Supervised Learning on Relational Databases with Quantum Graph Neural Networks

Martin Vogrin
Faculty of Natural Sciences and Mathematics, University of Maribor
Maribor, Slovenia
martin.vogrin@um.si

Rok Vogrin
Faculty of Electrical Engineering and Computer Science, University of Maribor
Maribor, Slovenia
rok.vogrin@um.si

Sven Groppe
Institut für Informationssysteme, Universität zu Lübeck
Lübeck, Germany
sven.groppe@uni-luebeck.de

Jinghua Groppe
Institut für Informationssysteme, Universität zu Lübeck
Lübeck, Germany
jinghua.groppe@uni-luebeck.de

ABSTRACT

Graph neural networks have shown great utility in streamlining and contextualizing machine learning on relational databases, due to their ability to automate integration and capture complex interactions between data. Encoding databases as graphs gives rise to large, heterogeneous graphs, as they originate from relational databases in real environments, therefore efficient (memory and complexity) techniques are needed for their analytics. In this work we propose a framework, which combines supervised machine learning on relational databases with graph neural networks and quantum graph neural network-based deep learning. We identify the key difficulties in developing and implementing such a framework, discuss possible techniques to solve them, and point out future research directions.

VLDB Workshop Reference Format:
Martin Vogrin, Rok Vogrin, Sven Groppe, and Jinghua Groppe. Supervised Learning on Relational Databases with Quantum Graph Neural Networks. VLDB 2024 Workshop: The Second International Workshop on Quantum Data Science and Management (QDSM'24).

1 INTRODUCTION

A relational database is a widely used type of database that stores and provides access to data points that are related to one another. Due to their inherent ability to link related data they have become the primary source for decision-making in millions of companies, and a prerequisite for (predictive) analytics of organizational data with machine learning. However, implementing machine learning on relational databases is, in practice, an increasingly complex problem. A major bottleneck is that ML models typically require single-table and flat data structures, requiring complex join operations and data transformations, subject to feature engineering. In industrial cases this is carried out on hundreds of tables and 10s of Terabytes of data [49], resulting in a complex, long, and error-prone process performed analogously with human-input. According to the study [10] only 22% of data-driven insights, a direct result of machine learning on company data, are utilized by decision-makers. This highlights the underlying problem of machine learning systems in industry (working mostly on large-scale relational data), which is inefficiency in implementing machine learning on relational databases and distrust in machine learning models to produce useful output.

To this end relational deep learning was introduced (see [20] and references therein, also [18]), as an effective technique for machine learning on relational databases. The idea is to map relational databases into graphs where entries become nodes and relationships are represented as edges, which creates a highly interconnected and richly annotated data structure. This structure can then be analysed using graph neural networks (GNNs), a specialized form of deep learning designed for graph data [38]. GNNs are incredibly efficient at extracting and analyzing data from relational databases, however the all-encompasing analytics brings also downsides. Analytics of graphs with GNNs is typically complex to execute. This is in part due to large size of graphs, as their complexity and complexity of their analytics grows exponentially with the number of nodes. At the same time graphs inherently lack (spatial) structure, combining long-range and short-range interactions.

Quantum graph neural networks (QGNNs) were introduced as the quantum analogue of classical GNNs [5, 42, 48]. Several different techniques for executing QGNN on Noisy Intermediate-Scale Quantum (NISQ) devices were subsequently introduced, constructing hybrid algorithms in different manners (e.g. by pooling, classical convolution, ego-decomposition, and similar). All approaches in some way reduce the dimensionality of the graph, to be able to encode it with the limited number of qubits, currently available. A notable example is exploring the feasibility of QGNNs to help track the particles produced by collisions in the LHC [46].

QGNNs show great promise in offering near- to medium- term improvement over classical, especially due to:

(1) Classical GNNs are closely related to random walks on graphs. Quantum random walks (on graphs) were shown to outperform classical in several scenarios [2, 3, 11, 12, 16, 28];

Proceedings of the VLDB Endowment. ISSN 2150-8097.

(2) Many of the graph problems are known to be in the NP-complete complexity class, while BQP-class problems (bounded-error quantum polynomial), the quantum analogue of P, are assumed to capture a larger class of NP problems than P;
(3) Graphs coming from relational databases are typically 10s of Terabytes large with millions of nodes and edges. Their analytics, especially at scale represents a significant burden for data infrastructure and a significant costs and environmental impact for corporations. Potential impact of improvements through quantum computing is large.

There are several challenges related to making QGNNs for relational database machine learning industrially useful, including:

- Build block-encoding-type algorithms for efficient quantum encoding for sparse matrices, allowing implementation of quantum convolutional and related graph neural networks. There is only a very limited body of work currently available on block encoding of sparse matrices [8];
- Building new types of QGNNs, and adapt classical frameworks, especially considering hardware constraints;
- Implement error correction adapted to QGNNs;
- Develop use cases and adapt QGNNs to relational deep learning. Benchmark against classical models.

The combination of relational deep learning and QGNNs has significant potential for both database research and industrial applications. In database research, these technologies can be used for query optimizations, data indexing and retrieval (e.g., quantum-enhanced indexing), pattern recognition and anomaly detection, and semantic data integration. In commercial and industrial applications, potential use cases include contextually segmenting customers, providing automated recommendations, predicting the next purchase time, identifying churn, fraud detection [27], and materials search [36].

1.1 Organization of the paper

The rest of this paper is organized as follows: We introduce the classical framework for supervised learning on relational databases with graph neural networks in Section 2. Inspired by the classical architecture, we propose a quantum graph neural network architecture for machine learning on relational databases in Section 3, and discuss recent developments in quantum graph neural networks and approaches to handling hardware limitations. We identify future research directions and briefly survey further related work in Section 4. Finally in Section 5 we conclude this paper.

