
Universal Parameterization in Variational Quantum Eigensolver for Hard Combinatorial Problems using Mean-Field Methods

Master's Thesis

Dongjin Suh

A thesis submitted for the degree of
MSc Physics
to
The Faculty of Mathematics, Computer Science and Natural Sciences at
RWTH Aachen University

Authored at the
Institute for Quantum Computing Analytics
Forschungszentrum Jülich

First examiner: Prof. Dr. Mario Berta
Second examiner: Dr. Tim Bode

June 2025

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Dr. Tim Bode for his continuous supervision throughout this thesis. His valuable advice, insightful ideas, and consistent motivation have been instrumental in shaping and guiding this work. The main credit and sincere thanks go to him.

I would also like to thank Prof. Dr. Mario Berta for supervising me and for providing the opportunity to carry out this research. My thanks also go to Prof. Dr. Frank Wilhelm-Mauch for enabling this project to be conducted in collaboration with the Quantum Computing Analytics (PGI-12) group at Forschungszentrum Jülich.

Furthermore, I wish to thank all members of the PGI-12 group for the collaborative and supportive environment. In particular, I am very grateful to Krish Ramesh and Rivan Rughubar for their exceptional support and help during the course of this work.

Contents

1	Introduction	4
2	Combinatorial Problems	6
2.1	Sherrington-Kirkpatrick model	7
2.2	MAX2SAT Problem	7
3	Adiabatic Quantum Computing	9
3.1	Adiabatic Theorem	9
3.2	Quantum Annealing: A Practical Heuristic	9
4	Mean-Field Methods	11
4.1	Mean-Field Approximation and Equations of Motion	11
4.2	Classical Dynamics of Spin Vectors	12
4.3	Instance Transformation via Mean-Field methods	13
4.3.1	Spin Frustration and Permutation Transformation	13
4.3.2	Shifting the Adiabatic Bottleneck	16
5	Algorithms for solving combinatorial optimization problems	17
5.1	Variational Quantum Eigensolver	17
5.1.1	Motivation and Principle	17
5.1.2	Hybrid Quantum-Classical Architecture	17
5.1.3	Ansatz Design in Variational Quantum Eigensolver	19
5.2	Quantum Approximate Optimization	22
5.3	Graph Neural Network	23
5.3.1	General Theory of Graph Neural Network	23
5.3.2	Multilayer Perceptron	23
5.3.3	Graph Representation of the SK Model for Optimization	24
5.3.4	GNN Architecture for SK Model	25
5.4	Evaluating the Solution Probabilities	26
6	Results	27
6.1	Eigenenergy Spectra and the Hardness of the Instances	27
6.2	Mean-Field Analysis and Transformation of Instances	30
6.2.1	Spin Label Permutation after Frustration Ordering	31
6.2.2	Adiabatic Gap Shift	32
6.3	Optimized Parameter Analysis for QAOA and VQE	32
6.3.1	Optimization Procedure	32
6.3.2	QAOA Parameter Behavior	35
6.3.3	VQE Parameter Behavior	36
6.4	Performance Analysis of QAOA and VQE	37
6.4.1	Cost Value and Average Index versus Number of Layers p for Fixed Spin Size N	37
6.4.2	Average State Index Evaluation	39
6.4.3	Success Probability Comparison	39
6.4.4	Comparison on Differently Transformed Instances	40
6.5	Performance of universal Parameterization for increased Spin sizes	43
6.6	Scalability and Practical Considerations	48
6.7	Benchmark Performance of GNN	49
6.7.1	Training Procedure	49
6.7.2	Performance of GNN against QAOA and VQE	50
7	Conclusion	54
	References	55

A Appendix	58
A.1 Derivation of the Mean-field equations of motion:	58

1 Introduction

Solving combinatorial optimization problems is a central challenge in both theoretical and applied mathematics and computer science, with wide-ranging implications in fields such as engineering and, of course, physics. Many of these problems are classified as NP-hard and thus remain intractable for exact classical algorithms as the system size grows. Whether it's scheduling, resource allocation, route planning, or solving physical models, these problems often require finding the best possible solution among many alternatives. While small problems can be solved using exact methods, many real-world problems are so complex that classical computers struggle to solve them in a reasonable amount of time.

This is especially true for so-called combinatorial optimization problems. These are problems where a solution must be chosen from a large, discrete set of possibilities, such as assigning values to variables or choosing paths in a graph. Many of these problems are known to be computationally hard, meaning that the time needed to solve them exactly grows very quickly with the problem size. Among the most studied examples are the MAX2SAT problem and the Sherrington-Kirkpatrick (SK) model, both of which serve as benchmark instances for the development of heuristic and quantum optimization methods.

Quantum computing offers a fundamentally new framework for approaching such problems. It is based on the laws of quantum physics and works very differently from classical computing. While the most prominent model is the digital quantum circuit model, based on sequential application of unitary quantum gates, it is not the only paradigm. In this thesis, we are particularly interested in quantum optimization strategies that aim to approximate the ground state of a given problem Hamiltonian. Over the past years, several quantum algorithms have been developed that aim to solve optimization problems by making use of quantum effects such as superposition and entanglement. A common goal in many of these algorithms is to find the ground state (i.e., the lowest energy state) of a certain mathematical object called a Hamiltonian. The idea is that the ground state of the Hamiltonian encodes the optimal solution to the problem at hand.

There are different ways to reach this ground state using quantum methods. One of them is *Adiabatic Quantum Computation* (AQC), which slowly changes the system from a simple starting point into a complex final state. The system begins in the ground state of an initial, well-known Hamiltonian and is then slowly evolved into the final Hamiltonian that represents the problem. According to the adiabatic theorem, if this evolution is done slowly enough and the energy gap remains open, the system will remain in the ground state throughout the process. At the end, the state of the system should represent the solution to the optimization problem [1, 2].

Another important method is the *Variational Quantum Eigensolver* (VQE). VQE uses a quantum circuit with adjustable parameters. These parameters are optimized with the help of a classical computer to minimize the energy of the quantum state. The hope is that, after optimization, the quantum state represents or approximates the ground state of the problem Hamiltonian. VQE has gained popularity because it is well-suited for today's noisy quantum hardware and can be adapted to many types of problems.

A related algorithm is the *Quantum Approximate Optimization Algorithm* (QAOA), which can be seen as a special case of VQE with a specific circuit structure. QAOA alternates between two types of quantum operations: one based on the problem Hamiltonian and one based on a simpler mixing Hamiltonian. The goal is to find the best sequence of these operations (defined by angles or parameters) that leads to a quantum state with low energy. Although conceptually simple, QAOA can become difficult to optimize as the number of layers increases, and finding good parameters can be challenging [3].

Because of this, there has been growing interest in strategies that allow VQE to work across many different problem instances without needing to start from scratch each time. One promising idea is to find *universal parameters*, parameter settings that perform well across a whole class of problems. Another idea is to apply *transformations* to the problem instances so that they become easier to solve, while keeping their essential properties. In some cases, this can be done using mean-field dynamics, which captures the average behavior of large systems and helps identify important features such as which parts of the system are most difficult to optimize.

In addition to physics-inspired approaches, machine learning is also being explored as a tool to

assist in quantum optimization. By training neural networks on examples of solved problems, one can hope to predict good parameter settings or even approximate solutions for new instances. Graph Neural Networks (GNNs) are particularly well-suited for this task, as they can naturally represent the structure of many optimization problems, including spin glasses and SAT problems.

This thesis combines all of these ideas. We focus on two representative problems: the Sherrington-Kirkpatrick model from statistical physics, and the MAX2SAT problem from theoretical computer science. We study how mean-field inspired instance transformations can make these problems easier to solve with quantum algorithms, how universal parameters can be developed, and how machine learning models, especially GNNs, can help in predicting solutions. Our goal is to develop methods that are robust, scalable, and capable of generalizing to new problem instances.

2 Combinatorial Problems

The quest to solve complex combinatorial optimization problems lies at the heart of numerous scientific and industrial applications. These problems appear in a wide range of contexts, including logistics, circuit design, machine learning, and material discovery. These problems typically involve searching for an optimal configuration among a large number of discrete possibilities. In many cases, especially when the number of variables becomes large, the problem quickly becomes intractable. This is because many combinatorial problems fall into the class of NP-hard problems, for which no efficient (i.e., polynomial-time) algorithms are currently known. Tackling such problems has therefore become a central challenge across computational physics, computer science, and operations research.

To make sense of the difficulty associated with such problems, it is helpful to briefly review the standard hierarchy of computational complexity classes. These are:

- **P:** Problems that can be solved in polynomial time using deterministic classical algorithms. Examples include sorting algorithms or computing the shortest path in a graph using Dijkstra’s algorithm [4].
- **NP:** Problems in this class allow efficient verification of proposed solutions, even though finding the solution itself may be computationally challenging. A canonical example is integer factorization.
- **NP-complete:** These problems are both in NP and as hard as any problem in NP. If one could solve any NP-complete problem in polynomial time, then all problems in NP could also be solved efficiently. Examples include the decision versions of Max-Cut, Knapsack, and Traveling Salesperson problems.
- **NP-hard:** This class contains problems that are at least as hard as NP-complete problems but not necessarily in NP themselves. These include general optimization versions of Max-Cut, Knapsack, and TSP. For these problems, no algorithm is known that guarantees a polynomial-time solution, and in practice, the runtime grows exponentially with the system size.

In short, NP-hard problems represent the frontier of what is computationally feasible. Classical algorithms often fail to solve large instances of these problems due to the exponential blow-up of resources needed. The computational cost becomes especially steep in systems with rugged energy landscapes and many local minima—typical features in models like the Sherrington-Kirkpatrick spin glass [5] or constraint satisfaction problems like MAX2SAT [6].

Given this context, there has been growing interest in quantum computing approaches to optimization. Quantum algorithms can, in principle, exploit superposition, interference, and tunneling to explore complex solution spaces more effectively. In particular, algorithms like the Quantum Approximate Optimization Algorithm (QAOA) [7, 6] and the Variational Quantum Eigensolver (VQE) [8] are designed to approximate ground states of Hamiltonians that encode the cost functions of such combinatorial problems.

These approaches are motivated not only by theoretical insights, but also by empirical results. Studies have shown that VQE and QAOA are capable of capturing low-energy states even for hard optimization instances, and there is evidence that quantum-inspired strategies such as mean-field approximations [9] or hybrid classical-quantum feedback loops [10] may offer practical improvements.

To solve a combinatorial optimization problem using quantum annealing or QAOA, which will be introduced later, the quantum algorithm must first interpret the problem in a form amenable to quantum processing. This is typically achieved by mapping the objective function to a classical Hamiltonian, most often written in the form of an Ising model or its equivalent Quadratic Unconstrained Binary Optimization (QUBO) representation [11, 12, 13]. The Ising Hamiltonian takes the general form

$$H = - \sum_{i < j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z - \sum_i h_i \hat{\sigma}_i^z, \quad (2.1)$$

where $\hat{\sigma}_i^z$ are Pauli-Z spin operators, J_{ij} represent pairwise couplings, and h_i are local fields. This formalism is highly expressive and serves as an unifying language for many discrete optimization

problems and the Sherrington-Kirkpatrick model is a paradigmatic example of such a Hamiltonian [5]. Based on this, the focus in this thesis lies on two archetypal NP-hard problems: The SK-model and MAX2SAT. These represent different types of hardness, one originating from disordered spin interactions and the other from clause-based logical constraints. The aim is to explore how quantum optimization methods, especially QAOA and VQE with universal parameterizations informed by mean-field analysis, can be leveraged to solve them efficiently.

As we will see in later chapters, both QAOA and VQE are applied to this model using its direct encoding into a time-dependent or variational Hamiltonian framework.

2.1 Sherrington-Kirkpatrick model

One of the mapped problems specifically, that is introduced here, is the Sherrington-Kirkpatrick model. It is a spin glass system and spin glasses are frustrated magnetic systems and a hallmark of their “glassiness” is the presence of a rugged energy landscape with many local minima. It appears obvious that in such an environment quantum effects might play an important role by opening new routes for relaxation due to quantum mechanical tunneling and indeed one observes experimentally a significant acceleration of the dynamics at low temperatures if quantum fluctuations are enhanced. It is also identified as a NP-hard problem. The origin of this problem point towards the natural Ising spin model, which can be seen from the Hamiltonian representation. The Sherrington-Kirkpatrick model is a paradigmatic example of a spin glass system. The Hamiltonian of the model is given by

$$\hat{H} = -\frac{1}{\sqrt{N}} \sum_{i < j \leq N} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z, \quad (2.2)$$

where $\hat{\sigma}^x$ and $\hat{\sigma}^z$ are Pauli spin operators. Every spin interacts with all others via random Gaussian couplings J_{ij} of zero mean and variance J^2/N . h and J are standard normal random variables representing local fields and spin-spin couplings, drawn from a Gaussian distribution, respectively.

Upon studying the problem’s structure, we observe that the energy depends solely on parity, i.e., it relies on the pairwise orientation of the spins ($S_i S_j$). Which creates a sense of symmetry with the states. Specifically, a bit string will possess the same energy as its conjugate (where all the spins are flipped). This property is known as Z2 symmetry. Now, while Z2 symmetry as such does not pose a problem, we can further optimize the problem by fixing the orientation of the last spin. By doing so, we introduce an asymmetry that alters the energy landscape, potentially reducing the convergence to redundant solutions.

$$\hat{H} = -\frac{1}{\sqrt{N}} \sum_{i=1}^N \left(h_i + \sum_{j>i} J_{ij} \hat{\sigma}_j^z \right) \hat{\sigma}_i^z, \quad (2.3)$$

This shows the new “effective Hamiltonian”, is one where the last column of the coupling matrix (J) acts as a local magnetic field. In other words, the impact of the fixed spin on the other spins can be envisioned as a local magnetic field acting on the new subsystem.

2.2 MAX2SAT Problem

The MAX2SAT problem is a canonical example of a constraint satisfaction problem (CSP) and represents a special case of the Boolean satisfiability problem (SAT), which is central in computational complexity theory. In the MAX2SAT variant, one is given a Boolean formula in conjunctive normal form (CNF) where each clause contains exactly two literals, and the task is to find a truth assignment to the variables that maximizes the number of satisfied clauses. Despite its simple formulation, MAX2SAT is NP-hard [14] and exhibits a rich structure of local optima and frustration, making it an ideal benchmark for optimization algorithms, including quantum heuristics like QAOA. The problem is naturally mappable to the Ising model, enabling its treatment within quantum annealing and variational frameworks by encoding logical constraints into spin interactions and local fields.

Mathematical Formulation. Each clause in a MAX2SAT instance involves two literals, i.e., Boolean variables or their negations. The goal is to find a variable assignment that satisfies the maximum number of clauses of the form:

$$(x_i \vee x_j), \quad (\neg x_i \vee x_j), \quad (x_i \vee \neg x_j), \quad (\neg x_i \vee \neg x_j),$$

where $x_i \in \{0, 1\}$ denotes a Boolean variable.

Ising Encoding. To enable treatment by quantum optimization algorithms, the Boolean variables are mapped to spin variables via the transformation:

$$x_i \mapsto \frac{1 - z_i}{2}, \quad \text{where } z_i \in \{-1, +1\}.$$

Each clause then contributes a local cost term to the Hamiltonian, and the full MAX2SAT problem can be written as minimizing the total energy of the effective Ising Hamiltonian:

$$H = - \sum_{\langle i, j \rangle} J_{ij} z_i z_j - \sum_i h_i z_i + \text{const.}, \quad (2.4)$$

where the coupling coefficients J_{ij} and local fields h_i are determined by the clause structure. This allows MAX2SAT to be natively embedded into the quantum variational framework and solved with algorithms such as QAOA, where the classical cost function becomes the problem Hamiltonian H_Z [11, 15].

Due to its clause-based logical structure and moderate ruggedness of the solution landscape, MAX2SAT serves as a useful complement to spin-glass-based models like SK in testing the effectiveness of universal parameterization and mean-field-inspired initialization schemes in quantum variational algorithms.

3 Adiabatic Quantum Computing

3.1 Adiabatic Theorem

Adiabatic Quantum Computation is grounded in the adiabatic theorem, which states that a quantum system initialized in the ground state of a Hamiltonian will remain in its instantaneous ground state throughout a slow evolution of the Hamiltonian. This concept is harnessed in AQC, where the time-dependent Hamiltonian is defined as:

$$H(s) = (1 - s)H_D + sH_P, \quad s \in [0, 1], \quad (3.1)$$

where:

- H_D : A Driver Hamiltonian with a simple, easily accessible ground state.
- H_P : The Problem Hamiltonian whose ground state encodes the solution to the optimization problem.

The system begins in the ground state of H_D and transitions adiabatically to the ground state of H_P [16, 13].

For the process to remain adiabatic, the evolution time τ must satisfy:

$$\tau \gg \max_s \frac{\langle \psi_1(s) | \frac{dH}{ds} | \psi_0(s) \rangle}{\Delta(s)^2}, \quad (3.2)$$

where

$$\Delta(s) = E_\alpha(s) - E_0(s) \quad (3.3)$$

is the energy gap between the ground state and any excited state α . For hard optimization problems, this gap often becomes exponentially small [17], necessitating prohibitively long evolution times [1].

3.2 Quantum Annealing: A Practical Heuristic

Building on the principles of AQC, Quantum Annealing (QA) provides a heuristic alternative for solving combinatorial optimization problems. Instead of strict adiabaticity, QA uses quantum tunneling to navigate rugged energy landscapes more efficiently than classical thermal processes [13, 10]. The indication was that quantum fluctuations in the SK model can perhaps lead to escape routes to ergodicity or quantum fluctuation-induced delocalization (at least in the low-temperature regions of the spin-glass phase) by allowing tunneling through macroscopically tall but thin barriers that are otherwise difficult to cross using classical fluctuations [10, 17]. This is based on the argument that escape probability due to quantum tunneling from a single valley with a barrier of height V and width w scales as

$$\exp(-\sqrt{Vw}/\Gamma),$$

where Γ is the quantum fluctuation strength. This is visualized in the Fig. 1.

In contrast, classical escape probability scales as $\exp(-V/T)$, where T is the system temperature. The dependence on the width w , which is absent in the classical case, significantly enhances tunneling in narrow-barrier regimes. For example, in a single narrow ($w \rightarrow 0$) barrier of height N , when Γ is slowly reduced to zero, the annealing time to reach the ground state becomes N -independent. [10] This quantum advantage becomes particularly relevant in combinatorial problems with tall and narrow barriers, where classical methods get trapped but quantum dynamics can traverse via tunneling. Such heuristics form the basis for embedding optimization into quantum systems and motivate hybrid quantum-classical algorithms such as VQE.

The theoretical formulation of quantum annealing also uses a Hamiltonian evolution of the form from Eq. (3.1), but without requiring strict adiabatic evolution. The problem Hamiltonian H_P is often expressed as an Ising Hamiltonian, mapping the optimization problem to a spin system [11]:

$$H_P = - \sum_{i < j} J_{ij} \hat{\sigma}_i \hat{\sigma}_j. \quad (3.4)$$

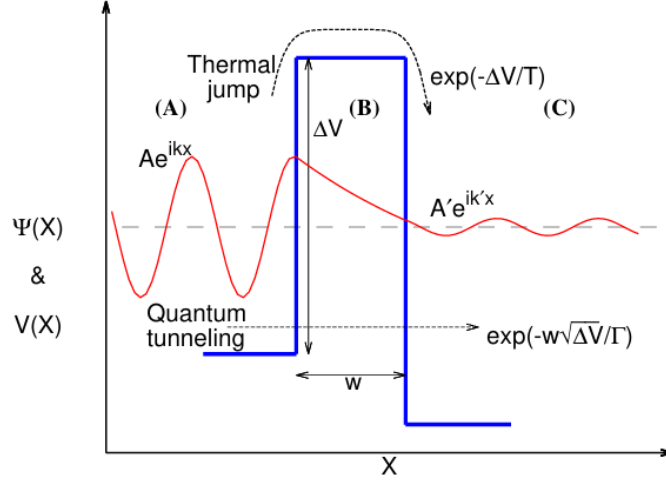


Figure 1: Illustration of barrier crossing in quantum annealing. A classical system would need a thermal fluctuation (dashed path) to jump over the barrier of height ΔV , whereas a quantum system can tunnel through it with probability amplitude $\exp(-w\sqrt{\Delta V/\Gamma})$. This mechanism allows quantum annealing to explore energy landscapes more efficiently than classical thermal methods. Adapted from [10].

Especially for the SK model, the classical spin glass Hamiltonian of Eq. (2.3) would be the problem Hamiltonian H_P . The driver Hamiltonian is then given as

$$H_X = -\Gamma \sum_{i=1}^N \hat{\sigma}_i^x, \quad (3.5)$$

where Γ is the strength of the transverse field. The full adiabatic Hamiltonian takes the form:

$$H(s) = (1 - s(t))H_X + s(t)H_Z, \quad (3.6)$$

where the annealing schedule is $s(t) = \frac{t}{T_f}$ with $t \in [0, T_f]$.

Critical Dynamics and Adiabatic Bottleneck

During the annealing evolution, the system encounters critical points where the gap $\Delta(s)$ (Eq. (3.3)) becomes minimal. The smallest such gap, or minigap, is defined as

$$\Delta = \Delta(s_{\min}),$$

with s_{\min} marking the location of the minimum. The criticality is analyzed using the adiabatic ratio:

$$g_\alpha(s) = \frac{|\langle \alpha | \partial_s H(s) | 0 \rangle|}{[E_\alpha(s) - E_0(s)]^2} = \frac{|\langle \alpha | H_Z - H_X | 0 \rangle|}{[E_\alpha(s) - E_0(s)]^2}, \quad (3.7)$$

where $|\alpha\rangle$ is an excited state (typically the first), and $|0\rangle$ the ground state. The most critical point is then:

$$s^* := \arg \max_{s \in [0,1]} g_\alpha(s), \quad (3.8)$$

typically for $\alpha = 1$ in the hardest instances.

The adiabatic condition becomes:

$$\dot{s}(t)g_\alpha(s^*) = \frac{g_\alpha(s^*)}{T_f} \ll 1, \quad (3.9)$$

ensuring the system remains adiabatic. The presence of bottlenecks - regions with extremely small gaps - poses serious challenges to scalability. Such challenges motivate the development of hybrid or variational approaches like QAOA [3, 7] and mean-field-based optimization strategies [9].

4 Mean-Field Methods

To address the ruggedness of the SK model, the mean-field approximation is applied which is a semi-classical method. We use this as an approach to simplifying the system by transforming the energy landscape of the SK model into a more uniform structure, which enables optimization. This provides a foundation for universal parameterization in quantum algorithms such as the VQE and QAOA, our actual main goal, which can exploit the uniform landscape for improved performance. The mean-field approximation replaces the interactions of all other spins on a given spin with an average field. The mean-field form of the adiabatic Hamiltonian is given by:

$$H(t) = -s(t) \sum_{i=1}^N \left(h_i + \sum_{j>i} J_{ij} n_j^z(t) \right) n_i^z(t) - (1-s(t)) \sum_{i=1}^N n_i^x(t), \quad (4.1)$$

where $n_i(t)$ represents classical spin vectors evolving on their respective Bloch spheres.

4.1 Mean-Field Approximation and Equations of Motion

The mean-field approximation offers a computationally efficient way to model many-body quantum systems by assuming that each spin interacts with an average field produced by the rest of the system. This simplification replaces the full quantum dynamics with a set of coupled classical equations for spin vectors $\vec{n}_i(t)$ evolving on the Bloch sphere. The key idea is to factorize the total quantum state into a product of single-qubit states, neglecting quantum entanglement while retaining non-linear interactions mediated by the mean field.

