Quantum Data Management in the NISQ Era

Rihan Hai
Web Information Systems Group
Delft University of Technology
r.hai@tudelft.nl

Shih-Han Hung
Department of Electrical Engineering
National Taiwan University
shihhanh@ntu.edu.tw

Tim Coopmans
QuTech
Delft University of Technology
t.j.coopmans@tudelft.nl

Tim Littau
Web Information Systems Group
Delft University of Technology
t.m.littau@tudelft.nl

Floris Geerts
Adrem Data Lab
University of Antwerp
floris.geerts@uantwerp.be

ABSTRACT

Quantum computing has emerged as a transformative force in the evolution of computing technology. Recent efforts have applied quantum techniques to classical database challenges, such as query optimization, data integration, index selection, and transaction management. In this paper, we shift focus to a critical yet underexplored area: *data management for quantum computing*. We are currently in the noisy intermediate-scale quantum (NISQ) era, where qubits, while promising, are fragile and still limited in scale. After differentiating quantum data from classical data, we outline current and future data management paradigms in the NISQ era and beyond. We address the data management challenges arising from the emerging demands of near-term quantum computing. Our goal is to chart a clear course for future quantum-oriented data management research, establishing it as a cornerstone for the advancement of quantum computing in the NISQ era.

PVLDB Reference Format:
Rihan Hai, Shih-Han Hung, Tim Coopmans, Tim Littau, and Floris Geerts. Quantum Data Management in the NISQ Era. PVLDB, 18(6): 1720 - 1729, 2025.
doi:10.14778/3725688.3725701

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/InfiniData-Lab/Quantum.

1 INTRODUCTION

Data management is crucial in our increasingly data-driven world, exemplified by the widespread use of database systems and the rapid advancement of big data systems [1, 129]. In recent years, the field of computer science has been energized by the transformative potential of quantum technologies. Quantum computing promises computational capacities far beyond what traditional computers can achieve [101]. However, quantum computing is still in a nascent stage, i.e., the *noisy intermediate-scale quantum (NISQ)* era, characterized by quantum computers that are constrained by noise and limited numbers of qubits [111].

With the ongoing development of quantum computing, new data management challenges naturally emerge. The fundamental differences between quantum and classical computing call for novel data representations to effectively preserve quantum information [101, 143]. Moreover, many quantum computing tasks are inherently *data- and computation-intensive* due to the need to handle large-scale quantum states [18], multidimensional quantum data structures [11, 153], and error correction codes [47, 48, 55, 131]. For example, simulating large-scale quantum computation on classical computers involves processing vast amounts of quantum information, leading to significant scalability and optimization challenges [18, 85, 151, 153]. By addressing data management challenges in the NISQ era, the database community has a unique opportunity to significantly enhance the scalability and reliability of quantum technologies, which will advance both research and real-world quantum applications.

These new data management challenges are not yet clearly defined, despite the recent efforts by the database community. Existing work has explored leveraging quantum computers as new hardware to address classical database challenges, such as query optimization [43, 44, 100, 118–121, 134, 147], data integration [49], index selection [58, 74], and transaction management [12, 57, 132]. However, fundamental questions about data management in the NISQ era remain unanswered. Specifically, how should we define and manage data in the context of near-term and future quantum advancements? Given the unique features of quantum computing, such as superposition and entanglement, what are the new considerations to be addressed for effective data management? What data structures and data management systems will best support the development of quantum technologies, particularly given a limited number of qubits, noise, and other challenges of the NISQ era?

While recent vision papers, tutorials, and surveys [23, 60, 148, 154, 155] have begun discussing the intersection of data management and quantum computing, they primarily focus on how quantum technologies can accelerate classical database operations. In contrast, our work delves into an equally important yet underexplored area: *data management for quantum computing*. At the moment, the fundamental concepts, research directions, and problem definitions in this area remain obscure within the database

Proceedings of the VLDB Endowment, Vol. 18, No. 6 ISSN 2150-8097.
doi:10.14778/3725688.3725701

community. This paper initiates the exploration of these critical aspects and lays the foundation for further deeper integration of data management and quantum computing. We will not dive into the basics of quantum computing and information, as ample resources are available [11, 113, 151, 153]. Instead, we focus on outlining the vision for data management paradigms in the NISQ era and on identifying potential research challenges that are particularly relevant to the database community, i.e., which could have a significant impact on the advancement of today's quantum technologies.
Contributions. Our contributions are summarized as follows:

- **Fundamentals**: We first explain quantum data and differentiate it from classical data (Sec. 2).
- **Roadmap**: We present our vision for data management research for quantum computing in three paradigms (Sec. 3).
- **Research problems**: We elaborate on near-term research questions in data management for quantum computing, and report on preliminary experimental results (Sec. 4).

2 DATA IN THE QUANTUM ERA

Classical data is the information that is collected, processed, and stored with traditional computing methods. Today, most of the classical data is stored and queried using database systems such as relational databases, document stores, graph databases, and vector databases [126]. We refer to *quantum data* as information collected and processed using *quantum computing devices*, i.e., computing devices that follow the rules of quantum mechanics to their advantage [101]. Quantum data is represented by qubits. Next, we list key differences between quantum and classical data below to help understand the unique features of quantum data.

1. Quantum data is probabilistic. Unlike a classical bit, which is 0 or 1, a quantum bit can be in *superposition*. Mathematically, a zero state and a one state may be represented by a unit vector in the *standard basis*. That is, the zero state $|0\rangle$ is represented by the vector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and the one state $|1\rangle$ is represented by the vector $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. A single qubit state, denoted $|\psi\rangle$, may be represented by a superposition, i.e., a linear combination of $|0\rangle$ and $|1\rangle$: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, for some pair of complex numbers α, β called *amplitudes*, which satisfy $|\alpha|^2 + |\beta|^2 = 1$. The probabilistic nature arises when a *measurement* is performed. When a measurement is performed on the state $|\psi\rangle$, the outcome 0 is obtained with probability $|\alpha|^2$ and the outcome 1 is obtained with probability $|\beta|^2$, and the state permanently changes to the obtained outcome. In classical computing, individual bits can be concatenated to form a bit string, e.g., the three-bit string "010." Similarly, we may concatenate qubits into a multi-qubit state. In this case, an n-qubit state can be represented as a superposition of $|x\rangle$ for $x \in \{0,1\}^n$, or equivalently a vector of 2^n components.

2. Quantum data is fragile. Quantum computers are anticipated to outperform classical computers in solving certain problems. However, with current quantum technology in the NISQ era, quantum resources remain scarce, and quantum data is prone to noise. *Decoherence* [110], is the process where quantum states lose their coherence due to environmental interactions, resulting in the gradual loss of quantum information. Quantum noise, resulting from unintended couplings with the environment, can significantly degrade the performance of quantum computers [136]. Commonly used noise models, such as amplitude damping, phase damping, bit-flip, and phase-flip, mathematically describe how various types of quantum noise lead to decoherence [101].

3. Quantum data can be entangled. Another difference between qubits and bits is *entanglement* [40], which means that multiple qubits are correlated such that measuring the state of one qubit immediately affects the others. A well-known entangled state is the *Bell's state*:

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

where knowledge of one qubit determines the other. Please see Appendix A of our online report [59] for more background.

3 OUR VISION: DATA MANAGEMENT FOR QUANTUM COMPUTING

To facilitate future research on data management for quantum computing, we first sketch the whole landscape in Fig. 1. We categorize this landscape into three distinct paradigms based on how quantum and classical data are transformed and utilized, and the type of hardware involved—whether a quantum or classical computer is employed. Our goal is to introduce data management researchers to quantum use cases, and by making a distinction based on the nature and role of data within cases.

***I Classical simulation of quantum computing paradigm**: classical data represents quantum states and operations.* This paradigm presents great potential for new database challenges. It focuses on using *classical data to represent and simulate quantum data*. For example, Bell's state from Sec. 2 is represented by a classical vector. This paradigm is more accessible as it relies on classical computers, not quantum ones.

