

Genetic optimization of ansatz expressibility for enhanced variational quantum algorithm performance

Manish Mallapur,^{1,*} Ronit Raj,^{1,†} and Ankur Raina^{1,‡}

¹*Department of Electrical Engineering and Computer Science,
Indian Institute of Science Education and Research, Bhopal, 462066, Madhya Pradesh, India*

(Dated: February 20, 2026)

Variational quantum algorithms have emerged as a leading paradigm that extracts practical computation from near-term intermediate-scale quantum devices, enabling advances in quantum chemistry simulations, combinatorial optimization, and quantum machine learning. However, the performance of variational quantum algorithms is highly sensitive to the design of the ansatze. To be effective, ansatze must be expressive enough to capture target states but shallow enough to be trainable. We propose a genetic algorithm-inspired framework for designing ansatze that achieve high expressibility while maintaining shallow depth and low parameter count. Our approach evolves ansatze through mutation and selection based on an expressibility metric. The circuit generated by our framework consistently demonstrates high expressibility at any target depth and performs comparably to traditional ansatz design approaches. This work presents a problem-agnostic, scalable solution for ansatz design, producing expressive, low-depth circuits that need to be designed only once and can serve a wide range of applications.

I. INTRODUCTION

Variational quantum algorithms [1] stand at the forefront of quantum computing’s transition from theoretical promises to practical applications. In the noisy intermediate-scale quantum (NISQ) era [2, 3], where quantum processors are characterized by their limited qubit counts and susceptibility to noise, variational quantum algorithms (VQAs) emerge as the leading algorithmic framework [4] for practical quantum computing applications. The primary advantage of VQAs is their ability to run shallow-depth circuits, which is crucial in NISQ devices, given their low coherence times and noise accumulation with an increase in circuit depth. The hybrid nature of this algorithm delegates the optimization process to classical computers while using the quantum computer for state preparations and state measurements. VQAs have demonstrated their practical utility across a range of applications that align with the capabilities of NISQ devices. For example, the QAOA [5] has been employed to solve combinatorial optimization, including its similar variants like the Max-Cut problem [6] and other constraint satisfaction problems [7], error correction [8, 9], and for solving linear systems of equations [10]. Additionally, Variational Quantum Eigen solver (VQE) [11] is a prominent algorithm for finding ground state energies of molecular systems.

However, the choice of ansatz determines the performance of VQAs [12]. This impact arises from an ansatz’s ability to explore the appropriate solution space, which determines how close the algorithm can approximate the true solution to the given problem. The performance of

the ansatz depends on balancing the competing requirements. On the one hand, the expressibility of an ansatz is crucial for representing the complex and wide range of states required for accurate solutions. The trainability of the ansatz is also crucial for the algorithm’s convergence. An ansatz with a huge number of quantum gates is supposedly highly expressive, but has poor trainability due to increased depth and parameter count. Several works have explored automated designs of ansatze to improve the performance of VQAs. These quantum architecture search methods aim to optimize variational circuit design, often through evolutionary strategies like genetic algorithms and gate-level optimization [13–15]. Reinforcement learning based quantum architecture search explores these quantum architectures through policy-driven agents [16]. Predictor-based QAS [17, 18] and SuperCircuit frameworks [19, 20] have gained attention due to improved scalability factors, but they still pose a high classical computational cost. The challenge is to find a balance between expressibility and trainability of the ansatz, keeping the scalability in mind.

Our work introduces a genetic algorithm-based ansatz design framework [21, 22] that addresses the limitations of current ansatz design schemes, which are often fixed and greedy in nature. Our method leverages the global search capabilities of evolutionary algorithms to explore the vast circuit design space. We guide the search towards circuits that maximize expressibility while operating on a fixed depth. In other words, we use a classical algorithm as a pre-processing step before using variational principles on an NISQ computer. We introduce depth-aware circuit optimization strategies where, at each depth, the framework identifies optimal configurations that balance gate efficiencies and expressibility. This systematic approach provides us a practical way to navigate the expressibility-trainability [23] trade-off, which is inherent in variational quantum algorithms. To validate the effectiveness of our framework, we apply

* manish21@iiserb.ac.in

† ronit21@iiserb.ac.in

‡ Corresponding author: ankur@iiserb.ac.in

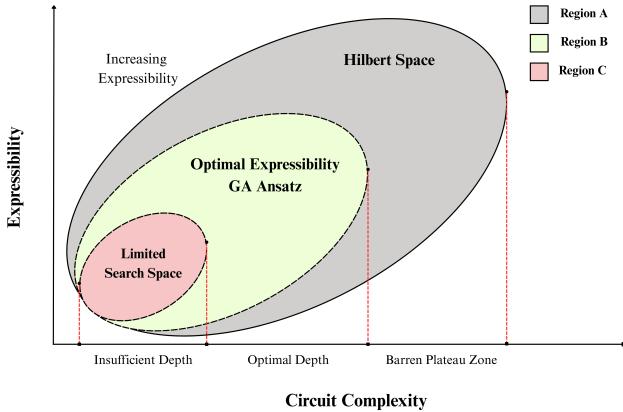


Figure 1. There is a trade-off between circuit complexity and expressibility in variational quantum circuits. The diagram shows three important areas in the expressibility-complexity landscape. Region C (red) shows a small search space because the circuit depth is not deep enough, which makes it hard to get close to the target states. The green area in Region B shows the best expressibility zone that our genetic algorithm (GA)-generated ansatz can find. It does this by choosing the best gate configurations at the right depths to balance expressibility and trainability. At high levels of expressibility and complexity, Region A (gray) shows the entire Hilbert space that can be reached. Circuits suffer from the barren plateau phenomenon, which makes them hard to train. Our depth-aware GA framework moves around these areas in real time to create resource-efficient ansatze that are highly expressive and converge quickly across a wide range of problem sizes.

the resulting ansatze to one of the most prominent VQA applications: finding the ground state energy of complex quantum systems using VQE. We benchmark their performance on a range of molecular Hamiltonians (IV B) as well as spin Hamiltonians (IV A 6), demonstrating practical utility for both quantum chemistry and condensed matter problems. Through this work, we demonstrate the potential of genetic algorithms as a ansatz design tool capable of generating resource-efficient, highly expressive circuits across various problems, sizes and depths.

This paper is organized as follows. In Section II, we briefly review the fundamentals of the VQE, including the critical tradeoff between ansatz expressibility and trainability. Section II C talks about the related works and established ansatz designs like UCCSD and ADAPT-VQE, which also serve as benchmarks. We then provide an overview of genetic algorithms in section III A and the evolutionary optimization techniques that form the basis of our method. Our novel genetic algorithm-based framework for designing highly expressive quantum circuits is detailed in Section III B. In Section IV, we present a comprehensive analysis of our framework, where we evaluate the impact of various parameters, benchmarking the generated ansatze on various Hamiltonians. Finally, we discuss the implications of our findings in Section IV E

and present our conclusion in Section V.

II. VARIATIONAL QUANTUM EIGENSOLVER (VQE)

The variational principle in quantum mechanics states that for any normalized trial wavefunction, the expectation value of the Hamiltonian operator will always give a value greater than the true ground state energy of the system. This principle has become the basis for many variational approaches, serving as a powerful theoretical framework for approximating quantum systems. Variational quantum algorithms tackle optimization problems by encoding the solution into a cost function $C(\theta)$. A parameterized quantum circuit (ansatz $U(\theta)$) is then proposed and a classical optimizer updates θ to minimize the cost function, where the minima corresponds to the problem solution. This ansatz is trained iteratively in a hybrid quantum-classical manner to solve the optimization task.

The cost function defines the solution landscape over which the hybrid loop iterates, directly influencing algorithm performance. It must be faithful (minima correspond to problem solutions), feasible to compute on quantum hardware, not classically trivial, operationally meaningful (lower cost means better solution), and smooth enough for optimization.

While the cost function provides direction for optimization, the variational ansatz defines the landscape to be explored. The ansatz acts as a template that dictates expressibility and resource efficiency. Expressibility is the ansatz's ability to span arbitrary space with respect to target states, directly impacting its ability to reach the desired solution. The ansatz structure fundamentally dictates both expressibility (how well it represents states) and trainability (how easily parameters can be optimized). However, practical issues like barren plateaus and local minima can hinder performance despite high expressibility [19]. Higher expressibility allows better representation of quantum states, which is necessary for accurate solutions, but trainability determines how easily circuit parameters can be optimized for convergence. As shown in [23], ansatze with many quantum gates become highly expressive, but noise accumulation often results in poor trainability. The goal is to produce sufficiently expressive ansatz that are resource efficient (i.e., have shallow depths) while keeping the circuits trainable.

A. System Hamiltonians of interest

To evaluate our ansatz circuits, we tested them on several molecular systems and compared results with state-of-the-art designs (UCCSD and ADAPT-VQE). The classical optimization was performed using Adam optimizers. We also tested on a non-molecular quantum system to check generalization.

1. Molecular Hamiltonians

We used H₂, LiH, BeH₂, and H₂O. Each molecular system is described by an electronic Hamiltonian in second quantization [24]:

$$\hat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s, \quad (1)$$

Where a_p^\dagger and a_p are the fermionic creation and annihilation operators that add or remove electrons in specific spin-orbitals, h_{pq} are one-electron integrals (kinetic energy and nuclear attraction), and h_{pqrs} are two-electron integrals (electron-electron repulsion).

These integrals are computed from molecular orbital data (molecular positions, bond lengths, etc.) using the STO-3G minimal basis set [25]. Since quantum computers work with qubits rather than fermions, we map this fermionic Hamiltonian to a qubit Hamiltonian using the Jordan-Wigner transformation [26, 27].

Our benchmark molecules [11] are: H₂ (simple two-electron system, four qubits), LiH (4 electrons, 12 qubits), BeH₂ (linear molecule, 6 electrons, 14 qubits), and H₂O (bent molecule, 10 electrons, 14 qubits).

