Model-Driven Engineering for Quantum Programming: A Case Study on Ground State Energy Calculation

Furkan Polat

Electrical and Electronics Engineering,

Koç University, & Department of Computer Science,

University of Antwerp, Antwerp, Belgium

fpolat17@ku.edu.tr

Hasan Tuncer

Department of Physics and Astronomy,

Purdue University, IN, USA

htuncer@purdue.edu

Armin Moin

Department of Computer Science,

University of Colorado,

Colorado Springs, CO, USA

amoin@uccs.edu

Moharram Challenger

Department of Computer Science,

University of Antwerp,

& Flanders Make Strategic Research Center, Antwerp, Belgium

moharram.challenger@uantwerpen.be

Abstract:

This study introduces a novel framework that brings together two main Quantum Programming methodologies, gate-based Quantum Computing and Quantum Annealing, by applying the Model-Driven Engineering principles. This aims to enhance the adaptability, design, and scalability of quantum programs, facilitating their design and operation across diverse computing platforms. A notable achievement of this research is the development of a mapping method for programs between gate-based quantum computers and quantum annealers, which can lead to the automatic transformation of these programs. Specifically, this method is applied to the Variational Quantum Eigensolver Algorithm and Quantum Annealing Ising Model, targeting ground state solutions. Finding ground-state solutions is crucial for a wide range of scientific applications, ranging from simulating chemistry lab experiments to medical applications, such as vaccine development. The success of this application demonstrates Model-Driven Engineering for Quantum Programming frameworks' practical viability and sets a clear path for Quantum Computing's broader use in solving intricate problems.

Keywords: Model-Driven Engineering, Quantum Computing Programming, Model Transformation, Variational Quantum Eigensolver

I. INTRODUCTION

Various models of computation exist for Quantum Computing (QC), such as quantum circuit models (i.e., Quantum gates) and adiabatic (i.e., quantum annealing), which are well-established ones[1]. However, it is already theoretically proven that any quantum circuit algorithm can be transformed into a quantum annealing algorithm with the same time complexity and vice-versa, hence showing they are polynomial equivalent [2].

Furthermore, the key technologies for the actual physical implementation of quantum processors comprise superconducting, ultra-cold atoms (e.g., trapped ions), quantum optics, and spin-based. For instance, Linke et al. [3] conducted experiments on two 5-qubit quantum computers with the quantum circuit model, which had two different technologies and architectures, namely superconducting and trapped ions. They illustrated that the superconducting system offered faster gate clock speeds and a solid-state platform, whereas the other one featured superior qubits and reconfigurable connections. They concluded that the performance of those systems reflected the topology of connections in the base hardware, thus supporting the idea that quantum software and hardware should ideally be co-designed [4].

The Model-Driven Engineering (MDE) paradigm has already been proposed as a reasonable choice to cope with the heterogeneity and complexity observed in the QC domain and its programming [4–6].

In this paper, the vision of applying MDE techniques for Quantum Programming (QP)[4, 7] is extended by proposing a new framework and applying it in the context of the ground state energy calculation problem, which has concrete use cases in various domains. This includes material science for developing new materials [8], chemistry for creating efficient catalysts [9], pharmaceuticals for advancing drug and vaccine development [10], and physics for exploring phenomena like superconductivity and quantum magnetism [11].

This paper is structured as follows: Section II provides a review of related work and presents an overview of QC, the MDE paradigm, and the ground-state energy calculation problem. In Section III, it is introduced that MDE framework for quantum programs, focusing on the strategy for developing Platform Independent Model for Ground State Calculation. Section IV outlines our proof of concept work for Model-Driven Engineering for Quantum Programs (MDE4QP). After implementing a ground state solution separately in

gate-based QPUs and Quantum Annealers, it demonstrates the results from abstract mapping. Finally, the paper concludes with a summary of our contributions and future research directions in Section V.

II. RELATED WORK AND BACKGROUND

Previous research in QC has explored various models of computation and technologies for physical implementation. Aharonov et al. [2] theoretically demonstrated the equivalence between quantum circuit algorithms and quantum annealing algorithms. This foundational work laid the groundwork for understanding the relationship between different QC models.

In terms of physical implementations, Linke et al. [3] conducted experiments using two different QC technologies: superconducting qubits and trapped ions. They highlighted the difference in performance and architecture between these systems, emphasizing the importance of co-designing quantum software and hardware.

The MDE paradigm has emerged as a promising approach for managing the complexity and heterogeneity in quantum programming [4–6]. Moin et al. [4] proposed using MDE techniques for quantum programming, laying the foundation for our work.

A. Quantum Computing

QC is a transformative approach to computing that integrates quantum mechanics principles into information processing This field's theoretical foundation was pioneered by Richard Feynman, who proposed the concept of quantum simulation in 1982. Feynman suggested that quantum systems could simulate physical phenomena more efficiently than classical computers, addressing complex quantum mechanical problems [12].

Significant progress in QC was marked by the development of Shor's algorithm. This algorithm, aimed at prime factorization, runs much faster on quantum computers than the fastest algorithms on classical computers, highlighting the advanced computational capabilities of quantum systems for certain tasks [13].

Following these early steps, quantum computing (QC) has undergone significant advancements, demonstrating its capability for speed improvements in various applications, a phenomenon known as quantum speedup. For instance, quantum algorithms can search an unsorted database with N entries in time proportional to $\sqrt{N}$, as shown by Grover [14], unlike classical computers, which take time proportional to N. Moreover, quantum computers can perform Fourier transforms, invert sparse matrices, and find their eigenvalues and eigenvectors in time that scales polynomially in $\log_2 N$, as demonstrated by Biamonte et al.

[15]. This contrasts with classical algorithms that scale as $N \log_2 N$.

The Sycamore processor by Google demonstrated the quantum advantage by performing a specific computational task exponentially faster than the current state-of-the-art classical supercomputers, highlighting the potential of quantum processors to achieve significant speed-ups for certain tasks [16].

The impact of QC spans various domains, from cryptography, where it could break current encryption schemes [17], to optimization challenges in logistics and manufacturing [18], and to detailed simulations in chemistry and physics [19]. The potential for QC to accelerate drug discovery and materials science highlights its implications across several fields [20].

B. Different Approaches in Quantum Computing

QC has a variety of computational paradigms, each distinguished by unique principles, methodologies, and quantum phenomena exploitation. There are various models of computation for QC, such as quantum circuit/logic gate, adiabatic/annealing, topological, quantum walks, one clean qubit and measurement-based. Since our work focuses on quantum circuit gate model and Annealing model mapping, this subsection provides an overview of the two primary approaches to them.

1. Quantum Gate-based Model

In this model, quantum operations are represented as sequences of quantum gates, each performing a specific transformation on the qubits. These gates include single-qubit gates, which act on individual qubits, and multi-qubit gates, which entangle multiple qubits. By applying sequences of these gates to the qubits in a quantum circuit, complex quantum computations can be performed. Notably, the gate-based model forms the foundation for various quantum algorithms, including Shor's algorithm [13] for prime factorization, Grover's algorithm for database searching [14] and Variational Quantum Algorithms (VQA) for optimization [21], eigenvalue estimation [22] and machine learning [23].

2. Quantum Annealing

This model originates from the adiabatic theorem to solve optimization problems by evolving an initial quantum state towards the ground state, which represents the problem's optimal solution. This evolution represents the problem's solution without manipulating the qubits through direct operators, marking a difference from the step-by-step operations seen in gate-based computing. AQC's method is inherently suited for various optimization challenges, encoding solutions in the system's lowest energy state [24]. Quantum Annealers efficiently addresses optimization problems such as logistical planning, financial portfolio optimization, and complex scheduling. Quantum annealing can often reach satisfactory solutions faster than classical computational methods by enabling the system to escape local minima and explore more of the solution landscape. This makes it a valuable tool in scenarios where speed is prioritized over absolute precision in finding the lowest possible energy state [25, 26].

C. Challenges in Quantum Computing

One of the main challenges in quantum computing is maintaining the coherence of quantum states. Qubits are extremely sensitive to their environment. Even small temperature fluctuations, electromagnetic waves, or other interactions with the environment can cause qubits to lose their quantum properties (decoherence) and the information they contain to become corrupted. Quantum noise can also interfere with the operation of qubits, causing errors in calculations [27].

Therefore, robust error correction methods are very important. Quantum error correction is more complex than classical error correction and requires significant additional qubits and computational resources to encode information in such a way that errors can be detected and corrected without disrupting the quantum state [28].