The representative task in this paradigm is *simulation*. Simulation is the process of emulating quantum computation, enabling researchers to model and analyze quantum processes as if they were operating on actual quantum hardware [151, 153].[1] Simulations are of paramount importance in the NISQ era [111]. Consider, for instance, the concept of quantum supremacy [18], which seeks to demonstrate the superior capabilities of quantum computers over classical computers. Given that large-scale quantum computers are not yet available in the NISQ era, simulation is essential for comparing the scalability of quantum computers with classical ones. Additionally, simulations play a crucial role in the development of new quantum algorithms, allowing researchers to design, debug, and validate the correctness of these algorithms before deploying them on expensive quantum devices. Moreover, simulations aid in the development of quantum hardware by evaluating error mitigation schemes, predicting algorithm runtimes, and more [7, 71, 140, 158, 160]. Simulation serves as a foundational tool across key areas of quantum computing, including quantum supremacy, quantum algorithms, quantum hardware, error correction, and the exploration of potential quantum applications [151].

The potential for database research in this paradigm is immense. First, imagine innovative database systems specifically designed to manage classical data that supports simulation. Second, efficient

[1] In this work, by simulation we refer to *classical simulation*. Another related term is *quantum simulation* [22, 53], which pertains to simulating quantum mechanics on a quantum computer, a subject studied in physics.

caching strategies for expensive operations, such as repeated simulations of quantum circuits with varying error parameters, present another important research direction. The third research direction in this paradigm involves the representation of quantum states and operations. This is useful for quantum design questions such as optimizing the number of quantum gates or gate depth [77, 159] and compiling textbook quantum circuits to only include the low-level operations that real quantum devices support [125]. As the representation of a quantum state (a vector) or quantum gate (a matrix) generally grows exponentially with the number of qubits, developing efficient methods to store and process these data structures is key to making simulations of quantum computation feasible, either in vector form or through more compressed structures like tensor networks [11, 103] or other advanced representations [20, 72, 137, 160].

II Joint Quantum-Classical Computing paradigm*: classical preprocessing & postprocessing for quantum computing.* This paradigm focuses on the situation involved in most quantum-technology applications: classical computers handle the preprocessing and postprocessing of data for quantum devices, such as quantum chips, quantum sensors, and quantum computers. In this paradigm, the focus is on data which is primarily stored and processed as *classical data*. For instance, a classical computer sends instructions (classical data) to a quantum chip, which performs computations and returns measurement outcomes—also classical data—back to the classical computer. Thus, the quantum device takes classical input and produces classical output.

Database research can enhance this process by developing efficient systems for managing, storing, and querying the classical data involved in preprocessing, postprocessing, and iterative feedback loops between classical computers and quantum devices, ensuring seamless integration and optimization of data workflows. For instance, the development of quantum error correction—a critical area in quantum computing—can benefit significantly from efficient graph data analytics techniques, as further discussed in Sec. 4.3.

We demonstrate the importance of this paradigm through three key categories of quantum applications: *First*, applications where the quantum computation is completed immediately upon returning a result. For example, two separated quantum computers can generate a secure, shared key (password) by encoding bits into qubits, then sending and measuring the qubits [10, 109]. This is followed by classical postprocessing on the bitstring (i.e. classical data) that the measurement returns [41]. The resulting bitstring is a secret key that can later be used for secure communication. In the *second* category, the quantum computation is stalled temporarily, and the quantum chip still holds quantum data on which the computation will continue later. An example is the detection of errors during a quantum computation, where at fixed timesteps during the computation check measurements are performed [55, 131], whose outcomes are then decoded to find out which error has most likely occurred [8, 92]. *Third*, efficient preprocessing is critical for quantum algorithms, which requires *loading* or *encoding* classical data into the quantum domain, typically by mapping classical bits onto quantum bits. This process often requires encoding classical data into a quantum circuit, either as part of the algorithm [130] or to output a quantum state with amplitudes representing the classical data [32], which is a challenging problem.

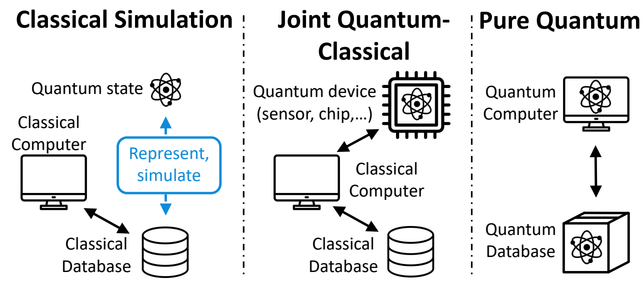


Figure 1: Landscape: Data management for quantum computing

III Pure quantum computing paradigm*: quantum-native data storage and processing.* Finally, we envision a future research paradigm beyond NISQ, when we have large-scale, fault-tolerant quantum computers. We will deal with pure *quantum data*, i.e., qubits. Here, the quantum hardware supports storing the qubits for long enough to perform other tasks in the meantime. This enables for example a cloud quantum computer [21] that rotates its resources among multiple clients, various types of quantum machine learning [24], and quantum random access memory [54], where data can be queried in superposition. A recent vision [67] shows a possible future: quantum data is collected from quantum sensing systems, e.g., for discovering a black hole; then stored and processed via the quantum memory of a quantum computer.

The data management research in this paradigm will be centered around handling quantum data, possibly developing quantum-native algorithms. Moreover, storing quantum data will remain challenging, as quantum data storage will still be costly in various ways (the number of qubits is not unbounded, robust storage of quantum data requires large-scale quantum error correction, which is computationally intensive, etc.). Another research topic is to efficiently allocate the available qubits. This includes, for example, various scheduling and allocation tasks [26, 45, 52, 64, 96, 123] and efficient use of different quantum hardware types each of which has speed-decoherence trade-offs [38]. Given the amount of quantum data in this paradigm, most research here calls for novel quantum data processing algorithms and systems beyond NISQ era.

4 NEAR-TERM RESEARCH PROBLEMS

We will now explore data management research questions across the three paradigms. In Sec. 4.1 and 4.2, we focus on a key challenge within Paradigm I: Classical simulation of quantum computing paradigm. In Sec. 4.3, we broaden the scope to include research opportunities across all three paradigms.

4.1 The Challenge of Simulating Quantum Computation

The input for a simulation is a quantum algorithm described as a quantum circuit,[2] consisting of input qubits upon which quantum gates are applied, ultimately resulting in a probability vector of the output states, i.e., measurement outcomes (see Sec. 2). Then, a simulation is a classical computation that computes that output

[2]A uniform family of circuits, to be precise, one for each input size.

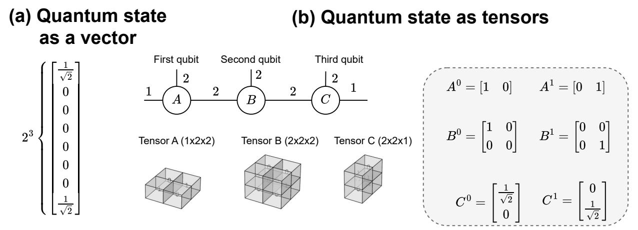
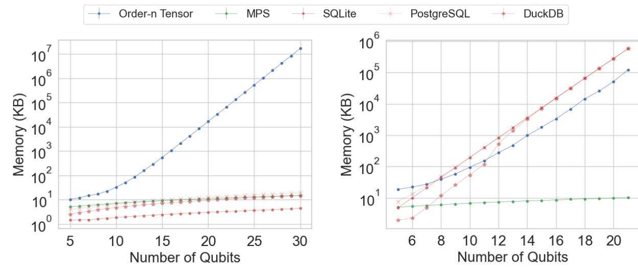


Figure 2: (a) The 3-qubit GHZ state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ represented as a state vector of size $2^3 = 8$. (b) GHZ state as a tensor network. For better visualization, in the grey box, we write the tensors in vector/matrix form by slicing the last dimension of A, B, and C. See Appendices A and B1 of [59] for details.



(a) GHZ State (sparse circuit) **(b) QFT (dense circuit)**

Figure 3: Memory usage comparison of three RDBMS-based solutions (red) against a general approach, i.e., order-n tensor baseline (blue) and the SotA MPS baseline (green), both implemented in Python. The x-axis shows increasing numbers of qubits, and the y-axis shows memory consumption (KB) in $\log_{10}$-scale. For sparse circuit (GHZ), RDBMS solutions demonstrate significantly lower memory usage than the order-n tensor baseline, with SQLite showing a slight advantage over the MPS baseline. For the dense circuit QFT, the memory usage of RDBMS solutions grows comparably to or exceeds the order-n tensor baseline, indicating the need for improvements in handling dense tensor computations.

distribution (so-called *strong* simulation), or samples from it (*weak* simulation) [138]. We primarily focus on strong simulation.