Let us consider the classical spin vector $\vec{n}_i(t)$ of the i -th qubit defined by its expectation values under the Pauli matrices:

$$\vec{n}_i(t) = (n_i^x(t), n_i^y(t), n_i^z(t))^T = (\text{Tr}[\hat{\rho} \hat{\sigma}_i^x], \text{Tr}[\hat{\rho} \hat{\sigma}_i^y], \text{Tr}[\hat{\rho} \hat{\sigma}_i^z])^T, \quad (4.2)$$

where the density matrix ρ is approximated as a tensor product over individual spin states:

$$\hat{\rho}(t) = \bigotimes_{i=1}^N \hat{\rho}^{(i)}(t). \quad (4.3)$$

Each spin experiences an effective magnetic field derived from a time-dependent mean-field Hamiltonian:

$$H(t) = -s(t) \sum_{i=1}^N \left(h_i + \sum_{j>i} J_{ij} n_j^z(t) \right) n_i^z(t) - (1-s(t)) \sum_{i=1}^N n_i^x(t), \quad (4.4)$$

where $s(t) \in [0, 1]$ is the interpolation schedule and h_i, J_{ij} define the local fields and spin-spin couplings. The evolution of the spin vector $\vec{n}_i(t)$ is governed by the classical precession equation:

$$\frac{d}{dt} \vec{n}_i(t) = \vec{n}_i(t) \times \vec{B}_i(t), \quad (4.5)$$

with the effective magnetic field

$$\vec{B}_i(t) = 2(1-s(t))\hat{e}_x + 2s(t)m_i(t)\hat{e}_z, \quad (4.6)$$

and local magnetization

$$m_i(t) = h_i + \sum_{j=1}^N J_{ij} n_j^z(t). \quad (4.7)$$

For a detailed derivation of this classical equations of motion for the spin vectors, see Appendix A.1. Finally, this leads to the following set of coupled non-linear differential equations:

$$\dot{n}_i^x(t) = 2s(t)m_i(t)n_i^y(t), \quad (4.8)$$

$$\dot{n}_i^y(t) = -2s(t)m_i(t)n_i^x(t) + 2(1-s(t))n_i^z(t), \quad (4.9)$$

$$\dot{n}_i^z(t) = -2(1-s(t))n_i^y(t). \quad (4.10)$$

These equations conserve the norm of the spin vector:

$$|\vec{n}_i(t)|^2 = (n_i^x(t))^2 + (n_i^y(t))^2 + (n_i^z(t))^2 = 1, \quad (4.11)$$

ensuring that each spin remains on the surface of the Bloch sphere throughout its evolution.

This mean-field formulation captures the interaction between quantum control (via the schedule $s(t)$) and spin interactions (through J_{ij}), forming the basis for analyzing variational protocols under semi-classical dynamics [9, 16].

4.2 Classical Dynamics of Spin Vectors

The classical dynamics of the spin vectors introduced in the previous section provide a useful analogy for understanding variational quantum algorithms within the mean-field approximation. In this setting, the spins are modeled as precessing vectors on the Bloch sphere, and their evolution is driven by an effective magnetic field derived from the interpolation of the driver and problem Hamiltonians.

We consider a piecewise-constant schedule, reminiscent of VQE dynamics, where time evolution alternates between two types of rotations: one generated by the transverse field term (driver Hamiltonian), and one by the longitudinal Ising term (problem Hamiltonian). For a total layers p of VQE, each layer consists of a unitary evolution under these two components:

$$\vec{n}_i(p) = \prod_{k=1}^p \hat{V}_D^{(i)}(k) \hat{V}_P^{(i)}(k) \vec{n}_i(0), \quad (4.12)$$

where $\vec{n}_i(0)$ is the initial spin vector and $\hat{V}_D^{(i)}(k)$ and $\hat{V}_P^{(i)}(k)$ are classical rotation matrices corresponding to the k -th driver and problem unitaries, respectively. These matrices take the explicit form:

$$\hat{V}_D^{(i)}(k) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(2\beta_k) & \sin(2\beta_k) \\ 0 & -\sin(2\beta_k) & \cos(2\beta_k) \end{pmatrix},$$

$$\hat{V}_P^{(i)}(k) = \begin{pmatrix} \cos(2m_i(t_{k-1})\gamma_k) & \sin(2m_i(t_{k-1})\gamma_k) & 0 \\ -\sin(2m_i(t_{k-1})\gamma_k) & \cos(2m_i(t_{k-1})\gamma_k) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Here, the angles β_k and γ_k represent the QAOA parameters at layer k , and $m_i(t_{k-1})$ denotes the effective magnetization experienced by spin i just before the k -th layer:

$$m_i(t_{k-1}) = h_i + \sum_{j=1}^N J_{ij} n_j^z(t_{k-1}). \quad (4.13)$$

This alternating sequence of rotations models the variational ansatz of QAOA in a classical setting. It retains key features of the quantum dynamics - namely, the accumulation of constructive interference through tailored phase evolutions - while making the evolution tractable via classical simulation.

Furthermore, one may define a consistent interpolation for the parameters based on an annealing schedule with total time τ . For example:

$$\gamma_k = \frac{\tau \cdot k}{p}, \quad \beta_k = \tau \left(1 - \frac{k-1}{p} \right),$$

which preserves the piecewise-linear character of the adiabatic schedule in the limit of large p [3, 9].

This representation provides a dynamical picture of QAOA as a classical precession problem and offers intuition for how specific parameter choices $(\vec{\gamma}, \vec{\beta})$ influence the trajectory of the system in Hilbert space. In subsequent sections, we will explore how this insight allows us to understand optimization hardness and motivate instance transformations.

4.3 Instance Transformation via Mean-Field methods

The Sherrington-Kirkpatrick model is a paradigmatic system in spin glass theory, characterized by its complexity of its low-energy landscape with numerous local minima. This complexity arises from the randomness of the couplings J and local fields h , making the model an excellent testbed for studying combinatorial optimization problems. Similar rugged energy landscapes are also encountered in other hard combinatorial problems, which makes it extremely difficult to find the global optimum.

To address this problem, we apply unitary transformations to the instances in order to reshape the energy landscape into a more uniform and manageable structure [9]. By doing so, we aim to achieve two key objectives. First, we seek to enable universal parametrization: through a uniform landscape, we expect to facilitate the development of universal parameters for the VQE and QAOA, which will be introduced later. At the same time, the transformations must retain the essential physical properties and characteristics of the system to ensure validity. For this, we focus on two physical phenomena of the spin glass model using mean-field methods [17]. Transforming hard instances of the SK model involves aligning spins with their frustration patterns and normalizing the adiabatic bottleneck. This process ensures that optimization schedules focus on the most challenging regions, paving the way for quantum methods. The transformation procedure involves several concrete steps:

- **Permutation of spin labels** to align with frustration ordering.
- **Identification of the critical point** s^* , which corresponds to phase transitions during annealing.
- **Adjustment of annealing schedules**, shifting the bottleneck s^* to $s = 0.5$ for uniformity across instances.

The transformations applied to the energy landscape, specifically spin frustration permutation and adiabatic bottleneck shift, are designed to reshape the optimization landscape for computational efficiency while maintaining the fundamental physical attributes of the system [17, 9].

4.3.1 Spin Frustration and Permutation Transformation

Frustration of Spins

Frustration is an important and ubiquitous phenomenon in disordered systems and spin glasses [18]. The stronger a spin is frustrated, the longer it remains undecided whether to align toward the north or south pole of its Bloch sphere. A graphical illustration of this behavior is shown in Fig. 2.

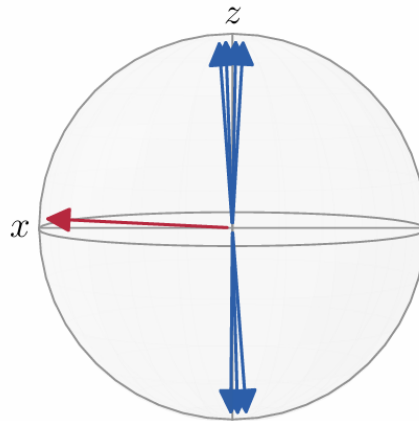


Figure 2: A graphical representation of a highly frustrated spin. the most frustrated spin remains near the equatorial plane throughout the evolution and only begins to polarize around the critical point. In contrast, the other spins - except for one other moderately frustrated case - tend to align toward $n_i^z = \pm 1$ significantly earlier. Adapted from [17].

This behavior is demonstrated for representative trajectories in Fig. 3: the orange trajectory corresponds to the most frustrated spin $i = 12$, which delays its alignment along the z -axis compared to other spins. The upper panel shows the evolution of the x -component $n_i^x(s)$, while the lower panel displays the z -component $n_i^z(s)$. The spin behavior shows that highly frustrated spins tend to remain in the equatorial plane until a critical point in the annealing process is reached, whereas other, less frustrated spins begin converging to values $n^z = \pm 1$ much earlier. The most frustrated spin is marked in orange and it only begins to polarize near the critical point s^* , highlighted by the shaded area, while others polarize significantly earlier. In such cases, the effective magnetic field m_{i^*} experienced by the frustrated spin remains close to zero throughout the evolution [17]. This visualization reinforces the

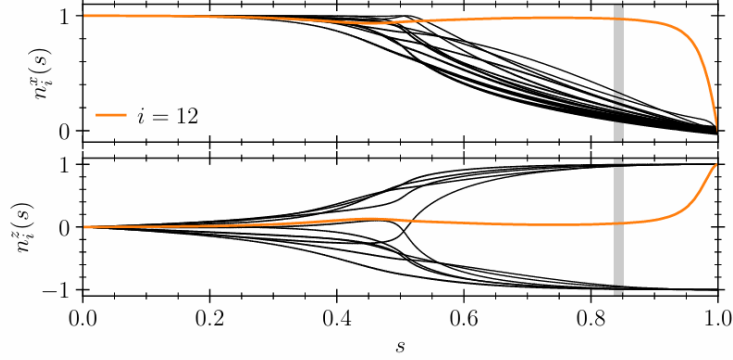


Figure 3: Dynamical evolution of spin components under mean-field equations for a single SK instance. The most frustrated spin (orange, $i = 12$) shows delayed polarization, remaining near the equatorial plane in $n_i^z(s)$ until close to the critical point $s^* \approx 0.8$. Other spins begin polarizing earlier. The shaded region indicates the onset of criticality. Adapted from [17].

quantitative definition of frustration used in our work, where the cumulative local order parameter $Q(i)$ serves as a metric for identifying spins that remain delocalized the longest during the annealing process. As we will show later, reordering the spins based on this metric leads to more uniform energy landscapes and facilitates optimization across instances.

To capture this phenomenon quantitatively, we first introduce the local order parameter:

$$q_i(s) := (n_i^z(s))^2, \quad (4.14)$$

from which the classical Edwards-Anderson (EA) order parameter [18] follows by summation:

$$q_{\text{EA}}(s) = \frac{1}{N} \sum_{i=1}^N q_i(s), \quad (4.15)$$

which can be interpreted as an indicator of the system's localization at a given point in the annealing schedule (see Fig. 3 for an example plot).

To quantify individual spin frustration, we define the cumulative local order parameter:

$$Q(i) = \int_0^1 ds q_i(s). \quad (4.16)$$

This metric reflects how long a spin remains delocalized during the annealing evolution. Alternatively, one may integrate the effective magnetic field [17] or observe how the ground-state energy $E_0(T_f)$ changes upon flipping each spin—both approaches yield comparable frustration assessments. Nevertheless, in this work, we adopt $Q(i)$ as our definition and identify the maximally frustrated spin as:

$$i^* := \arg \min_{i \in \{1, \dots, N\}} Q(i). \quad (4.17)$$

As we demonstrate below, focusing on this maximally frustrated spin allows us to infer the onset of the critical region in the annealing protocol with far greater precision than aggregate measures involving all spins.

Spin Frustration Permutation

Once the frustration ordering of spins is obtained from the mean-field dynamics [9], we apply the *Spin Frustration Permutation* to the instance. This transformation reorders the spins to align the frustration structure into a more uniform configuration, thereby reshaping the energy landscape without modifying the physical properties of the system. The goal is to facilitate optimization by reducing irregularities in the landscape.

The mathematical representation of the *Spin Frustration Permutation* is introduced in the following. The original Hamiltonian of the SK model is:

$$H = - \sum_{i=1}^N \left(h_i + \sum_{j>i} J_{ij} \hat{\sigma}_j^z \right) \hat{\sigma}_i^z,$$

where:

- J_{ij} are the pairwise couplings,
- h_i are local magnetic fields,
- σ_i^z are Pauli spin- z operators.

To perform the permutation, we define a permutation P on the spin indices. Applying P , the relabeled parameters become:

$$J'_{ij} = J_{P(i),P(j)}, \quad h'_i = h_{P(i)},$$

leading to the permuted Hamiltonian:

$$H' = - \sum_{i=1}^N \left(h'_i + \sum_{j>i} J'_{ij} \hat{\sigma}_j^z \right) \hat{\sigma}_i^z. \quad (4.18)$$

For a matrix representation, the couplings J_{ij} can be viewed as a symmetric matrix J , and the permutation is expressed as:

$$J' = PJP^T, \quad (4.19)$$

where P is a permutation matrix with exactly one entry of 1 in each row and column, satisfying $P^T P = I$. The spin frustration permutation corresponds to a unitary basis transformation. It preserves the eigenvalues of the Hamiltonian and thus leaves the energy spectrum invariant [9].

By reordering the spins, the couplings J'_{ij} can be arranged such that frustrated interactions—those with opposing coupling signs—are more evenly distributed. This reduces the variance and irregularity of the energy landscape, thereby making it more amenable to optimization techniques such as QAOA or quantum annealing [17].

In summary, the spin frustration permutation is formally defined as:

$$H' = - \sum_{i=1}^N \left(h_{P(i)} + \sum_{j>i} J_{P(i),P(j)} \hat{\sigma}_j^z \right) \hat{\sigma}_i^z, \quad (4.20)$$

or equivalently in matrix form:

$$J' = PJP^T. \quad (4.21)$$

This transformation preserves the fundamental physical properties of the system while reshaping the energy landscape in a way that improves the performance and generalization of variational quantum algorithms [9, 17].

4.3.2 Shifting the Adiabatic Bottleneck

A second key transformation targets the *adiabatic bottleneck* in the annealing dynamics, which has been introduced previously. The adiabatic bottleneck occurs at the point s^* where the energy gap $\Delta(s)$ between the ground state and the first excited state reaches its minimum during the interpolation between the driver and problem Hamiltonians. This bottleneck typically determines the required runtime for adiabatic evolution, as the minimal gap governs the adiabatic condition [19, 1].

For efficient optimization, aligning the bottleneck to a fixed interpolation point, typically $s^* = 0.5$, can simplify schedule design and facilitate the training of universal variational parameters [17]. To achieve this, we introduce a scaling factor r to the problem Hamiltonian H_Z , which preserves the overall adiabatic trajectory but shifts the location of the bottleneck to the target value.

The time-dependent Hamiltonian during annealing is generally expressed as given in Eq. (3.1):

$$H(s) = (1 - s)H_X + sH_Z,$$

where H_X is the driver Hamiltonian (often a transverse field), and H_Z is the problem Hamiltonian encoding the classical optimization cost function. The interpolation parameter $s \in [0, 1]$ controls the mixing.

The adiabatic bottleneck occurs at $s = s^*$, where the spectral gap $\Delta(s) = E_1(s) - E_0(s)$ (Eq. (3.3)) between the ground and first excited state reaches its minimum:

$$\Delta_{\min} = \min_{s \in [0, 1]} (E_1(s) - E_0(s)).$$

To shift this minimum to the midpoint of the annealing schedule, $s^* = 0.5$, we rescale the contribution of the problem Hamiltonian by a factor r , resulting in a new Hamiltonian:

$$H'(s) = (1 - s)H_X + s \cdot rH_Z. \quad (4.22)$$

Our goal is to derive an expression for r such that the new bottleneck position satisfies $s^* = 0.5$.

Energy Balance at the Bottleneck. At the bottleneck, the system is maximally sensitive to both Hamiltonian components; thus, we assume that the energy contributions of the driver and problem Hamiltonians are balanced:

$$(1 - s^*)H_X = s^*H_Z. \quad (4.23)$$

Rewriting this for the original bottleneck s_{old}^* , we obtain:

$$\frac{H_Z}{H_X} = \frac{1 - s_{\text{old}}^*}{s_{\text{old}}^*}. \quad (4.24)$$

After scaling $H_Z \rightarrow rH_Z$, and imposing that the new bottleneck occurs at $s^* = 0.5$, we equate:

$$(1 - 0.5)H_X = 0.5 \cdot rH_Z. \quad (4.25)$$

which simplifies to:

$$H_X = rH_Z. \quad (4.26)$$

Substituting the earlier ratio $H_Z/H_X = (1 - s_{\text{old}}^*)/s_{\text{old}}^*$ into this equation yields:

$$\frac{1}{r} = \frac{1 - s_{\text{old}}^*}{s_{\text{old}}^*}. \quad (4.27)$$

Solving for r , we find the required rescaling factor:

$$r = \frac{s_{\text{old}}^*}{1 - s_{\text{old}}^*}. \quad (4.28)$$

This ensures that the balance between H_X and rH_Z occurs precisely at $s = 0.5$, thereby shifting the bottleneck to the midpoint of the evolution.

Thus, the required scaling factor to shift the adiabatic bottleneck to $s^* = 0.5$ offers a principled way to relocate the adiabatic bottleneck across problem instances. This instance-wise normalization helps standardize the hardness profile of the optimization task and supports the development of transferable schedules or variational parameters [17, 20].

5 Algorithms for solving combinatorial optimization problems

5.1 Variational Quantum Eigensolver

In the field of quantum computing, the Variational Quantum Eigensolver (VQE) is one of the most promising hybrid quantum-classical algorithms for near-term devices. It exemplifies a paradigm shift in quantum algorithm design by embracing the limitations of current quantum hardware, namely, short coherence times and limited qubit connectivity, while exploiting the expressive power of quantum states to approximate solutions to complex problems. Initially introduced for quantum chemistry applications, VQE has since proven to be broadly applicable, including in condensed matter physics and combinatorial optimization [8].

5.1.1 Motivation and Principle

At its core, the VQE algorithm aims to find the ground-state energy of a Hamiltonian, a central problem in physics and adiabatic quantum computing. Solving this eigenvalue problem efficiently is often computationally intractable on classical hardware due to the exponential growth of Hilbert space with system size. For a general k -local Hamiltonian H acting on N qubits, the goal is to find its lowest eigenvalue:

$$E_0 = \min_{|\psi\rangle} \langle \psi | H | \psi \rangle, \quad \text{with} \quad \langle \psi | \psi \rangle = 1. \quad (5.1)$$

Many physically relevant systems exhibit structure that may render them tractable on quantum devices [21]. The VQE leverages the variational principle of quantum mechanics, which states that for any parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle$, the expectation value of the energy

$$E(\boldsymbol{\theta}) = \langle \psi(\boldsymbol{\theta}) | H | \psi(\boldsymbol{\theta}) \rangle \quad (5.2)$$

serves as an upper bound to the true ground-state energy E_0 . The goal is to minimize $E(\boldsymbol{\theta})$ over a set of classical parameters $\boldsymbol{\theta}$ until convergence is reached, yielding an approximation of E_0 and a quantum state close to the true ground state.

5.1.2 Hybrid Quantum-Classical Architecture

VQE is fundamentally a hybrid algorithm, combining the capabilities of quantum and classical processors in a feedback loop. As illustrated in Fig. 4, the workflow begins by preparing a parameterized trial state $|\psi(\boldsymbol{\theta})\rangle$ on a quantum device. The Hamiltonian is typically decomposed into a weighted sum of Pauli strings:

$$H = \sum_{\alpha} h_{\alpha} P_{\alpha}, \quad P_{\alpha} \in \{I, X, Y, Z\}^{\otimes N}. \quad (5.3)$$

The quantum processor estimates the expectation value of each Pauli term $\langle P_{\alpha} \rangle$ by repeatedly preparing the trial state and performing measurements. These expectation values are combined classically to compute the total energy:

$$E(\boldsymbol{\theta}) = \sum_{\alpha} h_{\alpha} \langle \psi(\boldsymbol{\theta}) | P_{\alpha} | \psi(\boldsymbol{\theta}) \rangle. \quad (5.4)$$

This value is minimized using classical optimization routines. The updated parameters are fed back to the quantum device, and the cycle repeats until convergence [22, 21].

Algorithmic Steps

The VQE algorithm proceeds through the following key steps:

1. **Hamiltonian Preparation:** Map the problem of interest (e.g., a fermionic Hamiltonian or classical Ising model) into a qubit Hamiltonian expressible as a sum of Pauli operators.

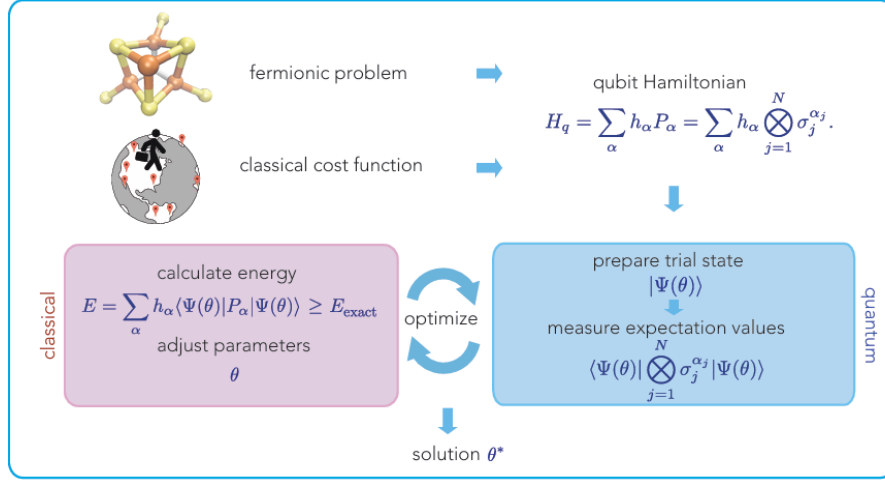


Figure 4: Hybrid quantum-classical architecture of the Variational Quantum Eigensolver. The quantum processor prepares trial states and measures expectation values, while the classical processor calculates the cost function and optimizes the parameters. Adapted from [21].

2. **Ansatz Construction:** Define a parameterized quantum circuit capable of expressing a sufficiently rich subset of Hilbert space. This is often problem-specific and constrained by hardware capabilities. A typical ansatz takes the form:

$$|\psi(\theta)\rangle = U(\theta) |0\rangle^{\otimes N}, \quad (5.5)$$

where $U(\theta)$ is a unitary operator composed of rotation and entangling gates.

3. **Measurement:** Decompose the Hamiltonian into Pauli strings and estimate each term's expectation value:

$$\langle P_{\alpha} \rangle = \langle \psi(\theta) | P_{\alpha} | \psi(\theta) \rangle. \quad (5.6)$$

4. **Cost Function Evaluation:** Compute the energy $E(\theta)$ by summing the measured terms.
5. **Classical Optimization:** Use a classical optimizer (e.g., ZYGOTE, COBYLA) to minimize $E(\theta)$, iteratively updating θ .
6. **Convergence:** Stop when $|E(\theta_{t+1}) - E(\theta_t)| < \epsilon$ or a maximum number of iterations is reached.

Application to fermionic and classical optimization systems

VQE is notably applicable to two broad problem classes: interacting fermionic systems (e.g. molecules) or classical optimization problems (e.g. MaxCut or SK model). In the former, the challenge lies in encoding fermionic operators into qubit representations using transformations like Jordan-Wigner or Bravyi-Kitaev [16]. In the latter, the problem Hamiltonian is typically a classical Ising-type model, requiring fewer qubits and gates. For classical problems, heuristic ansätze inspired by QAOA or problem-specific symmetries are typically used. The parameter landscape of such systems often makes classical optimization particularly challenging and motivates the need for better ansatz initialization and adaptive optimization strategies. For the classical Ising-type systems, the problem Hamiltonian is naturally diagonal in the computational basis.