2. Non-Molecular Hamiltonian: Transverse field Ising model

We also tested our approach on the transverse field Ising model (TFIM), a one-dimensional chain of spin- $\frac{1}{2}$ particles with nearest-neighbor interactions and uniform transverse magnetic field. The TFIM Hamiltonian [28, 29] is:

$$\hat{H}_{\text{TFIM}} = -J \sum_{\langle i,j \rangle} Z_i Z_j - h \sum_i X_i, \quad (2)$$

where Z_i and X_i are Pauli operators acting on qubit i , and $\langle i,j \rangle$ refers to nearest-neighbor interactions. The $Z_i Z_j$ term represents spin-spin interactions between neighbouring qubits, and the X_i term represents an external transverse magnetic field. We use $J = 1$ and $h = 1$ in our experiments.

B. Expressibility

Expressibility characterizes a PQC's ability to generate diverse quantum states across the Hilbert space. Several approaches to defining and measuring expressibility have been proposed in the literature [30–33]. In this work, we adopt the fidelity-based method introduced by [30], as it is an operationally meaningful and practically estimable measure of how well a circuit can explore the Hilbert space. A circuit is said to be highly expressible if it generates states that can approximate the uniform Haar distribution.

In practice, expressibility is estimated by sampling pairs of random parameters θ and ϕ , drawn uniformly from $[0, 2\pi]$, preparing the corresponding states $\psi(\theta)$ and $\psi(\phi)$, and computing their fidelity:

$$F = |\langle \psi(\theta) | \psi(\phi) \rangle|^2. \quad (3)$$

Repeating this procedure many times yields an empirical distribution $\hat{P}_{\text{PQC}}(F)$ of fidelities. This is compared against the known fidelity distribution of Haar-random states,

$$P_{\text{Haar}}(F) = (N-1)(1-F)^{N-2}, \quad (4)$$

where $N = 2^n$ is the Hilbert space dimension. Sim *et al.* originally proposed using the Kullback–Leibler (KL) divergence between $\hat{P}_{\text{PQC}}(F)$ and P_{Haar} distributions to quantify expressibility.

In our work, we use the Jensen–Shannon divergence, a symmetric and bounded alternative to the KL divergence.

$$E = D_{\text{JSD}}(\hat{P}_{\text{PQC}}(F) \| P_{\text{Haar}}(F)). \quad (5)$$

A lower divergence value indicates higher expressibility, reflecting a closer approximation to the Haar distribution.

C. Related work

Contemporary VQE ansatz design spans a broad spectrum, including hardware-efficient circuits tailored to device constraints [34], problem-specific constructions such as the Hamiltonian variational ansatz [35], and chemically motivated approaches that encode the molecular electronic Hamiltonian, notably Unitary Coupled Cluster with Singles and Doubles (UCCSD) [36] and adaptive schemes like ADAPT-VQE [37].

Unitary Coupled Cluster with Singles and Doubles (UCCSD) and adaptive schemes like ADAPT-VQE have undergone numerous variations [38–46]. We focus on the original UCCSD and ADAPT-VQE since we worked mainly on molecular Hamiltonians to compare them against established molecular ansatz designs.

1. UCCSD: Unitary Coupled Cluster with Singles and Doubles is a chemically inspired ansatz that adapts the accurate coupled cluster method from quantum chemistry into a unitary form suitable for quantum computation. The method constructs the ansatz by applying a unitary transformation to a reference Hartree–Fock state:

$$|\Psi(\theta)\rangle = e^{T-T^\dagger} |\Phi_0\rangle, \quad (6)$$

In the following, let $|\Phi_0\rangle$ denote the reference Hartree–Fock state; $|\Psi(\theta)\rangle$ and $|\psi(\theta)\rangle$ the parameterized ansatz wavefunctions; θ the vector of variational parameters; n the number of spin–orbitals;

a_p^\dagger and a_q the fermionic creation and annihilation operators; and \hat{H} the molecular Hamiltonian. For UCCSD, the excitation operator T is decomposed as $T = T_1 + T_2$ which includes single excitation operators:

$$T_1 = \sum_{i,a} \theta_i^a a_a^\dagger a_i, \quad (7)$$

and double excitation operators:

$$T_2 = \sum_{i < j, a < b} \theta_{ij}^{ab} a_a^\dagger a_b^\dagger a_j a_i. \quad (8)$$

with θ_i^a and θ_{ij}^{ab} the amplitude parameters for single and double excitations from occupied orbitals i, j to virtual orbitals a, b , respectively. Since direct implementation of the unitary exponential is challenging, it is approximated using a first-order Trotter–Suzuki decomposition [47] and mapped to qubits via fermion-to-qubit encodings. At the same time, UCCSD provides accurate results for small molecules, its $\mathcal{O}(n^3)$ scaling in parameterized gates with the number of spin-orbitals n limits scalability to larger systems.

2. ADAPT-VQE: Adaptive Derivative-Assembled Pseudo-Trotter Ansatz Variational Quantum Eigensolver is an iterative ansatz construction method that builds circuits one gate at a time based on energy gradients. The parameterized state is

$$|\psi(\boldsymbol{\theta})\rangle = U_N(\theta_N)U_{N-1}(\theta_{N-1}) \cdots |\Phi_0\rangle, \quad (9)$$

where each unitary

$$U_k(\theta_k) = e^{-i\theta_k \hat{A}_k}, \quad (10)$$

uses operators from a fermionic excitation pool. The algorithm iteratively computes gradients

$$g_j = \left| \langle \psi(\boldsymbol{\theta}) | [\hat{H}, \hat{A}_j] | \psi(\boldsymbol{\theta}) \rangle \right|, \quad (11)$$

and selects the operator with maximum g_j , appends its unitary to the ansatz, and optimizes all parameters until convergence. Although ADAPT-VQE can produce shallower circuits than fixed ansatze, computing gradients for all operators at each iteration incurs significant classical overhead.

III. THE ANSATZ DESIGN PROBLEM

The performance of VQAs is critically dependent on the design of the parameterized quantum circuit, or ansatz. On one hand, it requires sufficient expressibility to explore the relevant regions of the Hilbert space and accurately represent the target quantum state. On the other hand, increasing circuit complexity to boost expressibility often introduces significant challenges for Noisy Intermediate-Scale Quantum (NISQ) hardware, including increased susceptibility to noise and the emergence of optimization-hindering phenomena like barren plateaus. The task is therefore to find resource-efficient circuits that do not sacrifice performance. Given the vast, high-dimensional, and non-convex nature of the search space for circuit architectures, heuristic methods offer a promising path forward. In this section, we introduce an approach based on a genetic algorithm to systematically search for and evolve ansätze that balance these competing requirements.

A. Genetic Algorithm

Genetic algorithms [48, 49] are a heuristic evolutionary optimization algorithm inspired by natural selection, where in a population, only those individuals with the most suitable genetic traits are able to propagate and pass their characteristics to subsequent generations. This translates into a powerful computational framework that has proven remarkably effective in tackling complex optimization landscapes. Genetic algorithms [50] are found to work well in many cases where traditional optimization techniques struggle, such as problems with non-convex search spaces, scenarios requiring non-gradient optimization, or when working with high-dimensional data where classical methods often get trapped in local optima. For our specific case, the genetic algorithm is able to optimize the ansatz for expressibility given the constraints like the number of qubits and maximum circuit depth, essentially searching through the vast space of possible quantum circuit configurations to find those that best balance performance and resource requirements. Genetic algorithms emulate the process of evolution through biologically inspired operations like selection (where fitter individuals have higher chances of reproduction), crossover (where genetic material is exchanged between parent solutions to create offspring), and mutation (where small random changes are introduced to maintain diversity and prevent premature convergence) to produce high-quality solutions to the optimization problem. The iterative nature of this process allows the algorithm to gradually improve the population quality over successive generations, with each cycle potentially discovering better circuit designs that might be difficult to find through traditional optimization approaches. The general loop of a genetic optimization algorithm is as shown in Figure 2, which illustrates how these evolutionary principles work together to guide the search toward increasingly effective solutions.

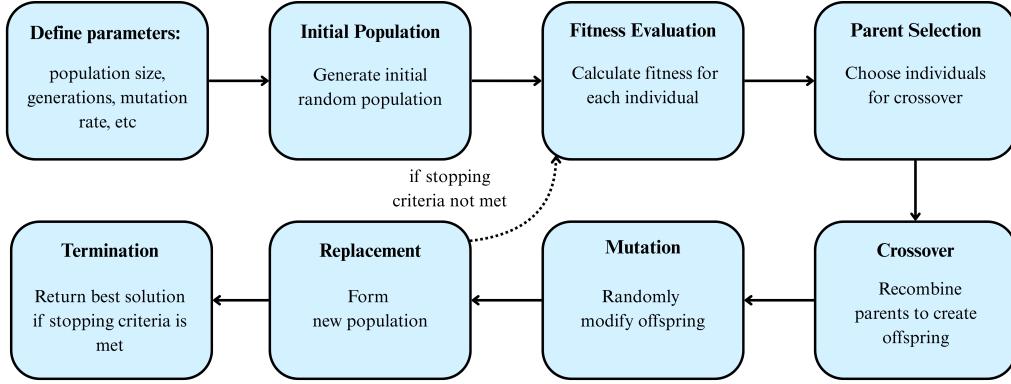


Figure 2. Overview of the general Genetic Algorithm: parameter initialization, generation of the initial population, iterative fitness evaluation, selection of parents, crossover to produce offspring, mutation for genetic diversity, population replacement, and termination upon meeting the stopping criteria.

B. Our ansatz design algorithm

In this work, we develop a framework inspired by genetic algorithms to identify highly expressible circuits (ansatze) with limited circuit depth. This evolutionary approach is well-suited to this task, as we have to search through a non-convex and high-dimensional landscape of quantum circuit designs. Our approach encodes quantum circuits as individuals in the population, where each circuit's expressibility is treated as its fitness score. Through the iterative process of selection, crossover, and mutation, the population evolves toward circuits with higher expressibility.

