EQUAL: Improving the Fidelity of Quantum Annealers by Injecting Controlled Perturbations

Ramin Ayanzadeh*
Georgia Tech
Atlanta, USA

Poulami Das
Georgia Tech
Atlanta, USA

Swamit Tannu
University of Wisconsin
Madison, USA

Moinuddin Qureshi
Georgia Tech
Atlanta, USA

Abstract—**Quantum computing is an information processing paradigm that uses quantum-mechanical properties to speedup computationally hard problems. Gate-based quantum computers and Quantum Annealers (QAs) are two commercially available hardware platforms that are accessible to users today. Although promising, existing gate-based quantum computers consist of only a few dozen qubits and are not large enough for most applications. On the other hand, existing QAs with few thousand qubits have the potential to solve some domain-specific optimization problems. QAs are single instruction machines and to execute a program, the problem is cast to a Hamiltonian, embedded in the hardware, and a single quantum machine instruction (QMI) is run. Unfortunately, noise and imperfections in hardware result in sub-optimal solutions on QAs even if the QMI is run for thousands of trials.**

Due to the limited programmability of QAs users execute the same QMI for all trials. This subjects all trials to a similar noise profile throughout the execution, resulting in a *systematic bias*. We observe that systematic bias leads to sub-optimal solutions and cannot be alleviated by executing more trials or using existing error-mitigation schemes. To address this challenge, we propose *EQUAL* (Ensemble QUantum AnneaLing). EQUAL generates an ensemble of QMIs by adding controlled perturbations to the program QMI. When executed on the QA, the ensemble of QMIs steers the program away from encountering the same bias during all trials and thus, improves the quality of solutions. Our evaluations using the 2041-qubit D-Wave QA show that EQUAL bridges the difference between the baseline and the ideal by an average of 14% (and up to 26%) without requiring any additional trials. EQUAL can be combined with existing error mitigation schemes to further bridge the difference between the baseline and ideal by an average of 55% (and up to 68%).

Index Terms—**Quantum Annealers, Quantum Computing**

I. Introduction

Quantum computing is an information processing paradigm that leverages quantum mechanical properties of quantum bits (qubits) to store and process information and promises significant computational advantages for many hard problems [1]–[4]. There exist different models for the physical realization of this computational paradigm [5], [6]. Currently, prototypes of both gate model [7]–[9] and annealing [10] types are available, and some of them can already outperform modern supercomputers for some tasks [11]–[14].

Gate-based quantum computers, such as IBM and Google machines, use discrete quantum gate operations to manipulate qubits such that the state of the qubits evolve to produce the desired outcome as the program proceeds. Such systems with about 50-plus qubits are already available [7], [15], [16]. To solve a problem on a gate-based quantum computer, we map it to an efficient quantum algorithm, map the high-level program qubits to the physical qubits of the device, translate the instructions into a series of low-level quantum gates, and execute them, as shown in Fig. 1(a). Although these types of systems promise significant computational advantages, they must grow in size for practical applications [5], [12], [17].

Unlike gate model quantum computers that can be programmed to solve different classes of problems, *Quantum Annealers (QAs)* are single-instruction machines that can only solve a specific discrete optimization problem by minimizing the energy of a physical system, called *Hamiltonian* [6], [18]. To solve a problem on a QA, (a) we cast it to a Hamiltonian, (b) embed it to match the topology of the QA device, (c) obtain the resulting single *Quantum Machine Instruction (QMI)*, (d) execute the single QMI, and (e) repeatedly run the same QMI multiple times [19], as shown in Fig. 1(b). The outcome with the lowest energy is deemed as the solution.

As QAs can only minimize a specific objective function, any other problem must be cast to this Hamiltonian. *Casting* computes the coefficients of the Hamiltonian such that the global minima of the Hamiltonian represent the global optima of the problem of interest [19]–[21]. *Embedding* maps the problem graph to the QA topology. As QAs have limited connectivity between qubits, embedding encodes a program qubit with higher connectivity by using a chain of physical qubits. The problem of limited connectivity exists even on most existing gate-based quantum computers and can be overcome by inserting SWAP operations [22]–[24]. However, a similar approach is impractical for QAs as they can only execute a single QMI. Unlike gate-based systems, QAs available today with 5000-plus qubits [10], [15], [25] are much larger, scale faster, and have the potential to power a wide range of real-world applications in many domains [21], [26]–[41].

Although promising, QA hardware suffers from various drawbacks such as noise, device errors, limited programmability, and low annealing time, which degrade their reliability [6], [19], [42]. Addressing these limitations requires device-level enhancements that may span generations of QAs. Therefore,

Ramin Ayanzadeh was supported by the NSF Computing Innovation Fellows (CI-Fellows) program. Poulami Das was supported by the Microsoft Research PhD fellowship. This research was partially supported by the Vice Chancellor Office for Research and Graduate Education at the University of Wisconsin–Madison with funding from the Wisconsin Alumni Research Foundation.