The major practical bottleneck of (strong) simulation is *scalability* [18, 85, 151, 153]. When simulating a quantum state of n qubits, the state is typically represented as a vector with a size of 2^n. The size of the state vector grows exponentially as n increases. For instance, an experiment demonstrating quantum supremacy required 2.25 petabytes for 48 qubits, reaching the memory limits of today's supercomputers [18]. To overcome the memory restriction, existing simulation tools have explored approximation [73, 94, 139], data compression [149], parallelization [66], and distributed computing [124]. In Fig. 2 we illustrate concepts mentioned in this section.

4.2 Databases to the Rescue

Simulation offers significant opportunities for database research, and conversely, database expertise can greatly enhance simulation.

> We envision a *classical-quantum simulation system* (CQSS) with the following capabilities: (i) automatically providing the most efficient simulation of the input circuit by selecting optimal data structures and operations based on available resources and circuit properties; (ii) operating inherently out-of-core to support the simulation of large circuits that exceed main memory capacity; (iii) ensuring consistency to prevent data corruption and enabling recovery in the event of large-scale simulation crashes; and (iv) improving the entire simulation workflow, including parameter tuning, data collection and querying, exploration and visualization.

At its core, a CQSS must, at a minimum, be capable of evaluating quantum circuits. This primarily involves performing linear algebraic operations, often described in terms of *tensor networks* [11].[3] A *tensor* is a multidimensional array, and the dimensions along which a tensor extends are its *indices*. A tensor network consists of vertices representing tensors, and edges the indices. Free indices are depicted as legs, i.e., edges that only connect to one vertex in the graph. Connecting two vertices by joining a leg corresponds to the contraction with the corresponding indices. Fig. 2b shows the tensor network representation of the GHZ state. Tensor contraction is the summation of shared indices between tensors and is a generalization of matrix multiplication [11]. See Appendix A of [59] for more details. The first question we ask is as follows.

> **Q1.** *Should we push the simulation workload to existing DBMSs?*

In other words, should we map the simulation workload to SQL queries and store the results in a DBMS? The advantage is that we can benefit from the efficiency and portability of modern database systems. And indeed, initial efforts from the database community [13, 133] have begun to address the out-of-core simulation challenge by mapping quantum states and gates to SQL queries. For instance, in [13], qubit states and gates are represented as tensor networks, with sparse tensors in COO format and Einstein summation operations mapped to SQL queries. Additionally, the recent abstract [133] introduces the idea of supporting dense vectors using SQL UNION ALL and CASE statements. Despite these efforts that support basic quantum states and gates up to 18 qubits and a circuit depth of 18 in separate experiments, the core challenges of representing general quantum states and operations in DBMSs and achieving scalability remain. Moreover, our preliminary results in Fig. 3 demonstrate that RDBMS-based simulations are efficient for quantum circuits involving sparse tensor computations. All experimental details, together with a space and time complexity analysis, can be found in Appendix B of [59]. An early prototype is described in [86]. RDBMS-based solutions are competitive with, and may even outperform, state-of-the-art representations like matrix product states (MPS) [108] in specific scenarios. Interestingly, this advantage does not extend to dense circuits such as quantum fourier transform (QFT) circuits, highlighting the need for further optimizations to enhance RDBMS systems for simulation workloads. In Appendix C of [59]

[3]For simplicity of exposition, we do not consider non-tensor-based simulation methods which represent quantum states by their symmetries, rather than by complex-valued vectors [56].

we also discuss the use of NoSQL databases. On the other hand, the connection to tensor computations suggests the following question.

Q2. *Should we instead leverage tensor-based database technologies for simulation?*

Indeed, the database community has a substantial body of work on tensor computation, especially with the recent synergies between databases and machine learning (ML) such as *learning ML models over relational data* [15, 76, 90, 104, 157]. These studies range from high-level representations to the runtime optimization of ML workflows. At the representation level, numerous studies represent data as tensors [80] and aim to integrate relational and tensor operations [68, 75, 115]. Meanwhile, system-building efforts from the database community [14, 14, 51, 62, 88, 97, 102, 107, 116, 117] focus on performance efficiency and scalability, employing optimization techniques during compilation (e.g., operator fusion [15, 16]) and runtime (e.g., parallelization [17]). Similar endeavours exist in the areas of compilers and high performance computing (HPC) [28, 70, 78, 114].

Our envisioned classical-quantum simulation system holds the potential to achieve optimal performance. By integrating relevant optimization techniques from the aforementioned existing systems, we can facilitate the automatic optimization of the underlying tensor network. The actual runtime performance will depend on several factors: the operations within the tensor network, the sparsity of the tensors involved, the memory layouts (i.e., how tensors are stored in memory, such as row-major or column-major formats), and the available hardware [15, 117]. To address challenges unique to quantum computing, we highlight two key gaps below.

4.2.1 Quantum-specific optimizations. Existing simulators for quantum computing are mostly in Python [34, 141], C++ [3, 141] or Julia [89]. There are also dedicated quantum programming languages and domain-specific compilation techniques. See the survey [63]. In addition, tensor network rewriting techniques have been considered in the ZX-calculus [30]. ZX-calculus is a graphical language for representing and reasoning about quantum computations, represented as tensor networks with specialized tensors called *spiders* [137]. Its graphical nature enables simplifications by locally merging connected vertices while preserving the original tensor, useful in various contexts including formal verification [37, 83, 137].

Quantum-specific optimizations share some conceptual similarities with classical database optimization, yet they diverge significantly due to the unique characteristics of quantum data and operations. First, classical databases rely on relational or NoSQL data models (graph, document-based, etc.) with indexing mechanisms like B-trees or hash tables to facilitate efficient access and retrieval. In simulations, however, quantum states are represented as state vectors (Fig. 2a) or by using more advanced data structures such as tensor networks (Fig. 2b). Second, in classical databases, query optimization involves techniques like join ordering to identify the optimal ordering of join operations between relations for an efficient query plan [84, 128, 135]. For tensor-based simulators, a defining feature is the optimization of *contraction order*, which determines the most efficient sequence to contract tensors representing quantum states [34, 153]. Contraction order optimization parallels the role of join order optimization in traditional databases but is more complex due to the exponential growth of quantum state dimensions and the unique features of quantum data, such as entanglement. Third, classical database systems rely heavily on cost estimation to predict and minimize resource use for query execution. Quantum computation simulators, however, lack cost estimation frameworks and instead use optimizations like parallelization, SIMD (Single-Instruction, Multiple-Data) processing, and matrix decomposition techniques (e.g., SVD) to improve performance [153]. Quantum-specific issues, such as quantum noise (Sec. 2), complicate simulation further. These differences lead to the need for fundamentally new optimization frameworks to manage and simulate quantum computation effectively, beyond classical structures and processes. The following question now arises.

Q3. *Can we build an optimizer for simulations?*

An optimizer can be designed, given a circuit, to determine the most efficient way to simulate it. For example, when a circuit allows for a tractable simulation (e.g., circuits of bounded treewidth), the optimizer should compile it into an efficient simulation. Similarly, based on the sparsity of (intermediate) quantum states, specialized sparse gate computations may be employed instead of the standard dense computations. Particular to the quantum setting is the noisy character of the quantum computation, which is known to impact the efficiency of simulation [2, 65, 66, 106, 112]. Hence, an optimizer needs to take noise levels into account. Moreover, the optimizer should consider the downstream task – whether it's computing precise probabilities for strong simulations or sampling measurement outcomes for weak simulations – and select the appropriate simulation strategy (algorithm) accordingly.

As part of the optimizer, we may also want to check whether two quantum states are exactly or approximately equivalent, analogous to checking equivalent conjunctive queries [27, 31, 79]. Having algorithms in place for testing equivalence may help to avoid redundant computations during simulation.

4.2.2 Quantum-specific data representations. Simulating quantum circuits requires precise and efficient representations of quantum states and operations, accounting for the complex nature of quantum mechanics. Ensuring accuracy and scalability as the system grows in complexity is crucial. We mentioned tensor networks as a way to represent the simulation, but various other data representations exist [143], e.g., based on algebraic decision diagrams.