A) Representation: Each individual in the population of the genetic algorithm (GA) is a unique quantum circuit ansatz. The individual is represented as a sequence of quantum gates arranged layer-by-layer, along with a set of corresponding parameters for the parameterized gates (RX, RY, and RZ). We treat the ordered set of all the gates in the circuit as the genome of the individual, with each individual gate, along with its parameters, functioning as a gene. This genome-based encoding facilitates genetic operations such as crossover and mutation in the evolutionary process.

B) Initialization: The population is initialized with random quantum circuits up to a specified depth. Each layer is built randomly, choosing gates from the predefined gate sets (gates = [RX, RY, RZ, H, I, and CNOT]). For gates with parameters, the parameters are uniformly sampled randomly from $[0, 2\pi]$.

C) Fitness Evaluation: The fitness of an individual is defined by the expressibility of corresponding ansatze. Expressibility is measured by calculating the Jensen-Shannon divergence (JSD) between the distributions of the fidelities [calculated from

random sampling of the circuit] and the Haar random distribution of the Hilbert space. A lower JSD value suggests higher expressibility, making the fitness inversely proportional to JSD.

D) Selection, Crossover, and Mutation:

- Selection: Individuals are ranked on the basis of their expressibility from most expressible (lowest JSD) to least expressible (highest JSD) and choose the top ' k ' individuals as parents for the next generations.
- Crossover: In generating new offspring, the circuits are split between layers and recombin ed. This introduces the structural variations in the offspring compared to their parents.
- Mutation: A random subset of gates is chosen to be mutated by replacing a gate with a random gate or changing the order of gates.

After several generations, the genetic algorithm framework converges towards ansatze with maximized expressibility for a given depth and qubit count.

The following section is a step-by-step implementation of our genetic algorithm to optimize for the expressibility of the ansatz for given constraints like number of qubits, circuit depth, or number of parameters, as shown in Fig. 3.

A) Constraints for optimization:

- Number of qubits
- Number of layers
- Population size
- Number of generations
- Mutation rate

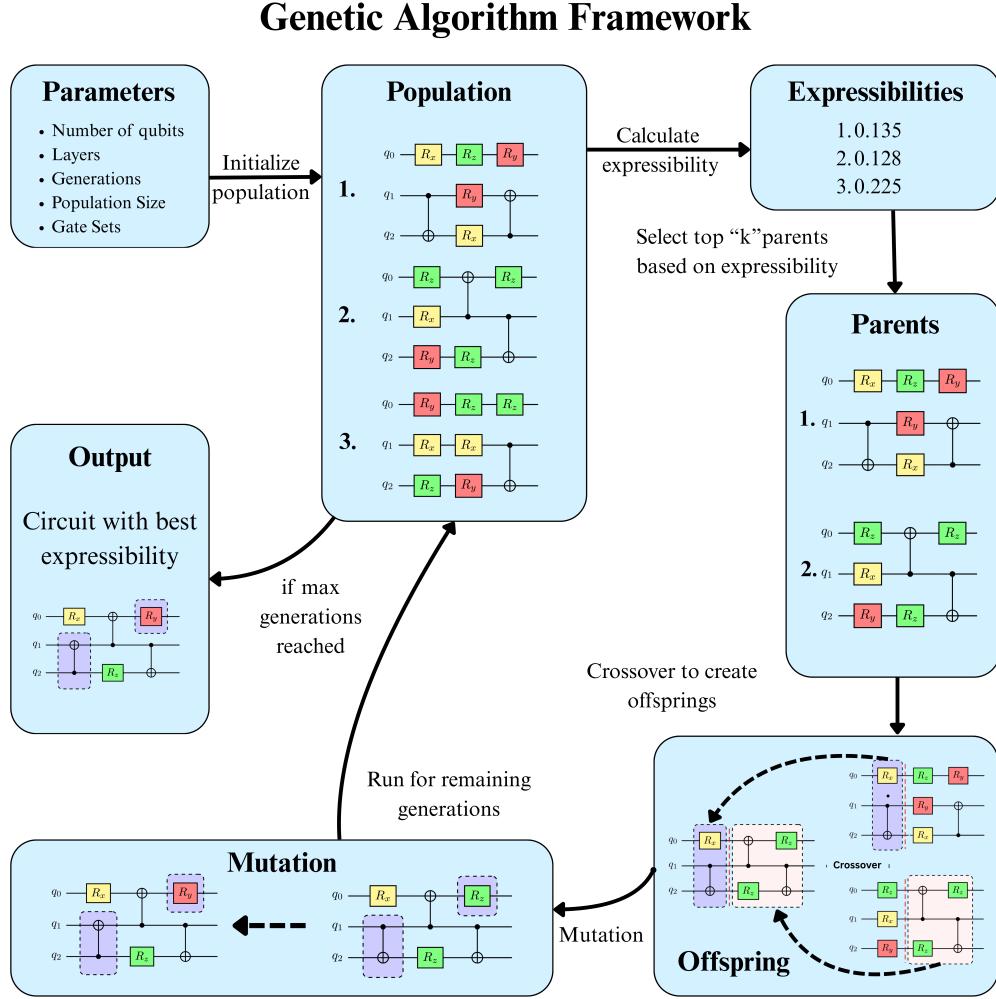


Figure 3. Genetic algorithm framework for ansatz optimization. A population of random quantum circuits is initialized and evaluated using expressibility as the fitness metric. Top-performing circuits are selected as parents for crossover, generating offspring by recombining gate sequences. Offspring undergo mutation to maintain population diversity. The process iterates over generations, yielding an ansatz with maximal expressibility under depth and qubit constraints.

- f) Number of samples for expressibility calculation
 - g) Available quantum gates: RX, RY, RZ, H, CNOT
- B) Initialize the population: Generate population size quantum circuits, each with a number of layers. Gates are randomly chosen from the available set. For parameterized gates (RX, RY, RZ), assign random parameters in the range $[0, 2\pi]$.
- C) Evaluate expressibility for each circuit:
- Sample two random parameter sets and compute the corresponding quantum states.
 - Compute fidelity $F = |\langle \psi(\theta) | \psi(\phi) \rangle|^2$.
 - Construct a histogram of fidelities and compare it to the Haar-random distribution using

- JSD divergence as described in Eq. 5 :
- $$E = D_{\text{JSD}}(P_{\text{circuit}}(F) \parallel P_{\text{Haar}}(F)).$$
- d) A lower expressibility score (E) indicates a better circuit.
- D) Select parents for the next generation: Rank circuits based on their expressibility scores and select the top k (e.g., 5) circuits.
- E) Generate offspring via crossover and mutation:
- Crossover: Select two parent circuits, and their genes are crossed to produce the offspring. This process can be done in two ways:
 - Single Point Crossover: We choose a random crossover point (layer index), and combine parts from both parents to form a child circuit.

- ii. N point crossover: Here, instead of choosing a single point to split the two parents, N random points are chosen on both parents, thus dividing them into N+1 parts that are then alternatively joined to give the offspring.
 - b) Mutation: With probability mutation rate, randomly modify a gate in the child circuit. These mutations are of two types:
 - i. Single qubit gate: If a single qubit gate is selected to be mutated, then it is replaced by another single qubit gate from the gate set.
 - ii. Two qubit gate (CNOT gate): If a CNOT gate is chosen to be mutated, then the control and target of the CNOT gate are swapped.
- F) Update population and iterate: Replace the old population with the new generation and repeat the process for the number of generations iterations.

Algorithm 1: The Ansatz design algorithm

Input: n (qubits), L (circuit depth), P (population size), S (samples), G (generations), k (parent count), m (mutation prob.)

Output: Ansatz with optimal expressibility

- 1 **Step 1: Initialize Population**
- 2 Create a population of P random circuits with the given input constraints and compute their expressibility. (Alg. 3)
- 3 **for** $g \leftarrow 1$ **to** G **do**
- 4 **Step 2: Selection and Crossover**
- 5 The best parents are selected on the basis of their expressibility. The parents are crossed and mutated to generate the offspring population using the current population. (Alg. 4)
- 6 **Step 3: Population Update**
- 7 Replace the current population with the offspring and track the best circuit. (Alg. 5)
- 8 **return** Best ansatz found over G generations

C. Generalizability of the GA-derived ansatz

Instead of designing a new ansatz for each individual problem, we aim to build one general ansatz. We can use this circuit for different Hamiltonians, reducing the overhead of running the genetic algorithm every time we look at a new system. Once the circuit is designed, it works as an ansatz for different target Hamiltonians with some parameter optimization.

It lowers the classical overhead because it avoids running the genetic algorithm repeatedly, making the pro-

cess simpler. It allows for faster experimentation as we can reuse the ansatz for different Hamiltonians.

Section IV A 6 and IV B show that the ansatz, which was not designed with a particular target Hamiltonian in mind, works well across different molecular and spin Hamiltonians.

IV. RESULTS

In this section, we provide a comprehensive analysis of the genetic algorithm (GA) framework developed to optimize ansatz expressibility. We systematically investigate the impact of various parameters, including the number of generations, circuit depth, gate set composition, and gate count, on the performance of the generated ansatze. Furthermore, we evaluate the real-world applicability of these GA-designed ansatze by benchmarking their performance on the variational quantum eigensolver (VQE) for both spin and molecular Hamiltonians. Our findings highlight key insights into designing efficient and effective ansatz for near-term quantum applications.

A. Genetic Algorithm (GA) framework

To characterize the performance of our GA framework, we systematically varied its parameters. We start by analyzing the convergence of expressibility across generations to determine the optimal stopping point of the algorithm (IV A 1). We then proceed to explore the relationship between circuit depth and expressibility, and identify the point of saturation where deeper circuits yield diminishing returns (IV A 2). A central part of our analysis also investigates how different gate sets can impact expressibility (IV A 3), resource requirement in terms of gate count (IV A 4), and classical runtime (IV A 5) of the framework to understand the tradeoffs involved in the ansatz design process.

Before presenting the results, we first define the gate sets used throughout this work. The performance of the genetic algorithm and the resulting ansatz depends strongly on the choice of gates and their connectivity constraints. All subsequent analyses assume the gate set definitions summarized in Table I.