Scalability is another important challenge. Building a quantum computer with enough qubits to solve practical problems is a major challenge. Increasing the number of qubits not only requires maintaining consistency across a larger system but also exponentially increases the complexity of error correction. The physical systems used to realize qubits (superconducting circuits, trapped ions, etc.) currently face significant engineering and

manufacturing challenges to scale up while maintaining fidelity [29, 30].

Quantum computers will need to work together with classical computers to perform practical tasks. This integration poses challenges in terms of creating efficient interfaces and communication protocols between quantum and classical systems [31].

Additionally, since quantum platforms take different approaches, their efficiency on different problems may differ. In this case, it will be important to develop a system that can run these platforms together. Our aim is to eliminate heterogeneity through Model-Driven Engineering (MDE).

Quantum computers will need to work together with classical computers to perform practical tasks. This integration poses challenges in terms of creating efficient interfaces and communication protocols between quantum and classical systems [31].

D. Model-Driven Engineering (MDE)

MDE is a software development approach that prioritizes creating and using DSMLs to represent the core functionality of a system. DSMLs capture the desired behaviour and properties at a higher level of abstraction instead of focusing on the final code. This focuses on the problem domain, rather than coding specifics, which leads to better communication and understanding among involved developers[32]. MDE offers several benefits, including enabling early validation and verification through techniques like model analysis and simulation, ultimately leading to a more robust and efficient software solution. Additionally, MDE can enhance development productivity by automating tasks such as code generation [33].

E. Ground State Calculation

The Ground State Calculation Problem is fundamental to understanding the physical and chemical properties of systems at the quantum level. Calculating the ground state of quantum systems enables scientists to predict behaviours and interactions, crucial for fields such as material science and drug discovery.

One area where ground-state energy calculations play a pivotal role is in the development of new pharmaceuticals. Quantum chemistry simulations, facilitated by Variational Quan-

tum Algorithms, are essential for discovering Hamiltonian spectra [22]. These algorithms are key to analyzing reaction rates and understanding catalytic processes, which are vital for creating new drugs. This method allows for the analysis of molecular energies, critical for predicting chemical reactions and interactions, and showcases the potential of near-term quantum hardware to enhance calculations in chemistry and drug development.

1. Classical Algorithms for Ground State Calculation

Classical algorithms for calculating the ground states of the quantum states are essential in understanding how atoms and molecules behave. These methods help scientists find the lowest energy state of quantum systems, which is important for predicting chemical reactions and material properties. Key techniques include the Variational Principle, Hartree-Fock Method [34], Monte Carlo Simulations [35]. Each traditional method has its own way of approaching the problem, but all aim to provide accurate predictions about the quantum.

a. Density Matrix Renormalization Group (DMRG) The development of the Density Matrix Renormalization Group (DMRG) algorithm marked a significant advancement in computational physics, addressing some of the limitations faced by classical algorithms. DMRG, primarily applied to one-dimensional quantum systems, optimizes the calculation of low-energy eigenstates by iteratively refining the system's state space, focusing on the most relevant degrees of freedom associated with entanglement. This process is significantly more efficient than traditional methods, offering a glimpse into the potential of specialized algorithms in overcoming computational challenges [36].

b. Tensor Networks Building upon the success of DMRG, tensor networks extend the concept of efficient quantum state representation to higher dimensions. These networks express the quantum state as a product of interconnected tensors, simplifying the representation of complex entanglements and correlations within the system:

$$\Psi_{i_1,i_2,\ldots,i_N} = \sum_{\alpha_1,\alpha_2,\ldots,\alpha_N} A^{[1]}_{i_1;\alpha_1} A^{[2]}_{i_2;\alpha_1\alpha_2} \ldots A^{[N-1]}_{i_{N-1};\alpha_{N-2}\alpha_{N-1}} A^{[N]}_{i_N;\alpha_N\alpha_{N-1}} \tag{1}$$

where $i_k;\alpha_k$ represents indices corresponding to local properties, and the bond indices labeled with α_k can encode correlations such as entanglement. The decomposition is a special case known as the matrix product state (MPS). The α_k indices' maximum number of values, known as the bond size, serves as a quantitative measure of entanglement.

c. Artificial Neural Networks in Quantum Physics The intersection of quantum physics and computational science has further expanded with the application of artificial neural networks (ANNs) to model quantum systems. ANNs, particularly through structures like the Restricted Boltzmann Machine (RBM), have shown promise in capturing the complex multi-particle wave functions of quantum states [37, 38]. The RBM employs a layered approach, where connections between visible and hidden units encode the quantum system's properties:

$$\psi_{\mathrm{RBM}}(\mathbf{j}) = \exp\left(\sum_{i=1}^{L} a_i j_i\right) \prod_{i=1}^{M} \cosh\left(b_i + \sum_{k} W_{ik} j_k\right), \tag{2}$$

This method,tensor networks, provides a useful tool for simulating quantum systems, Through the adaptation of ANNs, researchers can model systems with a level of efficiency and accuracy that opens new avenues for exploration and understanding in quantum physics [39–41].

The progression from classical algorithms through artificial neural networks, reflects the dynamic and interdisciplinary nature of computational research in quantum physics. Notably, NetKet emerges as a pivotal framework in this domain, offering a library for studying many-body quantum systems with machine learning approaches, highlighting the potential of neural network quantum states for quantum state tomography, supervised learning, and ground state searches [37]. Further, the explicit demonstration of initial state construction in ANNs using NetKet and validation with IBM Q Experience Platform underscore the practical applicability and accuracy of these models in simulating quantum computations [38]. With the release of NetKet 3, built on JAX, the toolbox for many-body quantum systems now supports advanced neural network ansätze, GPU/TPU accelerators, and discrete symmetry groups, marking significant advancements in the field [39].

Those classical methods encounter challenges as the system's complexity escalates with the addition of more particles. Real-world problems often involve large systems with many interacting particles, and the intricate interactions within such systems result in an exponential growth in computational workload. This presents a significant challenge that classical computers, with their inherent limitations, struggle to manage effectively.

Quantum computers, on the other hand, offer an advantage, and Qubits play a key role there. Adding a qubit to the system doubles the computational capacity. So, with n qubits,

the computational possibilities expand to 2^n. This scalability provides quantum computers with an edge, offering a promising solution for handling the complexities of calculating the ground states of the molecules [42, 43].

d. Quantum Phase Estimation Recent advancements in quantum computing have introduced new methods for addressing eigenvalue problems that are challenging for classical computational approaches. A key development in these quantum solutions is the Quantum Phase Estimation (QPE) algorithm, notable for its efficiency advantage over conventional computational methods. The QPE algorithm necessitates $O(p^{-1})$ quantum operations to achieve a specified level of precision, p, in estimating the eigenvalue linked to a known eigenvector of a Hermitian operator H.

The precision p highlights the accuracy in the eigenvalue estimation process, with the algorithm assuming initial access to the eigenvector in question. The aim is to uncover the associated eigenvalue. The quantum operations required for this process, directly related to $O(p^{-1})$, mandate sequential execution of the operation e^{-iHt}. In practical applications, this could require to use of millions or even billions of quantum gates—far exceeding the operational capacity of current quantum technologies, which are predominantly capable of managing only tens to hundreds of gates [10, 44, 45].

e. Variational Quantum Eigensolver (VQE) In hybrid VQE solvers, the crucial quantum component involves the utilization of Ising model solvers. Quantum computers work on the Ising model solution based on a given ansatz as illustrated in Fig. 1. The inputs include a cost function $C(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents parameters encoding the solution, an ansatz whose parameters are trained to minimize the cost, and potentially a set of training data $\{\rho_k\}$ used in optimization. Typically, the cost function can be expressed as a set of functions $\{f_k\}$.