Q4. *What are good data representations – possibly beyond tensors – for supporting simulations?*

An operation on a data structure is *tractable* if it executes in polynomial time with respect to the input size [36]. The choice of data representation can significantly impact whether certain quantum operations (e.g., gates, measurement explained in Sec. 2 and Appendix A of [59]) are tractable [143]. We emphasize three main requirements for data representation: *expressiveness*, *closure*, and *succinctness/tractability*.

For expressiveness, the data representation must be sufficiently rich to represent quantum states and measurements. Closure refers to the property that the result of quantum operations on the represented states can also be represented within the same data representation. This property ensures compositionality. Finally, it is crucial to minimize the space required to represent a quantum state, as the number of qubits and the level of entanglement can result in exponentially large states. Therefore, we seek data representations that are succinct, minimizing the space needed to store data while still allowing for efficient query processing. It may also be interesting to consider these questions beyond tensor network representations as well [20, 72, 137, 160]. A concrete idea for a novel data representation is to combine MPS and the recently-proposed *local-invertible map decision diagram* (LIMDD) [142]. LIMDD compresses a state vector by lumping together parts of the vector that are equivalent modulo simple quantum gates. In polynomial time and space in the number of qubits, LIMDDs can simulate circuits that MPS cannot and vice versa [143]. One approach to combine these strengths in a single data structure is applying the lumping to the MPS matrices instead of quantum state vectors, by building upon existing work on equivalence characterizations of MPS [81].

Q5. *Are there other benefits from relying on database technology?*

By using database technology we aim for the simulation of complex circuits that require significant memory, without relying on HPC infrastructures or the operating system's memory management. By controlling what data is pushed to secondary storage during simulations, we can achieve more efficient I/O behavior. Additionally, transaction management and recovery become important—where a transaction could, for example, represent a sequence of quantum gates—ensuring that computations can restart from saved checkpoints while maintaining the correctness of partially saved results. Parallel evaluation strategies in quantum circuit simulations also highlight the need for effective transaction scheduling, leading to more reliable computational processes. Undoubtedly, many challenges remain in this context when considering quantum simulation. We prioritize Q1–Q4, however, as first steps. A more extensive discussion can be found in Appendix D in [59].

4.3 More Data Management Opportunities

Beyond simulation for quantum computing, quantum-related research is broad, e.g., error correction [55, 122, 127], quantum networks [35, 146]. Next, we expand our discussion to uncover more opportunities spanning all three paradigms shown in Fig. 1.

Quantum error correction & Graph analytics. Quantum error correction (QEC) enables reliable execution of quantum computation on noisy quantum processors and is a crucial part of the roadmap to fault-tolerant quantum computation [131]. QEC process consists of two main aspects: coding and decoding. The input quantum information as well as the operations to be performed are coded using a quantum error-correcting code such as the surface code [19, 47, 48]. Intermittently, decoding is applied: errors are detected and corrected. Decoders [39, 46, 150] often leverage graph theory, solving tasks like a minimum-weight perfect matching (MWPM) problem on a graph where each node represents a check measurement. For MWPM, the number of nodes is of the same order as the number of qubits. Many interesting research problems arise how to model such graphs and design the data formats. Moreover, QEC has to be performed at high speeds to avoid decaying quantum states over time (see Sec. 2), e.g. at most several microseconds for some types of quantum hardware for decoding [8]. The scalability to many qubits, and speed required for QEC might benefit from advanced data management tools and techniques [9, 91, 93, 145, 152], offering a promising direction for further exploration.

Quantum experiments & Scientific data management. Quantum experiments, like any other scientific experiments, generate data and require data management. For instance, quantum mechanics experiments on supercomputers often apply HDF5 files to store data [33]. Another compelling direction is to explore how to build specialized data lakes [61, 98] or lakehouses [5, 6, 69] to support scientific data management for quantum computing.

ML & Quantum data management. i) Simulation: as explained in Sec. 4.2.1, a key challenge in optimizing tensor network based simulation is to find optimal tensor contraction order. A recent research direction is to apply machine learning (ML), specifically reinforcement learning (RL) and graph neural networks (GNN), to optimize tensor contraction order, addressing the computationally intensive nature of this challenge [87, 95]. ii) Error correction: decoding methods for quantum error correction like MWPM, face scalability challenges in large quantum systems, where rapid error detection within strict time limits is essential [144]. An interesting new direction is *data-driven QEC*, which employs ML techniques to quantum error correction, such as RL [4, 99, 156], multilayer perceptrons (MLP) [29], convolutional neural networks [25], and GNN [82]. iii) Improving quantum algorithm efficiency: Quantum algorithms are represented and implemented as quantum circuits, where efficiency can be improved by reducing costly gates like SWAP gate or by minimizing circuit depth and gate count. ML methods, specifically RL [42, 50] and MLP [105], have shown potential in optimizing circuit design to enhance the efficiency of quantum circuits on real devices.

5 CONCLUSION

We are at a privileged time in the evolution of data management, closely aligned with the rise of quantum computing. This convergence calls for innovative approaches to data representation, processing, and querying that are compatible with quantum computing. We here identify the unique features of quantum data compared to classical data, and advocate the exploration of three data management paradigms, which reveal a rich field of complex and significant challenges. As we continue to explore these paradigms, it becomes clear that the challenges we face, such as enhancing simulations with database technologies, are just the beginning. There exists a broader spectrum of challenges that remain unexplored, which require sustained and focused efforts from both the data management and quantum computing communities.

ACKNOWLEDGMENTS

This publication was supported (in part) by Dutch Research Council (VI.Veni.222.439). SHH acknowledges support from the National Science and Technology Council of Taiwan under Grant No. 114-2222-E-002-002-MY3, and from the National Taiwan University Core Consortium Project under Grant No. NTU-CC-114L895503. TC acknowledges the support received through the NWO Quantum Technology program (project number NGF.1582.22.035).

REFERENCES

[1] Daniel Abadi, Rakesh Agrawal, Anastasia Ailamaki, Magdalena Balazinska, Philip A Bernstein, Michael J Carey, Surajit Chaudhuri, Jeffrey Dean, AnHai Doan, Michael J Franklin, et al. 2016. The Beckman Report on Database Research. *Commun. ACM* 59, 2 (2016), 92–99.

[2] Dorit Aharonov, Xun Gao, Zeph Landau, Yunchao Liu, and Umesh Vazirani. 2023. A Polynomial-Time Classical Algorithm for Noisy Random Circuit Sampling. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*. 945–957.

[3] Quantum AI. 2021. qsim and qsimh. https://quantumai.google/qsim/overview. Accessed: 2024-08-05.

[4] Philip Andreasson, Joel Johansson, Simon Liljestrand, and Mats Granath. 2019. Quantum Error Correction for the Toric Code using Deep Reinforcement Learning. *Quantum* 3 (2019), 183.

[5] Michael Armbrust, Tathagata Das, Liwen Sun, Burak Yavuz, Shixiong Zhu, Mukul Murthy, Joseph Torres, Herman van Hovell, Adrian Ionescu, Alicja Łuszczak, et al. 2020. Delta Lake: High-Performance ACID Table Storage Over Cloud Object Stores. *Proceedings of the VLDB Endowment (PVLDB)* 13, 12 (2020), 3411–3424.

[6] Michael Armbrust, Ali Ghodsi, Reynold Xin, and Matei Zaharia. 2021. Lakehouse: A New Generation of Open Platforms That Unify Data Warehousing and Advanced Analytics. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, Vol. 8. 28.

[7] Koji Azuma, Stefan Bäuml, Tim Coopmans, David Elkouss, and Boxi Li. 2021. Tools for Quantum Network Design. *AVS Quantum Science* 3, 1 (2021), 014101.

[8] Francesco Battistel, Christopher Chamberland, Kauser Johar, Ramon WJ Overwater, Fabio Sebastiano, Luka Skoric, Yosuke Ueno, and Muhammad Usman. 2023. Real-Time Decoding for Fault-Tolerant Quantum Computing: Progress, Challenges and Outlook. *Nano Futures* 7, 3 (2023), 032003.

[9] Soheil Behnezhad, Mohammadtaghi Hajiaghayi, and David G Harris. 2023. Exponentially Faster Massively Parallel Maximal Matching. *Journal of the ACM (JACM)* 70, 5 (2023), 1–18.

[10] Charles H Bennett and Gilles Brassard. 1984. Quantum Cryptography: Public Key Distribution and Coin Tossing. *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing (ICCCSP)*, 175–179.