1. Number of generations

We began our analysis by determining the convergence behavior of the GA. As Fig. 5 illustrates, we observed the best expressibility scores for each gate set over ten generations. The plot (Fig. 5) clearly shows that the GA framework finds more expressive ansatze over generations. For nearly all gate sets, the improvement saturates and plateaus around the tenth generation. This indicated that 10 generations provide near optimal expressibility without unnecessary computational overhead.

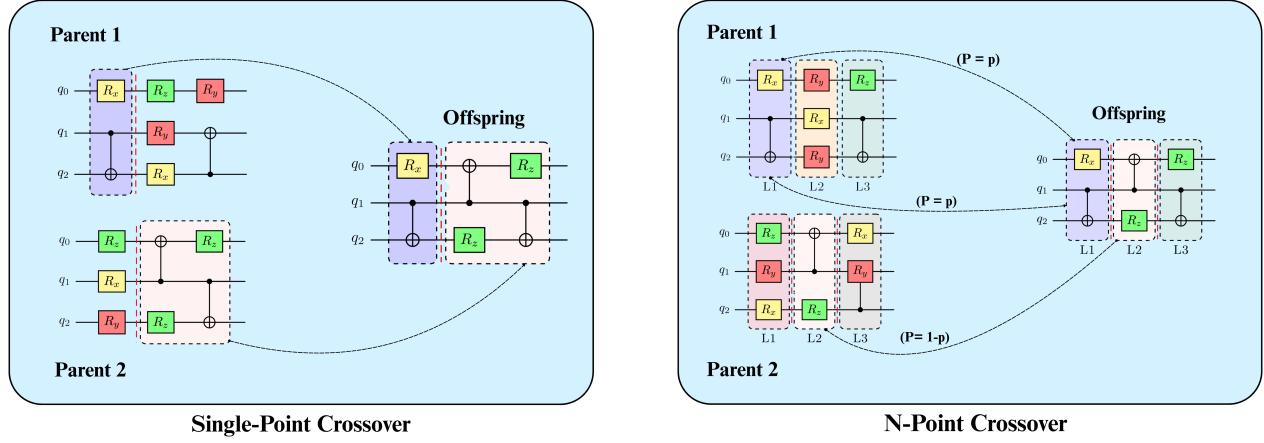


Figure 4. Single-point crossover. A single point splits the parent circuits, and the tail segments from both parents exchange to generate offspring that inherit partial gate sequences from both parents. N -point crossover. Multiple crossover points are selected across parent circuits, and segments are probabilistically swapped to create offspring that are diverse and preserve structural traits from both parents.

Table I. Description of Gate sets A–I. Each gate set specifies the allowed single- and two-qubit operations, including nearest-neighbour CNOT connectivity and the presence of an initial Hadamard layer. An asterisk (*) indicates that the CNOT gate is restricted to nearest-neighbor qubits. Gate sets marked with a double dagger (\ddagger) include an initial layer of Hadamard (H) gates.

Set	Gates
A	{RX, RY, RZ, H, CNOT*}
B	{RX, RY, H, CNOT*, I}
C	{RX, RY, RZ, H, CNOT, I}
D	{RY, RZ, H, CNOT, I}
E	{RY, RZ, CNOT, I}
F \ddagger	{RX, RY, RZ, CNOT, I}
G	{RX, RY, RZ, H, CNOT, I}
H \ddagger	{RX, RY, RZ, H, CNOT*}
I	{RX, RY, RZ, CNOT, I}

2. Circuit depth

After establishing the necessary generation convergence and we move to investigate the relationship between expressibility and circuit depth. Fig. 6 plots the expressibility for gate set A across systems of 4, 6, 10, 14 qubits. A consistent trend is observed across all systems: expressibility significantly improves as circuit depth increases, but only up to a saturation point. The saturation indicates that adding more layers yields diminishing results beyond a certain depth. Our results suggest that

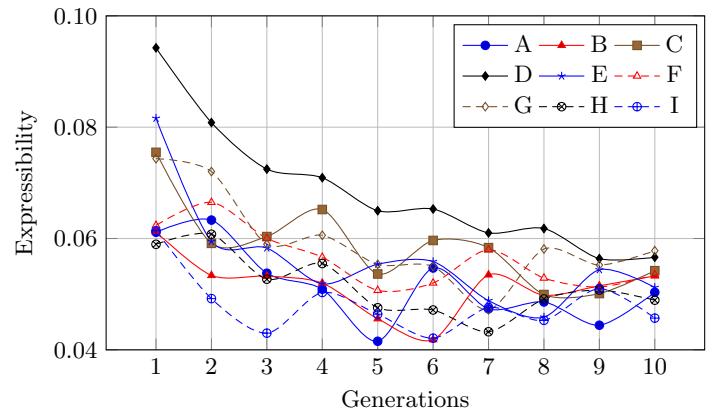


Figure 5. The plot shows the expressibility metric (y-axis) as a function of the GA framework generations (x-axis) for all gate sets (A–I) on a four-qubit system. A lower metric value indicates better expressibility. All curves show improvement in expressibility through generations, which consistently saturates by generation 10. We additionally verify this by extending the GA framework to 20 generations (see Figure 14 in the appendix), which shows that expressibility gains are minimal, confirming 10 generations as sufficient optimization duration.

this optimal depth scales linearly with the number of qubits, where optimality is defined empirically in terms of expressibility saturation, *optimal depth* $\approx 2n$. This finding is significant as it demonstrates that high expressibility can be achieved without requiring deeper circuits.

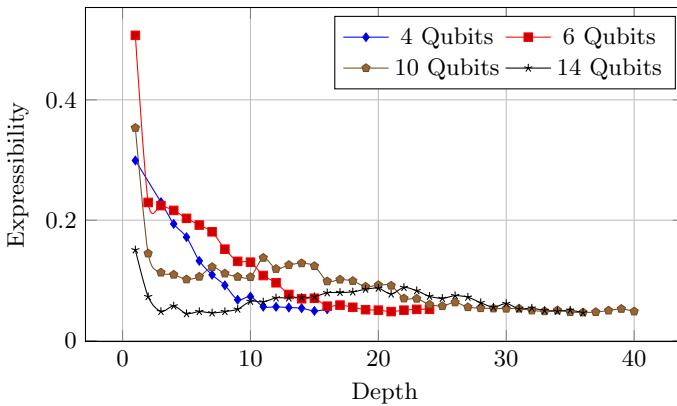


Figure 6. Expressibility saturation with increasing circuit depth. The plot shows the expressibility metric (y-axis) as a function of circuit depth (x-axis) for ansatze constructed using gate set A on systems of 4, 6, 10, and 14 qubits. The expressibility rapidly improves with initial increases in depth before reaching a plateau, which indicates that optimal expressibility is achieved at a circuit depth ($\approx 2n$ depth, where n is the number of qubits) that scales linearly with the number of qubits.

3. Gate set

The composition of the gates, single-qubit gates like (RX, RY, RZ, H) and two-qubit gates (CNOT) [51] is the primary factor influencing the GA’s search for an optimal ansatz. We evaluated nine gate sets (A-I), detailed in Table I, to understand this relationship. The resulting expressibility for each set is presented in Figure 7, which reveals a performance gap across all test system sizes and gate sets. Gate sets A, B, and H consistently produce ansatze with higher expressibility. The distinguishing feature of these top-performing sets is a structural constraint of the CNOT gate to be applied only on adjacent qubits. This locality constraint introduces more double and single-qubit gates, resulting in a high gate and parameter count. The higher count of entangling operations of CNOT allows the ansatz to more effectively and uniformly explore Hilbert space, which is the direct cause of its enhanced expressibility score.

In contrast, gate sets such as D, E, and I exhibit slightly lower expressibility compared to other gate sets while achieving this expressibility with significantly fewer gates and parameters. This improved resource efficiency leads to more favorable optimization landscapes, enhancing trainability and VQE performance.

4. Gate Count

To understand the resource cost associated with the expressibility of each gate set, we analyze the gate count composition of the resulting circuits. Fig. 8 illustrates the breakdown of parametrized, unparametrized, and to-

tal gate counts for the ansatze evolved by our GA framework on a 4-qubit, depth-16 system. We find a direct correlation between high expressibility and gate count. Notably, gate sets with superior expressibility- A, B, and H also yield circuits with the highest total number of gates. Conversely, gate set I is the most efficient, requiring the fewest total gates, but this efficiency comes at the cost of expressive power. This highlights a fundamental tradeoff in ansatz design: while a greater number of gates can improve expressiveness, it increases complexity and potential susceptibility to hindrance in trainability. Therefore, an optimal gate set must balance achieving sufficient expressibility with the minimum necessary gate count to ensure better trainability.

5. Time taken for classical optimization

Beyond the quantum resources required, the classical computational cost of the ansatz search is a key consideration. Figure 9 shows the runtime of our GA framework, revealing that the optimization time scales with both the number of qubits and depth for all gate sets. This scaling with both qubit count and depth is further quantified in Table IV. Gate sets also have an impact on the computational cost. However, the classical runtime should be viewed as a one-time investment rather than a recurring operational cost. Once the GA discovers an optimal circuit structure, that ansatz becomes a reusable tool for different Hamiltonian and VQA problems, without repeating the search process. This approach effectively amortizes the classical overhead.

6. Performance on VQE

After analyzing the structural properties of the ansatze, we evaluated their practical performance on the Variation Quantum Eigensolver (VQE). We used the Transverse-Field Ising Model (TFIM) as our test bed to assess how effectively each ansatz could find the ground-state energy of the Hamiltonian. Figure 10 plots the energy throughout the VQE optimization for an 8-qubit, 16-depth circuit. While most ansatze drive the energy toward the ground state, sets B, G, and H perform poorly, either converging slowly or getting trapped in local minima far from the true ground-state energy.