Measurement and Sampling Considerations

Energy estimation involves computing:

$$E(\boldsymbol{\theta}) = \sum_{\alpha} h_{\alpha} \langle P_{\alpha} \rangle.$$

To reduce measurement cost, Pauli terms are grouped into sets of commuting observables and measured simultaneously. Recent techniques such as classical shadows offer further improvements.

Hardware Constraints and Circuit Design

VQE circuits must respect hardware limitations such as qubit connectivity and gate fidelities. This led to the development of hardware-efficient ansätze, defined by alternating layers of single-qubit rotations and entangling gates:

$$U(\boldsymbol{\theta}) = \prod_{l=1}^d \left(\bigotimes_{i=1}^N R_i^{(l)}(\theta_i^{(l)}) \cdot U_{\text{ent}}^{(l)} \right), \quad (5.7)$$

where R_i are Euler rotations and U_{ent} are fixed entangling gates (e.g., CNOT). These designs minimize circuit depth while maintaining expressibility. However, such circuits can still encounter barren plateaus—regions in the parameter landscape where the gradient norm vanishes exponentially with system size:

$$\mathbb{E} \left[\|\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta})\|^2 \right] \sim \exp(-N),$$

hindering classical optimization. This motivates careful ansatz design and use of parameter initialization schemes.

5.1.3 Ansatz Design in Variational Quantum Eigensolver

The key part of the Variational Quantum Eigensolver is the design of the ansatz the parameterized quantum circuit used to generate trial wavefunctions. The expressive capacity, optimization landscape, and physical alignment of this circuit with the problem structure are all critical to the algorithm's success. Unlike fully quantum algorithms such as *Quantum Phase Estimation* (QPE), VQE relies heavily on classical-quantum co-design, making ansatz engineering a central part of the algorithm [22, 21].

In the context of spin glass models and combinatorial optimization, a key feature of our approach is that the VQE ansatz is explicitly designed with the *instance transformation* in mind, particularly the spin frustration permutation discussed in the previous section. This transformation, which reorders spin labels based on local frustration, reshapes the energy landscape to promote optimization uniformity across problem instances. However, for this transformation to be effective, the ansatz must reflect the identity and ordering of individual spins, something that is typically absent in QAOA.

Limitations of QAOA Parameterization

The standard QAOA employs alternating layers of unitaries generated by a driver Hamiltonian and a problem Hamiltonian. In its canonical form, QAOA introduces only two variational parameters per layer, denoted by β_k and γ_k , which are *global* in nature. They are applied uniformly across all qubits:

$$|\psi_p(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_{k=1}^p e^{-i\beta_k H_X} e^{-i\gamma_k H_Z} |+\rangle^{\otimes N}. \quad (5.8)$$

Because of this uniform parameterization, relabeling the spin indices has no effect on the output of the algorithm. Thus, applying a spin permutation based on frustration has no impact on the learned parameters in QAOA. This limitation prevents QAOA from exploiting instance-specific structure at the spin level and hinders to use the instance transformation effectively for the search for universal parameters across the instances.

Motivation for a Spin-Dependent Ansatz in VQE

To overcome this, we propose VQE ansätze in which the variational parameters are explicitly associated with individual spins. This design makes the ansatz sensitive to spin identity and order, thereby enabling it to fully exploit the effects of the frustration-based spin permutation. Since spins with higher frustration contribute more significantly to the ruggedness of the energy landscape, the ability to allocate expressive resources accordingly is highly advantageous.

This motivates the construction of spin-sensitive VQE ansätze that allow selective tuning of interactions and local fields. The goal is to create parameter structures that reflect the instance transformation and enable universality in parameter initialization or transfer.

Hamiltonian Components: X, Z, and ZZ Terms

A general time-dependent Hamiltonian problem, to be solved by the Variational Quantum Eigensolver, is typically composed of several physically and computationally meaningful terms in the context of combinatorial optimization and quantum variational algorithms. These include the *X-term* (driver), the *Z-term* (local fields), and the *ZZ-term* (pairwise couplings), each of which contributes to the total energy landscape. The combination of these terms defines the cost function whose ground state the algorithm seeks to approximate [3, 23].

X-Term (Driver Hamiltonian) The X-term is commonly referred to as the driver Hamiltonian (Eq. (3.5)) and is responsible for introducing quantum fluctuations into the system. It acts to delocalize the quantum state over the computational basis, facilitating transitions between classical configurations. It is defined as:

$$H_X = - \sum_{i=1}^N \hat{\sigma}_i^x, \quad (5.9)$$

where $\hat{\sigma}_i^x$ denotes the Pauli-X operator acting on qubit i . This term is used to initialize the system in a uniform superposition state $|+\rangle^{\otimes N}$, which is the ground state of H_X . In VQE circuits, its effect is typically implemented through single-qubit rotations of the form $R_X(\theta_i) = \exp(-i\theta_i \hat{\sigma}_i^x)$ [22].

Z-Term (Local Fields) The Z-term captures the effect of local fields or spin biases and contributes linearly to the system's energy. It appears in classical Ising models and maps directly onto combinatorial cost functions such as those found in MaxCut or the SK model. The Z-term is expressed as:

$$H_Z = - \sum_{i=1}^N h_i \hat{\sigma}_i^z, \quad (5.10)$$

where $h_i \in \mathbb{R}$ represents the strength of the local field at site i , and $\hat{\sigma}_i^z$ is the Pauli-Z operator. In variational circuits, this term is implemented via $R_Z(\theta_i) = \exp(-i\theta_i \hat{\sigma}_i^z)$ operations, either with global or spin-specific parameters [8].

ZZ-Term (Pairwise Coupling) The ZZ-term models the two-body interactions between qubits and is critical in capturing spin correlations and frustration. It is the main source of ruggedness in the energy landscape for disordered models like SK. The ZZ interaction term is given by:

$$H_{ZZ} = - \sum_{i < j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z, \quad (5.11)$$

where J_{ij} represents the coupling strength between qubits i and j , typically drawn from a problem-specific distribution (e.g., Gaussian in the SK model). In VQE, this term is implemented using two-qubit entangling gates of the form $\exp(-i\theta_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z)$, which are native to many quantum architectures [21].

Full Hamiltonian The complete problem Hamiltonian combining local fields and couplings is written as:

$$H = H_X + H_Z + H_{ZZ} = -\sum_{i=1}^N \hat{\sigma}_i^x - \sum_{i=1}^N h_i \hat{\sigma}_i^z - \sum_{i<j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z. \quad (5.12)$$

For state preparation and ansatz construction, it is common to alternate unitaries based on these components. For instance, a single-layer variational unitary may take the form:

$$U(\boldsymbol{\theta}) = \exp(-i\theta_X H_X) \exp(-i\theta_Z H_Z) \exp(-i\theta_{ZZ} H_{ZZ}), \quad (5.13)$$

or more generally, with spin- or bond-resolved parameterizations.

VQE Ansatz 1: Frustration-Aligned with $p(N+3)$ Parameters

The first ansatz we consider is designed to match the reshaped energy landscape created by the spin frustration permutation. Each layer includes:

- $N/2$ parameters for the most frustrated spins (X-term),
- 1 shared parameter for the remaining spins (X-term),
- $N/2$ parameters for the most frustrated spins (Z-term),
- 1 shared parameter for the remaining spins (Z-term),
- 1 shared parameter for the ZZ coupling term.

Parameters per layer: $\frac{N}{2}$ (X-term) + $\frac{N}{2}$ (Z-term) + 3 = $N+3 \implies$ Total: $p \times (N+3)$

This ansatz concentrates variational flexibility on the spins that dominate the nontrivial structure of the energy landscape. This targeting is made possible precisely because the spin label now carries physically meaningful information due to the frustration-based permutation. Without spin-specific parameterization, the benefit of reordering would be lost.

VQE Ansatz 2: Fully Localized with $p(2N+1)$ Parameters

The second ansatz maximizes expressibility by assigning each spin its own parameter for both the X and Z terms:

- N spin-specific parameters for the X-term,
- N spin-specific parameters for the Z-term,
- 1 shared parameter for the ZZ coupling term.

Parameters per layer: N (X-term) + N (Z-term) + 1 (ZZ coupling) = $2N+1 \implies$ Total: $p \times (2N+1)$

This ansatz allows maximum resolution of local features and is especially useful for modeling complex instances with localized frustration. Although it introduces a more difficult optimization landscape due to the increased dimensionality, it is particularly powerful when combined with instance-specific transformations.

Connection to Transferability and Universality

A major advantage of frustration-aware VQE ansätze is their potential for enabling *universal parameters* across problem instances. By reshaping diverse instances through frustration-based spin permutations, we can standardize the energy landscape such that similar variational structures emerge. The VQE ansatz, being sensitive to spin identity, allows these universal patterns to manifest as transferable parameter sets. This spin-dependent universality is not achievable with QAOA’s uniform parameter structure. In contrast, the proposed VQE ansätze are designed to leverage the reordering, thus enhancing generalization and performance across varied inputs [8, 22]. To maximize the power of instance transformation—specifically spin frustration permutation, the proposed VQE ansätze are designed to be sensitive to spin identity, enabling them to take full advantage of the transformation. This design principle is crucial for building transferable parameter sets and achieving strong performance across diverse (SK) instances. It marks a fundamental departure from the uniform structure of QAOA and positions VQE as a more flexible and physically grounded approach for solving hard optimization problems on NISQ devices.

5.2 Quantum Approximate Optimization

The Quantum Approximate Optimization Algorithm (QAOA) was first introduced by Farhi et al [3]. The QAOA is inspired by adiabatic quantum computing (AQC), and in particular its application to optimization problems in the context of quantum annealing. The quantum approximate optimization algorithm is a hybrid quantum-classical algorithm that allows for optimizing a cost function and finding approximate solutions. The QAOA can be considered a special case of the VQE of the preceding chapter [16]. The role of the quantum processor in QAOA is to prepare a trial state depending on some parameters. Then, the state is measured several times (upon repeated preparation of the same state), the expectation value of the cost function is computed, and the result is fed into a classical optimiser, which determines how to change the parameters for the next statepreparation stage, so that the expectation value of the cost function will be lowered. QAOA is designed to find approximate solutions to hard combinatorial optimization problems on quantum computers: it encodes the Hamiltonian related to the problem into a quantum circuit and leverages adiabatic time evolution and layering to optimize the variational parameters of the circuit, such that the approximate solution to the problem can be constructed by measuring the QAOA circuit with the optimal set of parameters.

The QAOA is based on the observation that the easiest way to practically implement the quantum annealing Hamiltonian evolution expressed by $H(t) = (1 - s(t))H_M + s(t)H_C$ is to Trotterize it, i.e. to decompose it in small time steps

$$\hat{U}(\tau) \equiv T \exp \left(-i \int_0^\tau \hat{H}(t) dt \right) \approx \prod_{k=1}^p \exp \left(-i \hat{H}(k\Delta t) \Delta t \right). \quad (5.14)$$

Here, $\hat{U}(\tau)$ is the evolution operator from 0 to τ , T is the time-ordering operator, and p is a large integer such that $\Delta t = \tau/p$ is a small time segment.

In QAOA, the evolution schedule is no longer a fixed function of time but is replaced by variational parameters: $(1 - s(t)) \rightarrow \beta$ and $s(t) \rightarrow \gamma$. These become tunable angles in a parameterized circuit. By using the Trotter formula and apply it to the discretized time evolution operator, one receives the corresponding unitary operators:

$$\hat{U}_X(\beta) = e^{-i\beta\hat{H}_X}, \quad \hat{U}_Z(\gamma) = e^{-i\gamma\hat{H}_Z} \quad (5.15)$$

for $k = 1, \dots, p$ with p -layer QAOA.

The QAOA Ansatz is then constructed from p pairs of these unitaries, acting on the initial state of quantum annealing, the plus state:

$$|\beta, \gamma\rangle = \hat{U}_X(\beta_p) \hat{U}_Z(\gamma_p) \cdots \hat{U}_X(\beta_1) \hat{U}_Z(\gamma_1) |+\rangle^{\otimes N}. \quad (5.16)$$

where $\vec{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_p)$ and $\vec{\beta} = (\beta_1, \beta_2, \dots, \beta_p)$ are the classical parameters to be optimized.

From the ansatz state, the expectation value of the cost function, the problem Hamiltonian H_Z , is computed as

$$F_p(\gamma, \beta) = \langle \gamma, \beta | \hat{H}_Z | \gamma, \beta \rangle. \quad (5.17)$$

which serves as the loss function to be optimized, i.e. the aim is to determine the values of γ and β that maximize the expectation value of cost function.

To execute QAOA in practice, one chooses a value for p and initializes the $2p$ parameters. The quantum circuit for $|\vec{\gamma}, \vec{\beta}\rangle$ is constructed and executed multiple times to sample bitstrings z . For each z , the classical cost function $C(z) = \langle z | H_Z | z \rangle$ is evaluated. The average of these over m measurements approximates the expected cost:

$$\frac{1}{m} \sum_{i=1}^m C(z_i). \quad (5.18)$$

The optimizer then updates $(\vec{\gamma}, \vec{\beta})$ and repeats the process. The best sampled bitstring z_{best} is retained along with its corresponding energy value.

The QAOA relies on finding optimal parameter angles γ and β for each instance of a problem. These angles govern the evolution of the quantum state and are crucial for approximating the ground state of the problem Hamiltonian. A key research challenge is the generalization of these angles: identifying a universal set of angles that can be applied across multiple instances of the same problem class, particularly for solving hard combinatorial problems like those represented by the Sherrington-Kirkpatrick model.

5.3 Graph Neural Network

5.3.1 General Theory of Graph Neural Network

Graph Neural Networks (GNN) are a class of machine learning models designed to operate on data structured as graphs. Unlike traditional neural networks which assume a regular, Euclidean domain such as images or sequences, GNNs are capable of leveraging the irregular structure inherent in graph data. This makes them suitable for a wide range of domains. For instance, Battaglia et al. [24] emphasized their use in physics and combinatorics, while Wu et al. [25] provided a thorough survey of their applications in machine learning.

A graph $G = (V, E)$ consists of a set of nodes or vertices V and a set of edges $E \subseteq V \times V$ that connect the nodes. Each node $v_i \in V$ is typically associated with a feature vector $x_i \in \mathbb{R}^d$, and edges $(v_i, v_j) \in E$ may also carry edge-specific features $e_{ij} \in \mathbb{R}^{d'}$. The key idea in GNNs is to propagate information through the graph via a process known as message passing, allowing each node to update its representation based on its neighbors.

The general framework of a GNN layer involves three stages: message computation, message aggregation, and node update. A commonly used formulation is given by:

$$h_i^{(k+1)} = \phi \left(h_i^{(k)}, \square_{j \in \mathcal{N}(i)} \psi \left(h_i^{(k)}, h_j^{(k)}, e_{ij} \right) \right), \quad (5.19)$$

where $h_i^{(k)}$ is the hidden state (embedding) of node i at layer k , $\mathcal{N}(i)$ denotes the set of neighboring nodes of i , ψ is a learnable message function (often an MLP), \square is a permutation-invariant aggregation function such as sum, mean, or max, and ϕ is the update function (often another MLP).

In matrix form, many GNNs adopt a simplified propagation rule that can be described as:

$$H^{(k+1)} = \sigma \left(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(k)} W^{(k)} \right), \quad (5.20)$$

where $\tilde{A} = A + I$ is the adjacency matrix with added self-loops, \tilde{D} is the diagonal degree matrix of \tilde{A} , $H^{(k)}$ is the node feature matrix at layer k , $W^{(k)}$ are learnable weight matrices, and σ is a non-linear activation function like ReLU. This spectral-based propagation is derived from a first-order approximation of the graph Laplacian and underlies Graph Convolutional Networks (GCNs) as described by Kipf and Welling [26].

5.3.2 Multilayer Perceptron

A Multilayer Perceptron (MLP) is a feedforward artificial neural network (ANN) composed of multiple layers of neurons. An MLP consists of at least three layers: the input layer, the output layer, and

one or more intermediate layers. The key distinction of MLPs compared to simpler networks is the presence of these intermediate layers, called *hidden layers*, which are situated between the input and output layers. Since MLPs are fully connected, each neuron in one layer is connected to every neuron in the next layer through a weighted connection. Thus, an MLP can be seen as a stack of fully connected layers, where each layer passes information to the next until the final output layer is reached and an output is generated.

Each input variable affects every neuron in the first hidden layer, and each of these neurons subsequently influences every neuron in the output layer. The operations performed at each neuron in the hidden and output layers can be described as follows:

$$\vec{h}(x) = \vec{s}(\vec{b}_1 + \vec{W}_1 x), \quad (5.21)$$

$$\vec{o}(x) = \vec{G}(\vec{b}_2 + \vec{W}_2 \vec{h}(x)), \quad (5.22)$$

where \vec{b}_1 and \vec{b}_2 are bias vectors, \vec{W}_1 and \vec{W}_2 are weight matrices, and \vec{s} and \vec{G} are activation functions. In the input layer, there is one neuron for each input variable, plus an additional so-called *bias neuron*, which always outputs the constant value one. This bias neuron allows the outputs of subsequent neurons to be shifted by adjusting the bias weight. This is necessary because the output of a neuron is a weighted linear combination of all its inputs. Each hidden layer also contains a bias neuron. The output layer typically consists of a single neuron that represents the overall output of the neural network.

5.3.3 Graph Representation of the SK Model for Optimization

As a paradigmatic example, the SK model can be formulated as a fully connected graph with random couplings. Each vertex represents a spin $\sigma_i \in \{-1, +1\}$, and each edge (i, j) is associated with a coupling strength $J_{ij} \in \mathbb{R}$. The goal is to find the spin configuration that minimizes the SK-Hamiltonian from Eq. (2.3). This is a combinatorial optimization problem over the discrete configuration space $\{-1, +1\}^N$, and can be represented as a graph with node features h_i and edge features J_{ij} . Fig. 5 illustrates how a Graph Neural Network interprets such an SK instance. Each spin is treated as a node, and message passing occurs across couplings to update the spin representations iteratively.

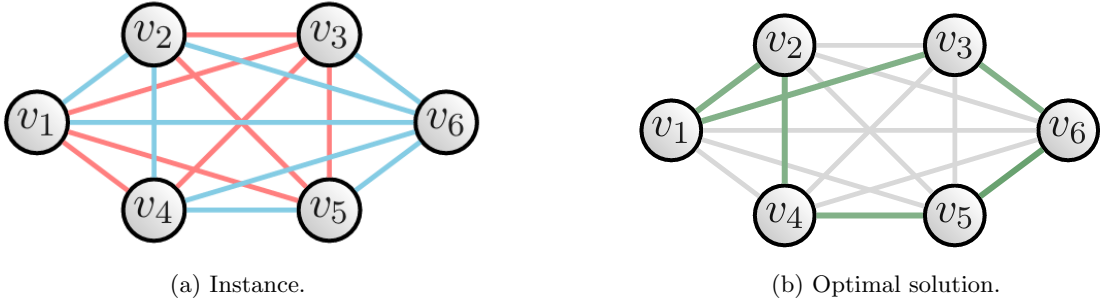


Figure 5: Graph representation of a typical SK instance for GNN processing. **(a)** Each node corresponds to a spin variable $\sigma_i \in \{-1, +1\}$, and each edge encodes a coupling J_{ij} between spins, with red and blue edges indicating negative and positive couplings respectively. **(b)** The corresponding optimal solution, highlighting active spin alignments with green edges. The GNN iteratively propagates messages along the coupling graph to update the spin embeddings and infer the ground-state configuration.

Graph-based representations like this are crucial for applying neural combinatorial optimization techniques to disordered systems like the SK model. GNNs learn to embed each spin by incorporating local coupling and field information, enabling data-driven inference of low-energy states across multiple instances. The figure demonstrates this by showing message propagation from neighboring spins, which is essential to solving highly frustrated, non-local energy landscapes like those found in the SK model.

GNNs differ by the design of their message, aggregation, and update functions. Common GNN architectures include among other things: Graph Convolutional Networks (GCN), which perform a weighted average of neighbor features using a normalized adjacency matrix [26].

Each of these architectures balances expressivity and computational efficiency differently. For example, GCNs are simple and scalable but may suffer from over-smoothing when stacked deeply. GATs offer improved expressiveness by learning attention weights but introduce higher computational complexity.

The expressiveness of GNNs has been linked to the Weisfeiler-Lehman (WL) graph isomorphism test. Xu et al. [27] showed that standard message-passing GNNs are at most as powerful as the 1-WL test. This motivates extensions such as higher-order GNNs or incorporating subgraph structures for improved discriminative power.

GNNs are widely used for tasks including:

- **Node classification:** Predict labels for individual nodes (e.g., user profiling in social networks).
- **Link prediction:** Predict the existence of edges between nodes (e.g., molecular bond prediction).
- **Graph classification:** Predict labels for entire graphs (e.g., molecule property prediction).

In addition to these classical ML tasks, GNNs have recently gained traction in combinatorial optimization. Gasse et al. [28] applied GNNs to learn heuristics for solving constraint satisfaction problems. Liu et al. [29] introduced automated architectures for large-scale optimization. GNNs have also proven useful in physics-informed domains due to their ability to capture local correlations and symmetries.

A key property of GNNs is their invariance to node ordering. This permutation invariance ensures that the output of the model depends only on the graph structure, not on how the nodes are indexed. This is essential for learning meaningful graph-level representations. The flexibility of GNNs allows them to be adapted and extended in various ways. For example, edge features can be included explicitly in the message function ψ , attention weights can be made edge-dependent, and node/edge updates can be conditioned on global graph attributes.

Overall, GNNs provide a powerful, general framework for learning over structured data. In the following section, the specific architectural choices made for our implementation will be detailed and how the GNN is adapted to represent the structure and physics of the optimization problems.

5.3.4 GNN Architecture for SK Model

In this section, we detail the design and training methodology for the GNN used to approximate ground state solutions of the SK model. Our goal is to predict the optimal spin configuration given the couplings J_{ij} and local fields h_i of an instance, thus solving the combinatorial optimization task in a supervised learning setup.

Model Architecture. The implemented GNN model, referred to as **SKGNN**, consists of three core modules:

- **Message Layer:** Implements edge-aware communication between nodes, using a multilayer perceptron (MLP) with `tanh` activation functions. Messages depend on both node features and edge weights.
- **Update Layer:** Updates each node’s embedding using its current state and the aggregated message from its neighbors. Another MLP is used here to refine local information.
- **Readout Layer:** Projects the final node embeddings to a single scalar, which corresponds to the predicted spin value $\sigma_i \in [-1, 1]$. This final activation is `tanh`, which naturally bounds predictions within the spin domain.

Formally, for each node i , the model operates in iterative rounds of message passing:

$$m_i^{(t)} = \sum_{j \in \mathcal{N}(i)} \psi \left(h_i^{(t)}, h_j^{(t)}, J_{ij} \right), \quad h_i^{(t+1)} = \phi \left(h_i^{(t)}, m_i^{(t)} \right), \quad (5.23)$$

where ψ and ϕ are learnable MLPs. After T iterations, the final node embeddings are passed through a readout network to obtain spin predictions:

$$\sigma_i^{\text{pred}} = \tanh \left(\text{MLP}(h_i^{(T)}) \right).$$

Loss Function. The GNN is trained to minimize a loss composed of two parts:

1. **Energy Regression Loss:** Measures the squared error between the predicted energy E (from the predicted spin configuration) and the true ground state energy E_0 :

$$L_{\text{energy}} = (E - E_0)^2. \quad (5.24)$$

2. **Binary Penalty:** Since spins should lie close to $\{-1, +1\}$, we penalize deviations from these ideal values with:

$$L_{\text{binary}} = \frac{1}{N} \sum_{i=1}^N \left(|\sigma_i^{\text{pred}}|^2 - 1 \right)^2.$$

The total loss is a weighted sum:

$$L_{\text{total}} = L_{\text{energy}} + \lambda L_{\text{binary}}, \quad \text{with } \lambda = 0.1.$$

5.4 Evaluating the Solution Probabilities

To evaluate the effectiveness of VQE, we define the relative energy deviation:

$$\Delta C = \frac{E_* - E_0}{E_0}, \quad (5.25)$$

where E_* is the expected energy from VQE and E_0 is the ground-state energy. A value of $\Delta C = 0$ indicates that VQE has exactly reached the ground state.

