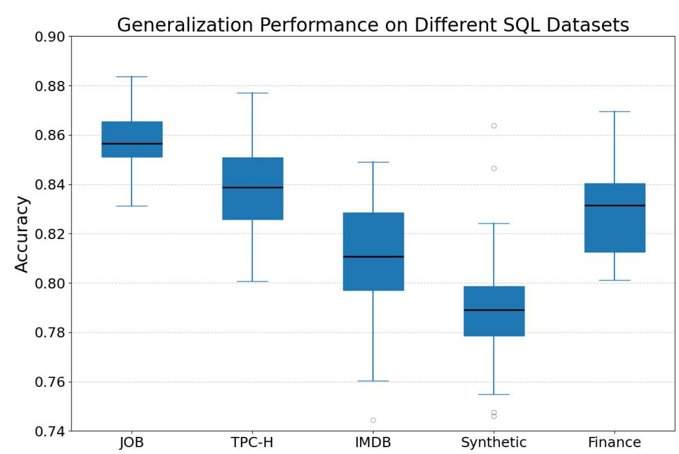

Figure 7. Generalization performance evaluation of model migration to other structured query datasets

As shown in Figure 7, the proposed model demonstrates generally good generalization performance across multiple structured query datasets. However, some fluctuations are observed between datasets. This reflects the impact of data distribution and query structure characteristics on the model's transferability. On the JOB and TPC-H datasets, accuracy scores are more concentrated. The median and upper quartile are both high. This indicates that the model adapts well to standard query workloads and has strong transfer capability.

On the IMDB dataset, the model shows more noticeable performance variation. The lower bound of accuracy drops significantly. This suggests that the dataset may include queries with deeper structures, more complex join paths, or higher levels of semantic noise. These factors pose challenges to the model's classification ability. The trend also suggests that, although the model performs robustly in most structured environments, there remains a risk of performance degradation under highly heterogeneous data sources.

On the Synthetic dataset, accuracy is generally lower and more dispersed. This reflects that in artificially generated queries, structural patterns may not align with the model's learning preferences. Features such as extreme nesting or redundant conditions reduce the model's generalization ability. The result highlights that, although transfer performance is strong on real datasets, more robustness is needed to handle artificially complex structures.

In contrast, results on the Finance dataset are more stable. The median accuracy is higher than on IMDB and Synthetic. This shows that the model adapts well to domain-specific query structures. Overall, the proposed model demonstrates strong cross-dataset generalization. It performs reliably across various typical SQL query scenarios. At the same time, the results suggest that further improvements are needed to handle highly heterogeneous structures.

5. Conclusion

This study addresses the problem of execution risk classification in structured query statements. It proposes a graph neural network-based risk classification model that integrates a Structure-Semantics Fusion Module (SSFM) and a Risk-aware Contrastive Learning mechanism (RCL). The model enables deep modeling of SQL query graphs and high-precision risk identification. By introducing graph structure awareness and class-level discriminative constraints, the model effectively captures risk signals in complex query structures and significantly improves classification performance and generalization under multi-class risk settings. Extensive empirical experiments show that the proposed method achieves strong performance across multiple metrics, including accuracy, F1 score, precision, and recall. These results confirm the model's adaptability and robustness in complex query tasks. In key experiments involving variations in query complexity, training data size, and cross-dataset transfer, the model consistently maintains stable classification capability. These findings demonstrate both the academic contribution of the model and its practical potential for deployment. The method can be applied to real-world scenarios such as database performance tuning, query scheduling, automatic rewriting, and failure prediction.

This research further extends the application scope of graph neural networks in database structure modeling. It provides an effective technical solution for intelligent risk control in database systems. Unlike traditional rule-based or heuristic optimization strategies, the proposed method offers a data-driven approach to understanding query structure and predicting performance. It lays a foundation for building adaptive and evolvable query optimization components. It also provides theoretical support for improving the stability and service quality of large-scale database systems. Future work may explore generalization across larger scales, multiple languages, or heterogeneous system environments. Possible directions include joint training on multi-source query logs and structural transfer learning between different database systems. In addition, reinforcement learning and generative models may be integrated to support proactive rewriting and risk avoidance strategies for high-risk queries. This could drive structured data systems toward greater intelligence and reliability and enhance their practical value in high-assurance domains such as finance, healthcare, and public services.

References

[1] Abbas, Shanza, et al. "A review of nlidb with deep learning: findings, challenges and open issues." IEEE Access 10 (2022): 14927-14945.

[2] Katsogiannis-Meimarakis, George, and Georgia Koutrika. "A survey on deep learning approaches for text-to-SQL." The VLDB Journal 32.4 (2023): 905-936.