[11] Jacob Biamonte. 2020. Lectures on Quantum Tensor Networks. https://arxiv.org/abs/1912.10049

[12] 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 (IDEAS)*. Article 21, 10 pages.

[13] Mark Blacher, Julien Klaus, Christoph Staudt, Sören Laue, Viktor Leis, and Joachim Giesen. 2023. Efficient and Portable Einstein Summation in SQL. *Proceedings of the ACM on Management of Data (PACMMOD)* 1, 2, Article 121 (jun 2023), 19 pages.

[14] Matthias Boehm, Iulian Antonov, Sebastian Baunsgaard, Mark Dokter, Robert Ginthör, Kevin Innerebner, Florijan Klezin, Stefanie N. Lindstaedt, Arnab Phani, Benjamin Rath, Berthold Reinwald, Shafaq Siddiqui, and Sebastian Benjamin Wrede. 2020. SystemDS: A Declarative Machine Learning System for the End-to-End Data Science Lifecycle. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*.

[15] Matthias Boehm, Matteo Interlandi, and Chris Jermaine. 2023. Optimizing Tensor Computations: From Applications to Compilation and Runtime Techniques. In *Companion of the 2023 International Conference on Management of Data (SIGMOD)*. 53–59.

[16] Matthias Boehm, Berthold Reinwald, Dylan Hutchison, Prithviraj Sen, Alexandre V Evfimievski, and Niketan Pansare. 2018. On Optimizing Operator Fusion Plans for Large-Scale Machine Learning in SystemML. *Proceedings of the VLDB Endowment (PVLDB)* 11, 12 (2018), 1755–1768.

[17] Matthias Boehm, Shirish Tatikonda, Berthold Reinwald, Prithviraj Sen, Yuanyuan Tian, Douglas Burdick, and Shivakumar Vaithyanathan. 2014. Hybrid Parallelization Strategies for Large-Scale Machine Learning in SystemML. *Proceedings of the VLDB Endowment (PVLDB)* 7, 7 (2014), 553–564.

[18] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. 2018. Characterizing Quantum Supremacy in Near-Term Devices. *Nature Physics* 14, 6 (2018), 595–600.

[19] Sergey Bravyi, Matthias Englbrecht, Robert König, and Nolan Peard. 2018. Correcting Coherent Errors with Surface Codes. *npj Quantum Information* 4, 1 (2018), 55.

[20] Sergey Bravyi and David Gosset. 2016. Improved Classical Simulation of Quantum Circuits Dominated by Clifford Gates. *Physical Review Letters* 116 (Jun 2016), 250501. Issue 25.

[21] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. 2009. Universal Blind Quantum Computation. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 517–526.

[22] Iulia Buluta and Franco Nori. 2009. Quantum Simulators. *Science* 326, 5949 (2009), 108–111.

[23] Umut Çalikyilmaz, 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. *Proceedings of the VLDB Endowment (PVLDB)* 16, 9 (2023), 2344–2353.

[24] Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. 2022. Challenges and Opportunities in Quantum Machine Learning. *Nature Computational Science* 2, 9 (2022), 567–576.

[25] Christopher Chamberland, Luis Goncalves, Prasahnt Sivarajah, Eric Peterson, and Sebastian Grimberg. 2023. Techniques for Combining Fast Local Decoders with Global Decoders under Circuit-level Noise. *Quantum Science and Technology* 8, 4 (jul 2023), 045011. https://doi.org/10.1088/2058-9565/ace64d

[26] Aparimit Chandra, Wenhan Dai, and Don Towsley. 2022. Scheduling Quantum Teleportation with Noisy Memories. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 437–446.

[27] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing (STOC)*. 77–90.

[28] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. 2018. TVM: An Automated End-to-End Optimizing Compiler for Deep Learning. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 578–594.

[29] Cheng Chu, Nai-Hui Chia, Lei Jiang, and Fan Chen. 2022. QMLP: An Error-Tolerant Nonlinear Quantum MLP Architecture using Parameterized Two-Qubit Gates. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*. 1–6.

[30] Bob Coecke and Ross Duncan. 2011. Interacting Quantum Observables: Categorical Algebra and Diagrammatics. *New Journal of Physics* 13, 4 (2011).

[31] Sara Cohen, Werner Nutt, and Yehoshua Sagiv. 2007. Deciding Equivalences Among Conjunctive Aggregate Queries. *Journal of the ACM (JACM)* 54, 2 (2007), 50 pages.

[32] John A. Cortese and Timothy M. Braje. 2018. Loading Classical Data into a Quantum Computer. https://arxiv.org/abs/1803.01958

[33] Raúl de la Cruz, Hadrien Calmet, and Guillaume Houzeaux. 2012. Implementing a XDMF/HDF5 Parallel File System in Alya. (Dec 2012). https://doi.org/10.5281/zenodo.6241986

[34] NVIDIA cuQuantum Team. 2024. NVIDIA cuQuantum. https://docs.nvidia.com/cuda/cuquantum/22.07.1/cutensornet/overview.html. Accessed: 2024-08-14.

[35] Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arian Stolk, Przemysław Pawełczak, Robert Knegjens, Julio de Oliveira Filho, et al. 2019. A Link Layer Protocol for Quantum Networks. In *Proceedings of the ACM special interest group on data communication (SIGCOMM)*. 159–173.

[36] Adnan Darwiche and Pierre Marquis. 2002. A Knowledge Compilation Map. *Journal of Artificial Intelligence Research (JAIR)* 17 (2002), 229–264.

[37] Niel de Beaudrap and Dominic Horsman. 2020. The ZX Calculus is a Language for Surface Code Lattice Surgery. *Quantum* 4 (2020), 218.

[38] Nathalie P De Leon, Kohei M Itoh, Dohun Kim, Karan K Mehta, Tracy E Northup, Hanhee Paik, BS Palmer, Nitin Samarth, Sorawis Sangtawesin, and David W Steuerman. 2021. Materials Challenges and Opportunities for Quantum Computing Hardware. *Science* 372, 6539 (2021), eabb2823.

[39] Antonio deMarti iOlius, Patricio Fuentes, Román Orús, Pedro M Crespo, and Josu Etxezarreta Martinez. 2024. Decoding Algorithms for Surface Codes. *Quantum* 8 (2024), 1498.

[40] Albert Einstein, Boris Podolsky, and Nathan Rosen. 1935. Can Quantum-mechanical Description of Physical Reality Be Considered Complete? *Physical review* 47, 10 (1935), 777.

[41] David Elkouss, Jesus Martinez-mateo, and Vicente Martin. 2011. Information Reconciliation for Quantum Key Distribution. *Quantum Information & Computation* 11, 3 (2011), 226–238.

[42] Hongxiang Fan, Ce Guo, and Wayne Luk. 2022. Optimizing Quantum Circuit Placement via Machine Learning. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*. 19–24.

[43] Tobias Fankhauser, Marc E Solèr, Rudolf M Füchslin, and Kurt Stockinger. 2021. Multiple Query Optimization Using a Hybrid Approach of Classical and Quantum Computing. (2021). https://arxiv.org/abs/2107.10508

[44] Tobias Fankhauser, Marc E Soler, Rudolf M Füchslin, and Kurt Stockinger. 2023. Multiple Query Optimization Using a Gate-Based Quantum Computer. *IEEE Access* (2023).

[45] Paolo Fittipaldi, Anastasios Giovanidis, and Frédéric Grosshans. 2023. A Linear Algebraic Framework for Dynamic Scheduling over Memory-equipped Quantum Networks. *IEEE Transactions on Quantum Engineering (TQE)* (2023).

[46] Austin G Fowler. 2015. Minimum Weight Perfect Matching of Fault-Tolerant Topological Quantum Error Correction in Average O(1) Parallel Time. *Quantum Information & Computation* 15, 1-2 (2015), 145–158.

[47] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. 2012. Surface Codes: Towards Practical Large-Scale Quantum Computation.