2 SUPERVISED LEARNING ON RELATIONAL DATABASES WITH GRAPH NEURAL NETWORKS

Graph neural networks are a class of neural networks designed to work directly with graph-structured data and respect their (permutation) invariance. Unlike traditional neural networks that operate on fixed-size vector inputs, GNNs can process graphs of varying sizes and shapes, making them suitable for tasks involving nodes, edges, and their relationships. Popular GNN architecture is the message-passing framework, generalizing classical architectures such as graph convolutional networks (GCNs) and graph attention networks (GATs).

The applications of graph neural networks span various fields. In social network analysis, they can be used to predict user behavior, detect communities, and identify influential nodes. In bioinformatics, graph neural networks facilitate the prediction of protein-protein interactions, the discovery of functional modules, and the identification of disease-related genes. In IT security, graph neural networks are applied to analyze code graphs in order to detect whether the code contains vulnerabilities [22, 23]. Recommendation systems benefit from graph neural networks by representing users and items as nodes in a bipartite graph, improving the accuracy of recommendations through better capturing of user-item relationships.

2.1 Deep learning on relational databases with GNNs

Traditional methods for machine learning on relational databases often rely on extensive manual feature engineering, which can be both time-consuming and prone to human error. To address this, the author of [18] introduced a systematic method to convert relational databases into graph structures, thereby enabling the direct application of GNNs. The method generalizes other supervised machine learning tasks on tables, such as regression or classification.

The core of their methodology is the RDBToGraph algorithm, which transforms the relational schema into a graph where each row of a table is represented as a node, and foreign key relationships define the edges between nodes (see Figure 1). This graph-based representation preserves the relational information inherently present in relational databases, which traditional feature engineering methods might overlook.

The experimental evaluation on three diverse datasets indicates that GNN-based approaches outperform traditional feature engineering methods in the selected cases. The article also demonstrates scalability of the proposed method and its applicability to various types of relational data, suggesting a broad potential impact on how machine learning tasks are approached in the context of relational databases.

2.2 Relational Deep Learning

The Relational Deep Learning framework was introduced in [20] and represents a significant advancement in the application of GNNs to relational databases. Traditional machine learning approaches require extensive feature engineering to convert relational data into a single table format, which can be time-consuming and suboptimal. Relational deep learning, however, views relational databases as temporal, heterogeneous graphs where rows are nodes and primary-foreign key relationships are edges (also other approaches to encoding databases as graphs are considered in the article). This representation allows the direct application of GNNs, which can learn from the data without manual feature engineering, leading to more accurate and efficient predictive models.

A critical aspect of relational deep learning is its treatment of temporal evolution. In relational databases, data evolves over time, and capturing this dynamic nature is essential for accurate predictions. The framework ensures that during the learning process, nodes

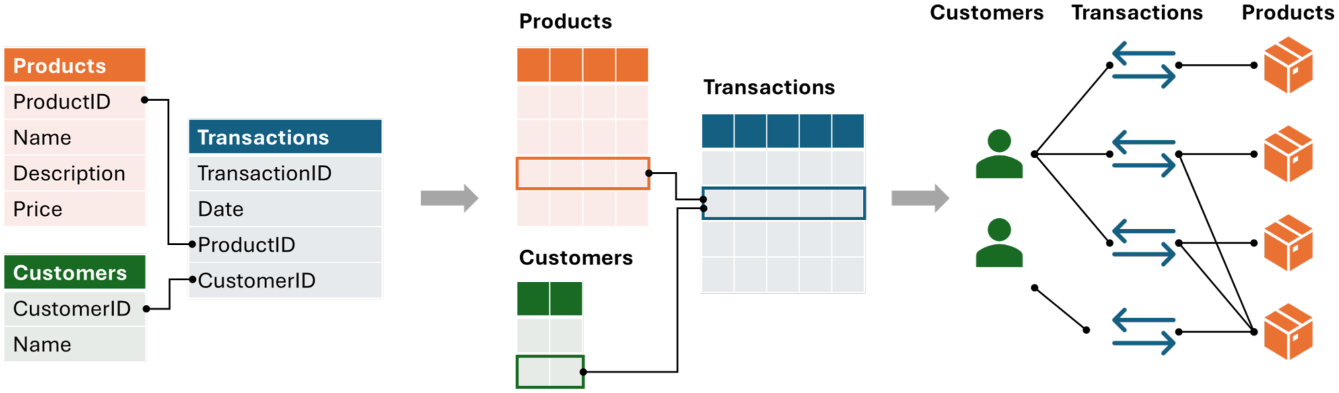

Figure 1: Relational database to graph encoding (see also [18]). In [20] in addition to the relational database tables, additional training table is considered storing node labels and training tasks, coming from temporal evolution of the database (graph).

only receive messages from other nodes with earlier timestamps, preventing information leakage and the creation of unrealistic shortcuts. This temporal message-passing strategy not only stabilizes model generalization across time but also allows the model to be dynamically updated as new data arrives, maintaining the relevance of predictions. Temporal tasks, in particular, benefit from the task table's ability to specify historical training labels and control data visibility based on timestamps, ensuring temporal consistency and robustness in the model training process.