Since each instance of a combinatorial problem (e.g., the SK model) has different couplings J_{ij} and fields h_i , it is crucial to investigate the generalization of QAOA/VQE parameters across instances. For this, parameters are first optimized per instance using gradient ascent (e.g., via Zygote.jl). These optimal parameters $(\vec{\gamma}^*, \vec{\beta}^*)$ can then be averaged:

$$\bar{\gamma} = \frac{1}{N} \sum_{i=1}^N \gamma_i^*, \quad \bar{\beta} = \frac{1}{N} \sum_{i=1}^N \beta_i^*, \quad (5.26)$$

where N is the number of training instances. These averaged values provide a universal initialization strategy.

To assess the performance of QAOA and VQE, the following procedure is applied to analyze the bitstrings generated:

Probabilities of Bitstrings: For each possible bitstring b , compute its probability p_i in the output state.

$$p_i = |\langle b | \psi_{\text{QAOA}} \rangle|^2,$$

where $|\psi_{\text{QAOA}}\rangle$ is the final state of the QAOA(VQE) circuit.

Identify the Most Probable Bitstring: Find the bitstring b_{max} with the highest probability.

$$b_{\text{max}} = \arg \max_i p_i.$$

Index of the Energy State: Define the index of b_{max} as the position of its corresponding energy state.

$$\text{Index} = \begin{cases} 0, & \text{if } b_{\max} \text{ corresponds to the ground state,} \\ 1, & \text{if } b_{\max} \text{ corresponds to the first excited state, and so on.} \end{cases}$$

Ensemble Averaging: Compute the average index over the entire ensemble of instances to evaluate the algorithm’s performance.

$$\langle \text{Index}_{state} \rangle = \langle \arg \max_i p_i \rangle_{ENS}. \quad (5.27)$$

$$\langle \text{Index} \rangle = \frac{1}{N} \sum_{k=1}^N \text{Index}_k, \quad (5.28)$$

where N is the number of problem instances.

6 Results

The preceding chapters established the theoretical foundation and methodology for quantum optimization using QAOA and VQE. In this chapter, we now present and analyze the original results obtained in the course of this thesis. The focus lies on evaluating the effect of instance transformation on variational methods, comparing their scalability and effectiveness in recovering ground states. This section presents the outcomes of our comprehensive investigation into the universal parameterization of Variational Quantum Eigensolver (VQE) and Quantum Approximate Optimization Algorithm (QAOA) methods when applied to Sherrington-Kirkpatrick and MAX2SAT combinatorial optimization problems. Utilizing mean-field transformations, we aim to simplify the complex energy landscapes inherent in these NP-hard problems, thereby enhancing the performance and robustness of quantum optimization techniques. We systematically explore eigenenergy distributions, mean-field landscape transformations, performance metrics of optimization algorithms, and analyze optimized variational parameters. We directly compare the performance of VQEs with mean-field transformed instances against standard QAOA implementations. The detailed results provided herein offer insights into how mean-field inspired transformations impact algorithmic efficiency and highlight pathways towards achieving universal parameterization for practical quantum computing implementations. Finally, we present the results obtained from the Graph Neural Network and compare its performance to that of the VQE approach.

The complete source code used for the implementation and analysis of the VQE and QAOA methods discussed in this thesis are available in the open-access repository [30].

6.1 Eigenenergy Spectra and the Hardness of the Instances

Understanding the energy spectra and adiabatic gap behavior provides critical insights into the complexity and hardness of quantum optimization instances. Here we examine typical instances of the Sherrington-Kirkpatrick model and MAX2SAT problems, illustrating their distinct characteristics and challenges.

To analyze a representative hard instance, we compute the spectrum of the time-dependent Hamiltonian $H(s) = (1 - s)H_X + sH_Z$, where H_X is the transverse driver Hamiltonian and H_Z encodes the problem instance. By doing that, the point in time when the energy gap closes can be determined.

The instantaneous eigenvalues are evaluated using the ARPACK library, which provides an efficient method for computing a subset of extremal eigenvalues using iterative Krylov subspace techniques. Although ARPACK does not directly return all eigenvalues, it can reliably approximate the lowest few states by targeting eigenvalues of largest magnitude.[31] By computing at several points along the annealing schedule $s \in [0, 1]$, we reconstruct the energy spectrum and locate the minimal gap region. To ensure accuracy, we verified that the computed spectrum was smooth and continuous.

Sherrington-Kirkpatrick Instance

For the SK model, instances are characterized by complex and rugged energy landscapes resulting from randomly distributed spin interactions. We focus specifically on instances classified as "hard", as determined by their minimal adiabatic gaps [17]. The hard instances exhibit characteristics that render it difficult for both mean-field (MF) methods and linear quantum annealing. In both cases, optimization fails to converge to the correct solution, underscoring the spectral complexity and degeneracy typical of hard SK instances. Fig. 6 shows the energy spectra of a representative hard SK instance. The upper panel illustrates the exact eigenenergies (black lines) and their corresponding Bogoliubov approximations (dashed gray lines), with the ground state energy E_0 subtracted. A clear clustering near the lowest energy levels indicates the proximity of multiple low-energy excited states. The lower panel plots $|\langle \alpha | H_Z - H_X | 0 \rangle|$ of the the adiabatic ratio for the first excited states, highlighting the location and severity of the adiabatic bottleneck. The sharp peak in the line marks the critical point s^* at 0.8, where the adiabatic gap becomes minimal and the system transitions from paramagnetic to spin-glass behavior.

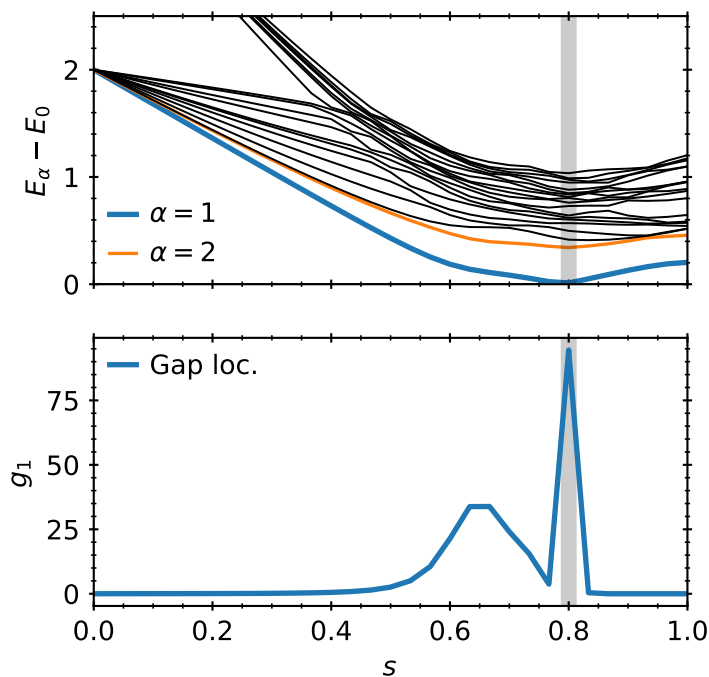


Figure 6: Upper panel: (Sherrington-Kirkpatrick) Eigenenergy spectrum $E_\alpha - E_0$ of a hard SK instance during adiabatic evolution. Lower panel: the critical point of the minimal adiabatic gap highlighted for $\alpha = 1$. The minimum gap occurs around $s(t) = 0.8$. This narrow gap region governs the difficulty of adiabatic quantum evolution.

To benchmark the correctness of approximate solutions from variational algorithms and neural predictors, we compute the ground state energy directly using brute-force enumeration over all 2^N spin configurations. For small sizes such as $N = 10$, this is feasible. The Tab. 1 lists the ten lowest-energy bitstrings for the selected instance, along with their associated energies:

These configurations provide a ground truth for assessing model quality. They also reveal a high level of degeneracy near the minimum energy, which contributes to the difficulty of optimization. As shown in the next sections, such hardness manifests in the spread and convergence behavior of QAOA, VQE, and neural approaches. The hard SK instances analyzed in this thesis are provided in Ref. [32].

Bitstring $\vec{\sigma}$	Energy $E(\vec{\sigma})$
[1, 1, 1, -1, 1, 1, 1, -1, 1, -1]	-6.50991
[-1, -1, -1, 1, -1, -1, -1, 1, -1, -1]	-6.3269
[1, -1, -1, 1, -1, -1, -1, 1, -1, -1]	-6.20022
[-1, -1, -1, 1, -1, -1, -1, 1, -1, 1]	-6.16702
[-1, -1, -1, 1, -1, -1, -1, -1, -1, -1]	-5.86399
[1, 1, 1, -1, 1, -1, 1, -1, 1, -1]	-5.80874
[-1, -1, -1, 1, -1, 1, -1, 1, -1, 1]	-5.72245
[1, -1, -1, 1, -1, -1, -1, -1, -1, -1]	-5.30015
[1, 1, 1, -1, 1, 1, 1, -1, 1, 1]	-5.27678
[[1, -1, -1, 1, -1, -1, -1, 1, -1, 1]	-5.24395

Table 1: Lowest-energy bitstrings at $s = 1$ for a hard SK instance of size $N = 10$ computed via brute-force.

MAX2SAT Instance

In parallel to the SK example previously discussed, we analyze a particularly challenging instance of the MAX2SAT problem selected from the dataset provided in Ref. [33]. It is important to note that in contrast to the SK dataset, where only instances with minimal adiabatic gaps were selected, the MAX2SAT dataset includes a mixture of soft and hard instances. As discussed in Ref. [33], MAX2SAT problems are not maximally difficult compared to the SK problem. This mixture of instances explains, at least in part, the comparatively stronger performance of classical QAOA on MAX2SAT, which will be shown throughout the evaluation later.

Fig. 7 illustrates the spectral structure of the instance. The upper panel presents the instantaneous energy spectrum calculated using ARPACK, where the 20 lowest eigenvalues are computed throughout the evolution. The lower panel shows the adiabatic ratio $g_1(t)$, as defined in Eq. (3.2), providing a metric for the severity of the adiabatic bottleneck. As shown, the minimal gap occurs near $s(t) \approx 0.47$, signaling a critical point in the evolution where transition probabilities between the ground and first excited states become non-negligible.

A convenient feature of the dataset from Ref. [33] is that all solutions are uniquely characterized by the all-minus configuration $\vec{\sigma} = [-1, -1, \dots, -1]$. This corresponds to setting all literals in the original SAT formula to false. For example, an instance of size $N = 10$ would have the exact solution $\vec{\sigma}_{\text{true}} = [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1]$, and similarly for larger sizes.

To assess the degeneracy and structure of low-energy configurations, we compute the energies of all 2^N bitstrings via brute-force enumeration. Tab. 2 shows the 10 configurations with the lowest energy values for this specific instance. While the all-minus configuration achieves the global minimum, numerous near-degenerate configurations lie just above it, differing by only a few spin flips.

Bitstring $\vec{\sigma}$	Energy $E(\vec{\sigma})$
[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1]	-22.0
[1, -1, -1, 1, 1, -1, 1, 1, 1, 1]	-18.0
[1, -1, -1, 1, 1, -1, -1, 1, 1, 1]	-18.0
[1, -1, -1, 1, 1, -1, 1, -1, 1, 1]	-18.0
[1, -1, -1, 1, 1, -1, -1, -1, 1, 1]	-18.0
[1, -1, -1, 1, -1, -1, -1, -1, 1, 1]	-18.0
[-1, -1, -1, 1, 1, 1, 1, 1, 1, -1]	-18.0
[-1, -1, -1, -1, 1, 1, 1, 1, 1, -1]	-18.0
[-1, -1, -1, 1, 1, 1, -1, 1, 1, -1]	-18.0
[-1, -1, -1, 1, 1, 1, 1, -1, 1, -1]	-18.0

Table 2: Lowest-energy bitstrings at $s = 1$ for a MAX2SAT instance of size $N = 10$, computed via brute-force. The ground state corresponds to the all-minus configuration. Several energetically close excitations form a dense subspace just above the minimum.

These results confirm that the hardness of MAX2SAT arises not from the ruggedness of the energy

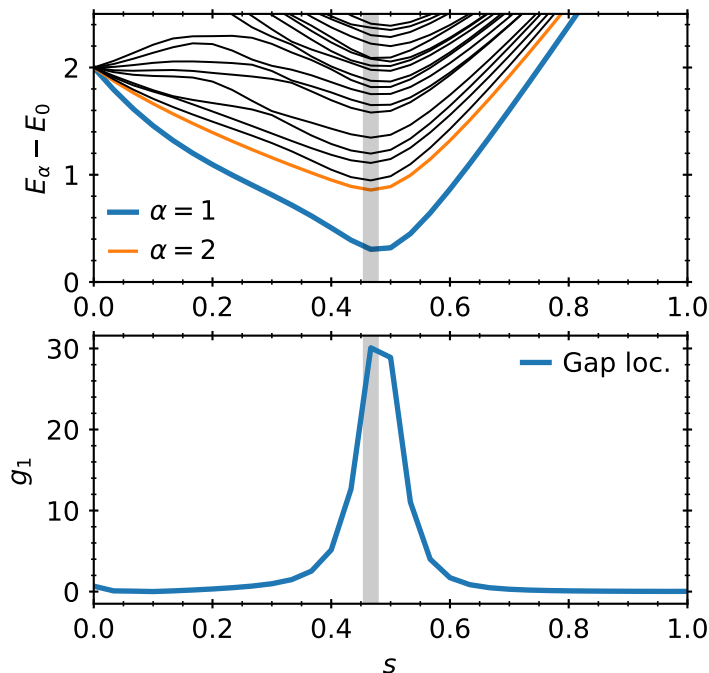


Figure 7: (MAX2SAT) Eigenenergy spectrum (top) and adiabatic ratio $g_1(t)$ (bottom), computed for the 20 lowest eigenstates using ARPACK. The minimum gap occurs around $s(t) \approx 0.47$, indicating the most non-adiabatic region during the evolution.

landscape—as in SK—but from its spectral degeneracy and dense low-energy manifold. Quantum optimization algorithms must therefore resolve between many nearly indistinguishable local minima to identify the true ground state. This property makes MAX2SAT instances particularly challenging in both adiabatic and variational settings.

6.2 Mean-Field Analysis and Transformation of Instances

The mean-field approximation provides a semiclassical perspective of the system’s evolution. Fig. 8 shows the dynamics of the spin projections $n_i^x(s)$ and $n_i^z(s)$ for each spin i in a representative SK instance. The orange line highlights spin $i = 11$, identified as the most frustrated spin due to its deviation from polar alignment near the critical point $s^* \approx 0.8$.

The lower panel of the Fig. 8 also plots the effective magnetizations $m_i(s)$, with spin $i = 11$ again highlighted. Its magnetization stays close to zero for a long time before diverging, marking it as a critical indicator of frustration and bottlenecks.

To improve optimization performance and enable universal parameterization, we investigate instance transformations that reshape or reinterpret the quantum landscape without altering the core problem structure by using those Mean-Field Analysis. These transformations are motivated by the desire to either simplify the energy profile or align the representation of spins in a way that enhances the generalizability of variational parameters. In particular, we explore two complementary strategies:

- **Adiabatic Gap Shift:** A method for relocating the critical point of the evolution, where the adiabatic gap is minimal, toward a region of the annealing schedule that is more favorable for optimization.
- **Spin Label Permutation:** A purely structural transformation that reorders spins based on their mean-field frustration signature. This reordering aims to create more consistent mappings across instances for learning-based parameter generation.

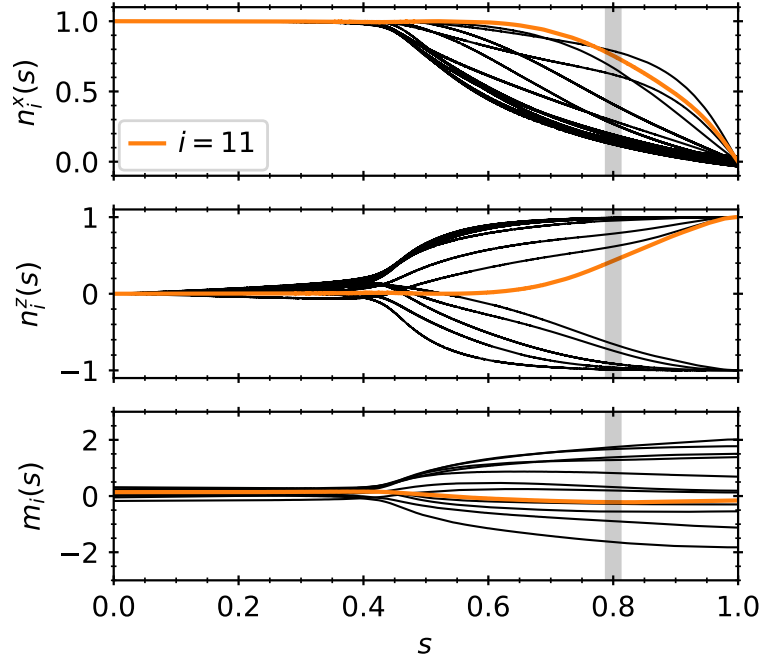


Figure 8: Upper panel: Mean-field dynamics of $n_i^x(s)$ (top) and $n_i^z(s)$ (bottom) for an example SK instance. The orange line tracks the most frustrated spin ($i = 11$), which remains in the equatorial plane until the system passes the critical point. Lower panel: Effective magnetization $m_i(s)$ for all spins. The most frustrated spin ($i = 11$, orange) shows minimal magnetization throughout the evolution until close to s^* , making it a signature of the bottleneck region.

6.2.1 Spin Label Permutation after Frustration Ordering

This transformation strategy involves reordering the spins based on their cumulative local order parameter, effectively permuting spin labels according to their frustration ranking. Importantly, this transformation does not affect the actual dynamics. Despite relabeling, the dynamical evolution of physical quantities, magnetization $m_i(s)$ and order parameter $q_i(s)$ remains unchanged. This invariance is illustrated by comparing the plots before and after transformation in Fig. 9a, spin $i = 11$ is tracked; after permutation, the same trajectory appears for spin $i = 1$, shown in Fig. 9b. The cumulative local order parameter $Q(i)$ is used to rank spins by frustration, and the permutation aligns the spin indices accordingly. This confirms that the transformation is a symmetry operation that preserves the physical trajectory and energy structure while enabling consistent instance alignment for universal parameter training.

This type of transformation preserves the system’s energy structure while aiding in more universal parametrization by aligning spin order with instance difficulty. The practical significance of this spin relabeling becomes clear when considering how quantum algorithms process problem instances. By arranging spins according to their frustration levels, with the most frustrated spins receiving the lowest indices, we effectively create a consistent ‘difficulty ranking’ that algorithms can learn to handle systematically. This is particularly important for VQE’s spin-specific parameterization, where the ansatz can allocate more resources to early spins (now guaranteed to be the most frustrated) through dedicated parameters. The invariance of physical quantities under this transformation ensures we’re not simplifying the problem artificially, but rather presenting it in a form that better matches the capabilities of near-term quantum devices.

6.2.2 Adiabatic Gap Shift

The result of another transformation strategy is introduced here. Applying a transformation to shift the critical point of the adiabatic gap to $s^* = 0.5$ alters the dynamics. In Fig. 9, the mean field trajectory is presented after applying spin label permutation and this gap-shift transformation.

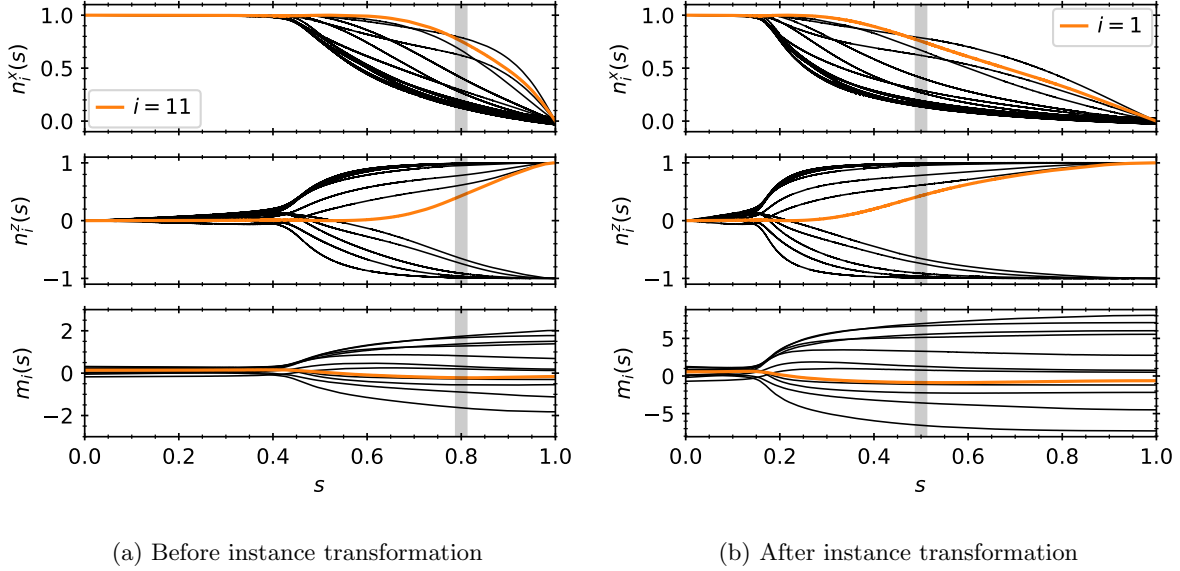


Figure 9: Mean-field dynamics of $n_i^x(s)$ and $n_i^z(s)$ after applying the spin label permutation and adiabatic gap shift transformation. The most frustrated spin ($i = 11 \rightarrow i = 1$) now exhibits critical behavior closer to $s = 0.5$. Effective magnetization $m_i(s)$ after adiabatic gap shift. The minimal magnetization region is moved to around $s = 0.5$, indicating successful relocation of the bottleneck.

A comparison of energy spectra before (Fig. 6) and after the transformation is shown in Fig. 10. While the overall spectrum remains unchanged, the position of the minimal gap shifts, confirming the effectiveness of the transformation.

This result demonstrates that targeted instance transformations can be used to steer the critical region of quantum dynamics into more favorable regimes. In particular, relocating the minimal gap to the center of the evolution range enables more symmetric and potentially more robust annealing schedules. Moreover, such transformations can be applied without altering the fundamental complexity of the problem, as the spectrum itself is preserved. Altogether, this approach exemplifies how physical insight into a system's structure, combined with mean-field analysis, can inform the design of instance transformations that systematically improve quantum optimization performance.

6.3 Optimized Parameter Analysis for QAOA and VQE

To understand the structure and universality of optimal variational parameters, we perform a detailed statistical analysis of optimized QAOA and VQE parameters for solving the SK model. For this, a optimization of the parameter angles must be carried out. We select a total of 100 hard SK instances and extract optimized parameters for both methods.

6.3.1 Optimization Procedure

The optimization is conducted in two main phases:

1. **Preliminary Optimization on Subset:** We begin by selecting 20 SK instances and perform QAOA/VQE optimization on each using a grid of initial parameters. For each instance, the initial parameter set yielding the best cost value is selected.