Physical Review A 86, 3 (2012), 032324.
[48] Austin G Fowler, Ashley M Stephens, and Peter Groszkowski. 2009. High-Threshold Universal Quantum Computation on the Surface Code. *Physical Review A—Atomic, Molecular, and Optical Physics* 80, 5 (2009), 052312.
[49] Kristin Fritsch and Stefanie Scherzinger. 2023. Solving Hard Variants of Database Schema Matching on Quantum Computers. *Proceedings of the VLDB Endowment (PVLDB)* 16, 12 (2023), 3990–3993.
[50] Thomas Fösel, Murphy Yuezhen Niu, Florian Marquardt, and Li Li. 2021. Quantum Circuit Optimization with Deep Reinforcement Learning. https://arxiv.org/abs/2103.07585
[51] Zekai J. Gao, Shangyu Luo, Luis Leopoldo Perez, and Chris Jermaine. 2017. The BUDS Language for Distributed Bayesian Machine Learning. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. ACM, 961–976.
[52] Scarlett Gauthier, Gayane Vardoyan, and Stephanie Wehner. 2023. A Control Architecture for Entanglement Generation Switches in Quantum Networks. In *Proceedings of the 1st Workshop on Quantum Networks and Distributed Quantum Computing*. 38–44.
[53] Iulia M Georgescu, Sahel Ashhab, and Franco Nori. 2014. Quantum Simulation. *Reviews of Modern Physics* 86, 1 (2014), 153–185.
[54] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. 2008. Quantum random access memory. *Physical Review Letters* 100, 16 (2008), 160501.
[55] Daniel Gottesman. 1997. Stabilizer Codes and Quantum Error Correction. https://arxiv.org/abs/quant-ph/9705052
[56] Daniel Gottesman. 1998. The Heisenberg Representation of Quantum Computers. https://arxiv.org/abs/quant-ph/9807006
[57] Sven Groppe and Jinghua Groppe. 2021. Optimizing Transaction Schedules on Universal Quantum Computers via Code Generation for Grover's Search Algorithm. In *Proceedings of the 25th International Database Engineering & Applications Symposium (IDEAS)*. 149–156.
[58] Le Gruenwald, Tobias Winker, Umut Çalikyilmaz, 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) (CEUR Workshop Proceedings)*, Vol. 3462.
[59] Rihan Hai, Shih-Han Hung, Tim Coopmans, Tim Littau, and Floris Geerts. 2025. Quantum Data Management in the NISQ Era: Extended Version. https://arxiv.org/abs/2409.14111
[60] Rihan Hai, Shih-Han Hung, and Sebastian Feld. 2024. Quantum Data Management: From Theory to Opportunities. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 5376–5381.
[61] Rihan Hai, Christos Koutras, Christoph Quix, and Matthias Jarke. 2023. Data Lakes: A Survey of Functions and Systems. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 35, 12 (2023), 12571–12590.
[62] Dong He, Supun Chathuranga Nakandala, Dalitso Banda, Rathijit Sen, Karla Saur, Kwanghyun Park, Carlo Curino, Jesús Camacho-Rodríguez, Konstantinos Karanasos, and Matteo Interlandi. 2022. Query Processing on Tensor Computation Runtimes. *Proceedings of the VLDB Endowment (PVLDB)* 15, 11 (2022), 2811–2825.
[63] Bettina Heim, Mathias Soeken, Sarah Marshall, Chris Granade, Martin Roetteler, Alan Geller, Matthias Troyer, and Krysta Svore. 2020. Quantum Programming Languages. *Nature Reviews Physics* 2, 12 (2020), 709–722.
[64] Oscar Higgott and Nikolas P. Breuckmann. 2021. Subsystem Codes with High Thresholds by Gauge Fixing and Reduced Qubit Overhead. *Physical Review X* 11 (Aug 2021), 031039. Issue 3.
[65] Cupjin Huang, Fang Zhang, Michael Newman, Junjie Cai, Xun Gao, Zhengxiong Tian, Junyin Wu, Haihong Xu, Huanjun Yu, Bo Yuan, Mario Szegedy, Yaoyun Shi, and Jianxin Chen. 2020. Classical Simulation of Quantum Supremacy Circuits. https://arxiv.org/abs/2005.06787
[66] Cupjin Huang, Fang Zhang, Michael Newman, Xiaotong Ni, Dawei Ding, Junjie Cai, Xun Gao, Tenghui Wang, Feng Wu, Gengyan Zhang, et al. 2021. Efficient Parallelization of Tensor Network Contraction for Simulating Quantum Computation. *Nature Computational Science* 1, 9 (2021), 578–587.
[67] Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, et al. 2022. Quantum Advantage in Learning from Experiments. *Science* 376, 6598 (2022), 1182–1186.
[68] Zezhou Huang, Rathijit Sen, Jiaxiang Liu, and Eugene Wu. 2023. JoinBoost: Grow Trees Over Normalized Data Using Only SQL. *Proceedings of the VLDB Endowment (PVLDB)* 16, 11 (2023), 3071–3084.
[69] Paras Jain, Peter Kraft, Conor Power, Tathagata Das, Ion Stoica, and Matei Zaharia. 2023. Analyzing and Comparing Lakehouse Storage Systems. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*.
[70] Zhihao Jia, Oded Padon, James Thomas, Todd Warszawski, Matei Zaharia, and Alex Aiken. 2019. TASO: Optimizing Deep Learning Computation with Automatic Generation of Graph Substitutions. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles (SOSP)*. 47–62.
[71] Tyson Jones, Anna Brown, Ian Bush, and Simon C Benjamin. 2019. QuEST and High Performance Simulation of Quantum Computers. *Scientific reports* 9, 1 (2019), 10736.
[72] Richard Jozsa and Akimasa Miyake. 2008. Matchgates and Classical Simulation of Quantum Circuits. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 464, 2100 (2008), 3089–3106.
[73] Bjarni Jónsson, Bela Bauer, and Giuseppe Carleo. 2018. Neural-Network States for the Classical Simulation of Quantum Computing. https://arxiv.org/abs/1808.05232
[74] Manish Kesarwani and Jayant R. Haritsa. 2024. Index Advisors on Quantum Platforms. *Proceedings of the VLDB Endowment (PVLDB)* 17, 11 (2024), 3615–3628.
[75] Mahmoud Abo Khamis, Hung Q. Ngo, XuanLong Nguyen, Dan Olteanu, and Maximilian Schleich. 2018. In-Database Learning with Sparse Tensors. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*. 325–340.
[76] Mahmoud Abo Khamis, Hung Q. Ngo, Xuanlong Nguyen, Dan Olteanu, and Maximilian Schleich. 2020. Learning Models Over Relational Data Using Sparse Tensors and Functional Dependencies. *ACM Transactions on Database Systems (TODS)* 45, 2 (2020).
[77] Aleks Kissinger and John van de Wetering. 2020. Reducing the Number of Non-Clifford Gates in Quantum Circuits. *Physical Review A* 102, 2 (2020), 022406.
[78] Fredrik Kjolstad, Shoaib Kamil, Stephen Chou, David Lugato, and Saman Amarasinghe. 2017. The Tensor Algebra Compiler. *Proceedings of the ACM on Programming Languages (PACMPL)* 1, OOPSLA (2017), 1–29.
[79] Phokion G. Kolaitis and Moshe Y. Vardi. 1998. Conjunctive-query containment and constraint satisfaction. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*. 205–213.
[80] Dimitrios Koutsoukos, Supun Nakandala, Konstantinos Karanasos, Karla Saur, Gustavo Alonso, and Matteo Interlandi. 2021. Tensors: An Abstraction for General Data Processing. *Proceedings of the VLDB Endowment (PVLDB)* 14, 10 (2021), 1797–1804.
[81] Guglielmo Lami and Mario Collura. 2024. Unveiling the Stabilizer Group of a Matrix Product State. *Physical Review Letters* 133 (Jul 2024), 010602. Issue 1. https://doi.org/10.1103/PhysRevLett.133.010602
[82] Moritz Lange, Pontus Havström, Basudha Srivastava, Valdemar Bergentall, Karl Hammar, Olivia Heuts, Evert van Nieuwenburg, and Mats Granath. 2023. Data-driven decoding of quantum error correcting codes using graph neural networks. https://arxiv.org/abs/2307.01241
[83] Adrian Lehmann, Ben Caldwell, and Robert Rand. 2022. VyZX : A Vision for Verifying the ZX Calculus. https://arxiv.org/abs/2205.05781
[84] Viktor Leis, Andrey Gubichev, Atanas Mirchev, Peter A. Boncz, Alfons Kemper, and Thomas Neumann. 2015. How Good Are Query Optimizers, Really? *Proceedings of the VLDB Endowment (PVLDB)* 9, 3 (2015), 204–215.
[85] Riling Li, Bujiao Wu, Mingsheng Ying, Xiaoming Sun, and Guangwen Yang. 2020. Quantum Supremacy Circuit Simulation on Sunway TaihuLight. *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 31, 4 (2020), 805–816.
[86] Tim Littau and Rihan Hai. 2025. Qymera: Simulating Quantum Circuits using RDBMS. In *Companion of the 2025 International Conference on Management of Data (SIGMOD)*. to appear.
[87] Xiao-Yang Liu and Zeliang Zhang. 2023. Classical Simulation of Quantum Circuits Using Reinforcement Learning: Parallel Environments and Benchmark. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*. 67082–67102.
[88] Shangyu Luo, Zekai J. Gao, Michael N. Gubanov, Luis Leopoldo Perez, and Christopher M. Jermaine. 2017. Scalable Linear Algebra on a Relational Database System. In *33rd IEEE International Conference on Data Engineering (ICDE)*. 523–534.
[89] Xiu-Zhe Luo, Jin-Guo Liu, Pan Zhang, and Lei Wang. 2020. Yao.jl: Extensible, Efficient Framework for Quantum Algorithm Design. *Quantum* 4 (2020), 341.
[90] Nantia Makrynioti and Vasilis Vassalos. 2019. Declarative Data Analytics: A Survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 33, 6 (2019), 2392–2411.
[91] Ioana Manolescu and Madhulika Mohanty. 2023. Full-Power Graph Querying: State of the Art and Challenges. *Proceedings of the VLDB Endowment (PVLDB)* 16, 12 (2023), 3886–3889.
[92] Satvik Maurya and Swamit Tannu. 2024. Managing Classical Processing Requirements for Quantum Error Correction. https://arxiv.org/abs/2406.17995
[93] Andrew McGregor. 2014. Graph Stream Algorithms: A Survey. *ACM SIGMOD Record* 43, 1 (may 2014), 9–20.
[94] Matija Medvidović and Giuseppe Carleo. 2021. Classical Variational Simulation of the Quantum Approximate Optimization Algorithm. *npj Quantum Information* 7, 1 (2021), 101.
[95] Eli Meirom, Haggai Maron, Shie Mannor, and Gal Chechik. 2022. Optimizing Tensor Network Contraction Using Reinforcement Learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research)*, Vol. 162. PMLR, 15278–15292.
[96] R. Van Meter, T. D. Ladd, W. J. Munro, and K. Nemoto. 2009. System Design for a Long-Line Quantum Repeater. *IEEE/ACM Transactions on Networking (TON)*