3 SUPERVISED LEARNING ON RELATIONAL DATABASES WITH QUANTUM GRAPH NEURAL NETWORKS

Machine learning on relational databases with quantum graph neural networks combines the recent progress in QGNN development with the usability of RDBToGraph encoding and relational deep learning and provides an alternative to classical deep learning on relational data. The system consists of the standard relational deep learning process, where the GNN step is replaced by the quantum machine learning pipeline, consisting of:

(1) Quantum encoding of the graph:
    (a) Quantum encoding of node and feature data
    (b) Quantum encoding of the graph structure (edge index/adjacency matrix)
(2) Variational quantum circuit and classical optimization
(3) Backend integration and adaptation
(4) Potential classical pre-processing

QGNNs [5, 42, 48] are specialized variational quantum circuits, which respect the structure of input data (permutation equivariance). Specialized QGNNs were considered in [4, 15, 31, 35, 54]. The constructions are broadly split into two classes, using a version of the quantum graph neural network ansatz (1) or constructing quantum circuits through block-encoding of sparse (adjacency) matrices, traditional approach to it being LCU [17] and modern approaches such as FABLE [9] and its specialization to sparse matrices, e.g. [8]. A promising approach to hardware limitations is executing quantum machine learning on subgraphs [1, 55] or hybrid models, containing classical and quantum layers [14, 30].

Quantum properties were also used to enhance classical GNNs [44]. Use cases range from drug response prediction [37], fraud detection [27], and travelling salesman problem learning [42], to particle track reconstruction [46].

3.1 Quantum Graph Neural Networks

Quantum systems can naturally be represented as graphs where nodes correspond to quantum subsystems (e.g., qubits) and edges represent interactions or entanglement between them. The total Hilbert space $\mathcal{H}$ of such a system is the tensor product of the Hilbert spaces of individual subsystems, $\mathcal{H}_N = \bigotimes_i \mathcal{H}_{\text{node},i}$. One can also consider a Hilbert space of edges $\mathcal{H}_E = \bigotimes_{(i,j)\in E} \mathcal{H}_{\text{edge},ij}$, encoding the edge features of the graph. The total Hilbert space of the graph is then $\mathcal{H} = \mathcal{H}_N \otimes \mathcal{H}_E$. Often different architectures limit their analysis of node features, reducing the Hilbert space to $\mathcal{H}_N$, and utilizing edge data solely for couplings between different quantum systems corresponding to graph nodes in the Hamiltonian. We will adopt this simplification in the following.

3.1.1 Quantum Graph Neural Network Ansatz. The QGNN ansatz [48] is a parameterized quantum circuit designed to operate on graph-structured data. It consists of sequences of Hamiltonian evolutions parameterized by trainable parameters, which dictate the interactions within the quantum system. These Hamiltonians are designed to reflect the graph topology, ensuring that the quantum states evolve in a manner consistent with the underlying graph structure.

The QGNN ansatz is given by the unitary operator

$$\hat{U}_{\text{QGNN}}(\eta,\theta) = \prod_{p=1}^{P}\left[\prod_{q=1}^{Q} e^{-i\eta_{pq}\hat{H}_q(\theta)}\right], \tag{1}$$

where Q is the number of Hamiltonian evolutions and P a number of repetitions of the sequence. The product is time-ordered and the Hamiltonians $\hat{H}_q(\theta)$ depend on a set of trainable parameters θ.

$$\hat{H}_q(\theta) \equiv \sum_{(j,k)\in E}\sum_{r\in\mathcal{I}_{jk}} W_{qrjk}\hat{O}_j^{(qr)} \otimes \hat{P}_k^{(qr)} + \sum_{j\in N}\sum_{r\in\mathcal{J}_j} B_{qrj}\hat{Q}_j^{(qr)}. \tag{2}$$

Here, W_{qrjk} and B_{qrj} are real-valued coefficients that are generally independent and trainable, forming the parameter set $\theta = \bigcup_{q,j,k,r}\{W_{qrjk}\} \cup \bigcup_{q,j,r}\{B_{qrj}\}$. The operators $\hat{O}_j^{(qr)}, \hat{P}_j^{(qr)}, \hat{Q}_j^{(qr)}$ are Hermitian and act on the Hilbert space of the j-th node of the graph. The sets $\mathcal{I}_{jk}$ and $\mathcal{J}_j$ index terms corresponding to edges E and nodes N, respectively. The first part of the Hamiltonian encodes the edge interactions, while the second part acts as update of the individual node, analogous to classical GNNs.

Since $\hat{H}_q(\theta)$ is hermitian and Kronecker products of Pauli matrices form a basis of even-dimensional hermitian matrices, $\hat{O}_j^{(qr)}$, $\hat{P}_j^{(qr)}$, $\hat{Q}_j^{(qr)}$ can be taken to be Pauli matrices by redefinition of the coefficients. The linear term then becomes an application of a parametrized rotation with a Pauli matrix, while the quadratic term is two-qubit rotation, which can be decomposed into universal gates. In [42], $\hat{O}_j^{(qr)}, \hat{P}_k^{(qr)}$ were chosen as Pauli-Z operators and $\hat{Q}_j^{(qr)}$ as Pauli-X.

Different GNN architectures can be realized within the QGNN setting by specialization of the ansatz, such as Quantum Graph Recurrent Neural Networks (QGRNNs) and Quantum Graph Convolutional Neural Networks (QGCNNs), each tailored for specific types of quantum processes and applications.

QGRNNs are designed to model dynamic processes on graphs by tying the temporal parameters across iterations. This approach mirrors classical recurrent neural networks, where parameters are shared over time steps. QGRNNs are particularly suited for tasks involving the learning of Hamiltonian dynamics in quantum systems, providing an efficient way to simulate and predict the evolution of quantum states over time. In QGRNNs the temporal parameters are tied between iterations ($\eta_{pq} \longrightarrow \eta_q$), the QGRNN ansatz thus becomes

$$\hat{U}_{QGRNN} = e^{-i\Delta \hat{H}_{\text{eff}}(\theta_p)}, \tag{3}$$

with $\hat{H}_{\text{eff}} = \Delta^{-1}\sum_q \eta_q H_q(\theta)$, and $\Delta = \sum_q |\eta_q|$.