The ansatz is depicted as a parameterized quantum circuit on the left, next to a neural network shown on the right. During each iteration, a quantum computer efficiently estimates the cost (or its gradients), with this information fed into a classical computer utilizing optimizers to navigate the cost landscape $\boldsymbol{\theta}$ and solve the optimization problem. Upon meeting termination conditions, the VQE provides an estimate of the solution, with the form of the output dependent on the specific task. The red-shaded box highlights common types of outputs. is a hybrid algorithm designed to estimate the ground state energy of a molecule by blending variational techniques with quantum mechanics. It operates iteratively, fine-tuning wave function guesses to minimize energy until it reaches the system's most stable

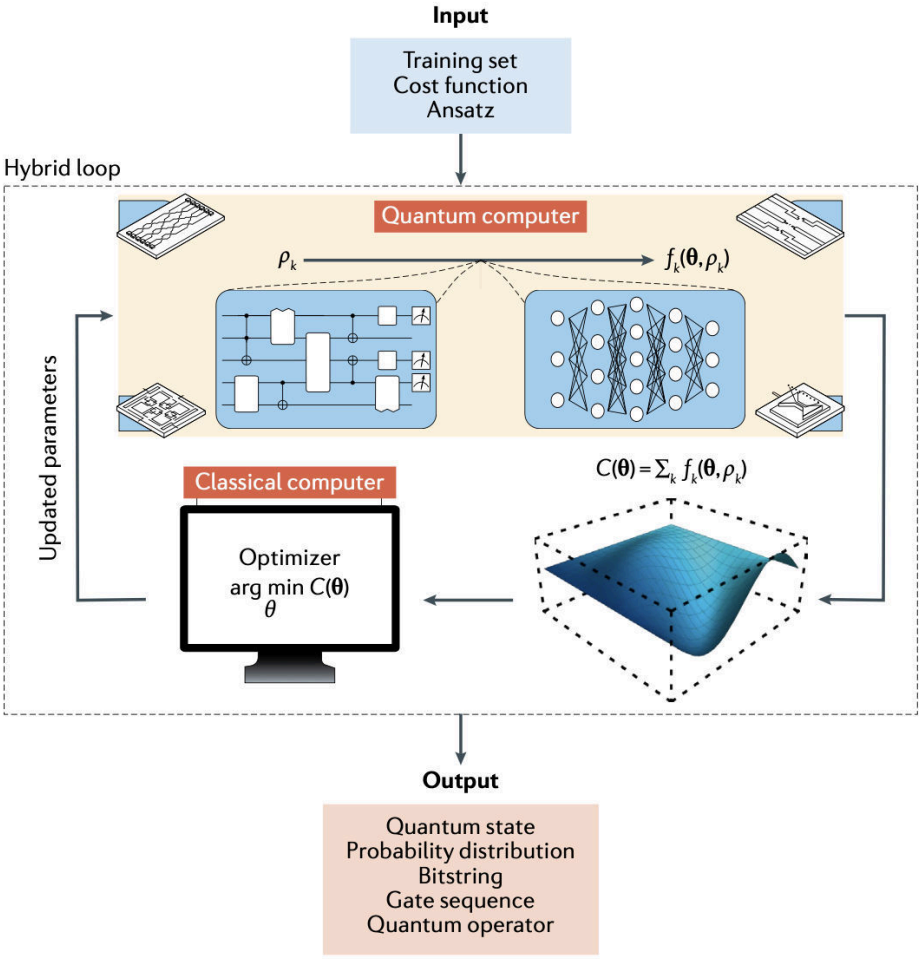


FIG. 1: The schematic diagram illustrates a variational quantum algorithm (VQA). [46]

state. This process involves both quantum and classical components: the quantum part computes the system's energy for a given Hamiltonian, while the classical part optimizes variational parameters [22, 46].

The fundamental equation at the heart of VQE for ground sate calculation is given by:

$$H\left|\psi\right\rangle = E\left|\psi\right\rangle, \tag{3}$$

where H represents the Hamiltonian acting on the wave function ψ, and E denotes the associated energy.

The Hamiltonian $H = \hat{T} + \hat{V}$ encompasses kinetic and potential energies, incorporating terms that describe electron-electron, nucleus-nucleus, and electron-nucleus interactions for electrons and nuclei. The representation of the wave function evolves from classical Slater determinants to second quantization. This methods expresses the states using fermionic

creation and annihilation operators. This formalism facilitates the expression of the Hamiltonian for computation on quantum computers [47].

VQE employs techniques like the Jordan-Wigner transformation to map fermionic operators to qubit operators that enables encoding of molecular systems into qubits [48]. The goal is to minimize the expectation value of the Hamiltonian:

$$E_\theta = \min_\theta \frac{\langle \Psi(\theta)|\hat{H}|\Psi(\theta)\rangle}{\langle \Psi(\theta)|\Psi(\theta)\rangle}, \tag{4}$$

where θ represents the parameters iteratively optimized through quantum gates. This iterative process aims to closely approximate the ground state of the molecular system, leveraging the precision and efficiency of quantum computation for energy calculations

f. Quantum Annealing is also employed for ground state calculations, as it uses the principles of adiabatic quantum mechanics to solve optimization problems. It progresses from an initial Hamiltonian, representing the problem at hand, to a final Hamiltonian that corresponds to the ground state solution.

g. Ising Model describes interacting spins in a system, where each spin interacts with its neighboring spins on a lattice grid. The interactions are defined by coupling constants, and there may also be local magnetic fields acting on each spin.[49] In VQE, the Ising model is used to map the terms of the molecular Hamiltonian onto a set of qubits, enabling quantum computers to simulate molecular systems and calculate their properties efficiently.[46]

Additionally, the Ising model finds widespread application in other contexts, such as in quantum annealers, where optimization problems are initially formulated as Ising models. The ground energy of these Ising models represents the optimal solution for various problems, including the traveling salesman problem and financial optimization.

III. PROPOSED APPROACH

A well-established MDE could provide a robust framework for developing quantum algorithms because it has shown its ability for automated code generation, model checking, and analysis in other domains. This paper presents a proof of concept for applying MDE to a quantum application to its hybrid/quantum equivalent or vice versa. As a first step, the paper is particularly focusing on ground-state calculations.

This paper proposes a new method that combines MDE with QC to tackle the complexities of systems with multiple particles. This approach aims to improve upon traditional methods and provide a new perspective for conducting ground-state calculations, where the complexity of systems involving multiple particles often makes traditional computational approaches insufficient[50]. With the MDE paradigms that resolve heterogeneity in both classical and QC, this research contributes to advancing QC applications for solving practical problems [4–6, 51].

The future of Classical and Quantum Computing will likely involve a mix of processors with different architectures and technologies. This will make software development more complex and require more abstraction. Modeling languages and Domain-Specific Modeling Languages (DSMLs) will be essential for providing this abstraction. These methods allow developers to model their applications without concerning about the underlying hardware or software. MDE and DSMLs will also be important for analyzing and verifying these applications, as the platforms will be complex and distributed. This will require formalizing the semantics of QC and the specific target platforms for QC. In other words, the knowledge of QC experts needs to be transferred to the modeling languages and tools.

MDE4QP builds upon the existing MDE tools in classical platforms. Developers can focus on their application's core functionality without getting involved with the intricacies of the underlying hardware or software. MDE4QP aspires to bring this same level of abstraction to QC platforms, particularly applications involving Quantum Machine Learning (QML).

Similar to how Domain- Specific Modeling (DSM) methodology offers a layer of abstraction, MDE4QP promises to do the same for Quantum program development. This approach is illustrated in Fig. 2. for the proof of concept case in ground state calculation.

The proposed approach prioritizes flexibility and adaptability for hardware platforms. Developers might not know the specific hardware (CPUs, GPUs). Still application will run

at runtime once they use DSMLs. MDE4QP aspires to extend this flexibility to the world of QC by covering various Quantum Processor Units (QPUs) with their diverse architectures and capabilities.

Additionally providing abstraction, MDE4QP aims to streamline the development process further through automated code generation. This will be achieved by implementing model-to-code transformations, essentially translating the high-level models created by developers into code that can run on the target platform. But the potential benefits go beyond mere code generation. The paper explores the intriguing possibility of using model-to-model transformations to achieve various functionalities at the modeling level. These functionalities includes the ability to:

- Convert between classical and quantum applications: This feature enables the seamless transition of applications between classical and QC environments, providing flexibility in the choice of computational model.

- Translate between different models of quantum computation: Such as quantum circuits and quantum annealing. This capability is essential for optimizing applications for specific problem types and uses the strengths of different QC approaches.

- Fine-tune applications for specific QPU architectures: This feature allows developers to optimize of applications to maximize performance and efficiency on a particular QPU with its unique characteristics. Developers can design applications with tailored optimizations for the target QPU, enhancing overall computational performance.

Also, MDE4QP recognizes the importance of early-stage analysis and verification. The approach aims to enable these activities at the modeling level, allowing developers to catch errors and optimize performance before the actual code is generated. This can significantly reduce development time and effort while ensuring the robustness and efficiency of the final application.