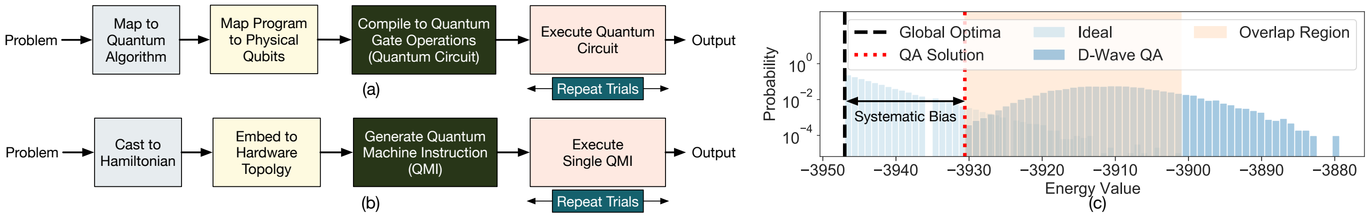

Fig. 1. Steps involved in solving a problem using (a) gate-model quantum computers and (b) quantum annealers. (c) Energy histogram of a 2000-qubit optimization benchmark executed on D-Wave QA (in logscale). The QA can quickly identify the region of the ground state energy (overlapping region), but the solution is far from the global optima due to systematic bias.

software techniques to improve the reliability of QAs is an important area of research [43]–[49].

Despite recent hardware and software enhancements, existing QAs may fail to find the global minima for certain problems [42]. For example, Fig. 1(c) shows the energy histogram of a 2000-qubit optimization problem on a D-Wave QA. We can think of QA as a machine that samples from a Boltzmann distribution such that samples with lower energy values are exponentially more likely to be observed [42], [50]. In theory, QA can find the optimal solution with a very high probability [51]. However, in this example, we observe that although the QA can quickly identify the region of the global optima, the best solution from the QA is far from the global optima. As users run only a single QMI, the program is subjected to a similar noise profile for all trials, resulting in a *systematic bias*. Our experiments show that running more trials or relying on existing error-mitigation schemes cannot overcome this bias. Unfortunately, systematic bias produces incorrect solutions far from the global optima and limits the reliability of QAs for practical applications.

In this paper, we propose *Ensemble Quantum Annealing (EQUAL)*, an effective scheme for mitigating systematic bias and improving the reliability of QAs by running an ensemble of QMIs with controlled perturbations. EQUAL is based on the insight that running the same QMI for all trials projects QAs to a very similar noise profile and bias. Instead, EQUAL uses an ensemble of QMIs that subjects the system to different noise profiles and biases. Generating effective ensembles of QMIs is nontrivial, and our design focuses on addressing it. [1]

To generate ensembles, EQUAL creates new Hamiltonians, called *Perturbation Hamiltonians*, and adds them to the original problem Hamiltonian. Every perturbation Hamiltonian adds noise to the original Hamiltonian and the QMI obtained from this process is a perturbed variation of the original QMI. The challenge in this step is that adding extremely small perturbations will have no impact on the systematic bias, whereas adding large perturbations can significantly change the landscape of the original Hamiltonian. In the worst case, the final perturbed Hamiltonian may correspond to a problem completely different from the one at hand. Thus, there exists a trade-off between the ability to eliminate systematic bias and the correctness of a Hamiltonian. Ideally, we want a perturbed Hamiltonian that can eliminate systematic bias without altering the characteristics of the problem Hamiltonian significantly. To address this challenge, EQUAL exploits the fact that QAs only allow a limited precision of coefficients for a Hamiltonian due to hardware limitations. For every ensemble, EQUAL draws the coefficients of the corresponding perturbation Hamiltonian randomly at a range just below the supported precision so that adding the Perturbation Hamiltonian may only shift the coefficients of the QMI (post truncation) to one of the neighboring quantization levels and not impose significant changes to the landscape of the original Hamiltonian.

We also analyze existing error-mitigation approaches for QAs. Our characterization experiments on D-Wave show that the SQC [42] postprocessing technique is highly effective for D-Wave. We compare EQUAL with SQC and show that the two schemes can be combined for even greater benefit. The resulting design, *EQUAL+*, provides significantly better fidelity than EQUAL and SQC standalone. As the SQC postprocessing relies only on classical computations, EQUAL+ does not incur any additional trials compared to EQUAL.

Our evaluations on D-Wave's 2041-qubit QA show that EQUAL bridges the difference between the baseline and the ideal by an average of 14% (and up to 26%). EQUAL+ further bridges the difference between the baseline and the ideal by an average of 55% (and up to 68%).

Overall, this paper makes the following contributions:

1) We show that there is a systematic bias associated with each QMI, running on QA, that deviates the annealing process from achieving the ground state of the corresponding Hamiltonian and produces sub-optimal solutions.
2) We propose *EQUAL (Ensemble Quantum Annealing)* to mitigate the bias by forming multiple perturbed copies of a given QMI and running each for a subset of trials.
3) We propose an effective method to generate the perturbed copies while retaining the structure of the problem by leveraging the hardware imperfections from limited precision.
4) We propose EQUAL+ that combines EQUAL with the existing SQC error-mitigation technique to further improve the reliability.