To assess the performance of GA ansatze, we analyze the energy error (percentage %) after convergence across multiple system sizes, as shown in Figure 11. The plot shows that gate set E and A consistently achieve the lowest energy errors, demonstrating their superior performance in the VQE task. These results lead to a crucial insight: high expressibility alone does not guarantee VQE success, as seen in the case of the gate sets B and H. The superior performance of set E suggests that an optimal ansatz for VQE requires a balance between sufficient expressibility and a resource-efficient gate set that promotes better trainability. This shows us that ansatz

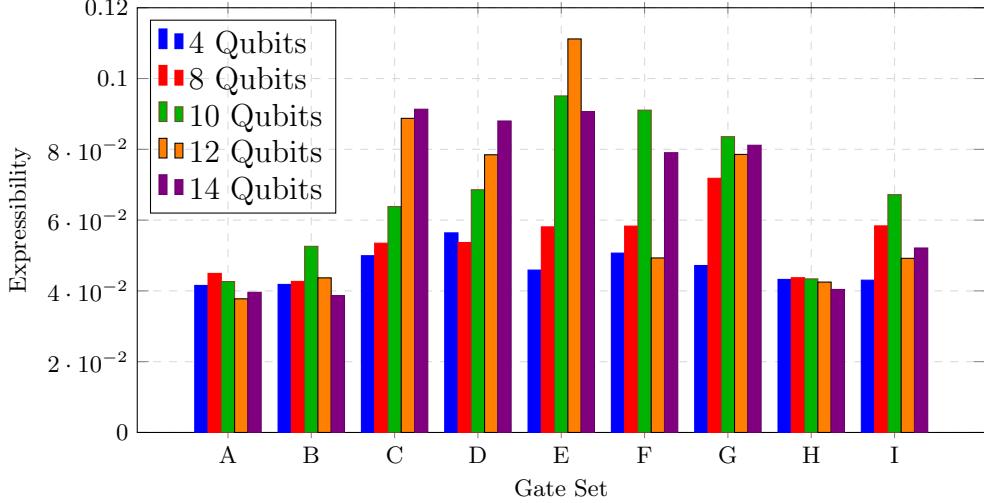


Figure 7. The chart compares the best expressibility (y-axis) achieved by the GA framework for nine gate sets (x-axis) across various system sizes, with their compositions defined in Table I. Gate sets A, B, and H consistently demonstrate superior performance. Their higher expressibility is linked to the adjacent-CNOT constraint, which results in the generation of circuits with a larger number of entangling gates.

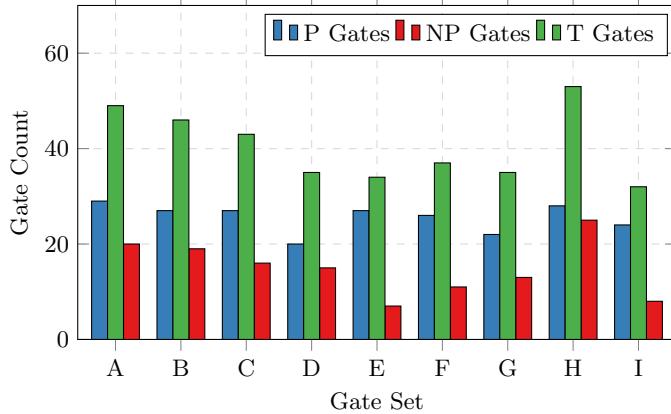


Figure 8. The bar plot shows the gate counts for GA designed ansatze on a 4-qubit, depth-16 circuit, categorized by gate set (x-axis). The bars represent the number of parametrized (P Gates), non-parametrized (NP Gates), and total (T Gates) gates. A strong correlation is observed between highly expressive gate sets (A, B, and H) and higher total gate counts, while more resource-efficient gate sets like E and I show lower expressibility.

design must properly balance expressibility and resource efficiency to perform well as a general-purpose ansatz.

B. Molecular Hamiltonian benchmarking:

To place this performance in context, Table II compares the final energies obtained using the GA ansatz with those achieved by two widely used approaches: the chemistry-inspired UUCSD and the adaptive ADAPT-VQE. The results clearly show that the GA ansatz per-

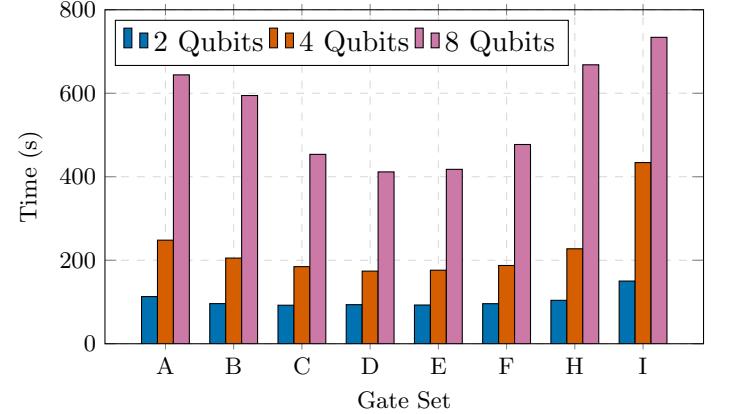


Figure 9. The classical execution time (y-axis) required to run the genetic algorithm for 10 generations is shown as a function of gate sets (x-axis) and system size (2, 4, and 8 qubits). The runtime scales with the number of qubits, the gate set, and depth.

forms competitively, yielding final energies comparable to these established methods. While chemical accuracy is achieved for the smallest system (H_2), for larger molecules the GA–VQE energies remain very close to the Exact Diagonalization values, with deviations on the order of 10^{-2} Hartree, despite the restricted circuit depth.

Notably, this accuracy is obtained with a reduced circuit depth. For LiH , BeH_2 , and H_2O , the GA ansatz requires depths of at most 48, whereas UCCSD depths exceed 1400 and ADAPT-VQE depths scale in the order of 200 – 300. This substantial reduction in circuit depth indicates that the GA ansatz is resource-efficient and scalable. These results demonstrate that the GA frame-

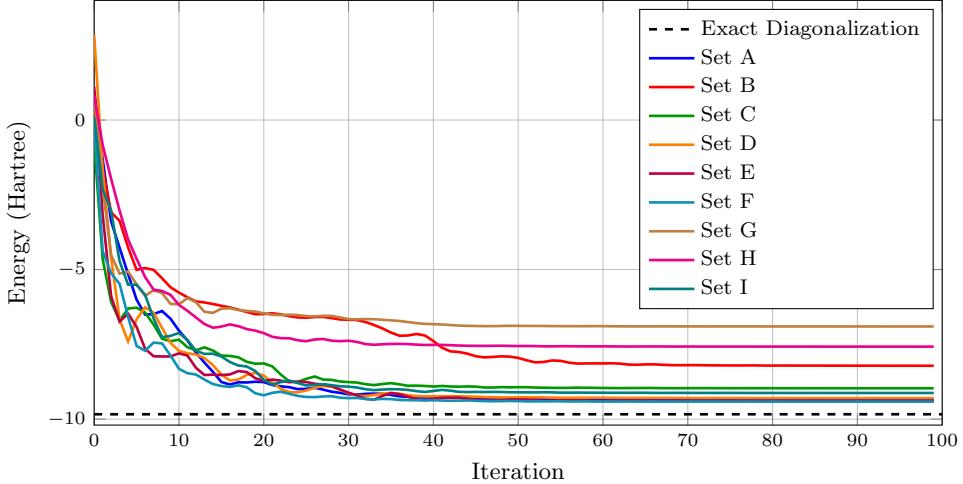


Figure 10. The plot tracks the energy (y-axis) as a function of optimization iterations (x-axis) for an 8-qubit TFIM Hamiltonian with a 16-depth circuit. The dashed line represents the analytically calculated ground-state energy. While most gate sets converge accurately, ansatze from gate sets B, G, and H struggle to reach the ground state, highlighting trainability issues.

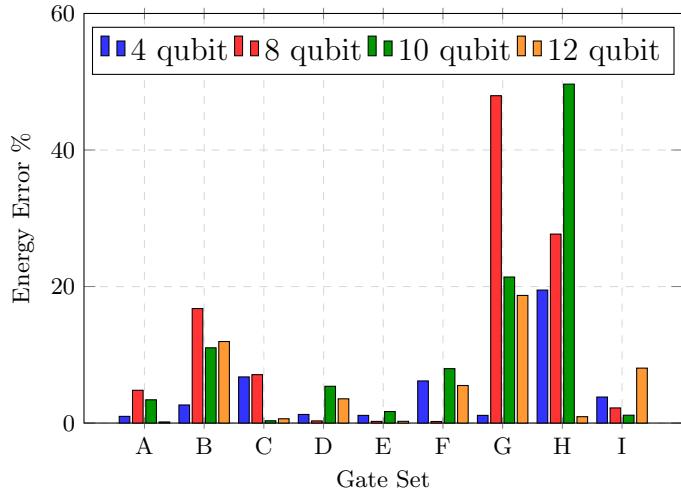


Figure 11. We test our GA ansatz on the TFIM with coupling ($J = 1$) and transverse field ($h = 1$) for different system size ranging from 4 qubits to 12 qubits. Energy errors for each gate set is reported. We see a consistent trend where most gate sets converge near to the ground state energy except B, G and H.

work can generate reusable, problem-agnostic ansatze that achieve systematic convergence and competitive accuracy while significantly reducing circuit complexity.

The final energy values of all three ansatze are compared in Table II.

C. Noisy simulations

To assess the performance of the ansatz produced by the genetic algorithm and compare it to ADAPT under noisy conditions, we repeat the H_2 simulation for both ansatze in the presence of noise. We use a noise

Table II. Performance benchmark against UCCSD and ADAPT-VQE. Comparison of final ground-state energies (Hartree) and circuit depths for different molecular systems using the GA-designed ansatz, UCCSD, ADAPT-VQE, and Exact diagonalization.

Mol	Metric	GA	UCCSD	ADAPT	Exact Diag.
H_2	Energy	-1.13608	-1.13608	-1.13608	-1.13608
	Depth	12	20	24	-
LiH	Energy	-7.86266	-7.8817	-7.8822	-7.88266
	Depth	48	1423	208	-
BeH ₂	Energy	-14.84842	-14.854	-14.876	-14.87803
	Depth	48	1423	192	-
H_2O	Energy	-73.212	-73.22	-73.2244	-73.226
	Depth	48	1442	284	-

model derived from a Qiskit backend with $N_{\text{shots}} = 1000$. The noise model is constructed from error data generated and sampled randomly from historical IBM backend data and incorporated into PennyLane as shown in their tutorial [52].