The first step towards the MDE4QP paradigm focuses on establishing a unified abstract mapping to bridge the gap between theoretical quantum algorithms and their practical implementation across various platforms. The work for this framework begins with the conceptual mapping of quantum computations for ground-state calculations. This initial

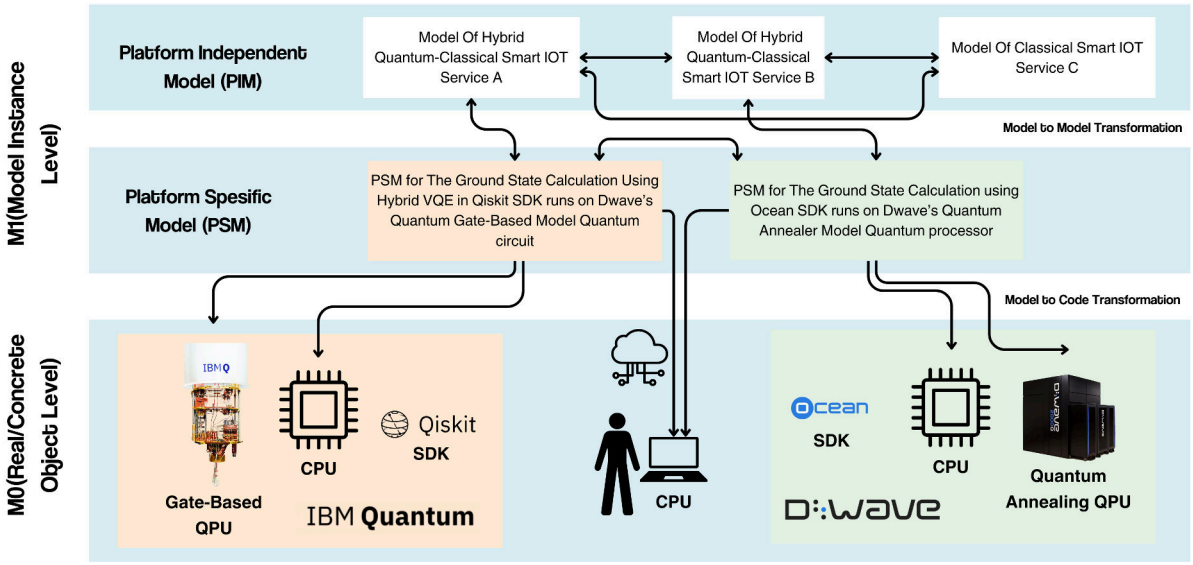


FIG. 2: The Proposed Vision of MDE4QP in the context of Model Driven Architecture for the ground state calculations in different quantum platforms

step lays the groundwork for designing and executing algorithms that can seamlessly operate across different platforms.

There are two paths for calculating he ground states solution of an Ising model: Adiabatic and VQE, as illustrated in 3. And Quantum Annealers uses the adiabatic path to evolve the Ising model.On the other side, VQE path includes the gate-based QPU processing of the model. The core objective of the methodology is to implement a Platform Independent Model (PIM) of the Ising model solution, which includes quantum gate-based models and quantum annealer models. By conceptualizing the solutions to the Ising model in a platform-agnostic manner, the flawless transition of quantum algorithms from one computational model to another is enabled

This work explores the relationship between quantum annealing ising model and Quantum Gate-Based VQE through the framework of abstract mapping. This analysis reveals the potential for cross-platform compatibility between different QC paradigms. Such compatibility facilitates the emergence of a new generation of versatile quantum applications. These applications would bridge the gap between diverse hardware platforms and enable the development of adaptive solutions.

a. Model-to-model Transformation The next step in this paradigm shift involves the design of domain-specific models (DSMs) that facilitate seamless transfer between diverse quantum and classical computing platforms. A system for model-to-model (M2M) transformation across various computation models is envisioned. This system would leverage PIMs (as illustrated in Fig. 2) to translate algorithms for execution on specific quantum hardware.

The ultimate goal lies in establishing a comprehensive framework that can dynamically adapt to the operational paradigms of both quantum annealing and gate-based architectures. This framework should ensure efficient algorithm translation and execution across platforms, minimizing the need for extensive re-engineering.

b. Model-to-code Transformation (Automated Code Generation) Related to designing Model-to-Model Transformation, Automated Code Generation through Model-to-code transformation is essential. This approach, as shown in Fig. 2, would enable the automatic generation of code from platform-specific models into desired application programming interfaces (APIs).

Automated code generation from high-level models into executable quantum code would significantly accelerate the development of quantum algorithms. This would, in turn, enhance the accessibility of QC for researchers and practitioners who might lack in-depth expertise in quantum programming languages.

To achieve this goal, the development of specialized tools and compilers is necessary. These tools would be responsible for interpreting high-level algorithmic descriptions and generating optimized quantum code tailored for various platforms. These tools will require a level of intelligence to navigate the complexities of quantum computation. This includes tasks like qubit allocation, gate selection, and the incorporation of error correction strategies. Ultimately, the aim is to produce efficient and reliable quantum programs.

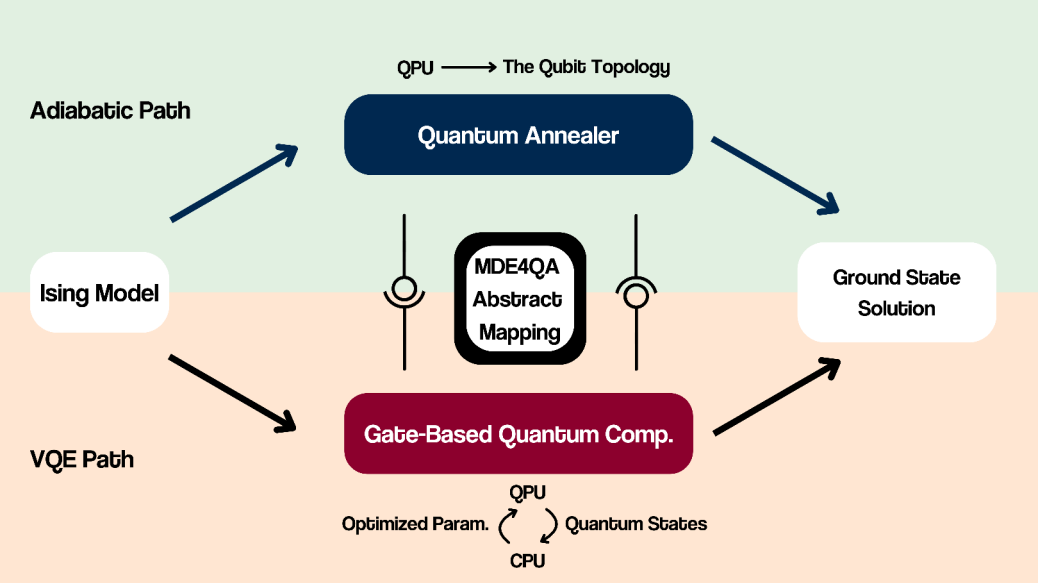


FIG. 3: The schematic diagram illustrates the abstract mapping of solutions from different platforms to ground state calculation. The upper green side represents the Quantum Annealing aproach, while the lower orange side depicts the Hybrid VQE approach in a Quantum Gate-based model. This conceptually simplifies the two methods for solving the same ground state calculation problem."

IV. CASE STUDY: GROUND STATE ENERGY CALCULATION USING GATE-BASED QC AND QUANTUM ANNEALER

This research explores a new method for applying the Ising model to both gate-based quantum computers and quantum annealers. This method uses a special mapping process to translate quantum algorithms between these two platforms. This allows researchers to easily switch between different types of quantum computers to solve problems, which can lead to faster discoveries and better solutions.

This method is tested by applying it to three different versions of the Ising model on both gate-based and quantum annealing machines. The goal is to find the lowest possible energy state (ground state) for each version of the model. Here's the Ising model's general Hamiltonian, which describes its energy:

$$H = \sum_{i,j} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z \tag{5}$$

where $J_{ij} = 1$ denotes uniform interaction strength between connected qubits i and j,

$h_i = 1$ represents a uniform external magnetic field applied to each qubit, and σ_i^z is the Pauli-Z operator acting on qubit i. The choice of $J_{ij} = 1$ is often made for simplicity and normalization purposes. It allows easier comparison and analysis of different systems. Additionally, this choice simplifies mathematical expressions and calculations while still capturing the essential behavior of the system.