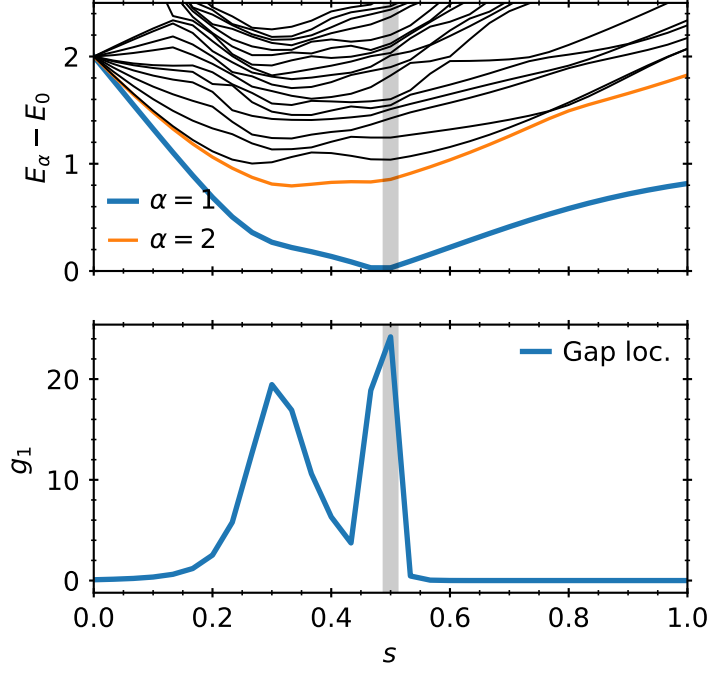


Figure 10: Adiabatic energy spectrum after shifting the gap to $s^* \approx 0.5$. Compared to the untransformed spectrum in Fig. 6, the gap location has been successfully modified without altering the spectrum’s shape.

2. **Batch Optimization on Full Ensemble:** The best-performing initial parameters from the first phase are used as a starting point for optimizing all 100 SK instances. The final optimal parameter for each instance is stored.
3. **Batch Optimization on Full Ensemble:** The best-performing initial parameters from the first phase are used as a starting point for optimizing all 100 SK instances. The final optimal parameter for each instance is stored.

We compute the average of each optimized parameter across all 100 instances. These averaged values form our final set of universal parameters for both QAOA and VQE.

To ensure efficient convergence, careful selection of starting parameters is essential. For the Variational Quantum Eigensolver (VQE), we conduct structured sweeps over initial angles and learning rates. The different VQE strategies and corresponding hyperparameters are summarized below.

VQE Hyperparameter Configuration

Phase	Instances	n_iter opt.	lr	lr step size	Start parameters	Param. step size
1 (coarse sweep)	20	100	0.001	+0.01 (3 steps)	0.01	+0.1 (10 steps)
2 (fine-tune)	100	100	0.0001	+0.001 (2 steps)	Best from Phase 1	-
3 (final)	100	0	-	-	Best from Phase 2	-

Table 3: Initial parameter sweep for VQE SK

Tab. 3 outlines the VQE training for the SK model using the original ansatz. The process begins with a coarse sweep (Phase 1) over 20 SK instances. The learning rate is initialized at 0.001 and increased in steps of +0.01 for three different values. Simultaneously, the circuit parameters are

initialized from 0.01 and swept in steps of +0.1 over 10 values. Each combination is evaluated over 100 optimization steps, and the best-performing configuration (in terms of energy minimization) is selected for the next phase.

In Phase 2, the selected configuration is fine-tuned on a larger set of 100 SK instances. The learning rate is refined to a smaller range starting from 0.0001 with two increments of +0.001. This phase helps to determine optimal convergence rates in the vicinity of the best initialization. Phase 3 serves as a final validation step where no further optimization is applied, applying a universal parameterization approach using the best configuration found during Phase 2.

Phase	Instances	n.iter opt.	lr	lr step size	Start parameters	Param. step size
1 (coarse sweep)	20	100	0.02	+0.02 (3 steps)	0.5	+0.1 (6 steps)
2 (fine-tune)	100	100	0.0001	+0.001 (2 steps)	Best from Phase 1	-
3 (final)	100	0	-	-	Best from Phase 2	-

Table 4: Initial parameter sweep for VQE v2 SK

Tab. 4 shows the hyperparameters used for the optimization of a modified VQE ansatz (v2) applied to the SK model. Compared to the original VQE, the initial learning rate is higher, starting at 0.02 and increased by +0.02 for three steps. The starting parameter values begin at 0.5 and are varied over 6 steps of +0.1 each. The optimization and evaluation phases follow the same logic and structure as the original variant, facilitating a direct comparison of ansatz expressiveness and robustness.

Phase	Instances	n.iter opt.	Learning rate	Start parameters	Param. step size
1 (coarse sweep)	20	100	0.001	0.1	+0.03 (10 steps)
2a (fine-tune)	100	100	0.001	Best from Phase 1	-
2b (fine-tune)	100	100	0.005	Best from Phase 1	-
3 (final)	100	0	-	Best from Phase 2	-

Table 5: Initial parameter sweep for VQE MAX2SAT

Tab. 5 outlines the training configuration for the MAX2SAT problem using the original VQE ansatz. In Phase 1, a grid of initial parameters is generated starting from 0.1, incremented by +0.03 over 10 steps. A constant learning rate of 0.001 is used during this coarse sweep, which is conducted over 20 instances. In Phase 2, the best parameter set is fine-tuned over 100 instances using two learning rates: the original 0.001 (Phase 2a) and a slightly increased rate of 0.005 (Phase 2b), allowing sensitivity analysis of convergence. The configuration with the lowest energy outcome across 2a and 2b is adopted for Phase 3, which is used for fixed evaluation on 100 unseen instances.

Phase	Instances	n.iter opt	Learning rate	Start parameters	Param. step size
1 (coarse sweep)	20	100	0.005	0.3	+0.1 (10 steps)
2a (fine-tune)	100	100	0.005	Best from Phase 1	-
2b (fine-tune)	100	100	0.01	Best from Phase 1	-
3 (final)	100	0	-	Best from Phase 2	-

Table 6: VQE version 2 MAX2SAT: Coarse parameter sweep and fine-tuning

The optimization flow for the second ansatz applied to MAX2SAT is provided in Tab. 6. This variant uses a more aggressive initialization with a learning rate of 0.005 and starting parameters from 0.3 in steps of +0.1 across 10 values. As in the other cases, the best configuration from the coarse sweep (Phase 1) is fine-tuned in two separate sub-phases: 2a retains the original learning rate (0.005), and 2b increases it to 0.01. The most effective configuration is selected for Phase 3, in which performance is evaluated without further updates.

These hyperparameter strategies are designed to balance exploration (through parameter sweeps) and exploitation (through finer tuning), ensuring robust convergence across both small and large SK instance ensembles.

QAOA Hyperparameter Configuration

The Quantum Approximate Optimization Algorithm (QAOA) requires a careful selection of two angle parameters (β and γ) per layer to approximate the ground state of the problem Hamiltonian. Given the non-convex optimization landscape and the sensitivity of performance to initial parameters, a multi-phase hyperparameter tuning strategy is used. This section outlines the parameter search protocols applied to the SK model and the MAX2SAT problem.

Phase	Instances	n_iter opt.	lr	lr step size	Init. param. range	Step Size β, γ
1 (coarse sweep)	20	100	0.006	-	$\beta, \gamma = 0.0$ to 0.5	+0.1 (25 steps)
2 (fine-tune)	100	100	0.006	+0.003 (4 steps)	Best from Phase 1	-
3 (final)	100	0	-	-	Best from Phase 2	-

Table 7: Initial parameter sweep for QAOA SK

Tab. 7 presents the hyperparameter configuration used to optimize QAOA parameters for the SK model. The optimization is conducted in three distinct phases. In Phase 1, a coarse grid search is performed over 20 SK instances. Each parameter set consists of a sequence of QAOA angles (β, γ) sampled from a uniform grid in the range $[0.0, 0.5]$ for each parameter angle, incremented by +0.1 for a total of 25 steps. Each instance is optimized over 100 iterations using a fixed learning rate of 0.006.

The best-performing parameter set from Phase 1—determined by energy minimization—is selected and used as the starting point for Phase 2. This phase conducts a fine-tuning of the learning rate by sweeping around the initial value with four additional steps of +0.003 while maintaining the same angle configuration. The optimization is scaled up to 100 SK instances to improve generalization. In the final phase (Phase 3), the best configuration from Phase 2 is used as a static set of angles across all instances for evaluation, thereby testing the universality of the optimized parameters.

Phase	Instances	n_iter opt.	lr	lr step size	Init. param. range	Step Size β, γ
1 (coarse sweep)	20	100	0.0005	-	$\beta, \gamma = 0.0$ to 0.5	+0.1 (25 steps)
2 (fine-tune)	100	100	0.0001	+0.0005 (2 steps)	Best from Phase 1	-
3 (final)	100	0	-	-	Best from Phase 2	-

Table 8: Initial parameter sweep for QAOA MAX2SAT

Tab. 8 outlines the optimization strategy for the MAX2SAT problem using QAOA. As in the SK model case, Phase 1 starts with a coarse parameter sweep across 20 instances. The angles (β, γ) are initialized from a uniform grid in $[0.0, 0.5]$ with a step size of +0.1, again producing 25 candidate parameter vectors. Each configuration is optimized for 100 steps using a learning rate of 0.0005.

In Phase 2, the best parameter vector from the coarse sweep is refined over 100 MAX2SAT instances. The learning rate is lowered to 0.0001 and explored further by two increments of +0.0005, allowing for a more stable convergence. Finally, in Phase 3, the top-performing parameter configuration from the fine-tuning phase is used without further updates across all problem instances.

6.3.2 QAOA Parameter Behavior

For QAOA, each of the p layers consists of one β and one γ parameter. Thus, the total parameter count is $2p$. In our analysis with $p = 20$, we observe a clear structure:

- The average β values decay monotonically with the layer index.
- The γ parameters show a complementary increasing trend.

This behavior is visualized in Fig. 11, confirming the smooth and interpretable structure of the QAOA ansatz. The error bars reflect the standard deviation across the 100 instances, showing low variance and indicating a high degree of universality in the optimized parameters.

Individual QAOA parameter profiles for each instance are shown in Fig. 12, further illustrating the consistency of the overall structure across different samples. Most parameter curves follow a similar

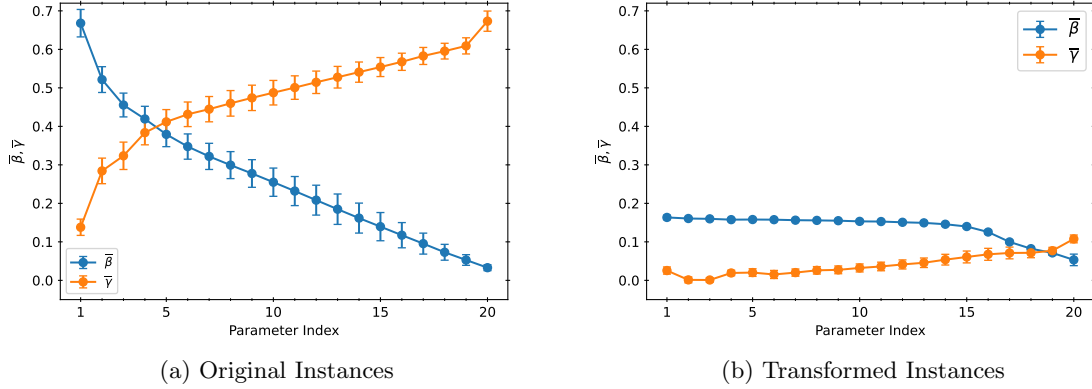


Figure 11: Comparison of average QAOA parameters β_k and γ_k over 100 SK instances for $p = 20$. For (a) original instances, the QAOA angles display a clear structural trend: β_k values decrease while γ_k increase across layers, reflecting the sequential encoding of mixing and cost Hamiltonians. For (b) transformed instances, the parameter curves become markedly smoother with reduced curvature and tighter variance, particularly in β_k . This smoothing is a direct consequence of applying the adiabatic gap shift transformation, which realigns the critical bottleneck to $s^* = 0.5$ and normalizes the annealing landscape across samples. The more uniform parameter landscape for transformed instances indicates enhanced generalizability and supports the notion of universal parameterization in QAOA.

symmetric pattern around the center layer, supporting the intuition that a fixed optimal template may exist for certain problem classes.

Fig. 11b shows the average parameters for the transformed SK instances using mean-field-informed gap shifting. The resulting parameter curves are significantly flatter and more homogeneous compared to the original instances. This smoother structure implies that the transformed instances are easier to optimize and admit simpler, more transferable parameter profiles.

The spread of QAOA parameters across the 100 transformed instances is presented in Fig. 12b. Unlike the original instances, the spread here is tightly concentrated around the average curve. This visual evidence reinforces the hypothesis that transformation enhances parameter regularity across problem instances.

6.3.3 VQE Parameter Behavior

The VQE ansätze used are described in Section 5.1.3 earlier. We consider two ansatz types:

- Ansatz 1 with $p(N + 3)$ parameters: assigns unique parameters to the most frustrated spins and shares parameters among others.
- Ansatz 2 with $p(2N + 1)$ parameters: assigns individual parameters to every spin for both X and Z terms.

Both VQE parameter profiles reflect the internal complexity of the ansatz and show potential for universality, particularly under appropriate instance transformations. The QAOA’s parameter structure shows a clear, interpretable form and becomes more universal across instances when transformations are applied. The transformation leads to both smoother curves and narrower distributions, suggesting more robust and generalizable parameters.

In contrast, VQE demonstrates more expressive power due to its finer parameter granularity. The key observation here is that VQE’s spin-specific parameters allow the ansatz to adapt to the problem’s structure in ways that QAOA’s uniform parameters cannot. This adaptability comes at a cost - more parameters and potentially harder optimization - but offers a path to better performance, especially when combined with the spin permutation transformation that ensures consistent ordering of difficult spins. Both VQE parameter profiles reflect the internal complexity of the ansatz and show potential for universality, particularly under appropriate instance transformations.

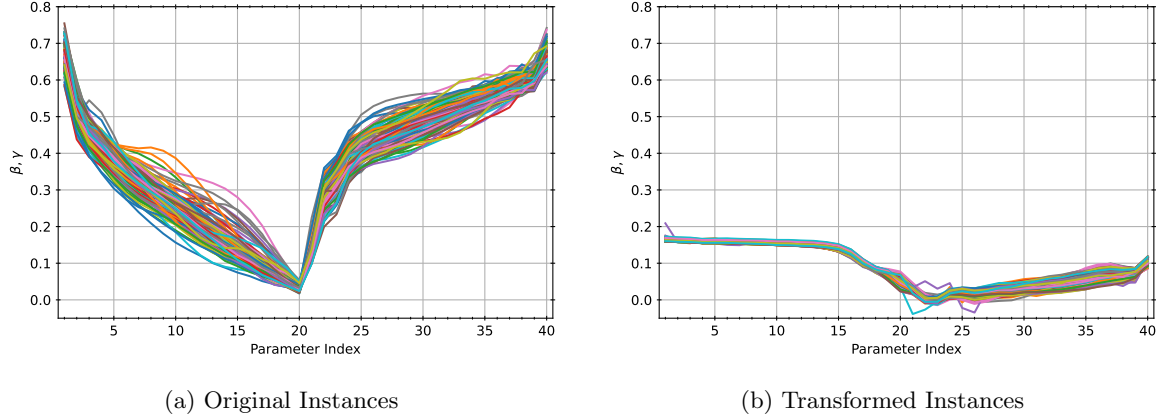


Figure 12: Instance-wise QAOA parameters (β_k, γ_k) for 100 different SK instances at depth $p = 20$. In (a) original instances, although each trajectory varies in scale, the general pattern, a characteristic "V"-shaped dip in the mid-layers, is consistently observed. This highlights a form of emergent universality across different problem realizations. For (b) transformed instances, the parameter trajectories exhibit significantly less variation across instances. This narrower spread illustrates enhanced alignment of optimal parameters, supporting the notion that the adiabatic gap shift transformation effectively standardizes the problem structure. The increased regularity facilitates the construction of transferable QAOA schedules, suggesting greater robustness and generalizability across instance ensembles.

Together, these results indicate that both QAOA and VQE exhibit promising universal trends in their optimal parameters, and that instance transformation is a powerful tool for enhancing parameter transferability and reducing optimization variance.

6.4 Performance Analysis of QAOA and VQE

To evaluate the efficacy of the optimized parameters discussed previously, we now investigate the cost performance and state index statistics across increasing circuit depths p for both QAOA and VQE, using SK and MAX2SAT instances. Our analysis is performed for a fixed spin size $N = 12$ and compares different transformations and ansatz structures. Both cost-related metrics and sampling-based indicators are analyzed to give a comprehensive performance picture.

The selection of evaluation metrics is critical for interpreting variational algorithm performance. The use of the cost difference provides a quantitative and continuous assessment of how close the variational state is to the true ground state. This metric is particularly important in the optimization process, as it directly reflects the objective function being minimized.

The average state index, while a discrete and coarser measure, provides insight into the output distribution generated by the variational circuit. It captures whether the algorithm tends to sample low-energy states even if the exact ground state is not always found. However, for MAX2SAT problems, this metric becomes less informative due to the highly degenerate nature of the low-energy spectrum (see Tab. 2), many states may have identical energies. In such cases, the average state index may fail to distinguish between good and poor performance, prompting us to instead rely on the ground state success probability P_{success} , which gives a more meaningful assessment of exact ground state targeting.

6.4.1 Cost Value and Average Index versus Number of Layers p for Fixed Spin Size N

The cost value is defined as the deviation between the optimized expectation value $\langle \psi(\vec{\theta}) | H | \psi(\vec{\theta}) \rangle$ and the true ground state energy E_0 . It provides a direct numerical criterion for assessing the algorithm's ability to approximate the ground state. Lower cost differences are indicative of better convergence. In Fig. 13, we present a comparison of the cost value ΔC for the VQE Ansatz 1 method on different MAS2SAT instances: one using only spin-label permutation and another incorporating the both mean field-based transformations (spin-label permutation and gap-shift). Across all depths p , the full transformation-enhanced VQE shows superior performance. In particular, the cost difference drops

rapidly and saturates at $p = 8$ for the permutation-only case, whereas the cost values of transformed variant drops slowly first but continues to improve more for increased layers. This is consistent with the hypothesis that transformation, which shifts the adiabatic gap to an unified critical point, is more favorable for optimization for the MAX2SAT instances. Based on these results, we will proceed the further investigation on the MAX2SAT problem with the full transformed instances.

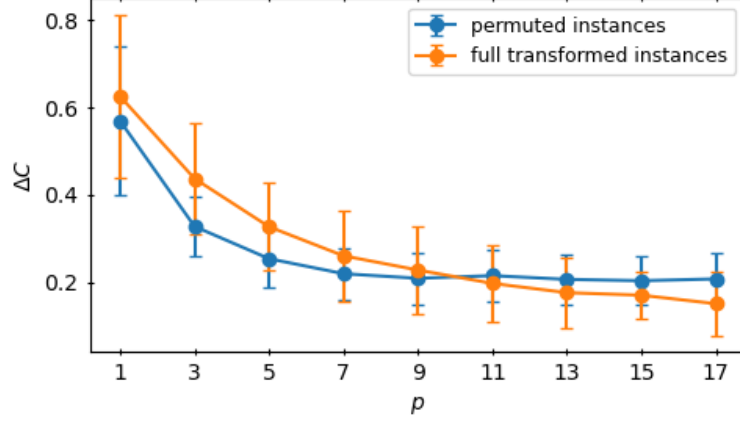


Figure 13: Cost value ΔC versus layer depth p for VQE with Ansatz 1 for two different MAX2SAT instances with each spin permutation and full transformation, which means both spin permutation and adiabatic gap-shift are applied. Error bars denote standard deviation over 100 instances. Full transformation accelerates convergence compared to only spin permutation.

Fig. 14 shows the convergence behavior for VQE with two different ansätze applied to the transformed SK instances. Since the gap shift transformation does not perform well at all on the SK problem in contrast to the MAX2SAT problem, which was shown before, only the spin label permutation is applied for the SK instance transformation. Ansatz 1 exhibits a smooth, nearly exponential convergence in cost with increasing p . Similarly, Ansatz 2 also shows a fast convergence at small p but converges at higher ΔC . This suggests that while a richer parameterization allows more rapid energy minimization, it may also induce a more rugged optimization landscape that increases the likelihood of suboptimal local minima.

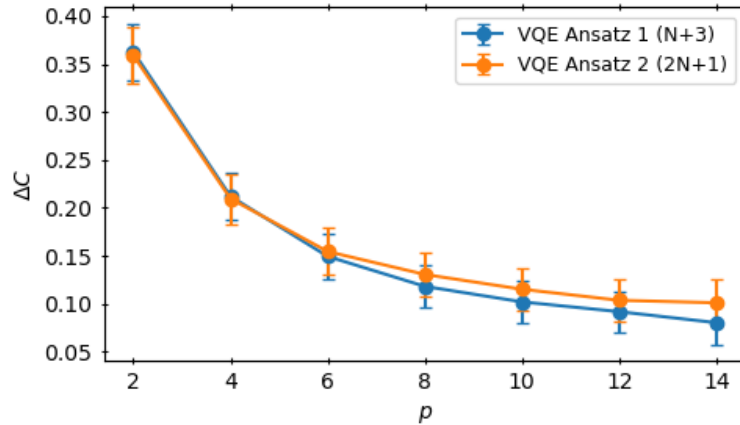


Figure 14: Cost difference versus p for SK instances using VQE Ansatz 1 ($p(N + 3)$) and VQE Ansatz 2 ($p(2N + 1)$). Error bars indicate standard deviation over 100 instances.

6.4.2 Average State Index Evaluation

The introduction of the term Average State Index is presented in Sec. 5.4. The average state index evaluation is used only for the SK instances whereas one has to come up with a different method of evaluation for the MAX2SAT as already mentioned before, which will be introduced in the next part (Sec. 6.4.3). The average state index offers an intuitive way to interpret the quality of the sampled output distribution. The index 0 corresponds to the true ground state, while higher indices reflect excited states. Hence, a lower average index suggests that the algorithm samples more frequently from low-energy configurations. Fig. 15 illustrates the average energy state index calculated from Eq. (5.28) of the two different VQE Ansätze applied on SK instances. The plot indicates that VQE in general shows steady increase in average state index as depth increases. This is inconsistent with the observed cost behavior from Fig. 14 and 13 and shows that lower energy approximations does not automatically translate into better sampling performance of finding the ground state.

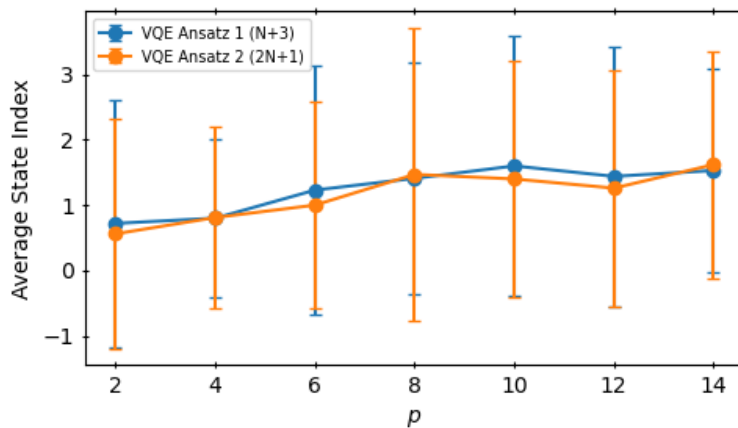


Figure 15: Average output state index versus p for VQE Ansatz 1 and Ansatz 2 on permuted SK instances.