Quantum Graph Convolutional Neural Networks (QGCNNs) extend the concept of convolutional neural networks to quantum systems. They enforce permutation invariance in the quantum domain by tying parameters across the graph, analogous to translational invariance in classical CNNs:

$$W_{qrjk} \longrightarrow W_{qr}, \quad B_{qrj} \longrightarrow B_{qr}. \tag{4}$$

The mixing term W_{qr} parameters can be fixed to encode the action of the adjacency (or related) matrix, while the linear term represents a parametrized rotation with parameters optimized through standard variational methods.

3.2 Construction of QGNNs via block-encoding

The authors of [15, 31] consider a different approach to "quantization" of graph neural networks, by directly applying the adjacency matrix to the qubit register. While such an approach directly incorporates the graph structure into the quantum circuit, adjacency matrix is typically non-unitary. To apply the adjacency matrix to the Hilbert space associated to the graph, the authors introduce an auxiliary register and block-encode the adjacency matrix to apply it to node channel register (part of register recording node labels).

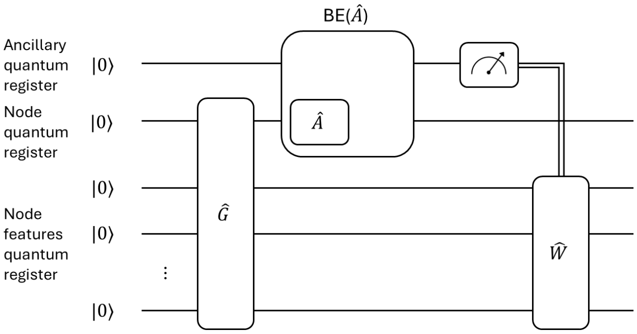

Figure 2: Schematic circuit for implementing QGCN of [15]. The operator $\hat{G}$ prepares the state $|\psi_X\rangle$ by encoding the node features. The architectures of QGCN, QGAN and QMP circuits are drawn in [31].

The variational circuit is then applied to the part of node register encoding node features.

Schematically, a feature matrix $X \in \mathbb{R}^{C\times N}$ associated to N nodes with C features can be encoded as:

$$|\psi_X\rangle = \sum_{i=1}^{N}\sum_{j=1}^{C} X_{ij}|i\rangle \otimes |j\rangle. \tag{5}$$

Here states $|i\rangle$ label node components, while states $|j\rangle$ encode the node features. To implement a QGCN, we first expand the node register with ancillary qubits to block encode the adjacency matrix and then act with the operator

$$\text{BE}(\hat{A}) \otimes \hat{W}, \tag{6}$$

where BE($\hat{A}$) is the block encoded adjacency matrix $\hat{A}$ and $\hat{W}$ is a parametrized quantum circuits, to be optimized. The action of the operator on the state $|0\rangle^{\otimes n_{\text{anc}}} \otimes |\psi_X\rangle$, where n_{anc} denotes the number of ancillary qubits needed for block encoding is

$$(\text{BE}(\hat{A}) \otimes \hat{W})(|0\rangle^{\otimes n_{\text{anc}}} \otimes |\psi_X\rangle) = \sum_{i=1}^{N}\sum_{j=1}^{C}(\hat{A}X\hat{W}^{\text{tr}})_{ij}|i\rangle \otimes |j\rangle, \tag{7}$$

after measuring the ancillary qubits. The resulting amplitudes $\hat{A}X\hat{W}^{\text{tr}}$ are exactly the transformed features under the classical GCNs [29]. Similar construction can be carried out for the quantum versions of GAN and MP architectures, with additional complications due to greater flexibility of the models.

The advantage of the approach is the freedom of encoding the quantum states. The most common encoding scheme for node features are linear-memory type encoding schema, like basis encoding, or angle encoding, which require $O(n)$ qubits to encode the data. More memory efficient encoding techniques like amplitude encoding are preferred, or even necessary for graph data, where in real applications, graphs consist of millions of nodes and edges. The approach of [15] and [31] allows for different encoding schema, which significantly lowers quantum memory constraints.

Additionally, the proposed circuits are fully quantum, and that the authors of [31] were able to develop analogues of prominent classical GNNs (GCNs, GANs and MP). While the existing quantum hardware is still unable to process such complicated quantum circuits for meaningful applications, the quantum nature of the circuits is important for performance in the long run.

3.3 Hybrid Quantum-Classical Graph Neural Networks

Hybrid circuits are a combination of classical and quantum processing layers. They offer a convenient way to process complex graphs on the current quantum hardware with limited amount of qubits. The main purpose of hybrid circuits is to reduce the dimensionality of graphs, where classical layers can compress high-dimensional node features into lower-dimensional embeddings before being fed to subsequent quantum layers. No essential structural information on graphs is lost in the process [26].

Typically, classical GNNs with fixed training parameters (learning rate, batch size, optimization algorithm) serve as a preprocessing step, embedding the graph into lower-dimensional space. A result of GNN-based processing are flat graphs, which can be further reduced by hierarchical representations of graphs via pooling. A state-of-the-art method is differentiable graph pooling that decomposes graphs by hierarchically clustering graph nodes [52]. Followed by efficient quantum encoding, and QGNN layers, this process has a potential to support analytics on very large databases [36].