The specific entanglement patterns are described as follows:

- Full-Entangled Ising Model: The model contains interactions among all pairs of qubits, which can be mathematically described as:

$$\sum_{i<j} \sigma_i^z \sigma_j^z \tag{6}$$

  This notation sums over all unique pairs i, j, indicating all-to-all connectivity.

- Linear-Entangled Ising Model: This model features a sequential chain of interactions, which can be expressed as:

$$\sum_{i=0}^{N-2} \sigma_i^z \sigma_{i+1}^z \tag{7}$$

  where N is the total number of qubits, illustrating a linear sequence of connections from qubit 0 to $N-1$.

- Circular-Entangled Ising Model: To represent the circular connectivity where the last qubit is also connected back to the first:

$$\sum_{i=0}^{N-1} \sigma_i^z \sigma_{(i+1) \mod N}^z \tag{8}$$

  Here, $\mod N$ ensures that after the last qubit, the summation loops back to the first qubit, forming a circular entanglement pattern.

For those interested in further exploration or contributions, the project's code is available on GitHub. [52].

A. VQE with Different Ising Models using Gate-Based Quantum Computing

The First step of the investigation highlights the VQE approach to tackle the Ising model with gate-based QC. VQE, a hybrid quantum-classical algorithm, uses the quantum mechanical strengths of superposition and entanglement, combined with classical optimization routines, to efficiently approximate the ground state of quantum systems.

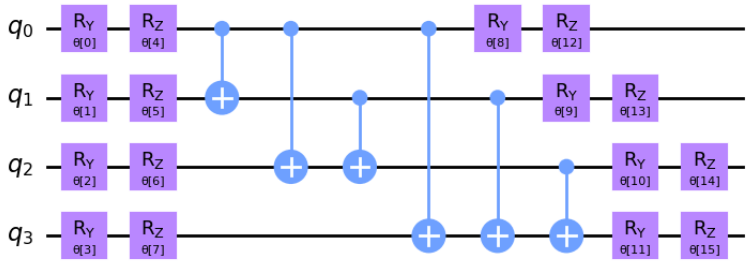
The analysis uses a 1-D Ising model, articulated through Pauli spin matrices, to explore the interplay between quantum spins under the influence of an external magnetic field (λ), set to a constant value of 1 for this study.

This Hamiltonian in 5 serves as the objective function for the VQE algorithm, guiding the search for the system's ground state.

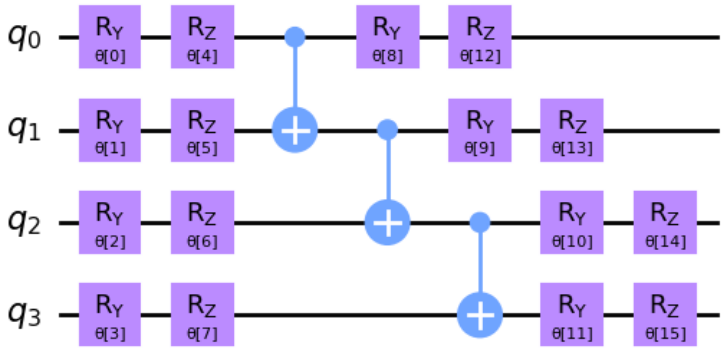
A 4-qubit system is used to explore various entanglement patterns within the ansatz designs: full-entangled (Fig. 4a), linear-entangled (Fig. 4b), and circular-entangled (Fig. 4c). These designs are instrumental in assessing the impact of entanglement on the optimization process and the algorithm's ability to approximate the ground state energy of the corresponding model.These ansatz circuits derived from [53] which were initially used in [54]. The convergence towards the ground state energy is visually tracked through energy plots for each ansatz, revealing the algorithm's effectiveness in approximating the Ising model's ground state.

- The **full-entangled ansatz** reached an energy state of -2.0, with the qubit configuration $|0, 1, 0, 1\rangle$.The iteration steps are shown in figure. 5a.
- The **linear-entangled ansatz** approached the ground state with an energy of -3.0, with the qubit configuration $|0, 1, 0, 1\rangle$.The iteration steps are are shown in figure. 5b.
- The **circular-entangled ansatz** demonstrated significant promise by converging to a ground state energy value of -4.0, with the qubit configuration $|0, 1, 0, 1\rangle$.The iteration steps shown in figure. **??**.

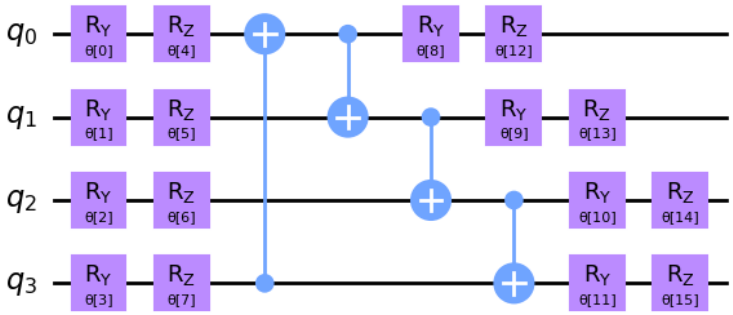
This VQE model investigates the 1-D Transverse Ising model using computational techniques to determine -state energy properties. The investigation via VQE with Qiskit demonstrates convergence to the target ground-state energies through optimization. Across all ansatz types, convergence to the ground-state energies (-2, -3, -4) for corresponding full linear and circular Ising models is observed.



(a) Full-entangled ansatz



(b) Linear-entangled ansatz

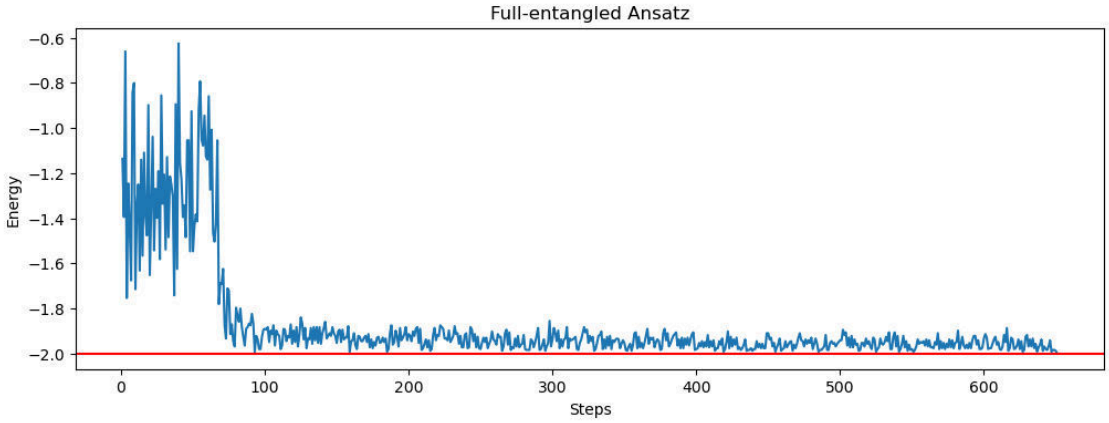


(c) Circular-entangled ansatz

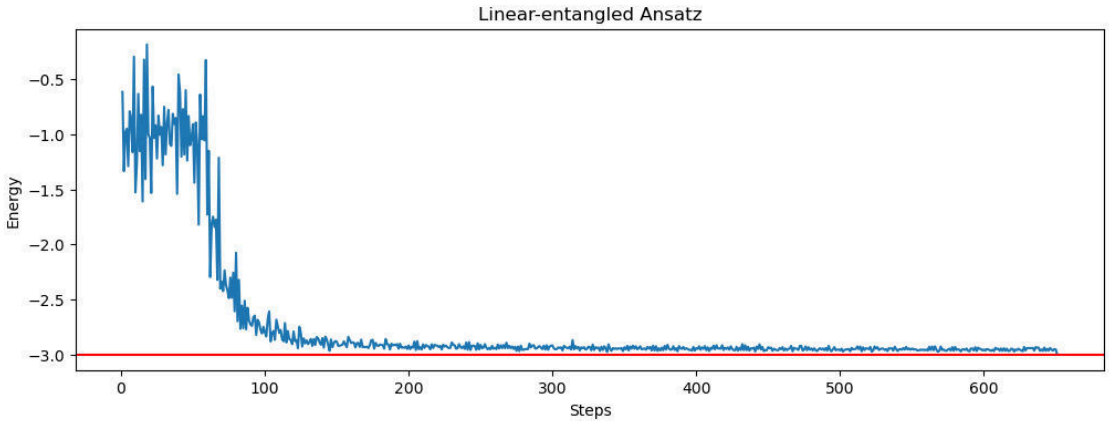
FIG. 4: Circuits of different entangled ansatzes. Each subfigure depicts a unique quantum circuit layout corresponding to its entanglement strategy.