[1] The problem of systematic bias is similar to correlated errors on gate-model quantum computers that can be addressed by mapping programs to different sets of physical qubits on the same machine [52], inserting different SWAP routes [52], or using different machines [53]. The equivalent step for QAs would be to use multiple embeddings. However, this is not viable for QA, and we discuss the details in Section VI.

II. Background and Motivation

A. Quantum Computing

Quantum computing is a computational paradigm that stores and processes information using quantum bits or *qubits*. The state of a qubit $|\psi\rangle$ can be represented as a superposition of its two basis states $|0\rangle$ and $|1\rangle$ using a vector: $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, where α and β are complex probability amplitudes associated with the basis states. Similarly, an N-qubit system exists in a superposition of 2^N basis states. This exponential scaling in state space with a linear increase in qubits enables quantum advantage. Currently, two types of quantum platforms are available to users through cloud services [7], [15], [16]—gate-based quantum computers and quantum annealers.

Gate-based Quantum Computers: A gate-based quantum computer executes a predefined sequence of quantum gate operations (known as a quantum circuit) to transform the initial state of qubits to the desired state by changing the superposition. Quantum computers from IBM, Google, and others use a gate-based model.

Quantum Annealers: Quantum annealing is a meta-heuristic for solving combinatorial optimization problems that runs on classical computers [18], [51], [54]–[57]. Quantum Annealers are single instruction machines for solving combinatorial optimization problems. Unlike gate model quantum computers, where we directly change the state of qubits via quantum gates, QAs control the environment, and qubits evolve to remain in the ground state (i.e., a configuration with the lowest energy value) of a Hamiltonian (or energy/cost function) [6], [18]. Quantum annealers, such as the ones from D-Wave, are analog systems that can only minimize the following energy function:

$$\mathbf{H}_p := f(\mathbf{z}) = \sum_{i=0}^{N-1} \mathbf{h}_i \mathbf{z}_i + \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} J_{ij} \mathbf{z}_i \mathbf{z}_j, \tag{1}$$

where N is the number of qubits, $\mathbf{h}_i \in \mathbb{R}$ specifies the linear coefficient of qubit i, $J_{i,j} \in \mathbb{R}$ represents the coupler weight between qubits i and j, and $\mathbf{z}_i \in \{-1, +1\}$ is the problem variable [19], [21], [42]. Ever since the introduction in 2011, QAs have rapidly scaled in size up to few thousand of qubits, as shown in Fig. 2(a) and promise significant computational advantage for a wide range of applications.

B. Operation Model of QA

To execute a program on a QA, the problem is cast to a Hamiltonian such that its global minimum represents the optimal solution to the problem at hand. This step computes the coefficients $\mathbf{h}$ and J, denoted in (1), corresponding to the *quantum machine instruction (QMI)* to be executed on the QA. Executing the QMI on a QA returns a sample $\mathbf{z} = \{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{N-1}\}$ as a potential minimum of the corresponding energy function. Unfortunately, executing a QMI only once may not result in the ground state of the Hamiltonian [42]. Thus, in practice, the process of executing the single QMI is repeated for thousands of trials. The sample with the lowest energy is reported as the solution.

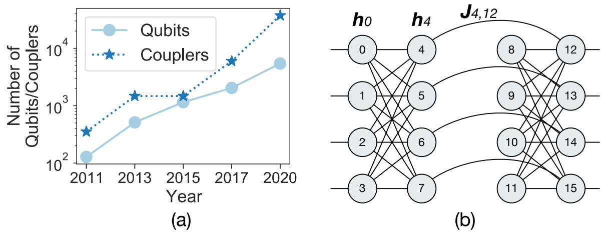

Fig. 2. (a) Evolution of Quantum Annealers (QAs) over time. (b) A cropped view of the connectivity graph for D-Wave 2000Q where the nodes denote qubits and edges represent couplers (or connectivity between two qubits).

C. The Opportunity: Solving Large Problems with QA

Google Sycamore is a state-of-the-art 54-qubit gate-based quantum computer that can outperform even the most powerful supercomputer for some tasks [11]. We compare the performance of the D-Wave 2041-qubit QA and Google Sycamore for 18 different Max-Cut problems. The Max-cut problems used in this evaluation correspond to the fully-connected Sherrington–Kirkpatrick (SK) Model [58] and uses up to 17 qubits [59]–[61]. These are some of the hardest benchmarks on Sycamore as fully connected graphs require many SWAP operations to overcome the limited connectivity. Running the same benchmarks on the D-Wave QA requires only 102 qubits (less than 5% of the qubits). Fig. 3 shows the value of the solution obtained from both machines.

We use the same weighted graphs from prior work [59] that result in negative cut values. Both quantum machines are successful in finding the optimal cut at small problem sizes. However, the performance of Google Sycamore degrades with increasing problem size [59] due to an increase in SWAPs and circuit depth. Furthermore, due to the limited capacity of 54-qubits, the problem size for Sycamore is limited to no more than 54 nodes. However, as QAs are much larger (2000–5000 qubits), we can use them to solve larger problems more relevant to real-world applications and exceed the size of near-term gate-based quantum computers. For example, Fig. 4 shows the performance of the D-Wave QA for Max-Cut problems corresponding to the SK Model using up to 60 qubits. For each problem, the D-Wave QA can find the optimal cut value. To determine the optimal cut value, we evaluate all possible combinations for problems using up to 25 qubits and use the best estimate for the larger problems.