4 FUTURE RESEARCH DIRECTIONS

Utilizing QGNNs for machine learning on relational data is opening a number of opportunities to apply quantum computing and quantum machine learning to problems in industry. However, to bring QGNNs to utility scale, there are still many challenges to be overcome. Besides the challenges of classical relational deep learning, such as efficient database encoding, adapting GNNs and integrating them with other downstream applications, there are several challenges originating in the quantum nature of computation.

4.1 Quantum Graph Encoding and Learning

On the quantum algorithm side one of the main challenges is developing efficient quantum graph encoding techniques, such as block encoding, edge index encoding, and especially leveraging graph sparsity in doing so, as classical models have excelled at doing. Another crucial area is the design of (further) variational quantum circuits (VQCs) adapted to graph structures. Balancing the expressibility and performance of these circuits is essential to capture the complexities of relational data while maintaining computational feasibility. Future research should focus on constructing VQCs that align with the topological features of graphs, thereby enhancing the accuracy and robustness of quantum graph representation learning models.

On the use cases side, building different classes of quantum graph neural networks (QGNNs) presents a promising research direction. Exploring various QGNN architectures (quantum analogues of classical GNN classes) and learning paradigms will provide deeper insights into their capabilities and limitations, paving the way for more sophisticated graph representation learning models.

For utility-scale quantum computing optimizing QPU performance with tools analogous to Pytorch DataLoader in classical machine learning frameworks is necessary. These tools should facilitate efficient data batching, loading, and preprocessing, maximizing the throughput and performance of quantum algorithms. By incorporating best practices from classical machine learning, these quantum-specific tools can significantly enhance the efficiency and scalability of quantum graph neural networks.

Developing concrete use cases for quantum machine learning on relational data will demonstrate the practical value of these techniques. Potential applications span domains such as social network analysis, bioinformatics, and financial modeling. Collaborations with industry partners to develop case studies and prototypes can showcase the real-world benefits of quantum-enhanced graph learning.

Benchmarking quantum graph neural network models on realistic datasests and against established benchmarks, like RelBench [20], is crucial for evaluating performance and demonstrating advantages over classical approaches. Standardized benchmarks provide a clear framework for comparing models, highlighting strengths, and identifying improvement areas. Comprehensive benchmarking efforts should be a focus of future research to validate the efficacy and efficiency of quantum approaches. This effort is currently in progress by the authors and will be published as a separate publication.

4.2 Quantum Query Optimization

Optimizing the join order is one of the most critical steps for query performance during query optimization. The authors of [53] provide a short survey about recent works that utilize quantum computers for query optimization. None of these approaches utilize QGNNs so far.

There have been several approaches to formulate the join order problem as quadratic unconstrained binary optimization (QUBO) that can run on quantum annealers or universal quantum computers using approaches like quantum approximate optimization algorithm (QAOA) and variational quantum eigensolver. While the approaches in [40] evaluated on quantum hardware and in [39] evaluated on quantum-inspired hardware are restricted to left-deep join trees based on a known transformation to mixed-integer linear programming, the solutions in [33, 34, 41] support general bushy join trees. Here, [33] can solve problems on current quantum hardware with [34] extending the number of joins that can be handled on today's quantum hardware by splitting the search space by sending more QUBO problems to the quantum hardware. In comparison, [41] has a lower complexity in the constraints of the QUBO problems for a large number of joins.

Multiple query optimization (MQO) determines a common optimal query plan for running a set of queries. While [45] proposed a QUBO formulation of the multiple query optimization problem including an extensive evaluation on a quantum annealer, [19] evaluated MQO using QAOA and discussed a multi-step reformulation of the MQO problem.

Quantum reinforcement learning has been utilized for join ordering during query optimization by returning the join tree within a single step [50, 51] and within multiple steps by incrementally constructing the join tree [21]. Both approaches are catching up with classical optimizers but with slightly better results by the multi-step approach. The approach in [47] estimates the database queries' cardinalities, costs, and execution times with a quantum natural language processing method.

Graph neural networks have been applied to optimize the join orders in queries (e.g., [13]), but the application of QGNNs to this problem is missing so far. Open challenges include the development of a quantum counterpart and an experimental evaluation measuring the benefits and drawbacks of such an approach.

4.3 Quantum Transaction Management

The work [53] performs a short survey about research efforts that utilize quantum computers for transaction scheduling, and the contribution in [56] provides a detailed analysis of these tasks' potential quantum accelerations when applying various quantum methods. We discuss other quantum accelerations of database tasks in Section 4.4.

Concurrently running conflicting transactions causes performance overhead in concurrency control mechanisms, which can be avoided by reordering the execution of transactions. This optimization is called *transaction scheduling*. So far transaction scheduling has been optimized by formulating transaction scheduling as QUBO problem and solving the problem on a quantum annealer [6, 7] and by introducing an oracle function to be used in Grover's search algorithm [24].

While the literature covers applying GNNs to the related flexible job-shop scheduling [43], there is a research gap in applying QGNNs to this and the transaction scheduling problem.

4.4 Other Issues

The authors in [25] discuss the potential of applying quantum machine learning to database index tuning for determining an index configuration that minimizes the workload processing time, but they have not investigated the possibilities of applying QGNNs for solving the addressed problem so far.

GNNs have also been applied to match subgraphs in graphs [32]. The development of a quantum counterpart utilizing QGNNs for this approach including the integration in a graph database and a rigorous investigation of the advantages, limits, and experimental analysis is missing so far.

5 SUMMARY AND CONCLUSIONS

In this paper, we explored the integration of supervised machine learning on relational databases with graph neural networks (GNNs) and the extension to quantum graph neural networks (QGNNs). The motivation for this integration stems from the inherent complexity and inefficiency of machine learning approaches on relational databases, which often require extensive feature engineering and complex data transformations, as well as machine learning on large graphs in the GNN applications. By leveraging the capabilities of quantum computing through QGNNs, we aim to enable more efficient large-scale relational data processing.