B. Quantum Annealer Ground State Solution for Different Ising Models

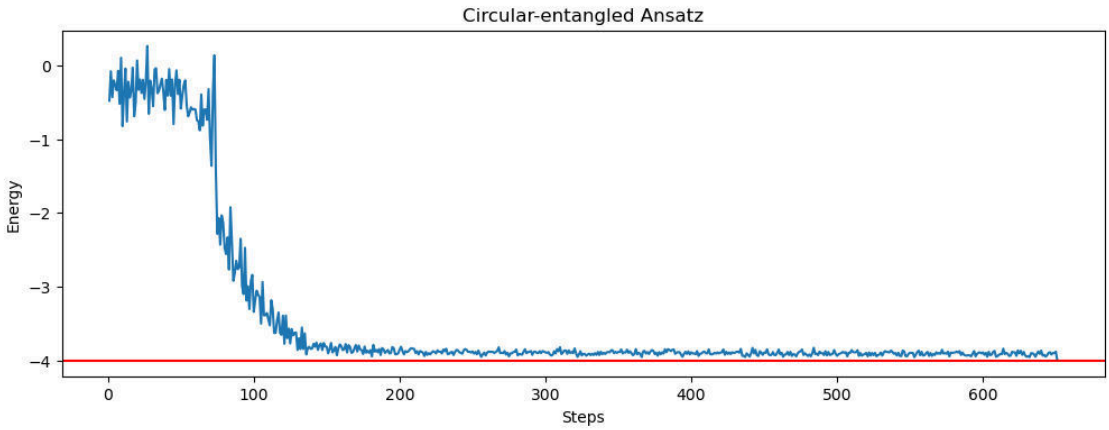
After gate-based QC, this study also investigates the application of quantum annealing to solve the Ising model using the D-Wave system. Quantum annealing uses quantum



(a) Full-entangled ansatz



(b) Linear-entangled ansatz



(c) Circular-entangled ansatz

FIG. 5: Step vs energy plots for different entangled ansatzes. Each subfigure illustrates the energy convergence with the red line indicating the exact ground state energy.

mechanical phenomena such as superposition and quantum tunneling, offering a specialized

and efficient way to find the system's ground state. This process is inherently suited to solving optimization problems, like finding the minimum energy configuration in the Ising model, by gradually transitioning from a superposition of all possible states towards the lowest energy state.

The D-Wave system facilitates this exploration through its Quantum Processing Unit (QPU), designed around a lattice of qubits that can be programmatically entangled and biased to represent the Ising model. The Ising model's Hamiltonian 5, representing the energy landscape of the system, is directly mapped onto the QPU. This mapping involves defining the external magnetic field (h) affecting individual qubits and the interaction strength (J) between qubits as circular, linear, and full-entangled models.

The implementation of each Ising model variants on the D-Wave system has a sequence of critical steps. The process begins with the initialization phase, where the Ising model parameters, specifically the h and J values, are configured to define the problem space accurately. Then, the `EmbeddingComposite` and `DWaveSampler` are employed to sample the Ising model multiple times. This step is essential for exploring the solution space thoroughly to identify low-energy states that approximate the ground state of the system. Furthermore, the following graphs for each Ising model variants— full-entangled (Fig.6c), Linear- entangled (Fig. 6b), and circular-entangled (Fig. 6a)—provide a visual representation of the qubit interactions and their implementation on D-Wave's QPU within the Pegasus topology. Finally, the culmination of this process involves the determination of the ground state by extracting the sample with the lowest energy from the result set, along with its corresponding energy value.

The ground state energy and corresponding qubit configurations for the different Ising models are as follows:

- Full-Entangled Model: Achieved a ground state energy of -2.0 with the qubit configuration $|0, 1, 0, 1\rangle$.
- Linear Model: Achieved a ground state energy of -3.0 with the same qubit configuration $|0, 1, 0, 1\rangle$.
- Circular Model: Achieved a ground state energy of -4.0 with the qubit configuration $|0, 1, 0, 1\rangle$.

These results highlight the D-Wave system's capability to efficiently solve the Ising model, with variations in the entanglement patterns affecting the ground state energy and qubit configurations. The ability to visualize the problem setup and solutions via the D-Wave Inspector helps in understanding the quantum annealing process and the dynamics of the Ising model on a quantum annealer.

C. Abstract Mapping For Platform Independent Model

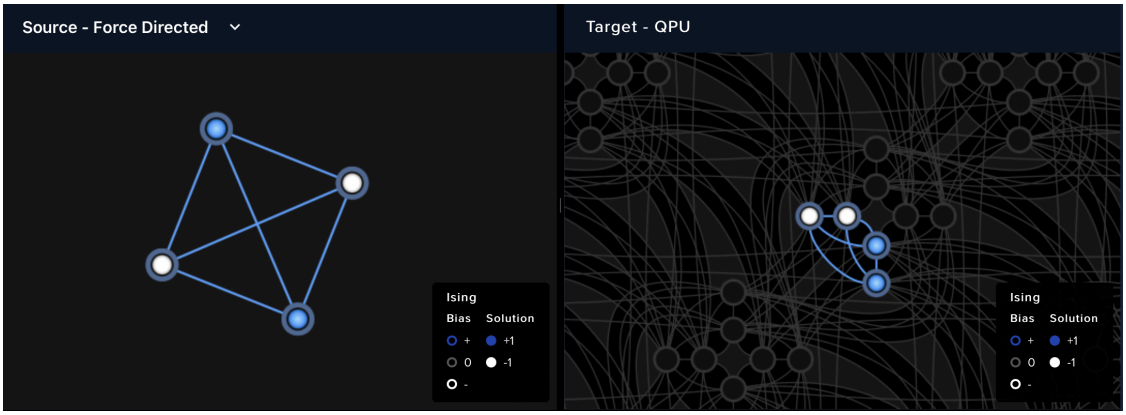
The primary objective of this study is to establish a conceptual framework and PIM that enables the integration of quantum annealing and gate-based quantum devices for solving the ground state energy for multi-partite quantum systems. PIM standardizes quantum computations for ground state calculation across different platforms like D-Wave's Leap SDK and IBM's Qiskit.

The mapping approach and the implementation of the PIM facilitate a seamless transition between computational paradigms and standardize methodology. Table I provides a simplified comparison between the quantum annealing approach and the VQE method in gate-based QC for solving the Ising model:

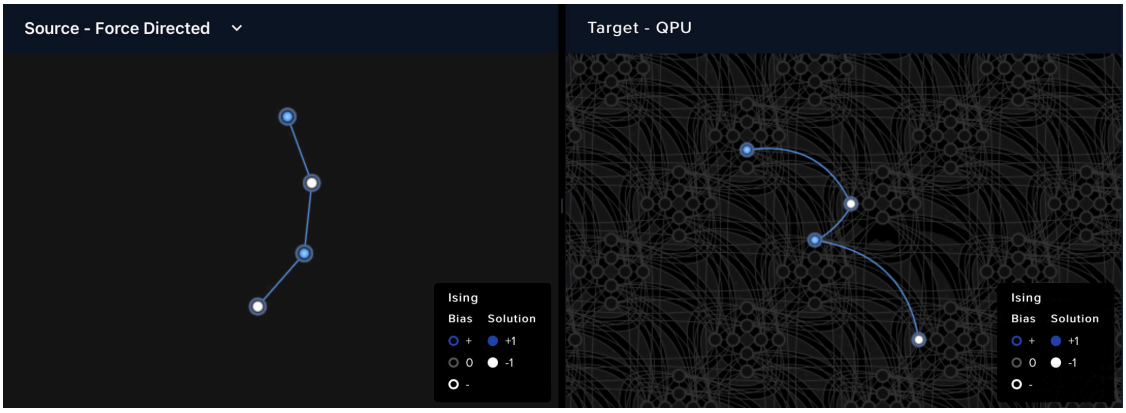
TABLE I: Comparison of Quantum Annealing and Quantum Gate-Based VQE Approaches in Ground State Calculation