17, 3 (2009), 1002–1013.
[97] Supun Nakandala, Karla Saur, Gyeong-In Yu, Konstantinos Karanasos, Carlo Curino, Markus Weimer, and Matteo Interlandi. 2020. A Tensor Compiler for Unified Machine Learning Prediction Serving. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 899–917.
[98] Fatemeh Nargesian, Erkang Zhu, Renée J Miller, Ken Q Pu, and Patricia C Arocena. 2019. Data Lake Management: Challenges and Opportunities. *Proceedings of the VLDB Endowment (PVLDB)* 12, 12 (2019), 1986–1989.
[99] Hendrik Poulsen Nautrup, Nicolas Delfosse, Vedran Dunjko, Hans J Briegel, and Nicolai Friis. 2019. Optimizing Quantum Error Correction Codes with Reinforcement Learning. *Quantum* 3 (2019), 215.
[100] Nitin Nayak, Jan Rehfeld, Tobias Winker, Benjamin Warnke, Umut Çalikyilmaz, 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 '23)*. 1–7.
[101] Michael A Nielsen and Isaac L Chuang. 2010. *Quantum Computation and Quantum Information*. Cambridge university press.
[102] Milos Nikolic and Dan Olteanu. 2018. Incremental View Maintenance with Triple Lock Factorization Benefits. In *Proceedings of the 2018 ACM International Conference on Management of Data (SIGMOD)*. ACM, 365–380.
[103] Román Orús. 2019. Tensor Networks for Complex Quantum Systems. *Nature Reviews Physics* 1, 9 (2019), 538–550.
[104] Matteo Paganelli, Paolo Sottovia, Kwanghyun Park, Matteo Interlandi, and Francesco Guerra. 2023. Pushing ML Predictions into DBMSs. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 35, 10 (2023), 10295–10308.
[105] Alexandru Paler, Lucian Sasu, Adrian-Cătălin Florea, and Răzvan Andonie. 2023. Machine Learning Optimization of Quantum Circuit Layouts. *ACM Transactions on Quantum Computing* 4, 2 (2023), 1–25.
[106] Feng Pan and Pan Zhang. 2021. Simulating the Sycamore Quantum Supremacy Circuits. https://arxiv.org/abs/2103.03074
[107] Kwanghyun Park, Karla Saur, Dalitso Banda, Rathijit Sen, Matteo Interlandi, and Konstantinos Karanasos. 2022. End-to-End Optimization of Machine Learning Prediction Queries. In *Proceedings of the 2022 ACM International Conference on Management of Data (SIGMOD)*. 587–601.
[108] David Pérez-Garcia, Frank Verstraete, Michael M. Wolf, and J. Ignacio Cirac. 2007. Matrix Product State Representations. *Quantum Information & Computation* 7, 5 (2007), 401–430.
[109] Stefano Pirandola, Ulrik L Andersen, Leonardo Banchi, Mario Berta, Darius Bunandar, Roger Colbeck, Dirk Englund, Tobias Gehring, Cosmo Lupo, Carlo Ottaviani, et al. 2020. Advances in Quantum Cryptography. *Advances in Optics and Photonics* 12, 4 (2020), 1012–1236.
[110] John Preskill. 1999. Lecture Notes for Physics 219: Quantum Computation. *Caltech Lecture Notes* 7 (1999), 1.
[111] John Preskill. 2018. Quantum Computing in the NISQ Era and Beyond. *Quantum* 2 (Aug. 2018), 79.
[112] Haoyu Qi, Daniel J. Brod, Nicolás Quesada, and Raúl García-Patrón. 2020. Regimes of Classical Simulability for Noisy Gaussian Boson Sampling. *Physical Review Letters* 124 (2020), 100502. Issue 10.
[113] Arend-Jan Quist, Jingyi Mei, Tim Coopmans, and Alfons Laarman. 2024. Advancing Quantum Computing with Formal Methods. In *International Symposium on Formal Methods (FM)*. Springer, 420–446.
[114] Jonathan Ragan-Kelley, Connelly Barnes, Andrew Adams, Sylvain Paris, Frédo Durand, and Saman Amarasinghe. 2013. Halide: A Language and Compiler for Optimizing Parallelism, Locality, and Recomputation in Image Processing Pipelines. *ACM SIGPLAN Notices* 48, 6 (2013), 519–530.
[115] Maximilian Schleich, Dan Olteanu, and Radu Ciucanu. 2016. Learning Linear Regression Models Over Factorized Joins. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. ACM, 3–18.
[116] Maximilian Schleich, Dan Olteanu, Mahmoud Abo Khamis, Hung Q. Ngo, and XuanLong Nguyen. 2019. A Layered Aggregate Engine for Analytics Workloads. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. ACM, 1642–1659.
[117] Maximilian Schleich, Amir Shaikhha, and Dan Suciu. 2023. Optimizing Tensor Programs on Flexible Storage. *Proceedings of the ACM on Management of Data (PACMMOD)* 1, 1 (2023), 1–27.
[118] Manuel Schönberger. 2022. Applicability of Quantum Computing on Database Query Optimization. In *Proceedings of the 2022 ACM International Conference on Management of Data (SIGMOD)*. 2512–2514.
[119] Manuel Schönberger, Stefanie Scherzinger, and Wolfgang Mauerer. 2023. Ready to Leap (by Co-Design)? Join Order Optimisation on Quantum Hardware. *Proceedings of the ACM on Management of Data (PACMMOD)* 1, 1 (2023), 1–27.
[120] Manuel Schönberger, Immanuel Trummer, and Wolfgang Mauerer. 2023. Quantum-inspired Digital Annealing for Join Ordering. *Proceedings of the VLDB Endowment (PVLDB)* 17, 3 (2023), 511–524.
[121] Manuel Schönberger, Immanuel Trummer, and Wolfgang Mauerer. 2023. Quantum Optimisation of General Join Trees. In *Joint Proceedings of Workshops at the 49th International Conference on Very Large Data Bases (VLDB) (CEUR Workshop Proceedings)*, Vol. 3462.
[122] Peter W Shor. 1995. Scheme for Reducing Decoherence in Quantum Computer Memory. *Physical Review A* 52, 4 (1995), R2493.
[123] Matthew Skrzypczyk and Stephanie Wehner. 2021. An Architecture for Meeting Quality-of-Service Requirements in Multi-User Quantum Networks. https://arxiv.org/abs/2111.13124
[124] Mikhail Smelyanskiy, Nicolas P. D. Sawaya, and Alán Aspuru-Guzik. [n.d.]. qHiPSTER: The Quantum High Performance Software Testing Environment. https://arxiv.org/abs/1601.07195
[125] Robert S Smith, Eric C Peterson, Mark G Skilbeck, and Erik J Davis. 2020. An Open-Source, Industrial-Strength Optimizing Compiler for Quantum Programs. *Quantum Science and Technology* 5, 4 (2020), 044001.
[126] Redgate Software. 2024. DB-Engines Ranking. https://db-engines.com/en/ranking Accessed: 2025-04-06.
[127] Andrew M. Steane. 1996. Error Correcting Codes in Quantum Theory. *Physical Review Letters* 77 (Jul 1996), 793–797. Issue 5.
[128] Michael Steinbrunn, Guido Moerkotte, and Alfons Kemper. 1997. Heuristic and Randomized Optimization for the Join Ordering Problem. *The VLDB journal* 6 (1997), 191–208.
[129] Michael Stonebraker and Andrew Pavlo. 2024. What Goes Around Comes Around... And Around... *ACM SIGMOD Record* 53, 2 (2024), 21–37.
[130] Christoph Sünderhauf, Earl Campbell, and Joan Camps. 2024. Block-Encoding Structured Matrices for Data Input in Quantum Computing. *Quantum* 8 (Jan. 2024), 1226. https://doi.org/10.22331/q-2024-01-11-1226
[131] Barbara M. Terhal. 2015. Quantum Error Correction for Quantum Memories. *Reviews of Modern Physics* 87 (Apr 2015), 307–346. Issue 2.
[132] 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.
[133] Immanuel Trummer. 2024. Towards Out-of-Core Simulators for Quantum Computing. In *Proceedings of the 1st Workshop on Quantum Computing and Quantum-Inspired Technology for Data-Intensive Systems and Applications (Q-Data '24)*.
[134] Immanuel Trummer and Christoph Koch. 2016. Multiple Query Optimization on the D-Wave 2X Adiabatic Quantum Computer. *Proceedings of the VLDB Endowment (PVLDB)* 9, 9 (2016).
[135] Immanuel Trummer and Christoph Koch. 2017. Solving the Join Ordering Problem via Mixed Integer Linear Programming. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1025–1040.
[136] William G Unruh. 1995. Maintaining Coherence in Quantum Computers. *Physical Review A* 51, 2 (1995), 992.
[137] John van de Wetering. 2020. ZX-Calculus for the Working Quantum Computer Scientist. https://arxiv.org/abs/2012.13966
[138] Maarten Van Den Nes. 2010. Classical Simulation of Quantum Computation, the Gottesman-Knill Theorem, and Slightly Beyond. *Quantum Information & Computation* 10, 3 (2010), 258–271.
[139] Guifré Vidal. 2003. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Physical Review Letters* 91, 14 (2003), 147902.
[140] Benjamin Villalonga, Sergio Boixo, Bron Nelson, Christopher Henze, Eleanor Rieffel, Rupak Biswas, and Salvatore Mandrà. 2019. A Flexible High-Performance Simulator for Verifying and Benchmarking Quantum Circuits Implemented on Real Hardware. *npj Quantum Information* 5, 1 (2019), 86.
[141] Trevor Vincent, Lee J O'Riordan, Mikhail Andrenkov, Jack Brown, Nathan Killoran, Haoyu Qi, and Ish Dhand. 2022. Jet: Fast Quantum Circuit Simulations with Parallel Task-Based Tensor-Network Contraction. *Quantum* 6 (2022), 709.
[142] Lieuwe Vinkhuijzen, Tim Coopmans, David Elkouss, Vedran Dunjko, and Alfons Laarman. 2023. LIMDD: A Decision Diagram for Simulation of Quantum Computing including Stabilizer States. *Quantum* 7 (2023), 1108.
[143] Lieuwe Vinkhuijzen, Tim Coopmans, and Alfons Laarman. 2024. A Knowledge Compilation Map for Quantum Information. https://arxiv.org/abs/2401.01322
[144] Suhas Vittal, Poulami Das, and Moinuddin Qureshi. 2023. Astrea: Accurate Quantum Error-Decoding via Practical Minimum-Weight Perfect-Matching. In *Proceedings of the 50th Annual International Symposium on Computer Architecture* (Orlando, FL, USA) *(ISCA '23)*. Association for Computing Machinery, New York, NY, USA, Article 2, 16 pages. https://doi.org/10.1145/3579371.3589037
[145] Ye Wang, Qing Wang, Henning Koehler, and Yu Lin. 2021. Query-by-Sketch: Scaling Shortest Path Graph Queries on Very Large Networks. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD)*. 1946–1958.
[146] Stephanie Wehner, David Elkouss, and Ronald Hanson. 2018. Quantum Internet: A Vision for the Road Ahead. *Science* 362, 6412 (2018), eaam9288.
[147] Tobias Winker, Umut Çalikyilmaz, 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 '23)*.
[148] 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 *Companion of the 2023 International Conference on Management of Data (SIGMOD)*. 45–52.