We change the optimizer to SPSA (gradient-free) for the GA ansatz, as gradient-free optimizers perform better in noisy conditions. For H_2 we observe that the GA circuits have a lower depth than the corresponding ADAPT circuits which translates into better noise resilience and lower mean final energy closer to the true ground-state energy, as we can see from the Table III.

D. Data Availability:

The quantum circuits produced by our Genetic Algorithm (GA) are provided as supplementary material [53]. Access to this repository is available upon request. Interested researchers may contact the authors for access.

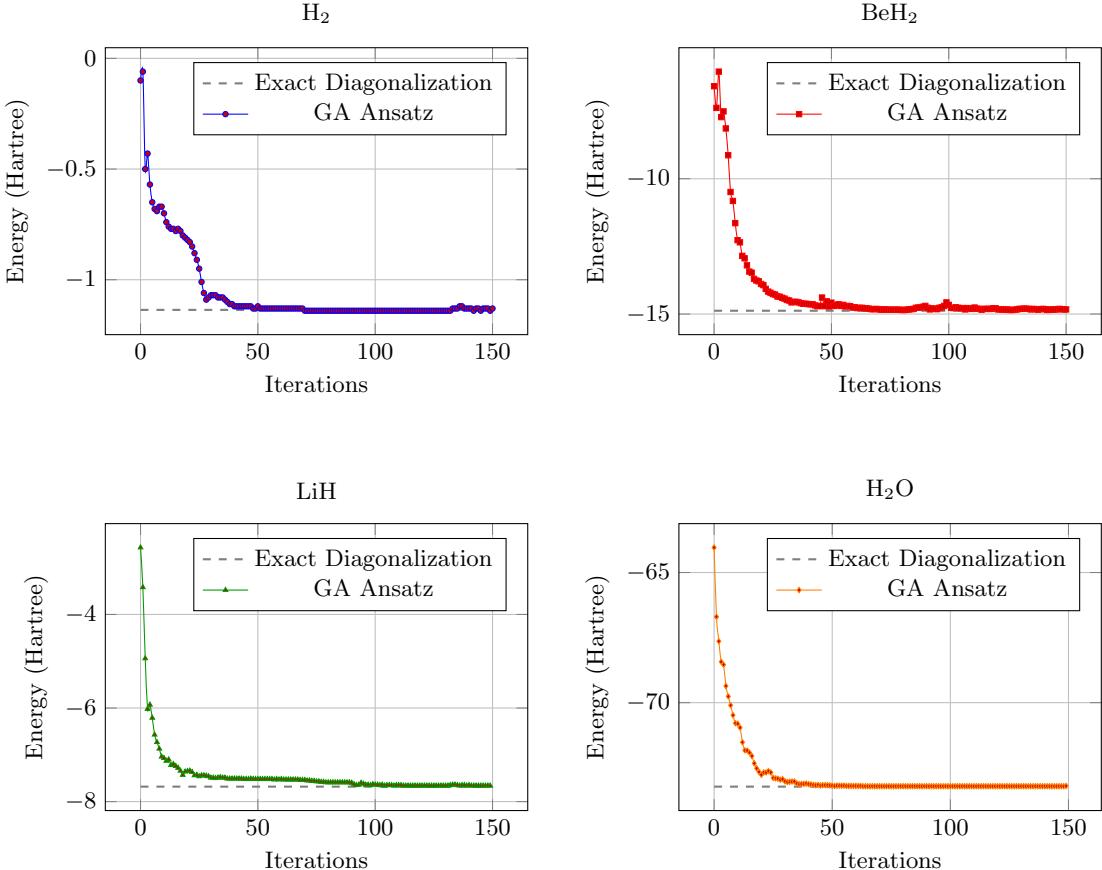


Figure 12. VQE convergence of the GA ansatz on Molecular Hamiltonians. VQE optimization for GA-designed ansatz on four benchmark molecules, H_2 , LiH , BeH_2 , and H_2O . Each plot shows the calculated energy (y-axis) converging to the exact ground state energy (dashed line) as a function of optimization iterations (x-axis). The rapid convergence across all four systems demonstrates the effectiveness and general applicability of the GA ansatz for quantum chemistry problems.

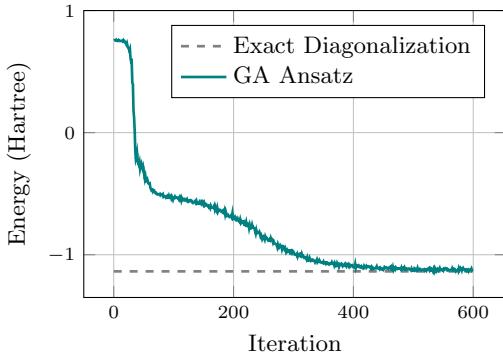


Figure 13. Convergence plot for the Genetic Ansatz optimization on H_2 under noisy conditions.

E. Discussion

In this work, we demonstrate that a genetic-algorithm-based framework can generate expressive, resource-efficient ansatze that are reusable across different Hamiltonians, eliminating the need to design a new circuit for

Table III. Comparison of noisy simulation results for ADAPT-VQE and the GA evolved ansatz on the H_2 Hamiltonian. SEM stands for standard error of mean, and the lowest observed absolute energy error is with respect to the exact ground-state energy obtained from diagonalization of the H_2 Hamiltonian.

Ansatz	Mean energy	SEM	Lowest energy error
ADAPT	-1.1414401	4.90×10^{-3}	4.40×10^{-3}
GA ansatz	-1.1352861	1.92×10^{-4}	4.17×10^{-4}

each system. By evolving a single ansatz and then directly applying it with only classical parameter optimization (for both molecular and non-molecular systems), we achieve several key strengths.

First, the reusability of our approach is demonstrated by the fact that the same evolved circuit structure was applied successfully to LiH , BeH_2 , H_2O , and the transverse-field Ising model(12 qubit (Set E)), converging within 1% of the exact ground-state energies without re-running the genetic search. Second, our method ex-

hibits competitive performance: across all benchmarks, our GA-derived ansatze perform almost on par with the problem-specific designs such as UCCSD and ADAPT-VQE. Notably, Gate Set E consistently yielded the lowest final energy errors, and our circuits match UCCSD's accuracy while using $O(n^2)$ parametrized gates rather than $O(n^3)$ for UCCSD. Third, the evolved circuits demonstrate resource efficiency by maintaining shallow depths and minimal gate counts. Finally, our approach provides reduced classical overhead since once the ansatz is generated, only classical optimizations are needed for new problems, saving the significant classical runtime otherwise spent on repeated architecture searches.

Therefore, we have obtained problem-agnostic circuits that balance expressibility and resource efficiency, perform robustly on various Hamiltonians, and greatly simplify the VQA workflow on NISQ devices.

V. CONCLUSION

Designing effective ansatze for variational quantum algorithms requires balancing expressibility and resource efficiency. Circuits must be able to represent target quantum states while remaining optimizable on noisy intermediate-scale quantum devices. Our genetic algorithm framework addresses this challenge by systematically evolving quantum circuit architectures that maximize expressibility within fixed depth constraints.

Our results demonstrate that genetically optimized ansatze achieve performance comparable to established

methods like UCCSD and ADAPT-VQE across multiple molecular systems, while using fewer parameterized gates. The key advancement lies in the reusability of evolved circuits: once generated, these ansatze can be applied across different Hamiltonians with only classical parameter optimization, reducing the computational overhead of repeated circuit design for new quantum problems.

With this work, we provide an ansatz design algorithm for variational quantum algorithms, where a single evolved circuit architecture serves multiple quantum simulation tasks without sacrificing accuracy while being resource efficient. The demonstrated transferability of our approach suggests that genetic optimization represents a viable strategy for developing practical ansatz for variational quantum algorithms on near-term quantum hardware.

ACKNOWLEDGMENTS

We are grateful to the High Performance Computing (HPC) facility at IISER Bhopal, where large-scale calculations in this project were run. We thank Mainak Bhattacharyya for his valuable insights and the members of the QuCIS Lab for their constructive discussions throughout the course of this project. The authors thank the funding received from the Department of Science and Technology under the National Quantum Mission.

-
- [1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, *Nature Reviews Physics* **3**, 625 (2021).
 - [2] J. Preskill, Quantum computing in the nisq era and beyond, *Quantum* **2**, 79 (2018).
 - [3] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, *et al.*, Noisy intermediate-scale quantum algorithms, *Reviews of Modern Physics* **94**, 015004 (2022).
 - [4] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. Akash-Narayanan, A. Asadi, *et al.*, PennyLane: Automatic differentiation of hybrid quantum-classical computations, arXiv preprint arXiv:1811.04968 (2018).
 - [5] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv preprint arXiv:1411.4028 (2014).
 - [6] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximate optimization algorithm for maxcut: A fermionic view, *Physical Review A* **97**, 022304 (2018).
 - [7] C. Y.-Y. Lin and Y. Zhu, Performance of qaoa on typical instances of constraint satisfaction problems with bounded degree, arXiv preprint arXiv:1601.01744 (2016).
 - [8] P. D. Johnson, J. Romero, J. Olson, Y. Cao, and A. Aspuru-Guzik, Qvector: an algorithm for device-tailored quantum error correction, arXiv preprint arXiv:1711.02249 (2017).
 - [9] M. Bhattacharyya and A. Raina, Quantum approximation optimization algorithm for the trellis based viterbi decoding of classical error correcting codes, arXiv preprint arXiv:2304.02292 (2023).
 - [10] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Physical review letters* **103**, 150502 (2009).
 - [11] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, A variational eigenvalue solver on a photonic quantum processor, *Nature communications* **5**, 4213 (2014).
 - [12] T. Hubregtsen, J. Pichlmeier, P. Stecher, and K. Bertels, Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability, *Quantum Machine Intelligence* **3**, 9 (2021).
 - [13] C. P. Williams and A. G. Gray, Automated design of quantum circuits, in *NASA International Conference on Quantum Computing and Quantum Communications* (Springer, 1998) pp. 113–125.
 - [14] L. Lamata, U. Alvarez-Rodriguez, J. D. Martin-Guerrero, M. Sanz, and E. Solano, Quantum autoencoders via quantum adders with genetic algorithms, *Quantum Science and Technology* **4**, 014007 (2018).