Aspect	**Quantum Annealer Approach**	**VQE Approach**
Hamiltonian Mapping	Directly onto hardware qubits	As quantum circuit operators
State Preparation	Physical qubits initialized in superposition	Dynamic state preparation
Entanglement	Fixed by hardware design	Configurable via software
Parameter Optimization	Annealing controls transitions	Algorithmic parameters
Ground State Search	Natural evolution	Iterative optimization
Result Analysis	Immediate hardware output	Computed post-process

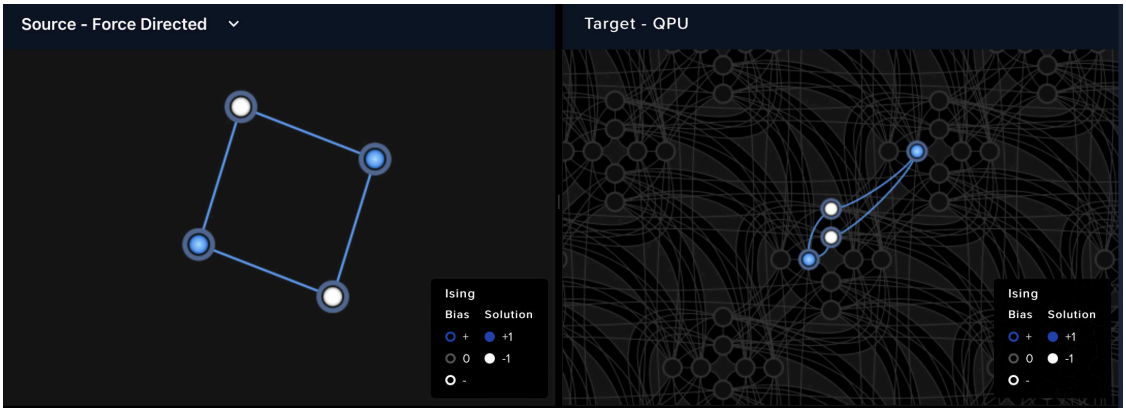
This comparative table illustrates the distinctive methodologies used in quantum annealing and the VQE method for the specific case, The Ground State Calculation problem. It highlights the shift from a more theoretical discussion to a practical application within the



(a) Full-entangled Ising Model



(b) Linear-entangled Ising Model



(c) Circular-entangled Ising Model

FIG. 6: Schematics of different Ising Models on a Quantum Annealer, constructed Leap SDK's default inspector tool, depicted across three entanglement configurations. Each subfigure shows the connections among qubits on the left and the implementation on D-Wave's QPU within Pegasus topology on the right.

PIM framework, which has been fully realized and now supports these two QC approaches effectively for ground state problem.

D. Platform Independent Model (PIM) for Ground State Calculations

This paper proposes a PIM for Ground state calculations to achieve standardized representation of quantum computations across various platforms, such as D-Wave's Leap SDK and IBM's Qiskit. The PIM allows users to define quantum systems using high-level parameters, thus eliminating the need for intricate details of the underlying hardware and SDK.

Unlike universal languages, quantum computers require specific instructions tailored to their unique capabilities. These Platform Specific Models (PSMs) act like custom blueprints, accounting for a machine's strengths, like qubit connections, and limitations, such as available operations. While PSMs unlock a quantum computer's full potential, they often demand in-depth knowledge of the underlying hardware, creating a hurdle for programmers.

By contrasting these platform-specific approaches, the need for a PIM becomes evident, as it abstracts these details into a unified model that can be understood and utilized without deep platform-specific knowledge.

1. Key PIM Parameters

The PIM simplifies the quantum system definition by standardizing the following key parameters:

- **Number of Qubits:** Defines the quantum system's size.
- **External Magnetic Field Constant:** Represents the average influence of external magnetic fields across the system.
- **Coupler Constant:** Captures the strength of interactions (couplings) between qubits.
- **Platform Name:** Identifies the targeted QC platform (e.g., 'dwave', 'qiskit').
- **Entanglement Type:** Describes the interconnection pattern of qubits (e.g., linear, circular, full).

2. Translation Functions between PSM and PIM

The system implements two primary function types:

A. **PSM to PIM Conversion Functions:** These functions (e.g., `PSM_to_PMI_dwave` and `PSM_to_PMI_qiskit`) translate PSMs into the platform-independent PIM format. This conversion is crucial for enabling uniform analysis and understanding of quantum systems across different platforms.

B. **PIM to PSM Conversion Function:** The `PIM_ground_state_calculator` function takes the platform-independent specifications and generates the necessary platform-specific model. This allows direct execution of quantum simulations or calculations on the designated hardware or simulation framework.

3. Use Cases and Benefits of PIM

The proposed PIM system offers several practical advantages:

- **Platform-Agnostic Quantum System Design:** Users can specify quantum systems using PIM parameters without being limited to a specific platform. This is ideal for theoretical and educational purposes where understanding core concepts outweighs platform-specific details.

- **Cross-Platform Translation:** Users working on one platform (e.g., D-Wave's Leap SDK) can effortlessly translate their models to another platform (e.g., IBM's Qiskit) using the conversion functions. This is particularly beneficial for comparing results across platforms or leveraging specific features offered by different platforms.

- **Simplified Interface for Quantum Developers:** By abstracting the complexities of each platform into a common set of parameters, PIM simplifies quantum software development. This can accelerate development cycles and reduce the learning curve for new quantum programmers.

V. CONCLUSION AND FUTURE WORK

A. Key Findings and Contributions

This study not only proposes a unified framework for integrating different quantum computing (QC) platforms but also successfully implements it for a specific use case of ground state calculation. A significant milestone in this framework is the implementation of the PIM for ground state calculation, which bridges the gap between quantum annealing and gate-based QC. It enables the use of a single set of high-level parameters to define and manipulate quantum systems across different platforms. Additionally, it allows transitions from one PSM to another between the Quantum Annealer model and the Gate-Based Quantum system.

B. Future Research and Potential Impact

The results outline possibilities for further research, particularly in refining these mapping techniques for even more sophisticated and broader QC challenges. Future work could explore extending the PIM to support emerging quantum technologies and integrating Domain-Specific Languages (DSLs) that could further simplify quantum algorithm development across platforms.

1. Model Enhancements

To further improve the PIM system, several aspects can be considered:

- **Validation and Error Handling:** The conversion functions should be robust in handling exceptional cases and provide informative error messages for unsupported configurations or invalid inputs.
- **Support for Additional Platforms:** As the field of QC evolves, incorporating support for emerging platforms will ensure the continued relevance and utility of the PIM system.
- **User-Friendly Documentation and Examples:** Comprehensive documentation and illustrative use cases will empower users to use the PIM system effectively. Tuto-

rials or interactive web applications could significantly enhance user engagement and understanding.

- **Extending Compatibility:** Currently, the framework is tailored only for Ising model solutions for the ground state problem. To fully tap into the potential of quantum technologies, it is essential to expand its applicability to encompass all programming algorithms in quantum computing.

These improvements will solidify MDE frameworks as valuable tools for facilitating standardized development and execution of quantum computations across diverse platforms for quantum or hybrid applications.

The impact of using MDE principles in QC is profound, potentially impact how quantum algorithms are developed and making QC more accessible to a wider audience.

REFERENCES

[1] P. Nimbe, B. Weyori, and A. Adekoya, "Models in quantum computing: a systematic review," *Quantum Information Processing*, vol. 20, p. 80, 2021.

[2] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, "Adiabatic quantum computation is equivalent to standard quantum computation," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 166–194, 2007. [Online]. Available: https://doi.org/10.1137/S0097539705447323

[3] N. M. Linke, D. Maslov, and et al., "Experimental comparison of two quantum computing architectures," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3305–3310, 2017.

[4] A. Moin, M. Challenger, A. Badii, and S. Günnemann, "Towards model-driven engineering for quantum ai," in *INFORMATIK 2022*, D. Demmler, D. Krupka, and H. Federrath, Eds. Gesellschaft für Informatik, Bonn, 2022, pp. 1121–1131.

[5] S. Ali and T. Yue, "Modeling quantum programs: Challenges, initial results, and research directions," ser. APEQS 2020. New York, NY, USA: ACM, 2020, p. 14–21.

[6] F. Gemeinhardt, A. Garmendia, and M. Wimmer, "Towards model-driven quantum software engineering," in *Second Int. Workshop on Quantum Software Eng.*, ser. ICSEW: Proceedings of the IEEE/ACM International Conference on Software Engineering Workshops, 2021.