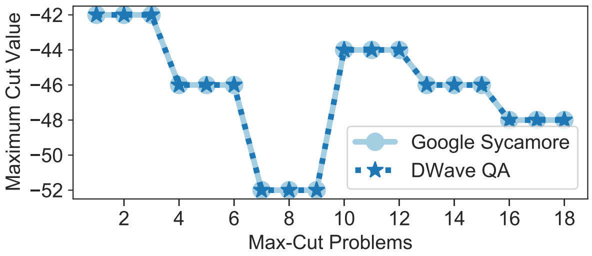

Fig. 3. Comparison of 54-qubit Google Sycamore [59] and 2041-qubit D-Wave Quantum Annealer.

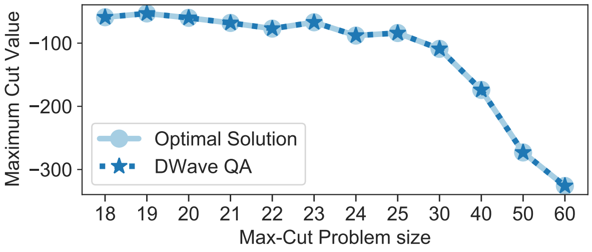

Fig. 4. Performance of D-Wave QA for larger Max-Cut problems corresponding to the SK Model.

D. The Challenge: Hardware and Software Limitations

Although QAs look promising for various applications, several challenges limit us from solving real-world problems on them.

1) Hardware-Level Challenges:

Limited coherence/annealing time: The probability of finding the ground state of a Hamiltonian using a QMI increases exponentially with increasing annealing time [6] and theoretically, many hard problems may require large annealing time. Unfortunately, the annealing time on current QAs is in the order of microseconds [19], [42] as qubits can retain their state only for a short span of time. Increasing the annealing time causes qubits to *decohere* and lose their state.

Noise and limited connectivity: Thermal noise and operational errors add unwanted perturbations during annealing and prevent QAs from reaching the ground state of a Hamiltonian [6]. QAs also suffer from sparse connectivity between qubits, as shown in Fig. 2(b). To address the same drawback on gate-model quantum computers, compilers insert SWAP instructions that interchange the state of physically adjacent qubits [22]–[24]. However, QAs cannot use a similar approach as they use only one QMI. Instead, we *embed* the problem graph to match the target device topology where multiple physical qubits represent a program qubit with higher connectivity. This can reduce the effective capacity of QAs.

Limited precision and range of coefficients: Casting a problem to a Hamiltonian and generating the corresponding QMI coefficients can require a double-precision representation. However, large precision impacts the performance of the digital-to-analog converters (DACs) used on the real QAs, which slows the annealing process. Therefore, existing QAs trade-off precision to achieve lower annealing times and truncate the QMI coefficients post casting to match the precision supported in hardware. This subjects the QMI to quantization errors, and the reduced precision QMI actually executed on a QA can be slightly different from the QMI that we originally intended to run, leading to a ground state that may not represent the solution of the problem at hand [42], [62], [63].

2) Software-level Challenges:

Limited programmability: QA can only minimize a specific objective function and any input problem must be *cast* to a Hamiltonian. Unfortunately, casting is nontrivial due to a lack of standardized algorithms and often comes with some approximations [19], [21]. Additionally, QAs can only execute a single QMI that performs the annealing step, and therefore, fine-grained optimizations at the instruction-level are infeasible.

Limitations of Embedding: To overcome the limited connectivity of the physical QAs, a problem graph is embedded in the QA to match the device topology. Finding the best embedding is NP-hard [43], [44], [47], [64] and existing algorithms can take several hours despite approximations. Moreover, our studies show they often fail large programs.

E. Impact of Trials on Energy Residual

In theory, QAs should find the global optima of a problem with high probability [6], [51]. However, in reality, QAs often fail to find the global optima for large problems due to noise and imperfect control. Moreover, the limited programmability of QAs forces users to run a single QMI for thousands of trials, resulting in a bias. As a user runs a single QMI for all trials, the noise profile is similar throughout execution, resulting in similar quality outcomes due to the inherent bias in the noise profile. We refer to this bias as *Systematic Bias*.

Fig. 5 shows the *Energy Residual (ER)* [65], [66] for an optimization problem on D-Wave QA. ER compares the gap between the energy of the solution from a noisy QA and the global minima. The energy of the best solution from a noisy QA remains far from the global optima even after running 1 million trials. This nonzero ER occurs due to systematic bias, and is particularly severe for large problems.

F. Goal of this Paper

Hardware and software limitations of QAs cause programs to encounter a systematic bias during execution which cannot be bypassed by executing more trials. Moreover, the operation model of QAs precludes leveraging proposed policies for gate-based quantum computers. Ideally, we want QAs to be free from systematic bias. In this paper, we propose Ensemble Quantum Annealing (EQUAL) that uses an ensemble of QMIs (with different biases) to mitigate systematic bias. We discuss the evaluation methodology before discussing our solution.