[149] Xin-Chuan Wu, Sheng Di, Emma Maitreyee Dasgupta, Franck Cappello, Hal Finkel, Yuri Alexeev, and Frederic T Chong. 2019. Full-State Quantum Circuit Simulation by Using Data Compression. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*. 1–24.

[150] Yue Wu, Namitha Liyanage, and Lin Zhong. 2022. An Interpretation of Union-Find Decoder on Weighted Graphs. https://arxiv.org/abs/2211.03288

[151] Xiaosi Xu, Simon Benjamin, Jinzhao Sun, Xiao Yuan, and Pan Zhang. 2023. A Herculean Task: Classical Simulation of Quantum Computers. https://arxiv.org/abs/2302.08880

[152] Xifeng Yan, Philip S Yu, and Jiawei Han. 2005. Substructure Similarity Search in Graph Databases. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. 766–777.

[153] Kieran Young, Marcus Scese, and Ali Ebnenasir. 2023. Simulating Quantum Computations on Classical Machines: A Survey. https://arxiv.org/abs/2311.16505

[154] Gongsheng Yuan, Yuxing Chen, Jiaheng Lu, Sai Wu, Zhiwei Ye, Ling Qian, and Gang Chen. 2024. Quantum Computing for Databases: Overview and Challenges. https://arxiv.org/abs/2405.12511

[155] Gongsheng Yuan, Jiaheng Lu, Yuxing Chen, Sai Wu, Chang Yao, Zhengtong Yan, Tuodu Li, and Gang Chen. 2023. Quantum Computing for Databases: A Short Survey and Vision. In *Joint Proceedings of Workshops at the 49th International Conference on Very Large Data Bases (VLDB) (CEUR Workshop Proceedings)*, Vol. 3462.

[156] Yexiong Zeng, Zheng-Yang Zhou, Enrico Rinaldi, Clemens Gneiting, and Franco Nori. 2023. Approximate Autonomous Quantum Error Correction with Reinforcement Learning. *Physical Review Letters* 131 (Jul 2023), 050601. Issue 5. https://doi.org/10.1103/PhysRevLett.131.050601

[157] Xuanhe Zhou, Chengliang Chai, Guoliang Li, and Ji Sun. 2020. Database Meets Artificial Intelligence: A Survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* (2020).

[158] Yiqing Zhou, E Miles Stoudenmire, and Xavier Waintal. 2020. What Limits the Simulation of Quantum Computers? *Physical Review X* 10, 4 (2020), 041038.

[159] Alwin Zulehner, Alexandru Paler, and Robert Wille. 2018. An Efficient Methodology for Mapping Quantum Circuits to the IBM QX Architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* 38, 7 (2018), 1226–1236.

[160] Alwin Zulehner and Robert Wille. 2018. Advanced Simulation of Quantum Computations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* 38, 5 (2018), 848–859.