Relational deep learning has shown great promise by mapping relational databases into graph structures, and utilizing historiacal data to automate machine learning task construction. This approach significantly improves the efficiency and accuracy of machine learning tasks on relational data by preserving the inherent relational information and enabling the automated extraction of complex interactions within the data. GNNs are particularly well-suited for this purpose due to their ability to handle graph-structured data and respect permutation invariance.

Quantum graph neural networks represent a promising advancement in this field. By leveraging quantum computing principles such as superposition and entanglement, QGNNs have the potential to handle the large, complex graphs derived from relational databases more efficiently than classical GNNs. This potential efficiency gain is crucial for managing the massive datasets often found in industrial applications, which can span tens of terabytes and involve millions of nodes and edges.

However, the development and implementation of QGNNs for relational database machine learning come with several challenges. Efficient quantum encoding of graph data, construction of variational quantum circuits, and handling the limitations of current quantum hardware are key hurdles that need to be addressed. Developing new QGNN architectures and optimizing quantum query processing are crucial areas for future research. Additionally, practical applications and benchmarking QGNNs against classical models in real-world scenarios will be essential for demonstrating their viability and advantages.

The potential environmental and societal benefits of QGNNs are significant. Quantum computing can help mitigate the high energy consumption associated with large-scale data processing, contributing to more sustainable practices in the field. Furthermore, QGNNs can enhance the interpretability and trust in machine learning models by effectively capturing complex relational data, making insights more actionable for decision-makers. This addresses a critical gap in current machine learning systems, which often struggle with trust and utility due to their inability to handle relational data contextually.

In conclusion, the integration of relational deep learning with QGNNs represents a significant step forward in the field of machine learning on relational databases. By leveraging the power of quantum computing, we can potentially overcome the limitations of classical approaches, leading to more efficient, scalable, and interpretable machine learning models on relational data. This research opens up new avenues for both theoretical exploration and practical application, paving the way for advancements in various domains reliant on relational data.

ACKNOWLEDGMENTS

The authors would like to thank Rene Markovič for discussion and collaboration on related work. The work of S.G. is funded by the German Federal Ministry of Education and Research within the funding program quantum technologies—from basic research to market—contract number 13N16090.

REFERENCES

[1] Xing Ai, Zhihong Zhang, Luzhe Sun, Junchi Yan, and Edwin Hancock. 2022. Decompositional quantum graph neural network. *arXiv preprint arXiv:2201.05158* (2022).

[2] Andris Ambainis, András Gilyén, Stacey Jeffery, and Martins Kokainis. 2020. Quadratic speedup for finding marked vertices by quantum walks. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 412–424.

[3] Simon Apers, András Gilyén, and Stacey Jeffery. 2019. A unified framework of quantum walk search. *arXiv preprint arXiv:1912.04233* (2019).

[4] Lu Bai, Yuhang Jiao, Lixin Cui, Luca Rossi, Yue Wang, Philip S. Yu, and Edwin R. Hancock. 2023. Learning Graph Convolutional Networks Based on Quantum Vertex Information Propagation. *IEEE Transactions on Knowledge and Data Engineering* 35, 2 (2023), 1747–1760. https://doi.org/10.1109/TKDE.2021.3106804

[5] Kerstin Beer, Megha Khosla, Julius Köhler, and Tobias J. Osborne. 2021. Quantum machine learning of graph-structured data. arXiv:2103.10837 [quant-ph]

[6] Tim Bittner and Sven Groppe. 2020. Avoiding blocking by scheduling transactions using quantum annealing. In *Proceedings of the 24th Symposium on International Database Engineering & Applications*. 1–10.

[7] Tim Bittner and Sven Groppe. 2020. Hardware Accelerating the Optimization of Transaction Schedules via Quantum Annealing by Avoiding Blocking. *Open Journal of Cloud Computing (OJCC)* 7, 1 (2020), 1–21. http://nbn-resolving.de/urn:nbn:de:101:1-2020112218332015343957

[8] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. 2024. Explicit quantum circuits for block encodings of certain sparse matrices. *SIAM J. Matrix Anal. Appl.* 45, 1 (2024), 801–827.

[9] Daan Camps and Roel Van Beeumen. 2022. FABLE: Fast Approximate Quantum Circuits for Block-Encodings. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 104–113. https://doi.org/10.1109/QCE53715.2022.00029

[10] Kevin Carillo, Nadine Galy, Cameron Guthrie, and Anne Vanhems. 2018. How to turn managers into data-driven decision makers: Measuring attitudes towards business analytics. *Business Process Management Journal* 25 (07 2018). https://doi.org/10.1108/BPMJ-11-2017-0331

[11] Shantanav Chakraborty, Leonardo Novo, Andris Ambainis, and Yasser Omar. 2016. Spatial search by quantum walk is optimal for almost all graphs. *Physical review letters* 116, 10 (2016), 100501.

[12] Shantanav Chakraborty, Leonardo Novo, and Jérémie Roland. 2020. Optimality of spatial search via continuous-time quantum walks. *Physical Review A* 102, 3 (2020), 032214.

[13] Jin Chen, Guanyu Ye, Yan Zhao, Shuncheng Liu, Liwei Deng, Xu Chen, Rui Zhou, and Kai Zheng. 2022. Efficient Join Order Selection Learning with Graph-based Representation *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 97–107. https://doi.org/10.1145/3534678.3539303

[14] Samuel Yen-Chi Chen, Tzu-Chieh Wei, Chao Zhang, Haiwang Yu, and Shinjae Yoo. 2021. Hybrid quantum-classical graph convolutional network. *arXiv preprint arXiv:2101.06189* (2021).