6.4.3 Success Probability Comparison

Finally, we consider the success probability P_{success} instead of the average state index for the MAX2SAT problem, defined as the frequency with which the algorithm returns the true ground state across multiple measurements. This is perhaps the most application-relevant metric for practical implementations. In Fig. 16, we compare P_{success} for MAX2SAT instances under two transformation schemes applied within the VQE framework: spin label permutation and mean-field landscape transformation. In contrast to the SK instances, the ground state index for MAX2SAT instances shows a marked improvement with increasing circuit depth. However, beyond a certain layer count, this improvement plateaus, and the success probability P_{success} stabilizes, indicating diminishing returns from further depth increases. The results show that while both methods benefit from increased circuit depth, the transformed variant achieves near-perfect ground state identification beyond $p = 8$. In contrast, the permutation-based method saturates at a lower success rate and shows much less consistent improvement. This confirms that the landscape simplification afforded by the transformation leads not only to improved energy convergence but also to more effective sampling of the true ground state. The sharp inflection in the success rate curve for transformed instances highlights the critical role of reshaping the optimization landscape.

Taken together, these metrics as cost value, average index, and success probability demonstrate that both the algorithmic structure (QAOA vs. VQE, Ansatz type) and instance transformation significantly influence optimization performance. Particularly, combining expressive variational forms with targeted instance reshaping provides the clearest route toward universal, scalable quantum optimization strategies.

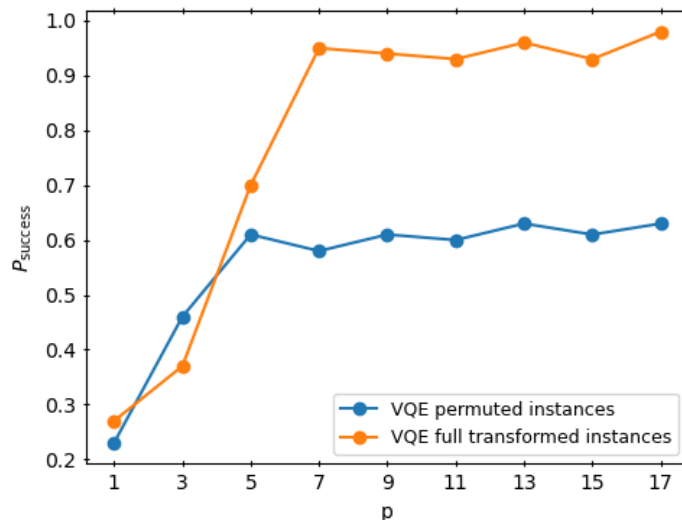


Figure 16: Success probability P_{success} versus p for MAX2SAT instances using two VQE transformations. The mean-field-based transformation drastically outperforms the permutation-based approach.

6.4.4 Comparison on Differently Transformed Instances

This section provides a comparative analysis of QAOA and VQE applied to SK and MAX2SAT problems, emphasizing the impact of various instance transformations. The aim is to evaluate how structural changes in the Hamiltonian, namely spin permutation and mean-field based transformations, influence the algorithm’s ability to sample low-energy or ground state configurations.

While an initial comparison between fully transformed and partially transformed instances (i.e., only spin label permutation applied) has already been touched upon in earlier sections, we now deepen this analysis by including results for the original, untransformed instances. This more comprehensive comparison allows us to investigate how each transformation component contributes to performance improvements, and whether the combined transformation offers a clear advantage over both partial and unmodified configurations.

For this analysis, the number of counts of each state indices are used for fixed spin size N and the number of layers p . All histograms shown below depict the distribution of the final energy state indices sampled by the quantum algorithm after optimization. The energy state index ranks the eigenstates by increasing energy: index 0 is the true ground state, index 1 the first excited state, and so on. For each algorithm and configuration, the energy state achieved after optimization is identified, and the frequency distribution over 100 instances is plotted.

These histograms are particularly useful because they provide discrete, interpretable insight into the performance beyond the average cost. A sharp peak at index 0 suggests frequent success in reaching the ground state, whereas a long tail toward higher indices indicates poor sampling behavior or convergence to suboptimal excited states. This also allows one to distinguish between high-precision convergence (many hits at index 0) versus shallow convergence (broad spread around low-index states).

QAOA: Original vs Transformed Instances

Fig. 17 shows the performance of QAOA with $p = 20$ for original and transformed SK instances with spin size of $N = 12$. For the original data (Fig. 17a), QAOA performs quite well: 29 of 100 trials reach the ground state (index 0), while more than 40 land on the first excited state (index 1), indicating good, but not perfect, ground state fidelity. The average index is 1.03, with most probability mass concentrated in the first two bins.

The transformed instances (Fig. 17b) also shows a solid performance: more ground state hits with over 40, but a higher average index (1.152), due to an increased number of samples falling into higher excited states. Notably, some samples even fall as far as index 12. This implies that while

transformation simplifies the landscape in principle, it may not suit QAOA’s expressiveness at moderate p . A likely reason for this reduced performance lies in the incompatibility of spin label permutation with QAOA’s parameter structure. QAOA parameters are typically assigned globally - i.e., they control entire layers of operators without dependence on specific spin indices. Therefore, permuting spins by frustration rank has no impact on how QAOA applies its gates, and the transformation may even disrupt correlations that QAOA implicitly relies on. In contrast to VQE, where parameter localization allows for more targeted adaptation to such structural changes, QAOA remains blind to the new ordering. This mismatch between the transformation and the algorithm’s structure leads to the observed loss in performance and that is the reason for bringing up the VQE ansatz for the transformed instances.

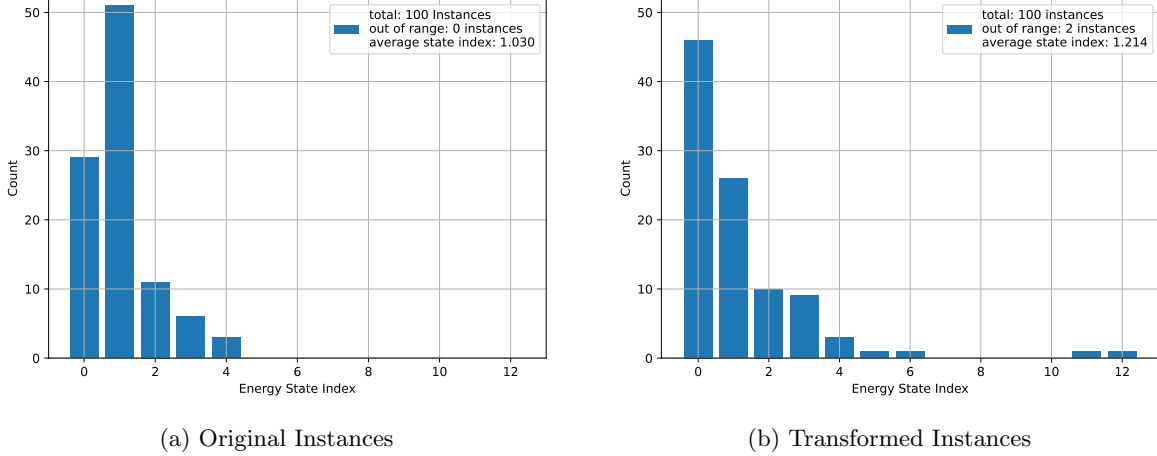


Figure 17: (a) Energy state index histogram for QAOA ($p = 20$) on original SK instances. Average state index: 1.03 over 100 samples. Peaks at indices 0 and 1 indicate efficient sampling from ground and near-ground states. (b) Energy state index histogram for QAOA ($p = 20$) on transformed SK instances. Average state index: 1.214. Every instance which lies outside of the selected index range are counted as out of range. The distribution broadens slightly with some outliers, including the 2 out of range instances, indicating a decrease in sampling fidelity.

VQE for SK Instances: Original, Permutation, Transformation

Fig. 18 compare VQE applied to SK instances with no transformation, spin permutation, and mean-field transformation, respectively. The number of layers is fixed at $p = 5$ for all further evaluations, as previous analyses in 6.4.2 indicated that increasing the circuit depth did not yield improved performance on SK instances. Therefore, a lower layer depth of $p = 5$ was chosen to minimize computational overhead, while achieving a relatively maximal performance. The original instance performance (Fig. 18a) exhibits a wide spread of sampled indices, with many counts beyond index 5. The average index is marked with 0.81, and no strong ground state peak is observed, showing weak optimization power at this depth. Spin permutation (Fig. 18b) reduces the spread and significantly increases the ground state hit rate. Also the average state index reaches the lowest value with 0.4. The value decreased by more than half relative to the original instance. It reorders the input representation to group more frustrated spins together, which likely improves the optimizer’s ability to target important parts of the energy landscape. In contrast, the transformed instances (Fig. 18c) show much poorer performance. The histogram is flatter and more dispersed, with fewer hits near index 0 and a greater portion of trials scattered across a wider range. The average index increases significantly compared to the original and permuted instances. This reflects that the mean-field gap-shift transformation, while theoretically attractive, destabilizes the energy structure in the SK model, creating deeper traps that the variational circuit cannot effectively navigate at shallow p .

In summary, while the spin label permutation improves the performance of VQE over the original instances, the adiabatic gap-shift degrades performance at least at low circuit depth.

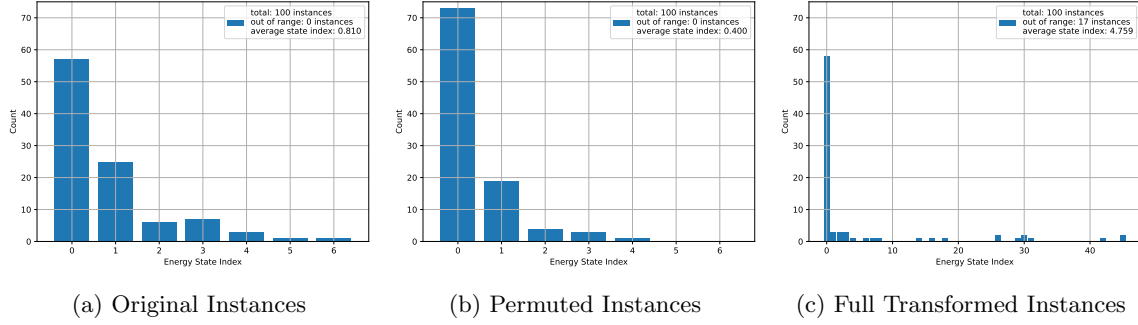


Figure 18: VQE (Ansatz 1) on three different SK instances with the layer depth $p = 5$. (a) VQE on original SK instances. Broad index distribution shows weak optimization and lack of state localization. (b) VQE on permuted SK instances. Spin label reordering significantly improves performance, shifting probability mass toward lower indices. (c) VQE on transformed SK instances. Histogram is widely scattered, indicating poor convergence and diminished fidelity due to destabilized optimization landscape.

VQE for MAX2SAT Instances: Permutation vs Transformation

This section extends the earlier success probability comparison in Sec. 6.4.3 by examining the energy index distributions in detail for VQE with a fixed layer depth of $p = 15$, since increasing the layer number beyond this point did not show a significant improvement anymore. The focus lies on understanding how transformation strategies impact optimization performance on MAX2SAT instances. Fig. 19 shows the energy index histograms for MAX2SAT under two transformation schemes with VQE using Ansatz 1. In the permutation-only case (Fig. 19a), the distribution peaks at index 0, but with significant counts still found up to index 15. This results in an average index of 2.87, acceptable but far from optimal.

After applying the complete mean-field transformation with spin label permutation and the adiabatic gap shift (Fig. 19b), the performance shows a clear improvement: more than 90 instances fall into the ground state, and the average index drops to 0.51. This sharp contrast highlights the effectiveness of this transformation for MAX2SAT. The energy landscape becomes smoother and easier to traverse, and the variational circuit can focus its resources on resolving meaningful degrees of freedom.

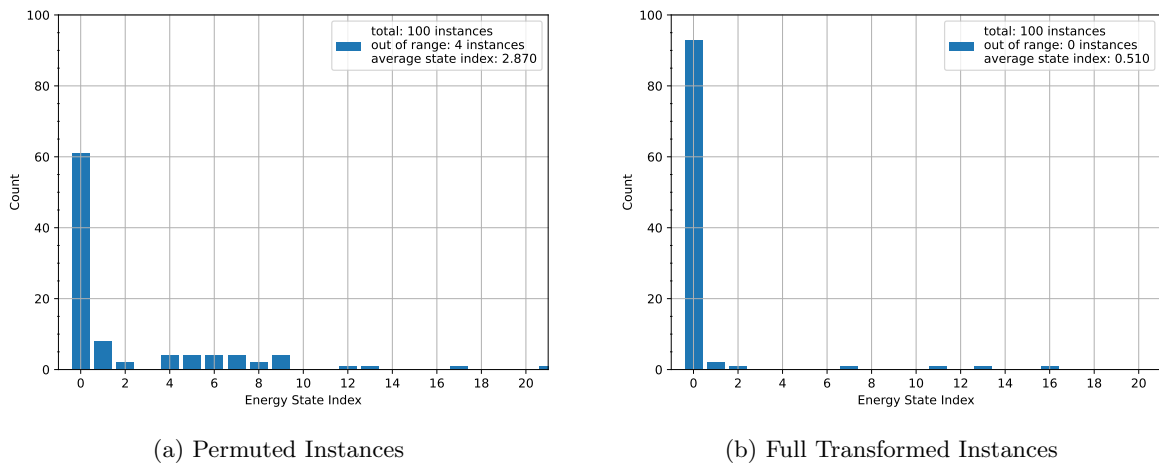


Figure 19: (a) VQE (Ansatz 1) on MAX2SAT with spin permutation only ($p = 15$). Moderate spread with several excited state samples. Average index: 1.784. (b) VQE (Ansatz 1) on transformed MAX2SAT ($p = 15$). Sharp peak at ground state. Average index 0.515, showing strong localization and successful optimization.

These results confirm the importance of aligning the transformation strategy with the nature of

the underlying optimization problem. For problems like MAX2SAT, with fewer long-range correlations and structured clause interactions, mean-field transformations enable meaningful simplification.

6.5 Performance of universal Parameterization for increased Spin sizes

To assess the scalability and generalization capacity of universal parameterization, we now analyze the performance of both QAOA and VQE when the system size N increases. Here it should be noted once again that for QAOA original instances are used, whereas VQE is applied to transformed instances, since QAOA cannot effectively exploit the spin permutation as explained in Sec. 5.1.3 under Limitations of QAOA Parameterization. The variational parameters used in this section are not re-optimized per instance or per system size. Instead, they are derived from a fixed training distribution and then applied to new SK instances of varying sizes. This tests the robustness and portability of the learned parameters in unseen problem regimes and evaluates whether a universal template can efficiently approximate ground state solutions across scaling system complexities.

Parameter Extension Strategy for higher N

A core distinction between the QAOA and VQE ansätze lies in the scaling of their variational parameter spaces. The QAOA circuit has a fixed number of parameters, determined solely by the depth p , independent of system size N . This design allows one to define a compact parameter set, for example, $2p$ values for the alternating γ and β angles, and deploy it uniformly across instances with different spin counts. This parameter economy provides simplicity and universality but also limits expressivity, especially as the complexity of the Hamiltonian grows with N .

On the other side, VQE circuits typically exhibit parameter scaling with N , reflecting their layered gate structure with qubit-local operations. For instance, Ansatz 1 uses $p(N + 3)$ parameters per instance, while Ansatz 2, being more expressive, uses $p(2N + 1)$. This size-dependence presents a challenge for universal parameterization, as direct transfer of a trained parameter set from one system size to another is not straightforward. To overcome this, we implement an incremental strategy:

1. First, full optimization is performed at $N = 12$ using the previously introduced two-stage method: a coarse grid search across initial parameters followed by fine-tuned optimization using a local optimizer over 100 random instances.
2. The resulting average parameter set becomes the “seed” for larger N values.
3. For each $N > 12$, we retain the optimized parameters for the first 12 spins and initialize the new parameters (for the additional spins) using heuristic values. Only the newly added parameters are re-optimized, while the original subset remains frozen.

This approach constrains the optimization to a lower-dimensional subspace, reducing computational cost and preserving the structural integrity of the previously optimized configurations. It ensures that the VQE maintains performance continuity while adapting to expanded Hilbert spaces. In doing so, we achieve a hybrid model of universal parameterization: fixed core parameters augmented by a size-dependent extension layer.

Sherington-Kirkpatrick model

Fig. 20 displays the average energy state index as a function of spin size N for QAOA ($p = 20$, $p = 40$) and VQE ($p = 4$). The trend across all methods shows a gradual increase in average index with growing system size. This is expected due to the exponentially expanding Hilbert space and corresponding increase in state density, making ground state identification more challenging.

Importantly, VQE consistently maintains a lower average index across all system sizes compared to both QAOA variants. For instance, at $N = 18$, the average index for QAOA with $p = 20$ rises to approximately 1.62, whereas the VQE remains below 1.0. This gap illustrates the relative resilience of the VQE ansatz under scaling, despite its growing parameter count. It suggests that the incremental parameter extension strategy effectively accommodates the increased complexity introduced by additional spins.

Interestingly, QAOA performance with $p = 40$ shows only marginal improvement over $p = 20$, indicating diminishing returns at higher depths under fixed parameterization. This hints at the limited capacity of shallow QAOA to generalize beyond its training regime without re-optimization, especially when the optimized schedule no longer aligns with the problem landscape at higher N .

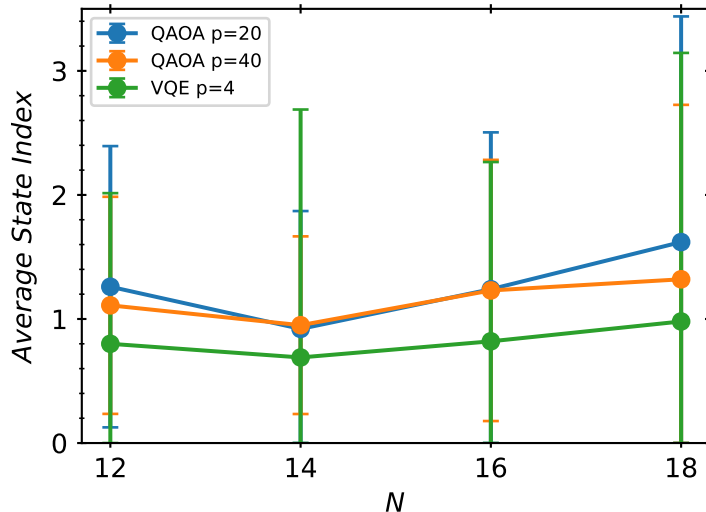


Figure 20: Average energy state index versus spin size N . VQE ($p = 4$) shows consistently lower indices than QAOA for both $p = 20$ and $p = 40$, highlighting its robustness under fixed and partially extended parameterization. Error bars represent standard deviation over 100 instances.

To obtain a more granular view of algorithm performance, we examine the full histograms of energy state index distributions in Fig. 21 and 22. Each panel shows the frequency of sampled state indices for $N = 12, 14, 16, 18$, comparing QAOA and VQE across varying depths and ansätze. In Fig. 21, we observe that for $N = 12$, both QAOA and VQE produce strongly peaked histograms at index 0, with QAOA showing slightly more spread. As N increases, QAOA’s peak flattens and shifts rightward, indicating that the original parameters become increasingly misaligned with the new energy landscape. VQE, however, retains a high sampling rate from the ground state, showing a smoother degradation in performance. For $N = 18$, while both methods show broadened distributions, VQE maintains its relative advantage with most counts still concentrated in the ground state index.

Fig. 22 expands the comparison to include both QAOA depths ($p = 20, p = 40$) and both VQE ansätze. The additional degrees of freedom in VQE Ansatz 2 ($p(2N+1)$) allow it to slightly outperform Ansatz 1 at larger N , as seen by the lower average indices and higher counts near index 0. On the other side, QAOA with $p = 40$ only marginally improves over $p = 20$, reinforcing the idea that parameter schedule quality (not just depth) is the limiting factor. This observation is particularly important when considering the number of parameter angles. QAOA typically has fewer parameters per layer than VQE, meaning that to achieve a comparable total parameter count, QAOA must operate at a higher layer depth p . Therefore, it is significant to demonstrate that even when QAOA is run with a much higher p , and thus more parameters, VQE applied to transformed instances still outperforms classical QAOA on original instances. Furthermore, the histograms for QAOA show a shift in peak index as N increases, moving from index 1 toward index 2-3 by $N = 18$, while the VQE distributions remain more sharply peaked at index 0. This robustness of VQE’s extended parameter strategy suggests that shallow universal QAOA may require depth scaling and re-optimization to remain competitive at higher N .

In conclusion, the hybrid parameterization approach used for VQE and the instance transformation demonstrates strong potential for scalability. By freezing the core parameters and only adapting extensions for higher N , VQE on tranformed instances maintains low state indices and high ground state fidelity, while QAOA on original instances shows worse results across the entire N compared to VQE and the static parameters of QAOA are progressively misaligned. Therefore, the universal VQE

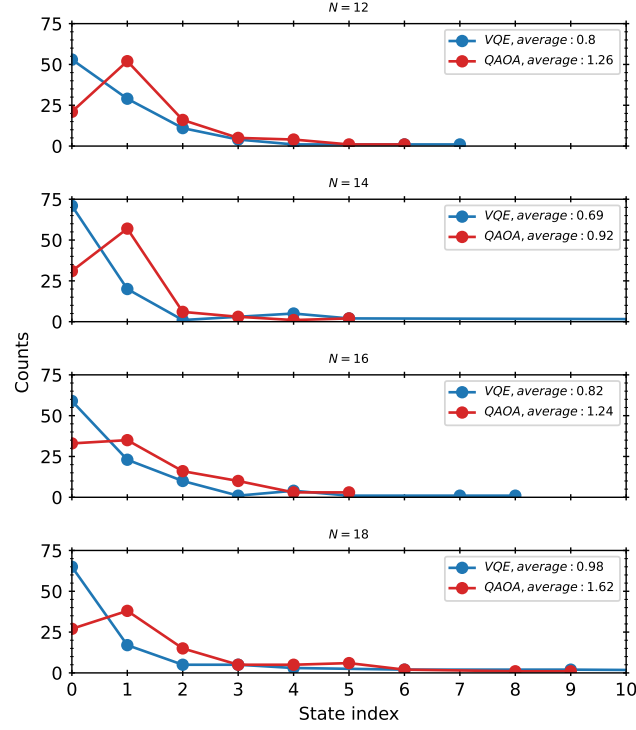


Figure 21: Energy index histograms for QAOA ($p = 20$) and VQE Ansatz 1 ($p = 4$) applied to SK instances with increasing system size N . Histogram broadening in QAOA indicates loss of alignment under fixed parameters. VQE demonstrates higher localization.

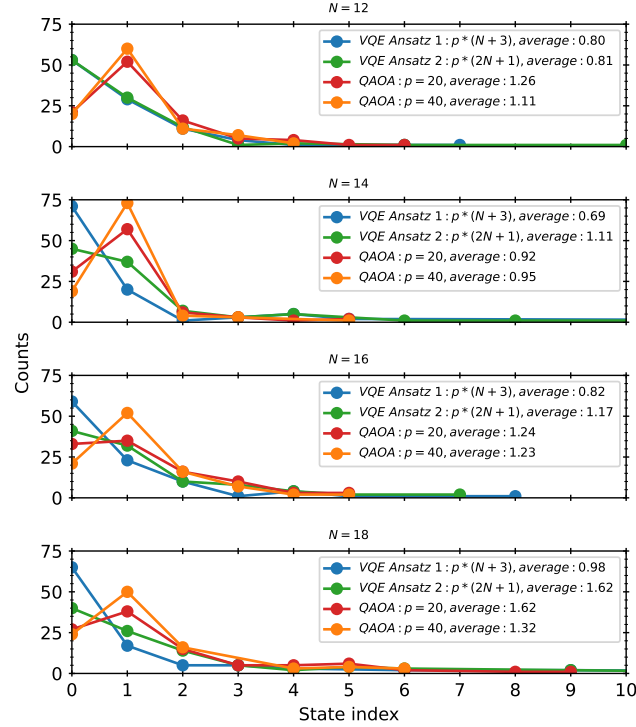


Figure 22: Histograms comparing all methods: VQE Ansatz 1 with $p=4$, VQE Ansatz 2, QAOA $p = 20$, and QAOA $p = 40$ for $N = 12$ to $N = 18$. VQE distributions are more localized, especially for Ansatz 2. QAOA shows broader profiles as N increases.

approach with the mean-field based transformation of the instances offers a practical advantage for the SK problem under realistic constraints, offering a promising direction for scalable quantum variational algorithms.