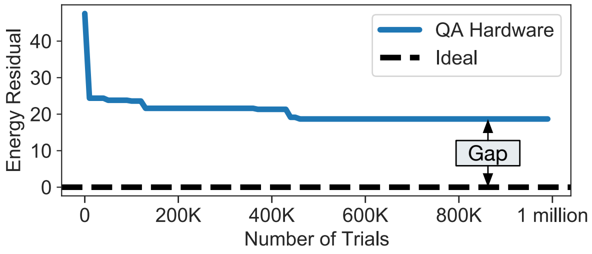

Fig. 5. Energy Residual of an optimization problem on D-Wave QA with increasing number of trials.

III. Evaluation Methodology

We discuss the evaluation infrastructure used in this paper.

A. Quantum Platform and Baseline

For our evaluations, we use the 2041-qubit quantum annealer from D-Wave Systems via Amazon BraKet cloud service [15]. We use the default annealing time (i.e., 20 μseconds) and schedule recommended for this system. For the baseline, we use 100,000 trials for each benchmark. Such a large number of trials reduces sampling errors and therefore, this serves as a strong baseline. For EQUAL, trials are equally split between QMIs. Thus, EQUAL requires the same number of trials as the baseline.

B. Benchmarks

We use *random weighted Max-Cut* problems, similar to Quantum Approximate Optimization Algorithms [67] used on gate-based quantum computers. For the benchmarks, we draw the Hamiltonian coefficients of the QMIs from the standard normal distribution (a mean of 0 and a standard deviation of 1). This approach is a common practice used in prior works related to benchmarking QAs [18], [42], [46], [63], [68]. To avoid the impact of embedding on our evaluations, we directly use the connectivity graph of the D-Wave QA. Thus, the number of program qubits in benchmarks is equal to the number of physical qubits on the QA. As the size of benchmarks significantly exceeds the size of existing gate-model quantum computers, we cannot compare our results with them.

C. Figure-of-Merit

We evaluate the reliability of QA using **Energy Residual (ER)**. The best solution from a QA is the outcome with the minimum energy. ER computes the energy gap between the minimum energy (E_{min}) obtained on a QA with respect to the global energy minimum (E_{global}) of the application as follows:

$$\text{Energy Residual (ER)} = \left|E_{min} - E_{global}\right|. \qquad (2)$$

Ideally, when the best solution obtained on a QA corresponds to the ground state of the problem Hamiltonian, ER is zero. Thus, a lower value (closer to zero) for ER is desirable.

The challenge in computing the ER for random large benchmarks spanning 2000+ qubits is that finding the ground state of the Hamiltonian is nontrivial. To overcome this challenge and still enable a fair comparison, we perform intensive classical computations using state-of-the-art tools [69] and approximate the global optimum of our benchmark problems. Recent studies have shown that this algorithm can estimate the ground state of Chimera based Hamiltonians [10], [43] (such as the ones considered in our paper) with a high probability. The techniques used to derive the best estimate of the ground state energy of a Hamiltonian require intensive classical computing resources and could take up to days for problem sizes with a few thousand qubits. We discuss more on this in Section VI.

IV. EQUAL: Ensemble Quantum Annealing

The vulnerability of a program to systematic bias results from limited programmability and the current operating model of QAs where the same QMI is executed for thousands of trials. This subjects each trial to a similar noise profile on the QA, and the entire execution suffers from the same inherent bias. Our proposed solution *EQUAL*—Ensemble Quantum Annealing—takes a different approach. Instead of a single QMI, EQUAL generates an ensemble of QMIs that subjects the program execution to different noise profiles and, therefore, different systematic biases. When results are aggregated, ensembles enable us to improve the quality of solutions.

A. Challenges in Generating Ensembles in EQUAL

There is potential to generate ensembles during any one of the three phases that a problem goes through before execution on a physical QA hardware: (1) casting, (2) embedding, and (3) QMI generation. Generating ensembles during casting was previously studied in the context of Boolean satisfiability (SAT) [21] and binary compressive sensing [37] problems on QAs. Unfortunately, these methods exploit the features of the application-specific casting algorithms. Therefore, this approach has limited applicability and is hard to generalize for QAs. The other alternative approach is to use an ensemble of embeddings for a given problem. However, this approach too has its limitations. Firstly, finding the best embedding is an NP-hard problem in itself [43], [44], [47], [64]. Secondly, current embedding schemes for QAs use several approximations and may or may not be able to determine an ensemble of embeddings of similar quality [43], [44], [47], [64]. Our studies show that existing embedding algorithms often fail to find an adequate number of embeddings, particularly for problems at scale that require 2000+ qubits. Thirdly, even if it is possible to find multiple embeddings, they are often of inferior quality and require larger chains of physical qubits to represent a program qubit with higher connectivity. This makes the embedding significantly more vulnerable to noise compared to the best embedding. Thus, generating ensembles at the embedding step is nontrivial. Instead, EQUAL focuses on generating ensembles at the instruction-level and produces multiple QMIs.