[3] Timadi, M. E., and E. C. M. Obasi. "Integrating Zero-Trust Architecture with Deep Learning Algorithm to Prevent Structured Query Language Injection Attack in Cloud Database." University of Ibadan Journal of Science and Logics in ICT Research 13.1 (2025): 52-62.

[4] Milicevic, Bogdan, and Zoran Babovic. "A systematic review of deep learning applications in database query execution." Journal of Big Data 11.1 (2024): 173.

[5] Alghawazi, Maha, Daniyal Alghazzawi, and Suaad Alarifi. "Deep learning architecture for detecting SQL injection attacks based on RNN autoencoder model." Mathematics 11.15 (2023): 3286.

[6] Marcus, Ryan, and Olga Papaemmanouil. "Plan-structured deep neural network models for query performance prediction." arXiv preprint arXiv:1902.00132 (2019).

[7] Kalyanpur, Aditya, et al. "Structured data and inference in DeepQA." IBM Journal of Research and Development 56.3.4 (2012): 10-1.

[8] Zhang, Dongdong, et al. "Combining structured and unstructured data for predictive models: a deep learning approach." BMC medical informatics and decision making 20 (2020): 1-11.

[9] Ryan, Mark. Deep learning with structured data. Manning, 2020.

[10] Wang, Wenjuan, et al. "A systematic review of machine learning models for predicting outcomes of stroke with structured data." PloS one 15.6 (2020): e0234722.

[11] Liao, Wenlong, et al. "A review of graph neural networks and their applications in power systems." Journal of Modern Power Systems and Clean Energy 10.2 (2021): 345-360.

[12] Lee, Jaekoo, et al. "Transfer learning for deep learning on graph-structured data." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 31. No. 1. 2017.

[13] Zhao, Ying, and Jinjun Chen. "A survey on differential privacy for unstructured data content." ACM Computing Surveys (CSUR) 54.10s (2022): 1-28.

[14] Sharifani, Koosha, and Mahyar Amini. "Machine learning and deep learning: A review of methods and applications." World Information Technology and Engineering Journal 10.07 (2023): 3897-3904.

[15] Bhatti, Uzair Aslam, et al. "Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence." International Journal of Intelligent Systems 2023.1 (2023): 8342104.

[16] Xu, Xinzheng, et al. "A comprehensive review of graph convolutional networks: approaches and applications." Electronic Research Archive 31.7 (2023): 4185-4215.

[17] Phan, Huyen Trang, Ngoc Thanh Nguyen, and Dosam Hwang. "Aspect-level sentiment analysis: A survey of graph convolutional network methods." Information Fusion 91 (2023): 149-172.

[18] Cao, Pingping, et al. "Applications of graph convolutional networks in computer vision." Neural computing and applications 34.16 (2022): 13387-13405.

[19] Liu, Xin, et al. "Sampling methods for efficient training of graph convolutional networks: A survey." IEEE/CAA Journal of Automatica Sinica 9.2 (2021): 205-234.

[20] Ahmad, Tasweer, et al. "Graph convolutional neural network for human action recognition: A comprehensive survey." IEEE Transactions on Artificial Intelligence 2.2 (2021): 128-145.

[21] Wang, Yuxia, et al. "Generating population migration flow data from inter-regional relations using graph convolutional network." International Journal of Applied Earth Observation and Geoinformation 118 (2023): 103238.

[22] Chen, Qian, et al. "DeFedGCN: Privacy-Preserving Decentralized Federated GCN for Recommender System." IEEE Transactions on Services Computing (2025).

[23] LIU, Jun-qi, Wen-xuan TU, and En ZHU. "Survey on graph convolutional neural network." Computer engineering & science 45.08 (2023): 1472.

[24] Zhang, Zhaoning, et al. "A quick survey on large scale distributed deep learning systems." 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2018.

[25] Smagulova, Kamilya, and Alex Pappachen James. "A survey on LSTM memristive neural network architectures and applications." The European Physical Journal Special Topics 228.10 (2019): 2313-2324.

[26] Mascarenhas, Sheldon, and Mukul Agarwal. "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification." 2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON). Vol. 1. IEEE, 2021.

[27] Lin, Tianyang, et al. "A survey of transformers." AI open 3 (2022): 111-132.

[28] Cheng, Xing, et al. "Mltr: Multi-label classification with transformer." 2022 IEEE international conference on multimedia and expo (ICME). IEEE, 2022.

[29] Liu, Shilong, et al. "Query2label: A simple transformer way to multi-label classification." arXiv preprint arXiv:2107.10834 (2021).