MAX2SAT

We now turn our attention to MAX2SAT instances and evaluate the performance of universal parameterization as the system size N increases. Unlike the SK model, MAX2SAT problems possess a more structured clause-based formulation and tend to exhibit higher ground state degeneracy. As a result, metrics such as average energy index are less meaningful for these instances to evaluate their performance. Instead, we evaluate performance using the success probability P_{success} , defined as the frequency of correctly identifying one of the ground state configurations. So we do not measure how far the energy state outcomes from the true ground state are. To evaluate the portability of learned VQE parameters, we first analyze Fig. 23, which compares the performance of VQE models initialized from two different base system sizes, $N = 10$ and $N = 12$. Both use the universal parameterization strategy, where for base size of $N = 10$, parameters trained at a smaller N are extended for larger N by initializing additional parameters while keeping the existing ones fixed. For base system size $N = 12$, we also go down to lower system size, so some parameters are being removed instead of additional optimization.

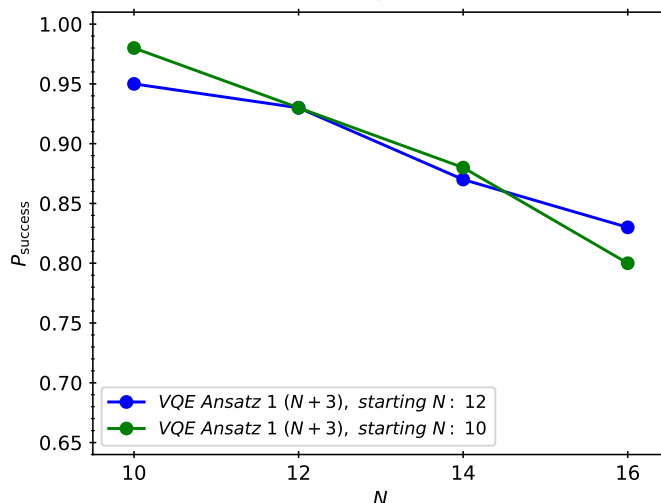


Figure 23: Success probability P_{success} versus N for MAX2SAT instances under VQE. Parameters are initialized from either $N = 10$ or $N = 12$ and extended to higher N .

As seen in Fig. 23, both VQE strategies show a decline in performance with increasing N , consistent with the growing complexity of the solution space. However, the parameterization extended from $N = 12$ consistently outperforms the one extended from $N = 10$, with a particularly noticeable gap at $N = 16$. This result indicates that starting from a moderately larger base system provides richer initial structure and parameter correlations that generalize more effectively across increasing N .

We now broaden the comparison to include QAOA, as shown in Fig. 24, where we evaluate P_{success} for QAOA at depths $p = 5$ and $p = 10$, along with the two VQE strategies. Fig. 24 shows the success probability for VQE and QAOA applied to MAX2SAT problems across $N = 10, 12, 14, 16$, using universal parameterizations derived from either $N = 10$ or $N = 12$.

Fig. 24 compares the performance of QAOA and VQE on MAX2SAT problems across increasing system sizes $N = 10, 12, 14, 16$, using universal parameterizations. Two QAOA depths ($p = 5$ and $p = 10$) and two VQE ansätze with increasing parameter counts ($p = N + 3$ and $p = 2N + 1$) are considered. The success probability P_{success} , the fraction of instances for which the algorithm recovers a valid ground-state configuration, serves as the evaluation metric. Across all system sizes, QAOA

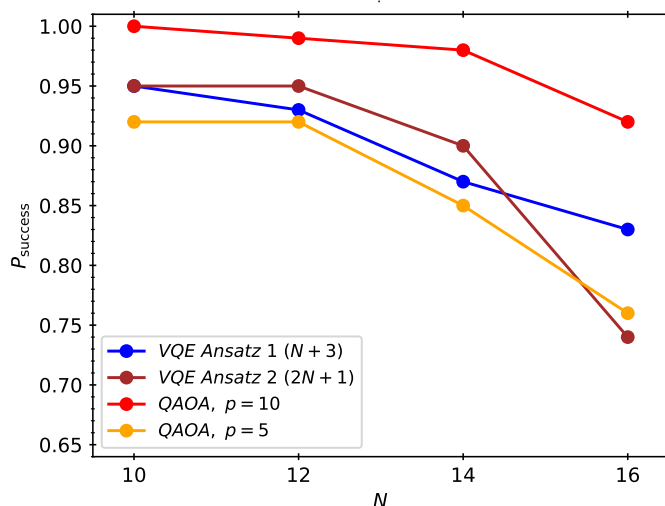


Figure 24: Success probability P_{success} versus system size N for 100 MAX2SAT instances each N . VQE and QAOA methods using fixed or partially extended parameters. While all methods degrade with N , VQE initialized from $N = 12$ retains the highest success rates.

with $p = 10$ consistently outperforms every VQE variant, maintaining a success probability above 0.95 up to $N = 14$, and only dropping below that threshold at $N = 16$. This is in contrast to both VQE variants, which show a more rapid decay of performance with increasing N . The superior performance of QAOA in this setting can be attributed to several interrelated factors:

Problem Structure Suitability: MAX2SAT, a constraint satisfaction problem composed of binary clauses, naturally maps into the QAOA framework. The QAOA cost Hamiltonian consists of commuting Pauli-Z terms that directly encode the clause structure, while the mixing Hamiltonian consists of transverse Pauli-X terms that explore the state space. This structure aligns very well with the clause logic of MAX2SAT, making QAOA particularly suited to capturing its solution landscape efficiently [3] [7].

Energy Degeneracy: MAX2SAT instances often have a large degenerate ground space, meaning many solutions yield the same optimal cost. QAOA benefits from this, as its measurement-based output can successfully hit a ground-state bitstring even when its probability distribution is only moderately concentrated. In contrast, VQE’s gradient-based optimization relies on minimizing expected energy and may struggle with flat energy landscapes where many states yield similar low energies but only a few are true ground states.

Hardness of the instances: The contrasting results between SK and MAX2SAT can be partially attributed to differences in instance hardness. The SK instances used in this thesis were specifically chosen for their difficulty, often characterized by very small minimum energy gaps, which pose a significant challenge for classical QAOA. In contrast, the MAX2SAT instances were not filtered for hardness and include a range of difficulties, many of which are relatively simple. This likely explains the strong performance of QAOA on MAX2SAT and its generally weaker results on SK. One can therefore assume that classical QAOA performs better on easier instances, while VQE combined with instance transformation may offer advantages on harder cases. If MAX2SAT instances were filtered to include only the most challenging examples, it is plausible that QAOA performance would degrade significantly, potentially falling below that of VQE applied to transformed instances.

Circuit Regularity and Depth: QAOA circuits for fixed depth p have a clean, alternating structure of unitaries that can implicitly encode time evolution under the problem Hamiltonian. As depth increases, this structure can better approximate the adiabatic evolution required to transition toward the ground state. The $p = 10$ QAOA circuit is deep enough to capture meaningful correlations among variables without overfitting, which may explain its strong performance.

Parameter Sensitivity and Transferability: Despite being a variational method, QAOA’s param-

eters, when trained over an ensemble of problem instances, tend to generalize well across different problem sizes, especially for clause-based problems like MAX2SAT [34]. In contrast, the VQE ansätze used here may suffer from overparameterization or insufficient expressivity depending on the choice ($N + 3$ or $2N + 1$), leading to poorer generalization.

Optimization Landscape: The cost function landscape of the MAX2SAT instances is often smoother and more structured, especially when optimized at fixed depth across similar clause densities. The transformed instances, on the other hand, might have a more complex optimization landscape due to its entangling structure and deeper parameterization, making it more susceptible to getting stuck in local minima during training.

Importantly, while QAOA performs better in this specific comparison of MAX2SAT problem, VQE should not be dismissed. In more disordered settings like the Sherrington-Kirkpatrick model, VQE can outperform QAOA due to its adaptability, layer-wise structure, and integration with instance-specific mean-field transformations. The strong performance of QAOA on MAX2SAT in Fig. 24 shows its effectiveness in solving clause-based optimization problems with less harder instances when equipped with sufficiently expressive depth and properly optimized parameters. VQE, while generally powerful and scalable, may require more sophisticated ansatz design or training strategies to match this performance in highly structured and degenerate problem domains.

6.6 Scalability and Practical Considerations

This thesis primarily presents an internal comparative analysis of quantum variational algorithms, namely QAOA and VQE, applied to both original and transformed instances of the SK and MAX2SAT problems. Rather than benchmarking absolute performance against classical high-performance computing (e.g. state-of-the-art HPC) methods, the emphasis is on how instance structure and transformations affect the performance and generalizability of these quantum methods.

The transformations introduced in this work, inspired by mean-field properties and adiabatic insights, reshape the energy landscape to reduce local traps and align problem hardness features. This enables a more favorable environment for variational quantum algorithms, particularly VQE, which benefits from adaptive circuit structures and gradient-based optimization.

Importantly, once a universal parameterization has been successfully established, typically by optimizing on a single, well-chosen or transformed instance, future evaluations require minimal computational effort. That is, for unseen problem instances of similar size and structure, no further optimization is necessary. The ground state can then be approximated directly by reusing pre-optimized parameters, leading to dramatic reductions in runtime compared to conventional methods.

This is in contrast to classical solvers, where each new instance typically requires a fresh optimization routine. Moreover, classical mean-field methods tend to fail for hard spin-glass instances due to their inability to capture quantum correlations or correctly resolve degenerate ground states [17]. The transformed VQE approach, by contrast, retains quantum fidelity and exhibits higher success probabilities across increasing system sizes, especially for difficult instances where classical methods break down.

In terms of scalability, several considerations apply:

- **QAOA:** While QAOA features shorter circuit depth and is more suitable for near-term quantum hardware, it lacks adaptivity in parameter schedules. As system size increases, performance degrades unless deeper circuits are used, which increases resource demands and hampers convergence.
- **VQE:** VQE offers architectural flexibility through custom ansätze and allows fine-tuning of layers. When combined with instance transformation, this flexibility results in improved scalability and cross-instance performance. However, VQE's optimization is more computationally demanding and more sensitive to quantum noise.

Compared to high-performance classical computation, real quantum hardware remains costly and error-prone. Nevertheless, the use of instance transformation and universal parameter schedules has the potential to significantly lower the total cost of quantum optimization by removing the need for per-instance retraining. This positions transformed VQE as a promising approach for scalable quantum optimization, particularly when targeting hard instances that resist classical solvers.

In conclusion, while large-scale deployment of these quantum algorithms still awaits advances in quantum hardware, the internal comparison in this work illustrates a viable path forward. VQE and Instance transformation with universal parameterization stands out as a practical candidate for future use, especially when the target problems are computationally hard and classical methods reach their limits.

6.7 Benchmark Performance of GNN

In addition, as in the last part, we benchmark this GNN model’s performance against QAOA and VQE results for the SK and MAX2SAT model, highlighting its ability to generalize over disorder realizations and capture low-energy configurations effectively. The complete implementation of the Graph Neural Network (GNN) models used for training and evaluation on the SK and MAX2SAT instances, including data preprocessing, model architecture, training loop, and loss functions, is available in Ref. [35]. The GNN code is implemented using the Julia packages `Flux.jl` and `GeometricFlux.jl`, and is structured for modular experimentation across different spin-glass model types and system sizes.

6.7.1 Training Procedure

The model is trained using the ADAM optimizer with a fixed learning rate and mini-batch gradient descent. Each training instance consists of a tuple $(J, h, \vec{\sigma}_{\text{true}})$, where $J \in \mathbb{R}^{N \times N}$ encodes couplings, $h \in \mathbb{R}^N$ are local fields, and $\vec{\sigma}_{\text{true}} \in \{-1, +1\}^N$ is the known ground state.

Training follows these steps:

- Randomly shuffle the dataset each epoch.
- Divide into mini-batches of size B .
- For each batch, compute average loss and update parameters using backpropagation.
- Track loss history over epochs for convergence assessment.

The model iterates message-passing and updates over a fixed number of steps (typically $T = 10$), and spin predictions are extracted after the final iteration. Empirically, this architecture balances representational capacity with training stability across larger problem sizes.

The Hyperparameters used for the training of GNN is given in the following Tab. 9, 10:

Parameter	Symbol	Value
Learning rate	η	0.01
Batch size	B	50
Epochs	E	20
Message passing iterations	T	10
Hidden dimension (MLP layers)	d_{hidden}	64
Loss penalty coefficient (binary spin)	λ	0.1
Activation function	σ	tanh (for MLP), relu (readout)
Training dataset size	–	1000 instances
System sizes of training dataset	N	10, 12, 14, 16

Table 9: Hyperparameters used in GNN training of MAX2SAT instances

The learning rate η controls the step size in parameter updates and is set to a moderately small value (0.01) for stable convergence. The batch size B determines how many training samples are processed per weight update. A total of $E = 20$ epochs provides enough iterations to ensure convergence. The number of message-passing steps T is fixed to 10 or 5, which enables sufficient context propagation across the graph. The hidden dimension ensures sufficient expressivity of the MLPs within the GNN. The binary penalty weighted by $\lambda = 0.1$ encourages predicted spins to lie close to physical values ± 1 . The **tanh** activation function is used in the message and update layers to enforce bounded outputs, while the readout layer uses **relu** followed by **tanh** to predict continuous spin values constrained to the interval $[-1, 1]$.

Parameter	Symbol	Value
Learning rate	η	0.0017
Batch size	B	50
Epochs	E	50
Message passing iterations	T	5
Hidden dimension (MLP layers)	d_{hidden}	32
Loss penalty coefficient (binary spin)	λ	0.1
Activation function	σ	tanh (for MLP), relu (readout)
Training dataset size	–	3360 instances
System sizes of training dataset	N	10, 12, 14, 16, 18

Table 10: Hyperparameters used in GNN training of Sherrington-Kirkpatrick instances

6.7.2 Performance of GNN against QAOA and VQE

MAX2SAT

Fig. 25 shows the training loss over 20 epochs for the MAX2SAT - GNN model. The corresponding loss function is taken from Eq. 5.24. The loss exhibits a steep decline during the first three epochs, dropping from over 350 to below 10, after which it flattens out and stabilizes. This behavior indicates that the model quickly learns a low-energy representation of the SK instances early in training. The convergence plateau suggests that the model’s parameters reach a local minimum that generalizes well across the diverse input instances. The early convergence also confirms that the chosen hyperparameters provide sufficient learning capacity without overfitting, even when training over instances of varying system sizes.

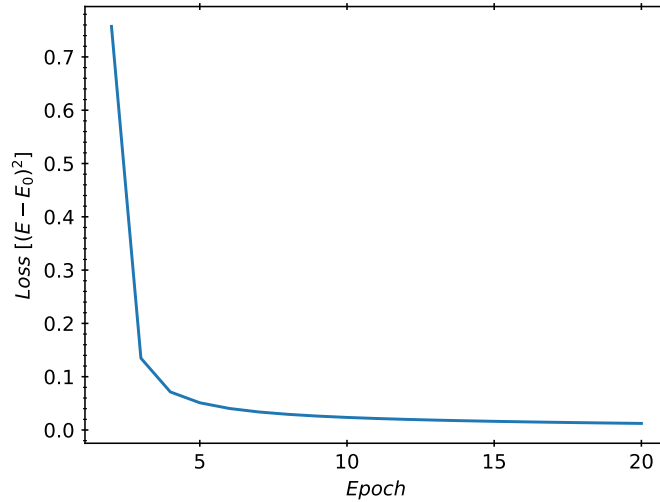


Figure 25: Training loss of the MAX2SAT - GNN model over 20 epochs. The sharp decline in the first few epochs indicates rapid convergence.

Fig. 26 presents the success probability P_{success} of solving MAX2SAT instances across increasing system sizes $N = 10, 12, 14, 16$ using Graph Neural Networks (GNNs), compared to QAOA (with $p = 10$), and two VQE configurations: one with $2N + 1$ parameters and one with $N + 3$ parameters.

The GNN consistently achieves perfect success across all values of N , demonstrating its superior generalization ability. It is test over 100 instances for each N . At every N , the GNN attains $P_{\text{success}} = 1$, outperforming both QAOA and VQE, which show more rapid degradation in performance. QAOA, while competitive at smaller N , declines to below 0.90 for $N = 16$. VQE, in both parameterizations, exhibits the steepest performance drop, indicating a stronger dependence on system size and a limited

capacity to generalize universal parameters compared to GNN.

This trend highlights the flexibility and scalability of GNN-based models. Unlike variational quantum circuits, GNNs can learn to extract instance-invariant features by processing structured input representations (e.g., clause graphs), making them well-suited for problems with high ground-state degeneracy like MAX2SAT. Furthermore, the message-passing mechanism in GNNs enables the model to effectively adapt to topological variations in the input graph, which is difficult for fixed-structure quantum circuits.

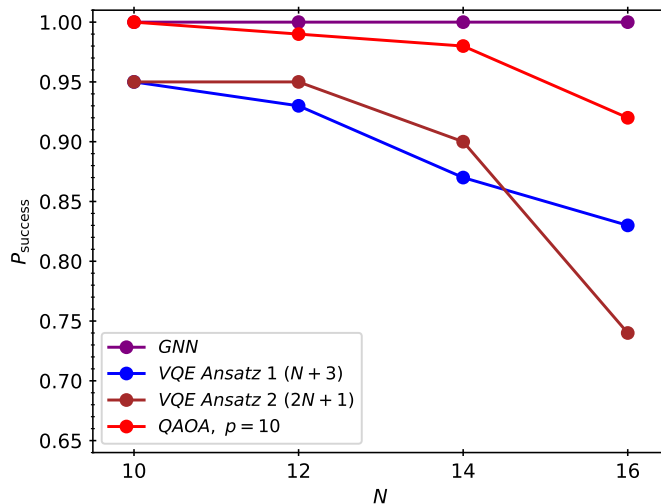


Figure 26: Success probability P_{success} versus system size N for MAX2SAT. GNN demonstrates superior scalability compared to QAOA and VQE, retaining high accuracy even for $N = 16$.

Sherrington-Kirkpatrick

The performance of the GNN model on the SK instances is significantly weaker compared to MAX2SAT. This section serves primarily to illustrate the limitations of the current GNN implementation in capturing the highly frustrated and complex energy structure of SK instances. Fig. 27 shows the loss function during training. After a steep drop within the first few epochs, the loss quickly stagnates at a relatively high value, suggesting that the model gets trapped in a local minimum from which it cannot escape. Through continued training, a second drop is observed to less than a loss value of 5. From this point on, no substantial improvement is observed for further epoch flow, indicating limited learnability of the SK ground states with the current training setup and the amount of the training data.

The loss function does not reach a necessary value at the end to obtain a meaningful result for the GNN. The consequences of this training failure are reflected in Fig. 28, which plots the average state index over test instances for increasing system sizes N . Across all sizes, the average index remains very high, well above the first few excited states. For $N = 16$ and $N = 18$, the index exceeds 40, meaning the GNN is rarely, if ever, identifying the ground state or even low-energy configurations. The error bars further suggest wide variation and instability in the output predictions.

A deeper analysis for SK problem is not meaningful given the lack of successful predictions. These outcomes are likely rooted in the intrinsic difficulty of the SK model and the explicit selection of hard instances, characterized by rugged energy landscapes, small adiabatic gaps, and strong spin frustration. In contrast to MAX2SAT, where the clause structure imposes local constraints, the SK model requires the GNN to capture long-range correlations across all spins - something that is difficult to achieve with limited training data.

In this work, the GNN was trained on 3360 SK instances from different spin sizes N that are available for the training. While this number might seem substantial, it is rather small for training neural

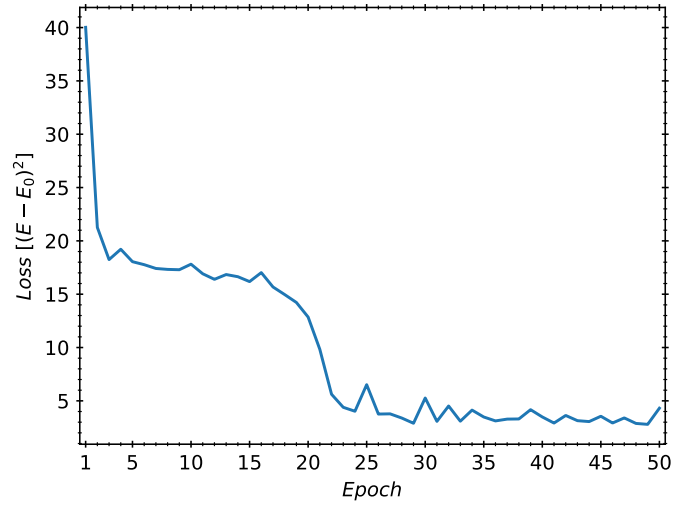


Figure 27: Loss curve of the GNN during training on 3360 SK instances of different spin sizes N . The model converges prematurely into multiple local minima and shows no further significant recovery over 50 epochs.

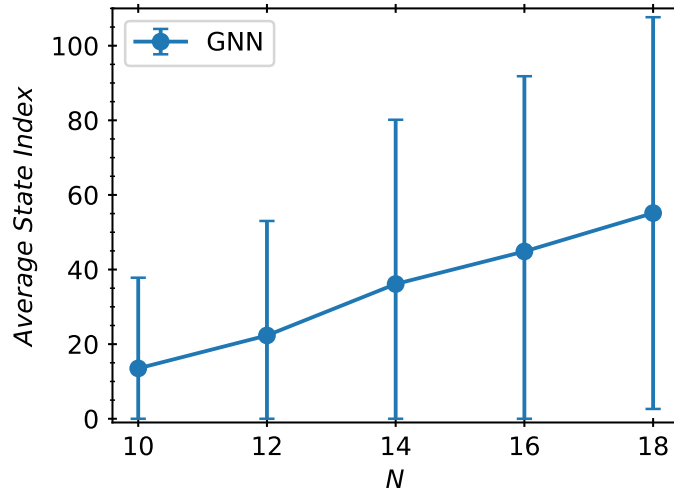


Figure 28: Average state index predicted by the GNN across different SK instance sizes. The results demonstrate poor performance with large deviation and no meaningful ground state recovery.

networks, especially for learning fine-grained features needed to locate ground states in disordered systems [36]. The size of the training set, in combination with a modest GNN architecture, is insufficient for generalization to higher- N instances [37]. In particular, the number of training samples for $N = 16$ and $N = 18$ was limited to only 360 out of 3360 available instances, which likely contributes to the weaker performance observed at these larger spin sizes compared to smaller N .

As conclusion, the poor performance of the GNN on SK instances highlights the limitations of learning-based approaches when confronted with complex, non-local spin glass structures. While the results were not particularly promising, we did observe some improvement in loss optimization when increasing the size of the training dataset and fine-tuning hyperparameters such as the learning rate and increasing the number of training epochs. This suggests that there is still potential for improvement with more extensive training and parameter tuning. The easiest way of improvements could be achieved by increasing the size and diversity of the training dataset. Additionally, incorporating more sophisticated GNN architectures capable of long-range message passing (e.g. attention mechanisms, deeper layers), or using physics-informed loss functions or curriculum learning strategies to gradually expose the model to increasing complexity might also be advanced approaches to improve the results.

Overall, while GNNs show promise for structured problems like MAX2SAT, the SK model poses significant challenges that require further architectural and methodological advances to overcome.

7 Conclusion

In this thesis, we explored the potential of improving the performance and universal generalizability of variational quantum algorithms through instance transformation techniques. The primary focus was placed on the Variational Quantum Eigensolver, enhanced by semiclassical preprocessing methods motivated by mean-field theory. As a comparative baseline, we included classical QAOA operating on untransformed original instances, serving primarily to contextualize the benefits of the VQE-based approach.