B. Overview of Design

Fig. 6 shows an overview of EQUAL. It relies on adding controlled perturbations to the original QMI. For each ensemble, EQUAL generates a *Perturbation Hamiltonian*, denoted by δ. Each of these Perturbation Hamiltonians creates a new QMI when added to the original Hamiltonian. For example, if EQUAL generates m ensembles of QMIs, it generates m perturbation Hamiltonians, namely $\delta_1, \delta_2, \ldots, \delta_m$. The ensemble QMIs—QMI$_1$, QMI$_2$ to QMI$_m$—are obtained by adding the original Hamiltonian (say $\mathbf{H}$) and the respective perturbation Hamiltonians. In other words, the ensemble of QMIs now corresponds to the perturbed versions of the original Hamiltonian.

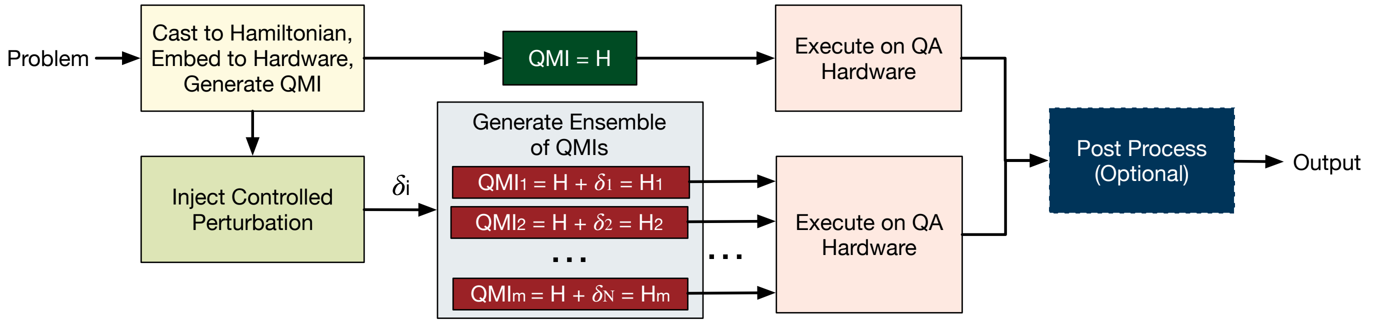

Fig. 6. Overview of EQUAL. EQUAL creates an ensemble of QMIs by adding controlled perturbations to the original QMI. It executes the original QMI as well as the ensemble of QMIs separately on the QA hardware and returns the outcome with the lowest energy value. EQUAL can also optionally leverage the benefits of existing postprocessing error mitigation schemes (EQUAL+).

C. Generating Ensembles via Controlled Perturbations

Creating an effective perturbation Hamiltonian is nontrivial. If the perturbations add too little noise, the resulting Hamiltonian will be too close to the problem Hamiltonian and encounter a similar bias. Alternatively, too large perturbations result in a Hamiltonian significantly different from the problem of interest and can produce infeasible results. For example, Fig. 7(a) shows the landscape of an example optimization problem— $\min_{x,y} x^2 + y^2$ for $x, y \in [-2, +2]$. Fig. 7(b) shows that injecting an extremely noisy perturbation Hamiltonian significantly changes the landscape of the original problem. Thus, there is a trade-off between the effectiveness of a perturbation Hamiltonian to reduce bias and its ability to alter the problem Hamiltonian. To address this challenge and generate an effective ensemble of QMIs, EQUAL exploits the device-level characteristics of QAs.

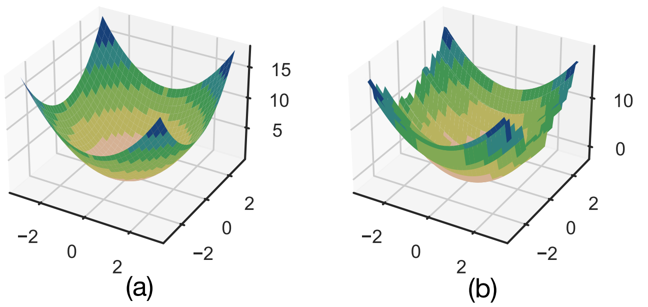

Fig. 7. (a) Landscape of an example optimization problem. (b) The resultant landscape differs significantly from (a) when an extremely noisy perturbation is imposed. (This figure is for illustrative purposes only. Hamiltonians and QAs can only deal with discrete optimization problems.)

1) Exploiting Hardware Characteristics of QAs: Recollect that casting a problem to a Hamiltonian can require a double-precision representation of the Hamiltonian coefficients. Unfortunately, real QAs can only support a small range and precision of coefficients due to the limitations imposed by the digital to analog converters (DACs) used on QAs. If the precision of the coefficients are too large, the DACs are too slow, which eventually slows the controlling modules of QAs and is not desirable. To bridge this gap, post the casting step, the coefficients of the QMI are truncated to match the precision supported by the hardware. While this is a limitation on QAs, EQUAL leverages it to its advantage and draws the coefficients of the perturbation Hamiltonian randomly at a range that is below the supported precision so that adding the perturbation Hamiltonian only shifts the coefficients of the QMI (post truncation) to one of the neighboring quantization levels and thus, does not significantly alter the problem landscape. More specifically, let b be the number of bits used for representing coefficients of a physical QA. For every ensemble, EQUAL draws a uniform random number $r \in \left[\frac{1}{2^{b+1}}, \frac{1}{2^b}\right]$ and set all coefficients of the Perturbation Hamiltonian to be r.