[15] Yanhu Chen, Cen Wang, Hongxiang Guo, et al. 2022. Novel architecture of parameterized quantum circuit for graph convolutional network. *arXiv preprint arXiv:2203.03251* (2022).

[16] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. 2003. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. 59–68.

[17] Andrew M Childs and Nathan Wiebe. 2012. Hamiltonian simulation using linear combinations of unitary operations. *arXiv preprint arXiv:1202.5822* (2012).

[18] Milan Cvitkovic. 2020. Supervised learning on relational databases with graph neural networks. *arXiv preprint arXiv:2002.02046* (2020).

[19] Tobias Fankhauser, Marc E. Solèr, Rudolf Marcel Füchslin, and Kurt Stockinger. 2023. Multiple Query Optimization Using a Gate-Based Quantum Computer. *IEEE Access* 11 (2023), 114031–114043. https://doi.org/10.1109/ACCESS.2023.3324253

[20] Matthias Fey, Weihua Hu, Kexin Huang, Jan Eric Lenssen, Rishabh Ranjan, Joshua Robinson, Rex Ying, Jiaxuan You, and Jure Leskovec. 2023. Relational Deep Learning: Graph Representation Learning on Relational Databases. arXiv:2312.04615 [cs.LG]

[21] Maja Franz, Tobias Winker, Sven Groppe, and Wolfgang Mauerer. 2024. Hype or Heuristic? Quantum Reinforcement Learning for Join Order Optimisation. *arXiv* 2405.07770 (2024). https://doi.org/10.48550/arXiv.2405.07770

[22] Jinghua Groppe, Sven Groppe, and Ralf Möller. 2023. Variables are a Curse in Software Vulnerability Prediction. In *The 34th International Conference on Database and Expert Systems Applications (DEXA 2023)*. Springer Nature, 516–521.

[23] Jinghua Groppe, Sven Groppe, Daniel Senf, and Ralf Möller. 2024. There Are Infinite Ways to Formulate Code: How to Mitigate the Resulting Problems for Better Software Vulnerability Detection. *Information* 15, 4 (2024), 216. https://www.mdpi.com/2078-2489/15/4/216

[24] Sven Groppe and Jinghua Groppe. 2021. Optimizing Transaction Schedules on Universal Quantum Computers via Code Generation for Grover's Search Algorithm. In *25th International Database Engineering & Applications Symposium*. ACM. https://doi.org/10.1145/3472163.3472164

[25] Le Gruenwald, Tobias Winker, Umut Çalıkyılmaz, Jinghua Groppe, and Sven Groppe. 2023. Index Tuning with Machine Learning on Quantum Computers for Large-Scale Database Applications. In *Joint Proceedings of Workshops at the 49th International Conference on Very Large Data Bases (VLDB 2023) - International Workshop on Quantum Data Science and Management (QDSM'23), Vancouver, Canada*. https://ceur-ws.org/Vol-3462/QDSM5.pdf

[26] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[27] Nouhaila Innan, Abhishek Sawaika, Ashim Dhor, Siddhant Dutta, Sairupa Thota, Husayn Gokal, Nandan Patel, Muhammad Al-Zafar Khan, Ioannis Theodonis, and Mohamed Bennai. 2024. Financial fraud detection using quantum graph neural networks. *Quantum Machine Intelligence* 6, 1 (Feb. 2024). https://doi.org/10.1007/s42484-024-00143-6

[28] Viv Kendon. 2020. How to compute using quantum walks. *arXiv preprint arXiv:2004.01329* (2020).

[29] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[30] Yan-Ying Liang, Si-Le Tang, Zhe-Hao Yi, Hao-Zhen Si-Tu, and Zhu-Jun Zheng. 2024. Learnability of a hybrid quantum-classical neural network for graph-structured quantum data. *arXiv preprint arXiv:2401.15665* (2024).

[31] Yidong Liao, Xiao-Ming Zhang, and Chris Ferrie. 2024. Graph Neural Networks on Quantum Computers. arXiv:2405.17060 [quant-ph]

[32] Zhaoyu Lou, Jiaxuan You, Chengtao Wen, Arquimedes Canedo, Jure Leskovec, et al. 2020. Neural subgraph matching. *arXiv preprint arXiv:2007.03092* (2020).

[33] Nitin Nayak, Jan Rehfeld, Tobias Winker, Benjamin Warnke, Umut Çalıkyılmaz, and Sven Groppe. 2023. Constructing Optimal Bushy Join Trees by Solving QUBO Problems on Quantum Hardware and Simulators. In *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments (BiDEDE), Seattle, WA, USA*. https://doi.org/10.1145/3579142.3594298

[34] Nitin Nayak, Tobias Winker, Umut Çalıkyılmaz, Sven Groppe, and Jinghua Groppe. 2024. Quantum Join Ordering by Splitting the Search Space of QUBO Problems. *Datenbank-Spektrum* 24, 1 (2024), 21–32. https://doi.org/10.1007/s13222-024-00468-3

[35] Seunghyeok Oh, Jaeho Choi, and Joongheon Kim. 2020. A tutorial on quantum convolutional neural networks (QCNN). In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 236–239.

[36] Ju-Young Ryu, Eyuel Elala, and June-Koo Kevin Rhee. 2023. Quantum Graph Neural Network Models for Materials Search. *Materials* 16, 12 (2023). https://doi.org/10.3390/ma16124300

[37] Asel Sagingalieva, Mohammad Kordzanganeh, Nurbolat Kenbayev, Daria Kosichkina, Tatiana Tomashuk, and Alexey Melnikov. 2023. Hybrid quantum neural network for drug response prediction. *Cancers* 15, 10 (2023), 2705.