Our study was conducted across two problem domains: the Sherrington-Kirkpatrick spin glass model and the MAX2SAT constraint satisfaction problem. We introduced a novel instance transformation framework based on semi-classical mean-field analysis. This framework included spin-label permutation and adiabatic gap-shift techniques, which were designed to reduce the optimization complexity without changing the spectral properties of the problem Hamiltonian. As shown in Chapter 6, these transformations significantly improved the ground-state sampling of VQE on SK instances. In particular, the transformed instances exhibited clearer bottleneck alignment and enabled a more effective use of the variational parameters.

For the SK model with hard instances, the proposed transformations proved highly effective. When using the VQE with transformed instances, we observed a significant improvement in ground state sampling and success probability, particularly in comparison to classical QAOA applied directly to the original instances. Notably, this advantage persisted even when QAOA was run at much higher circuit depths, highlighting the efficiency of the VQE when guided by problem-informed preprocessing.

In contrast, for MAX2SAT, where problem instances are structurally soft and often include many degenerate solutions, classical QAOA already achieves high success rates. In this case, instance transformation offers only modest benefits for VQE. However, these results also suggest that transformation techniques are most valuable when applied to hard and non-trivial instances, where naive quantum optimization struggles and classical mean-field methods alone fail.

We further explored the idea of universal parameterization in VQE, demonstrating that variational parameters trained on smaller instances could be transferred to larger systems with minimal retraining. This method, especially when paired with transformed problem instances, enables significant computational savings by avoiding repeated optimization for each new instance size.

Finally, we attempted to use Graph Neural Networks to predict ground state configurations for the SK model. While the performance for MAX2SAT was promising, GNNs failed to generalize effectively for SK. This underperformance is attributed to the complexity of the SK landscape and the limited size of the training data. As shown in Chapter 6.7, the model’s performance stagnated in local minima, indicating the need for significantly larger datasets for hard problem instances.

To conclude, this thesis demonstrates that combining classical problem transformations with variational quantum algorithms such as VQE can substantially improve their performance and achieve universal parameterization over higher N on hard combinatorial problems. These insights suggest a promising direction for hybrid quantum-classical workflows, where physics-informed preprocessing paves the way for scalable and efficient quantum optimization.

In future work, several promising directions remain to be explored. One natural extension is the integration of neural parameter prediction methods with instance transformation techniques, forming hybrid schemes that could further improve initialization and reduce optimization overhead. Lastly, the GNN approach can be extended by scaling both the architecture and training dataset. This may ultimately allow for more reliable prediction of ground state configurations for hard instances of the SK problem. Additionally, further investigation into VQE performance on MAX2SAT problems is also of interest, particularly to close the observed gap against QAOA and to assess whether refined ansatz designs or training procedures could increase effectiveness. Another important direction involves benchmarking the universal parameterization strategies under realistic noise models, as implemented on near-term quantum hardware. This would provide insights into their robustness and practical utility beyond idealized simulations.

References

- [1] Tameem Albash and Daniel A. Lidar. “Adiabatic quantum computation”. In: *Rev. Mod. Phys.* 90 (1 Jan. 2018), p. 015002. DOI: 10.1103/RevModPhys.90.015002. URL: <https://link.aps.org/doi/10.1103/RevModPhys.90.015002>.
- [2] Nicholas Chancellor. “Modernizing quantum annealing using local searches”. In: *New Journal of Physics* 19.2 (Feb. 2017), p. 023024. ISSN: 1367-2630. DOI: 10.1088/1367-2630/aa59c4. URL: <http://dx.doi.org/10.1088/1367-2630/aa59c4>.
- [3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. “A Quantum Approximate Optimization Algorithm”. In: (2014). arXiv: 1411.4028 [quant-ph]. URL: <https://arxiv.org/abs/1411.4028>.
- [4] Edsger W Dijkstra. “A note on two problems in connexion with graphs”. In: *Numerische mathematik* 1.1 (1959), pp. 269–271. URL: <https://doi.org/10.1007/BF01386390>.
- [5] A. Andreanov and M. Müller. “Long-Range Quantum Ising Spin Glasses at $T=0$: Gapless Collective Excitations and Universality”. In: *Phys. Rev. Lett.* 109 (17 Oct. 2012), p. 177201. DOI: 10.1103/PhysRevLett.109.177201. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.109.177201>.
- [6] Thorge Müller et al. “Limitations of Quantum Approximate Optimization in Solving Generic Higher-Order Constraint-Satisfaction Problems”. In: (2024). arXiv: 2411.19388 [quant-ph]. URL: <https://arxiv.org/abs/2411.19388>.
- [7] Leo Zhou et al. “Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices”. In: *Phys. Rev. X* 10 (2 June 2020), p. 021067. DOI: 10.1103/PhysRevX.10.021067. URL: <https://link.aps.org/doi/10.1103/PhysRevX.10.021067>.
- [8] Abhinav Kandala et al. “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”. In: *Nature* 549.7671 (2017), pp. 242–246. DOI: 10.1038/nature23879. URL: <https://doi.org/10.1038/nature23879>.
- [9] Aditi Misra-Spieldenner et al. “Mean-Field Approximate Optimization Algorithm”. In: *PRX Quantum* 4.3 (Sept. 2023). ISSN: 2691-3399. DOI: 10.1103/prxquantum.4.030335. URL: <http://dx.doi.org/10.1103/PRXQuantum.4.030335>.
- [10] Bikas K Chakrabarti and Sudip Mukherjee. *Quantum Annealing and Computation*. 2022. arXiv: 2203.15839 [cond-mat.stat-mech]. URL: <https://arxiv.org/abs/2203.15839>.
- [11] Andrew Lucas. “Ising formulations of many NP problems”. In: *Frontiers in Physics* 2 (2014), p. 5. DOI: 10.3389/fphy.2014.00005.
- [12] Catherine McGeoch. *Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice*. Vol. 5. July 2014, pp. 1–93. DOI: 10.2200/S00585ED1V01Y201407QMC008.
- [13] Arnab Das and Bikas K. Chakrabarti. “Colloquium: Quantum annealing and analog quantum computation”. In: *Rev. Mod. Phys.* 80 (3 Sept. 2008), pp. 1061–1081. DOI: 10.1103/RevModPhys.80.1061. URL: <https://link.aps.org/doi/10.1103/RevModPhys.80.1061>.
- [14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. First Edition. W. H. Freeman, 1979. ISBN: 0716710455. URL: <http://www.amazon.com/Computers-Intractability-NP-Completeness-Mathematical-Sciences/dp/0716710455>.
- [15] Stuart Hadfield et al. “From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz”. In: *Algorithms* 12.2 (2019). ISSN: 1999-4893. DOI: 10.3390/a12020034. URL: <https://www.mdpi.com/1999-4893/12/2/34>.
- [16] Anton Frisk Kockum et al. *Lecture notes on quantum computing*. 2025. arXiv: 2311.08445 [quant-ph]. URL: <https://arxiv.org/abs/2311.08445>.
- [17] Tim Bode and Frank K. Wilhelm. “Adiabatic bottlenecks in quantum annealing and nonequilibrium dynamics of paramagnons”. In: *Physical Review A* 110.1 (July 2024). ISSN: 2469-9934. DOI: 10.1103/PhysRevA.110.012611. URL: <http://dx.doi.org/10.1103/PhysRevA.110.012611>.

- [18] S F Edwards and P W Anderson. “Theory of spin glasses”. In: 5.5 (May 1975), p. 965. DOI: 10.1088/0305-4608/5/5/017. URL: <https://dx.doi.org/10.1088/0305-4608/5/5/017>.
- [19] Edward Farhi et al. “A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem”. In: *Science* 292.5516 (Apr. 2001), pp. 472–475. ISSN: 1095-9203. DOI: 10.1126/science.1057726. URL: <http://dx.doi.org/10.1126/science.1057726>.
- [20] Lucas T. Brady et al. “Optimal Protocols in Quantum Annealing and Quantum Approximate Optimization Algorithm Problems”. In: *Physical Review Letters* 126.7 (Feb. 2021). ISSN: 1079-7114. DOI: 10.1103/physrevlett.126.070505. URL: <http://dx.doi.org/10.1103/PhysRevLett.126.070505>.
- [21] Nikolaj Moll et al. “Quantum optimization using variational algorithms on near-term quantum devices”. In: *Quantum Science and Technology* 3.3 (June 2018), p. 030503. ISSN: 2058-9565. DOI: 10.1088/2058-9565/aab822. URL: <http://dx.doi.org/10.1088/2058-9565/aab822>.
- [22] Jarrod R McClean et al. “The theory of variational hybrid quantum-classical algorithms”. In: *New Journal of Physics* 18.2 (Feb. 2016), p. 023023. ISSN: 1367-2630. DOI: 10.1088/1367-2630/18/2/023023. URL: <http://dx.doi.org/10.1088/1367-2630/18/2/023023>.
- [23] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature Communications* 5.1 (July 2014). ISSN: 2041-1723. DOI: 10.1038/ncomms5213. URL: <http://dx.doi.org/10.1038/ncomms5213>.
- [24] Peter W. Battaglia et al. “Relational inductive biases, deep learning, and graph networks”. In: (2018). arXiv: 1806.01261 [cs.LG]. URL: <https://arxiv.org/abs/1806.01261>.
- [25] Zonghan Wu et al. “A Comprehensive Survey on Graph Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (Jan. 2021), pp. 4–24. ISSN: 2162-2388. DOI: 10.1109/tnnls.2020.2978386. URL: <http://dx.doi.org/10.1109/TNNLS.2020.2978386>.
- [26] Thomas N. Kipf and Max Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: (2017). arXiv: 1609.02907 [cs.LG]. URL: <https://arxiv.org/abs/1609.02907>.
- [27] Keyulu Xu et al. *How Powerful are Graph Neural Networks?* 2019. arXiv: 1810.00826 [cs.LG]. URL: <https://arxiv.org/abs/1810.00826>.
- [28] Maxime Gasse et al. *Exact Combinatorial Optimization with Graph Convolutional Neural Networks*. 2019. arXiv: 1906.01629 [cs.LG]. URL: <https://arxiv.org/abs/1906.01629>.
- [29] Yang Liu et al. *Combinatorial Optimization with Automated Graph Neural Networks*. 2024. arXiv: 2406.02872 [cs.LG]. URL: <https://arxiv.org/abs/2406.02872>.
- [30] Dongjin Suh. *VQE.jl – Variational Quantum Eigensolver in Julia package*. <https://github.com/dongjinsuh/VQE.jl>. Accessed: 2025-06-11. 2024.
- [31] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users’ Guide*. Society for Industrial and Applied Mathematics, 1998. DOI: 10.1137/1.9780898719628. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9780898719628>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9780898719628>.
- [32] PGI-12 Forschungszentrum Jülich. *SpinFluctuations.jl: data directory*. <https://github.com/FZJ-PGI-12/SpinFluctuations.jl/tree/main/data>. Accessed: 2025-06-14. 2024.
- [33] Puya Mirkarimi et al. “Comparing the hardness of MAX 2-SAT problem instances for quantum and classical algorithms”. In: *Physical Review Research* 5.2 (June 2023). ISSN: 2643-1564. DOI: 10.1103/physrevresearch.5.023151. URL: <http://dx.doi.org/10.1103/PhysRevResearch.5.023151>.
- [34] Daniel J. Egger, Jakub Mareček, and Stefan Woerner. “Warm-starting quantum optimization”. In: *Quantum* 5 (June 2021), p. 479. ISSN: 2521-327X. DOI: 10.22331/q-2021-06-17-479. URL: <http://dx.doi.org/10.22331/q-2021-06-17-479>.
- [35] Dongjin Suh. *GNN.jl – Graph Neural Network in Julia package*. <https://github.com/dongjinsuh/GNN.jl>. Accessed: 2025-06-15. 2025.

-
- [36] Giuseppe Carleo and Matthias Troyer. “Solving the quantum many-body problem with artificial neural networks”. In: *Science* 355.6325 (Feb. 2017), pp. 602–606. ISSN: 1095-9203. DOI: 10.1126/science.aag2302. URL: <http://dx.doi.org/10.1126/science.aag2302>.
 - [37] Vijay Prakash Dwivedi et al. *Benchmarking Graph Neural Networks*. 2022. arXiv: 2003.00982 [cs.LG]. URL: <https://arxiv.org/abs/2003.00982>.

A Appendix

A.1 Derivation of the Mean-field equations of motion:

The given Hamiltonians are

$$H_P = - \sum_{i=1}^N \left[h_i + \sum_{j>i} J_{ij} \hat{\sigma}_j^z \right] \hat{\sigma}_i^z = - \sum_{i=1}^N h_i \hat{\sigma}_i^z - \sum_{i,j=1}^N J_{ij} \hat{\sigma}_j^z \hat{\sigma}_i^z$$

$$H_X = - \sum_{i=1}^N \Delta_i \hat{\sigma}_i^x, \quad \Delta_i > 0$$

$$\Rightarrow H(t) = \gamma(t)H_P + \beta(t)H_D$$

Apply von Neumann equation:

$$i\hbar \frac{d\varphi}{dt} = [H, \varphi]$$

$$\text{Let } \langle \hat{\sigma}_i^\alpha \rangle = \text{Tr}(\rho \hat{\sigma}_i^\alpha), \quad \text{where } \alpha = x, y, z$$

X-component:

$$\begin{aligned} [H_D, \hat{\sigma}_j^x] &= - \sum_{j=1}^N \Delta_j [\hat{\sigma}_j^x, \hat{\sigma}_j^x] = 0 \\ [H_P, \hat{\sigma}_i^x] &= - \sum_{j=1}^N h_j [\hat{\sigma}_j^z, \hat{\sigma}_i^x] - \sum_{j=1}^N \sum_{k>j} J_{jk} [\hat{\sigma}_j^z \hat{\sigma}_k^z, \hat{\sigma}_i^x] \end{aligned}$$

$$\text{Since } [\hat{\sigma}_i^z, \hat{\sigma}_j^x] = \begin{cases} 0 & \text{if } i \neq j \\ 2i\sigma_i^y & \text{if } i = j \end{cases}$$

$$\begin{aligned} [H_P, \hat{\sigma}_i^x] &= -2ih_i \hat{\sigma}_i^y - \sum_{j=1}^N \sum_{k>j} J_{jk} [\hat{\sigma}_k^z \hat{\sigma}_j^z, \hat{\sigma}_i^x] \\ &= -2ih_i \hat{\sigma}_i^y - \sum_{j=1}^N \sum_{k>j} J_{jk} (\hat{\sigma}_k^z [\hat{\sigma}_j^z, \hat{\sigma}_i^x] + [\hat{\sigma}_k^z, \hat{\sigma}_i^x] \hat{\sigma}_j^z) \\ [\hat{\sigma}_j^z, \hat{\sigma}_i^x] &= 0 \quad \text{for } j \neq i \\ &= -2ih_i \hat{\sigma}_i^y - 2i \sum_{k>i} J_{ik} \hat{\sigma}_k^z \hat{\sigma}_i^y - 2i \sum_{j<i} J_{ji} \hat{\sigma}_j^z \hat{\sigma}_i^y \\ &= -2i \hat{\sigma}_i^y \left(h_i + \sum_{k>i} J_{ik} \hat{\sigma}_k^z - \sum_{j<i} J_{ji} \hat{\sigma}_j^z \right) \end{aligned}$$

$$\begin{aligned} \Rightarrow [H, \hat{\sigma}_i^x] &= \beta(t)[H_D, \hat{\sigma}_i^x] + \gamma(t)[H_P, \hat{\sigma}_i^x] \\ &= -2i \hat{\sigma}_i^y \gamma(t) \left(h_i + \sum_{k>i} J_{ik} \hat{\sigma}_k^z - \sum_{j<i} J_{ji} \hat{\sigma}_j^z \right) \\ \Rightarrow i\hbar \frac{d\hat{\sigma}_i^x}{dt} &= [H, \hat{\sigma}_i^x] = -2i \hat{\sigma}_i^y \gamma(t) \left(h_i + \sum_{k>i} J_{ik} \hat{\sigma}_k^z - \sum_{j<i} J_{ji} \hat{\sigma}_j^z \right) \end{aligned}$$

Set $\hbar = 1$:

$$\begin{aligned}\Leftrightarrow \frac{d}{dt}\langle\hat{\sigma}_i^x\rangle &= 2\langle\hat{\sigma}_i^y\rangle\gamma(t)\left(h_i + \sum_{k>i}J_{ik}\langle\hat{\sigma}_k^z\rangle - \sum_{j<i}J_{ji}\langle\hat{\sigma}_j^z\rangle\right) \\ &= 2\langle\hat{\sigma}_i^y\rangle\gamma(t)\left(h_i - \sum_{k<i}J_{ik}\langle\hat{\sigma}_k^z\rangle\right)\end{aligned}$$

Y-component:

$$\begin{aligned}[H_D, \hat{\sigma}_i^y] &= -\sum_{j=1}^N \Delta_j [\hat{\sigma}_j^x, \hat{\sigma}_i^y] \\ [\hat{\sigma}_j^x, \hat{\sigma}_i^y] &= 2i\delta_{ij}\hat{\sigma}_i^z \\ \Rightarrow [H_D, \hat{\sigma}_i^y] &= -\sum_{j=1}^N \Delta_j \cdot 2i\delta_{ij}\hat{\sigma}_i^z = -2i\Delta_i\hat{\sigma}_i^z \\ [H_P, \hat{\sigma}_i^y] &= -\sum_{j=1}^N h_j [\hat{\sigma}_j^z, \hat{\sigma}_i^y] - \sum_{j=1}^N \sum_{k>j} J_{jk} [\hat{\sigma}_j^z \hat{\sigma}_k^z, \hat{\sigma}_i^y] \\ &= -\sum_{j=1}^N h_j (-2i\delta_{ij}\hat{\sigma}_i^x) - \sum_{j=1}^N \sum_{k>j} J_{jk} (\hat{\sigma}_k^z [\hat{\sigma}_j^z, \hat{\sigma}_i^y] + [\hat{\sigma}_k^z, \hat{\sigma}_i^y] \hat{\sigma}_j^z) \\ &= 2ih_i\hat{\sigma}_i^x + 2i\sum_{k>i} J_{ik}\hat{\sigma}_k^z\hat{\sigma}_i^x + 2i\sum_{j<i} J_{ji}\hat{\sigma}_j^z\hat{\sigma}_i^x \\ &= 2i\hat{\sigma}_i^x \left(h_i + \sum_{k>i} J_{ik}\hat{\sigma}_k^z + \sum_{j<i} J_{ji}\hat{\sigma}_j^z\right) \\ \Rightarrow [H, \hat{\sigma}_i^y] &= \beta(t)[H_D, \hat{\sigma}_i^y] + \gamma(t)[H_P, \hat{\sigma}_i^y] \\ &= -2i\beta(t)\Delta_i\hat{\sigma}_i^z + 2i\gamma(t)\hat{\sigma}_i^x \left(h_i + \sum_{k>i} J_{ik}\hat{\sigma}_k^z + \sum_{j<i} J_{ji}\hat{\sigma}_j^z\right)\end{aligned}$$

Equation of motion using von Neumann equation:

$$\frac{d}{dt}\langle\hat{\sigma}_i^y\rangle = \frac{1}{i\hbar}\langle[H, \hat{\sigma}_i^y]\rangle$$

Assume $\hbar = 1$:

$$\Rightarrow \frac{d}{dt}\langle\hat{\sigma}_i^y\rangle = 2\beta(t)\Delta_i\langle\hat{\sigma}_i^z\rangle - 2\gamma(t)\langle\hat{\sigma}_i^x\rangle \left(h_i + \sum_{k>i} J_{ik}\langle\hat{\sigma}_k^z\rangle\right)$$

Z-component:

$$\begin{aligned}
[H_D, \hat{\sigma}_i^z] &= - \sum_{j=1}^N \Delta_j [\hat{\sigma}_j^x, \hat{\sigma}_i^z] \\
&= - \sum_{j=1}^N \Delta_j (-2i\delta_{ij} \hat{\sigma}_i^y) = 2i\Delta_i \hat{\sigma}_i^y \\
[H_P, \hat{\sigma}_i^z] &= - \sum_{j=1}^N h_j [\hat{\sigma}_j^z, \hat{\sigma}_i^z] - \sum_{j=1}^N \sum_{k>j} J_{jk} [\hat{\sigma}_j^z \hat{\sigma}_k^z, \hat{\sigma}_i^z] \\
[\hat{\sigma}_j^z, \hat{\sigma}_i^z] &= 0 \\
[\hat{\sigma}_j^z \hat{\sigma}_k^z, \hat{\sigma}_i^z] &= 0 \\
\Rightarrow [H_P, \hat{\sigma}_i^z] &= 0 \\
\Rightarrow [H, \hat{\sigma}_i^z] &= 2i\beta(t)\Delta_i \hat{\sigma}_i^y
\end{aligned}$$

Equation of motion using von Neumann equation:

$$\frac{d}{dt} \langle \hat{\sigma}_i^z \rangle = \frac{1}{i\hbar} \langle [H, \hat{\sigma}_i^z] \rangle$$

Assume $\hbar = 1$:

$$\frac{d}{dt} \langle \hat{\sigma}_i^z \rangle = -2\beta(t)\Delta_i \langle \hat{\sigma}_i^y \rangle$$

So we have:

$$\frac{d}{dt} \langle \hat{\sigma}_i^x \rangle = 2\langle \hat{\sigma}_i^y \rangle \gamma(t) \left(h_i - \sum_{k \neq i} J_{ik} \langle \hat{\sigma}_k^z \rangle \right) \quad (\text{A.1})$$

$$\frac{d}{dt} \langle \hat{\sigma}_i^y \rangle = 2\Delta_i \langle \hat{\sigma}_i^z \rangle \beta(t) - 2\langle \hat{\sigma}_i^x \rangle \gamma(t) \left(h_i - \sum_{k \neq i} J_{ik} \langle \hat{\sigma}_k^z \rangle \right) \quad (\text{A.2})$$

$$\frac{d}{dt} \langle \hat{\sigma}_i^z \rangle = -2\beta(t)\Delta_i \langle \hat{\sigma}_i^y \rangle \quad (\text{A.3})$$

The effective magnetic field $\vec{B}_i(t)$ is defined as

$$\vec{B}_i(t) = 2\beta(t)\Delta_i \vec{e}_x + 2\gamma(t)m_i(t) \vec{e}_z$$

with self-consistent magnetization:

$$m_i(t) = h_i + \sum_{k \neq i} J_{ik} n_k^z(t)$$

The classical spin vectors are defined as

$$\vec{n}_i(t) = \begin{pmatrix} n_i^x(t) \\ n_i^y(t) \\ n_i^z(t) \end{pmatrix} = \begin{pmatrix} \langle \hat{\sigma}_i^x \rangle \\ \langle \hat{\sigma}_i^y \rangle \\ \langle \hat{\sigma}_i^z \rangle \end{pmatrix}$$

The vector equation of motion is governed by

$$\partial_t \vec{n}_i(t) = \vec{n}_i(t) \times \vec{B}_i(t)$$

Component Form

We write the components of the cross product:

$$\partial_t n_i^x(t) = \frac{d}{dt} \langle \hat{\sigma}_i^x \rangle = n_i^y(t) B_i^z(t) - n_i^z(t) B_i^y(t) \quad (\text{A.4})$$

$$\partial_t n_i^y(t) = \frac{d}{dt} \langle \hat{\sigma}_i^y \rangle = n_i^z(t) B_i^x(t) - n_i^x(t) B_i^z(t) \quad (\text{A.5})$$

$$\partial_t n_i^z(t) = \frac{d}{dt} \langle \hat{\sigma}_i^z \rangle = n_i^x(t) B_i^y(t) - n_i^y(t) B_i^x(t) \quad (\text{A.6})$$