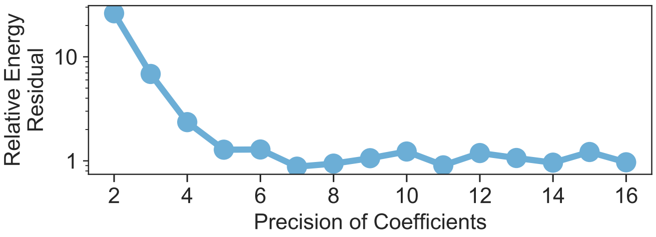

Fig. 8. Relative Energy Residual of QMIs with truncated coefficients with respect to the original problem QMI for bits values of precision. (Lower is better)

2) Profiling QAs to estimate Hardware Precision: Unfortunately, the precision of the coefficients supported on real devices is unavailable to programmers. Determining this precision is vital for the performance of EQUAL. Drawing the perturbation Hamiltonian coefficients far below the supported precision introduces large noise and may alter the Hamiltonian landscape significantly. Alternately, drawing them far above the supported range may not have any effect post truncation. To tackle this challenge, EQUAL profiles the QA using random benchmarks to estimate the precision supported by QAs. In this experiment, we truncate all coefficients of the benchmark for $2, 3, \ldots, 16$ bits precision and execute the corresponding QMIs. Fig. 8 shows the relative Energy Residual of the truncated QMIs with respect to the original

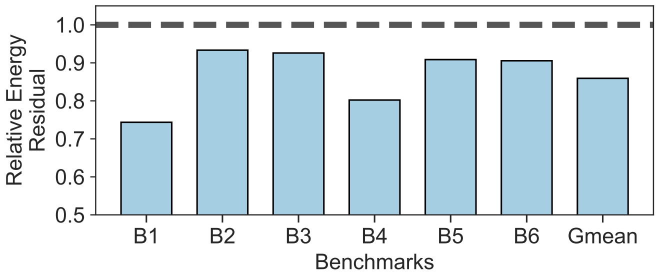

Fig. 9. Energy Residual of six random benchmarks on D-Wave QA hardware using EQUAL relative to the baseline.

problem (without truncation). Our profiling experiments with multiple benchmarks show that the hardware is likely limited by 7–8 bits of precision. Thus, EQUAL generates coefficients of ensembles in $\left[\frac{1}{2^9}, \frac{1}{2^8}\right]$.

D. Execution on QA Hardware

EQUAL splits the trials between the ensemble of QMIs equally, including the original QMI (without perturbation), and executes them separately on the QA hardware. Our default design uses 10 ensembles of QMIs, and allocates 10,000 trials for every ensemble. We do a more rigorous sensitivity analysis for the number of trials and ensembles in Section IV-H.

E. Aggregating Results

By default, the outcome with the lowest energy is deemed as the solution for problems executed on QAs. In the baseline, this corresponds to the outcome with the lowest energy obtained by executing the original QMI. As EQUAL executes multiple QMIs, the outcome with the lowest energy among all the QMIs is returned as the solution. Also, as EQUAL runs the ensemble of perturbed QMIs in addition to the original program QMI, the final solution is guaranteed not to perform worse than the baseline, assuming there are no sampling errors. Note that the solution with the minimum energy corresponds to an outcome that may come from a single QMI. For the baseline, this corresponds to the original QMI, whereas for EQUAL it comes from one or more of the QMIs in the ensemble. However, which QMI corresponds to the best solution is not known a-priori and EQUAL must execute the entire ensemble.

F. Results for Energy Residual

Energy Residual (ER) computes the gap between the energy obtained from the best outcome on a QA with the global optima. Fig. 9 shows the ER of EQUAL for our benchmarks executed on D-Wave 2041-qubit QA relative to the baseline. We observe that EQUAL bridges the difference between the baseline and the ideal by an average of 14% (and up to 26%). QAs deal with industry-scale optimization problems where even a minuscule improvement has a tremendous impact in terms of practical advantage, such as saving millions of dollars [39], [70], [71] in the context of scheduling and planning applications or finding better candidates for drug discovery [41] and material science [40]. Thus, the quality of solutions is of utmost importance.

Fig. 10 compares the ER of the individual benchmarks for baseline and EQUAL. We observe that the ER quickly saturates in the baseline for all benchmarks, whereas improves for EQUAL as more QMIs are executed. As the QMIs are generated using random controlled perturbations, some of them may result in higher ER compared to the baseline due to a different noise profile at run time. However, the ensemble overall enables EQUAL to reach a better solution. In the worst case, EQUAL performs similar to baseline as the original program QMI is executed too. We observe that the fidelity of the baseline saturates with more trials, whereas the diversity of EQUAL helps it keep on improving with additional trials.