- [15] Y. Huang, Q. Li, X. Hou, R. Wu, M.-H. Yung, A. Bayat, and X. Wang, Robust resource-efficient quantum variational ansatz through an evolutionary algorithm, *Physical Review A* **105**, 052414 (2022).
- [16] E.-J. Kuo, Y.-L. L. Fang, and S. Y.-C. Chen, Quantum architecture search via deep reinforcement learning, arXiv preprint arXiv:2104.07715 (2021).
- [17] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Neural predictor based quantum architecture search, *Machine Learning: Science and Technology* **2**, 045027 (2021).
- [18] Z. He, X. Zhang, C. Chen, Z. Huang, Y. Zhou, and H. Situ, A gnn-based predictor for quantum architecture search, *Quantum Information Processing* **22**, 128 (2023).
- [19] Y. Du, T. Huang, S. You, M.-H. Hsieh, and D. Tao, Quantum circuit architecture search for variational quantum algorithms, *npj Quantum Information* **8**, 62 (2022).
- [20] H. Wang, Y. Ding, J. Gu, Y. Lin, D. Z. Pan, F. T. Chong, and S. Han, Quantumnas: Noise-adaptive search for robust quantum circuits, in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* (IEEE, 2022) pp. 692–708.
- [21] F. M. Creevey, C. D. Hill, and L. C. L. Hollenberg, Gasp: a genetic algorithm for state preparation on quantum computers, *Scientific Reports* **13**, 11956 (2023).
- [22] L. Sünkel, D. Martyniuk, D. Mattern, J. Jung, and A. Paschke, Ga4qco: genetic algorithm for quantum circuit optimization, arXiv preprint arXiv:2302.01303 (2023).
- [23] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, *PRX Quantum* **3**, 10.1103/prxquantum.3.010313 (2022).
- [24] T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (Wiley, 2000).
- [25] W. J. Hehre, R. Ditchfield, and J. A. Pople, Self-consistent molecular orbital methods. xii. further extensions of gaussian-type basis sets for use in molecular orbital studies of organic molecules, *The Journal of Chemical Physics* **56**, 2257 (1972).
- [26] P. Jordan and E. Wigner, Über das paulische Äquivalenzverbot, *Zeitschrift für Physik* **47**, 631 (1928).
- [27] J. T. Seeley, M. J. Richard, and P. J. Love, The bravyi-kitaev transformation for quantum computation of electronic structure, *The Journal of Chemical Physics* **137**, 224109 (2012).
- [28] S. Sachdev, *Quantum Phase Transitions*, 2nd ed. (Cambridge University Press, 2011).
- [29] A. Dutta, G. Aeppli, B. K. Chakrabarti, U. Divakaran, T. F. Rosenbaum, and D. Sen, *Quantum Phase Transitions in Transverse Field Spin Models: From Statistical Physics to Quantum Information* (Cambridge University Press, 2015).
- [30] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, *Advanced Quantum Technologies* **2**, 1900070 (2019).
- [31] K. Nakaji and N. Yamamoto, Expressibility of the alternating layered ansatz for quantum computation, *Quantum* **5**, 434 (2021).
- [32] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, *Physical Review A* **103**, 032430 (2021).
- [33] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, *PRX quantum* **3**, 010313 (2022).
- [34] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, *Nature* **549**, 242 (2017).
- [35] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, *Physical Review A* **92**, 042303 (2015).
- [36] A. G. Taube and R. J. Bartlett, New perspectives on unitary coupled-cluster theory, *International Journal of Quantum Chemistry* **106**, 3393 (2006).
- [37] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, *Nature Communications* **10**, 3007 (2019).
- [38] P. Ollitrault, K. Sugisaki, and W. Mizukami, Multi-reference uccsd variational quantum algorithm for strongly correlated molecules, *arXiv* (2024).
- [39] W. Mizukami, K. M. Nakanishi, and K. Yamazaki, Orbital optimized unitary coupled cluster theory for quantum computer, *Physical Review Research* **2**, 033421 (2020).
- [40] J. Lee, W. Huggins, M. Head-Gordon, and K. B. Whaley, Generalized unitary coupled cluster wavefunctions for quantum computation, *Journal of Chemical Theory and Computation* **15**, 6474 (2019).
- [41] F. A. Evangelista, G. K.-L. Chan, and G. E. Scuseria, Unitary selective coupled-cluster method, *Quantum* **6**, 703 (2022).
- [42] H. Tang, H. Jean-Ruel, J. Williams, and K. Whaley, Qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansätze, *Physical Review X Quantum* **1**, 020310 (2020).
- [43] V. Shkolnikov and S. Woerner, Overlap-adapt-vqe: practical quantum chemistry on near-term devices, *npj Quantum Information* **9**, 56 (2023).
- [44] I. Yordanov, X. Yuan, and G. Chan, Qubit-excitation-based adaptive variational quantum eigensolver, *Physical Review Research* **3**, 043161 (2021).
- [45] C. Wang, P. D. Johnson, and W. Scherer, Reducing the resources required by adapt-vqe using coupled exchange operators, *Nature Communications* , 1234 (2025).
- [46] A. Smith, B. Jones, and C. Lee, Tetris-adapt-vqe: An adaptive algorithm that yields shallower circuits, *Physical Review Research* **6**, 013254 (2024).
- [47] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).
- [48] D. E. Goldberg and J. H. Holland, Genetic algorithms and machine learning, *Machine Learning* **3**, 95 (1988).
- [49] S. Mirjalili, Genetic algorithm, in *Evolutionary Algorithms and Neural Networks: Theory and Applications* (Springer International Publishing, Cham, 2019) pp. 43–55.
- [50] A. Zhang and S. Zhao, Evolutionary-based searching method for quantum circuit architecture, *Quantum Information Processing* **22**, 283 (2023).
- [51] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Physical review A* **52**, 3457 (1995).
- [52] U. Azad, How to import noise models from qiskit,

- https://pennylane.ai/qml/demos/tutorial_how_to_import_qiskit_noise_models (2024).
- [53] M. Mallapur, Ansatz design using genetic algorithm, <https://github.com/ManishMallapur/ansatz-design-using-genetic-algorithm> (2025), accessed: 2024-05-20.

APPENDIX

This appendix provides supplementary information to support the main text. Appendix A details the pseudocode for our genetic algorithm framework along with information on how long it takes to generate different circuits in table IV and Figure 14 showing the genetic algorithm framework being run for 20 generations. Appendix B provides the details for replicating the molecular Hamiltonians used for our experiment. In Appendix C, we have provided some of the ansatze that we have used in our experiments.

Appendix A: Genetic Algorithm Framework

This section provides pseudo-code for all the steps involved in our genetic algorithm framework for ansatz search. Algorithm 2 details the pseudocode for our fidelity-based method for calculating the expressibility of a parametrized quantum circuit. Further pseudocodes outline the core components of our GA-framework-based ansatz search, including population initialization(algorithm 3), parent selection and offspring generation(algorithm 4), and population update after each generation(Algorithm 5).

Algorithm 2: Calculating Expressibility of a PQC

Input: $n = \#$ qubits, $S = \#$ samples
Output: $\mathcal{E} =$ Expressibility

- 1 Define parametrized circuit on n qubits
- 2 Initialize **fidelities** list
- 3 **for** $i \leftarrow 1$ **to** S **do**
- 4 Sample $\theta_1, \theta_2 \sim \mathcal{U}[0, 2\pi]$
- 5 $\psi_1 \leftarrow \text{circuit}(\theta_1)$, $\psi_2 \leftarrow \text{circuit}(\theta_2)$
- 6 $F \leftarrow |\langle \psi_1 | \psi_2 \rangle|^2$
- 7 Append F to **fidelities**
- 8 Build histogram of **fidelities** (bin size b)
- 9 Compute Haar distribution over the same bins
- 10 Normalize both to probability vectors
- 11 $\mathcal{E} \leftarrow D_{\text{KL}}(\text{fidelities} \parallel \text{Haar})$
- 12 **return** \mathcal{E}

Algorithm 3: Initialize Population and Evaluate Expressibility

Input: n (qubits), L (layers), P (pop. size), S (samples)
Output: Expressibility scores for each of P circuits

- 1 $\mathcal{G} \leftarrow \{\text{RX}, \text{RY}, \text{RZ}, \text{CX}, \text{CRX}, \text{H}\}$
- 2 **for** $i \leftarrow 1$ **to** P **do**
- 3 Randomly build circuit $_i$ with L gates from \mathcal{G}
- 4 Compute $\mathcal{E}_i \leftarrow \text{Expressibility}(\text{circuit}_i, S)$ (Alg. 2)

Algorithm 4: Select Parents and Generate Offspring

Input: Population of circuits + their \mathcal{E}
Output: Offspring population

- 1 Sort circuits by ascending \mathcal{E}
- 2 Select top k as parents
- 3 **for** each offspring slot **do**
- 4 Pick two parents p_1, p_2
- 5 Choose crossover layer $\ell \in [1..L]$
- 6 offspring \leftarrow layers[1: ℓ] of p_1 + layers[$\ell+1$: L] of p_2
- 7 Assign new random parameters
- 8 **if** $\text{Uniform}(0, 1) < m$ **then**
- 9 Mutate one gate or parameter
- 10 Compute offspring's \mathcal{E} (Alg. 2)

Algorithm 5: Update Population and Return Best Ansatz

Input: Current population, Offspring population
Output: Best ansatz found

- 1 Replace population with offspring
- 2 Record best expressibility \mathcal{E} of this generation
- 3 **if** not last generation **then**
- 4 **Go to Alg 4**
- 5 **return** Ansatz with lowest \mathcal{E}

Table IV. Average time (in seconds) to generate variational quantum circuits for different numbers of qubits n and circuit depths $d = kn$, where $k \in \{1, 2, 3, 4\}$. Each entry shows the average time taken by the ansatz generation algorithm for given n and d .