[38] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80. https://doi.org/10.1109/TNN.2008.2005605

[39] Manuel Schönberger, Stefanie Scherzinger, and Wolfgang Mauerer. 2023. Ready to Leap (by Co-Design)? Join Order Optimisation on Quantum Hardware. *Proc. ACM Manag. Data* 1, 1, Article 92, 27 pages.

[40] Manuel Schönberger, Immanuel Trummer, and Wolfgang Mauerer. 2023. Quantum-Inspired Digital Annealing for Join Ordering. In *Proc. VLDB Endow.*, Vol. 16.

[41] Manuel Schönberger, Immanuel Trummer, and Wolfgang Mauerer. 2023. Quantum Optimisation of General Join Trees. In *VLDBW'23*.

[42] Andrea Skolik, Michele Cattelan, Sheir Yarkoni, Thomas Bäck, and Vedran Dunjko. 2023. Equivariant quantum circuits for learning on weighted graphs. *npj Quantum Information* 9, 1 (2023), 47.

[43] Wen Song, Xinyang Chen, Qiqiang Li, and Zhiguang Cao. 2023. Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning. *IEEE Transactions on Industrial Informatics* 19, 2 (2023), 1600–1610. https://doi.org/10.1109/TII.2022.3189725

[44] Slimane Thabet, Romain Fouilland, Mehdi Djellabi, Igor Sokolov, Sachin Kasture, Louis-Paul Henry, and Loïc Henriet. 2023. Enhancing Graph Neural Networks with Quantum Computed Encodings. *arXiv preprint arXiv:2310.20519* (2023).

[45] Immanuel Trummer and Christoph Koch. 2016. Multiple query optimization on the D-Wave 2X adiabatic quantum computer. *Proceedings of the VLDB Endowment* 9, 9 (May 2016), 648–659.

[46] Cenk Tüysüz, Federico Carminati, Bilge Demirköz, Daniel Dobos, Fabio Fracas, Kristiane Novotny, Karolos Potamianos, Sofia Vallecorsa, and Jean-Roch Vlimant. 2020. A quantum graph neural network approach to particle track reconstruction. *arXiv preprint arXiv:2007.06868* (2020).

[47] Valter Uotila. 2023. SQL2Circuits: Estimating Metrics for SQL Queries with A Quantum Natural Language Processing Method. arXiv:2306.08529 (June 2023). https://doi.org/10.48550/arXiv.2306.08529 arXiv:2306.08529 [quant-ph].

[48] Guillaume Verdon, Trevor McCourt, Enxhell Luzhnica, Vikash Singh, Stefan Leichenauer, and Jack Hidary. 2019. Quantum Graph Neural Networks. arXiv:1909.12264 [quant-ph]

[49] Adrian Vogelsgesang, Michael Haubenschild, Jan Finis, Alfons Kemper, Viktor Leis, Tobias Muehlbauer, Thomas Neumann, and Manuel Then. 2018. Get Real: How Benchmarks Fail to Represent the Real World. In *Proceedings of the Workshop on Testing Database Systems* (Houston, TX, USA) *(DBTest'18)*. Association for Computing Machinery, New York, NY, USA, Article 1, 6 pages. https://doi.org/10.1145/3209950.3209952

[50] Tobias Winker, Sven Groppe, Valter Uotila, Zhengtong Yan, Jiaheng Lu, Maja Franz, and Wolfgang Mauerer. 2023. Quantum Machine Learning: Foundation, New Techniques, and Opportunities for Database Research. In *Proceedings of ACM SIGMOD/PODS International Conference on Management of Data (SIGMOD)*.

[51] Tobias Winker, Umut Çalıkyılmaz, Le Gruenwald, and Sven Groppe. 2023. Quantum Machine Learning for Join Order Optimization using Variational Quantum Circuits. In *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments (BiDEDE), Seattle, WA, USA*. https://doi.org/10.1145/3579142.3594299

[52] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. *CoRR* abs/1806.08804 (2018). arXiv:1806.08804 http://arxiv.org/abs/1806.08804

[53] Gongsheng Yuan, Jiaheng Lu, Yuxing Chen, Sai Wu, Yao Chang, Zhengtong Yan, Tuodu Li, and Gang Chen. 2023. Quantum Computing for Databases: A Short Survey and Vision. In *International Workshop on Quantum Data Science and Management (QDSM'23) (CEUR Workshop Proceedings)*. CEUR.

[54] Jin Zheng, Qing Gao, and Yanxuan Lü. 2021. Quantum graph convolutional neural networks. In *2021 40th Chinese Control Conference (CCC)*. IEEE, 6335–6340.

[55] Kaixiong Zhou, Zhenyu Zhang, Shengyuan Chen, Tianlong Chen, Xiao Huang, Zhangyang Wang, and Xia Hu. 2022. QuanGCN: Noise-Adaptive Training for Robust Quantum Graph Convolutional Networks. *arXiv preprint arXiv:2211.07379* (2022).

[56] Umut Çalıkyılmaz, Sven Groppe, Jinghua Groppe, Tobias Winker, Stefan Prestel, Farida Shagieva, Daanish Arya, Florian Preis, and Le Gruenwald. 2023. Opportunities for Quantum Acceleration of Databases: Optimization of Queries and Transaction Schedules. *Proc. VLDB Endow.* 16, 9 (2023), 2344–2353. https://doi.org/10.14778/3598581.3598603