[7] A. Moin, A. Badii, and M. Challenger, "Model-driven quantum federated learning (qfl)," in *Companion Proceedings of the 7th International Conference on the Art, Science, and Engineering of Programming*, ser. Programming '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 111–113. [Online]. Available: https://doi.org/10.1145/3594671.3594690

[8] F. Aryasetiawan and O. Gunnarsson, "The gw method," *Reports on Progress in Physics*, vol. 61, pp. 237–312, 1997.

[9] G. Kresse and J. Hafner, "Ab initio molecular dynamics for liquid metals." *Physical Review B*, vol. 47, pp. 558–561, 1993.

[10] A. Aspuru-Guzik and P. Walther, "Photonic quantum simulators," *Nature Physics*, vol. 8,

pp. 285–291, 2012. [Online]. Available: https://doi.org/10.1038/nphys2253

[11] Affleck, Kennedy, Lieb, and Tasaki, "Rigorous results on valence-bond ground states in antiferromagnets." *Physical review letters*, vol. 59 7, pp. 799–802, 1987.

[12] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, no. 6/7, 1982.

[13] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," pp. 124–134, 1994.

[14] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, pp. 325–328, 1997.

[15] J. Biamonte, P. Wittek, and N. P. et al., "Quantum machine learning," *Nature*, vol. 549, pp. 195–202, 2017.

[16] F. Arute, K. Arya, and R. B. et al., "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, 2019.

[17] J.-P. Aumasson, "The impact of quantum computing on cryptography," *Computer Fraud & Security*, vol. 2017, pp. 8–11, 2017.

[18] A. Ajagekar and F. You, "Quantum computing for energy systems optimization: Challenges and opportunities," *Energy*, 2019.

[19] M. Motta and J. Rice, "Emerging quantum computing algorithms for quantum chemistry," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 12, 2021.

[20] Y. Cao, J. Romero, and A. Aspuru-Guzik, "Potential of quantum computing for drug discovery," *IBM J. Res. Dev.*, vol. 62, pp. 6:1–6:20, 2018.

[21] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[22] A. Peruzzo, J. McClean, and P. S. et al., "A variational eigenvalue solver on a photonic quantum processor," *Nat. Commun.*, vol. 5, p. 4213, 2014.

[23] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[24] V. Bergholm, J. Izaac, and M. S. et al., "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.

[25] S. Boixo, T. F. Rønnow, and S. V. I. et al., "Evidence for quantum annealing with more than one hundred qubits," *Nat. Phys.*, vol. 10, pp. 218–224, 2013.

[26] A. D. King, S. Suzuki, and J. R. et al., "Coherent quantum annealing in a programmable

2,000 qubit ising chain," *Nat. Phys.*, vol. 18, pp. 1324–1328, 2022.

[27] D. Cory, W. Mass, M. Price, E. Knill, R. Laflamme, W. Zurek, T. F. Havel, and S. Somaroo, "Experimental quantum error correction," *Physical Review Letters*, vol. 81, pp. 2152–2155, 1998.

[28] E. Knill, R. Laflamme, and L. Viola, "Theory of quantum error correction for general noise," *Physical review letters*, vol. 84 11, pp. 2525–8, 1996.

[29] N. D. de Leon, K. Itoh, D. Kim, K. Mehta, T. Northup, H. Paik, B. Palmer, N. Samarth, S. Sangtawesin, and D. Steuerman, "Materials challenges and opportunities for quantum computing hardware," *Science*, vol. 372, 2021.

[30] D. Franklin and F. Chong, "Challenges in reliable quantum computing," pp. 247–266, 2004.

[31] R. V. Meter and S. Devitt, "The path to scalable distributed quantum computing," *Computer*, vol. 49, pp. 31–42, 2016.

[32] H. Lee, K. J. Brown, A. K. Sujeeth, H. Chafi, T. Rompf, M. Odersky, and K. Olukotun, "Implementing domain-specific languages for heterogeneous parallel computing," *IEEE Micro*, vol. 31, pp. 42–53, 2011.

[33] A. Idani, Y. Ledru, and G. Vega, "Alliance of model-driven engineering with a proof-based formal approach," *Innovations in Systems and Software Engineering*, pp. 1–19, 2020.

[34] A. Becke, "A new mixing of hartree-fock and local density-functional theories," *Journal of Chemical Physics*, vol. 98, pp. 1372–1377, 1993.

[35] R. Swendsen and J.-S. Wang, "Nonuniversal critical dynamics in monte carlo simulations." *Physical review letters*, vol. 58 2, pp. 86–88, 1987.

[36] U. Schollwoeck, "The density-matrix renormalization group in the age of matrix product states," *Annals of Physics*, vol. 326, pp. 96–192, 2010.

[37] G. Carleo, K. Choo, D. Hofmann, J. E. Smith, T. Westerhout, F. Alet, E. J. Davis, S. Efthymiou, I. Glasser, S.-H. Lin, M. Mauri, G. Mazzola, C. B. Mendl, E. P. van Nieuwenburg, O. O'Reilly, H. Théveniaut, G. Torlai, F. Vicentini, and A. Wietek, "Netket: A machine learning toolkit for many-body quantum systems," *SoftwareX*, vol. 10, 2019. [Online]. Available: https://doi.org/10.1016/j.softx.2019.100311

[38] A. Dash, S. Sahu, S. Kar, B. Behera, and P. Panigrahi, "Explicit demonstration of initial state construction in artificial neural networks using netket and ibm q experience platform," 2019, rG.2.2.30229.17129. [Online]. Available: https://www.researchgate.net/publication/

335159432

[39] F. Vicentini, D. Hofmann, A. Szabó, D. Wu, C. Roth, C. Giuliani, G. Pescia, J. Nys, V. Vargas-Calderón, N. Astrakhantsev, and G. Carleo, "NetKet 3: Machine Learning Toolbox for Many-Body Quantum Systems," *SciPost Phys. Codebases*, p. 7, 2022. [Online]. Available: https://scipost.org/10.21468/SciPostPhysCodeb.7

[40] X. Gao and L.-M. Duan, "Efficient representation of quantum many-body states with deep neural networks," *Nature Communications*, vol. 8, 2017. [Online]. Available: https://doi.org/10.1038/s41467-017-00705-2

[41] K. Choo, G. Carleo, N. Regnault, and T. Neupert, "Symmetries and many-body excitations with neural-network quantum states," *Phys. Rev. Lett.*, vol. 121, p. 167204, Oct 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.121.167204

[42] D. S. Abrams and S. Lloyd, "Simulation of many-body fermi systems on a universal quantum computer," *Phys. Rev. Lett.*, vol. 79, no. 13, pp. 2586–2589, 1997.

[43] S. L. D. S. Abrams, "Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors," *Phys. Rev. Lett.*, vol. 83, no. 24, pp. 5162–5165, 1999.

[44] N. C. Jones, J. D. Whitfield, P. L. McMahon, M.-H. Yung, R. Van Meter, A. Aspuru-Guzik, and Y. Yamamoto, "Faster quantum chemistry simulation on fault-tolerant quantum computers," *New Journal of Physics*, vol. 14, p. 115023, Nov 2012.

[45] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, "Simulation of electronic structure hamiltonians using quantum computers," *Molecular Physics*, vol. 109, no. 5, pp. 735–750, 2011.

[46] M. Cerezo, A. Arrasmith, and R. B. et al., "Variational quantum algorithms," *Nat. Rev. Phys.*, vol. 3, pp. 625–644, 2021.

[47] S. Weinberg, "The quantum theory of fields: The cluster decomposition principle," pp. 169–190, 1995.

[48] V. Havl'ivcek, M. Troyer, and J. Whitfield, "Operator locality in the quantum simulation of fermionic models," *Physical Review A*, vol. 95, 2017.

[49] S. Haley, "Ii. exact solutions of disordered ising spin chains in a magnetic field," *Journal of Mathematical Physics*, vol. 19, pp. 1187–1191, 1978.

[50] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, pp. 602 – 606, 2016.

[51] A. Moin, M. Challenger, A. Badii, and S. Günnemann, "Mde4qai: Towards model-driven

engineering for quantum artificial intelligence," *arXiv preprint arXiv:2107.06708*, 2021.

[52] https://github.com/micss-lab/MDE4QP/.

[53] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms," *Advanced Quantum Technologies*, vol. 2, no. 12, p. 1900070, 2019.

[54] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Phys. Rev. A*, vol. 101, p. 032308, Mar 2020.