Qubits n	$d = n$ (s)	$d = 2n$ (s)	$d = 3n$ (s)	$d = 4n$ (s)
2	24.257	32.877	46.744	52.203
3	40.426	70.095	87.764	107.556
4	65.067	101.094	143.742	186.507
5	87.083	149.503	208.842	269.538
6	115.716	214.969	304.193	400.754
7	146.054	289.292	503.385	659.768
8	241.702	726.306	1184.670	1563.758
9	559.752	619.953	884.411	1190.944
10	722.713	1460.488	2113.906	2767.386

Appendix B: Molecular Hamiltonians

Table V summarizes the molecular geometries and active-space definitions used to construct the chemical Hamiltonians. All Hamiltonians are generated using the STO-3G basis set. For BeH₂ and H₂O, a frozen-core approximation is employed, where the lowest-energy core orbital (corresponding to two core electrons) is frozen, and only the remaining valence orbitals are included in the active space. All other molecules are treated without freezing any core orbital.

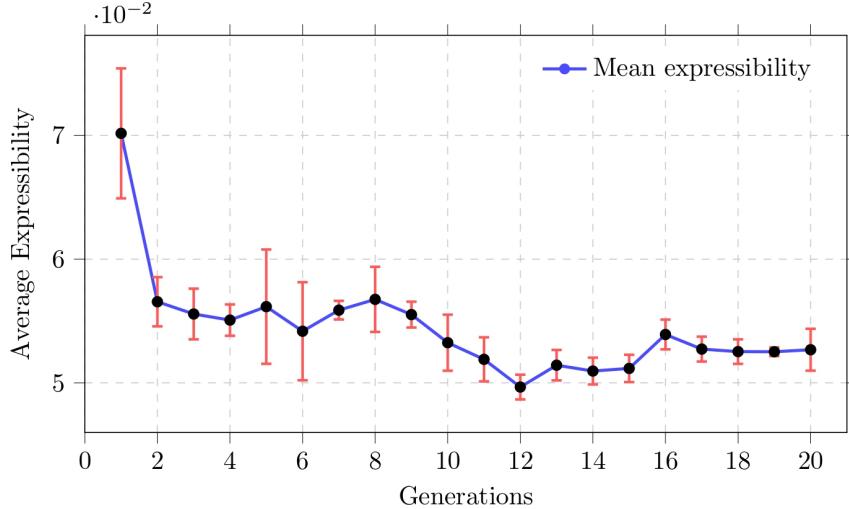


Figure 14. Genetic ansatz evolution over 20 generations over four different seeds. Extending the GA to 20 generations yields no significant improvement in expressibility beyond generation 10, indicating that additional generations only increase computational cost.

Table V. Molecular geometries and frozen-core settings used for constructing the chemical Hamiltonians. All distances are in Å and angles in degrees.

Molecule	Geometry	Bond length(s) / Angle	Frozen core	Atomic coordinates (Å)
H ₂	Linear	H–H = 1.32	None	H : (0, 0, -0.66) H : (0, 0, +0.66)
LiH	Linear	Li–H = 2.9693	None	Li : (0, 0, 0) H : (0, 0, 2.9693)
BeH ₂	Linear	Be–H = 1.3264	2 e ⁻ , 1 orb.	Be : (0, 0, 0) H : (0, 0, +1.3264) H : (0, 0, -1.3264)
H ₂ O	Bent	O–H = 0.957 \angle HOH = 104.52	2 e ⁻ , 1 orb.	O : (0, 0, 0) H : (+0.757, 0, 0.586) H : (-0.757, 0, 0.586)

Appendix C: Evolved Genetic Ansatze

In this section, we present a subset of quantum circuits constructed with 8 qubits and a circuit depth of 16 for each considered gate set. These circuits are used for benchmarking in various instances throughout the paper.

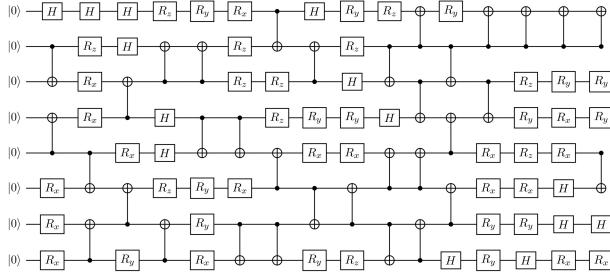


Figure 15. Gate Set A

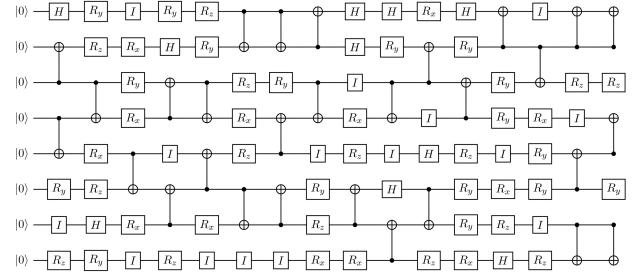


Figure 16. Gate Set B

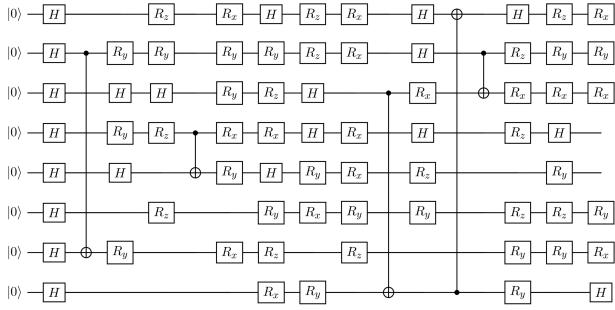


Figure 17. Gate Set C

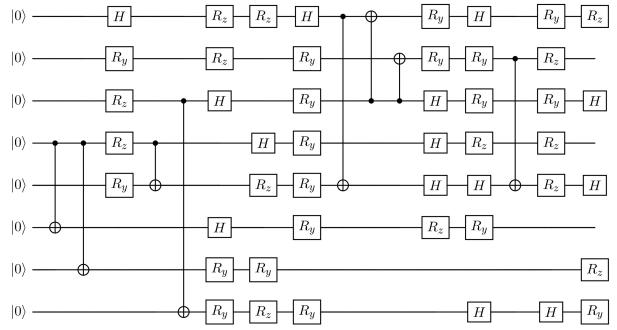


Figure 18. Gate Set D

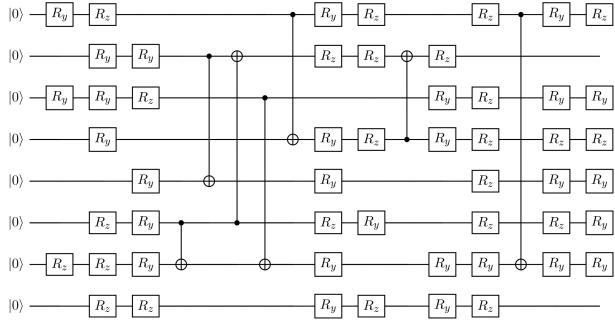


Figure 19. Gate Set E

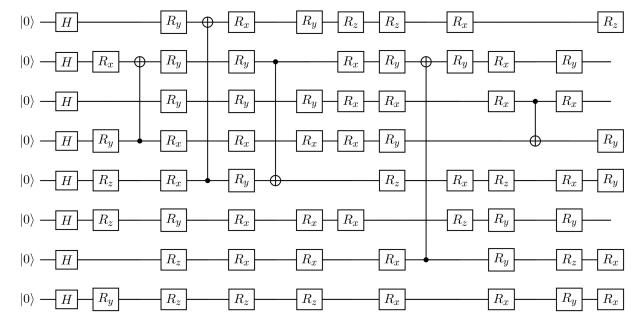


Figure 20. Gate Set F

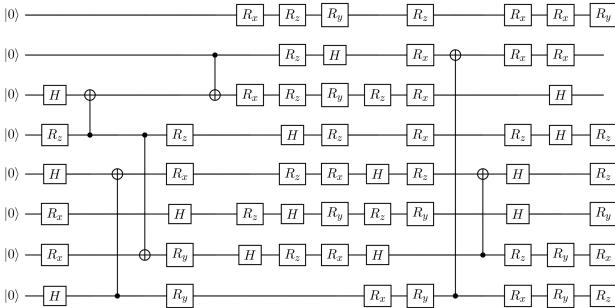


Figure 21. Gate Set G

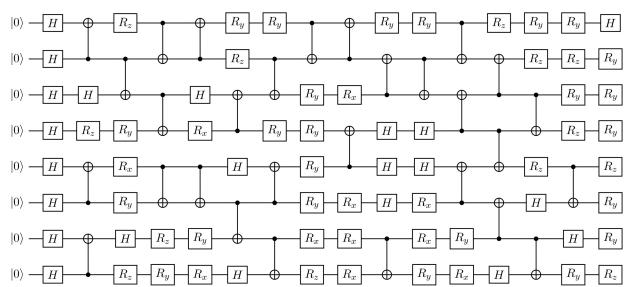


Figure 22. Gate Set H

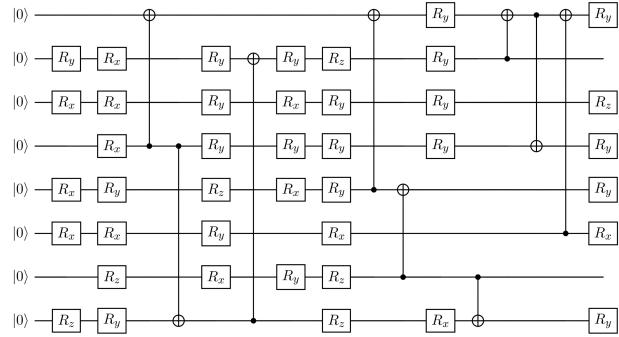


Figure 23. Gate Set I