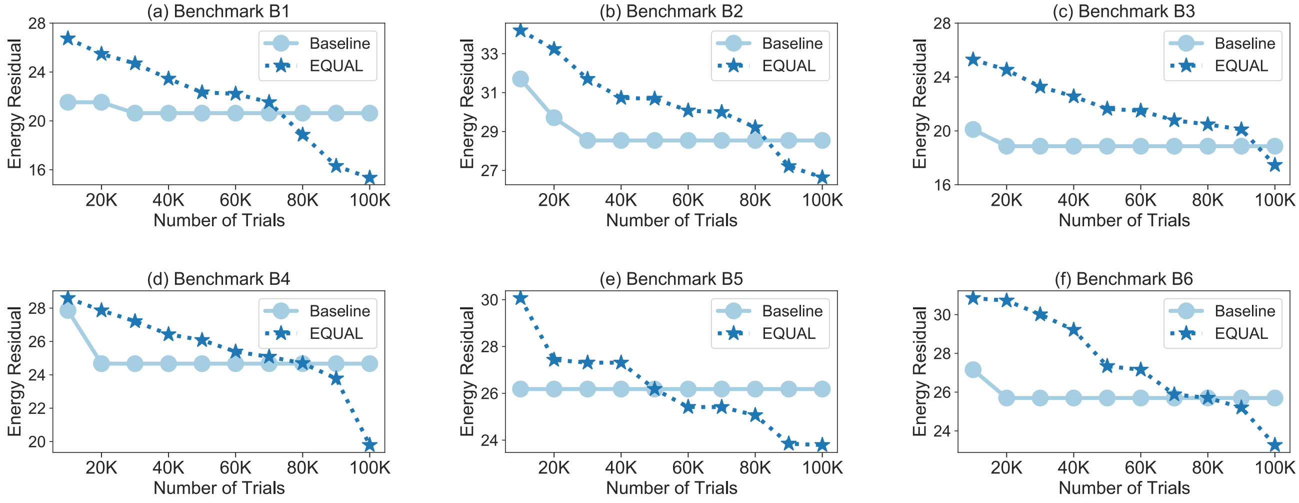

Fig. 10. Trends in Energy Residual for the baseline and EQUAL for all the benchmarks.

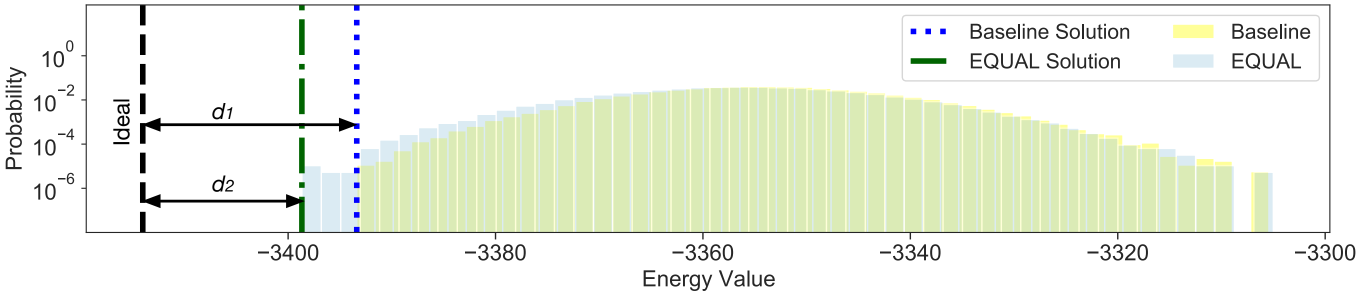

Fig. 11. Histogram of energy values from the outcomes on the QA for benchmark B1 using the Baseline and EQUAL. The solution from EQUAL is closer to the ideal solution compared to the baseline solution ($d_2 < d_1$). The histograms for the baseline and EQUAL largely overlaps which indicates that EQUAL does not significantly alter the problem Hamiltonian.

G. Case-Study: How EQUAL Reduces Systematic Bias

Fig. 11 shows the histograms of the energy values obtained by running benchmark B1 for the baseline and EQUAL. The goal of QAs is to obtain the outcome corresponding to the ground state energy. We observe that the optimal solution is at a distance d_1 from the ground state, and EQUAL produces a solution at a distance d_2 that is closer to the ground state energy ($d_2 < d_1$) by minimizing the impact of bias. We also observe that distributions for both the baseline and EQUAL overlap largely, indicating that the ensemble of QMIs do not largely alter the original Hamiltonian corresponding to our problem. We make similar observations for other benchmarks.

The best solution obtained by a QA depends on the overlapping region between the ideal and the noisy distributions. From Fig. 11, we observe two potential approaches to get closer to the global optima. First, by *flattening* the energy histogram of the Hamiltonian such that it covers a broader search space. Second, by *shifting* the energy histogram towards the ideal solution. Note that both of these techniques must ensure that the properties of the original program Hamiltonian remain unaltered. EQUAL uses the first approach. The performance of EQUAL can be improved further if we could shift the histogram closer to the ideal solution. We explore combining EQUAL with existing error mitigation schemes to obtain the advantage of both flattening the histogram and shifting the histogram towards the ground state.

H. Impact of Number of Ensembles

We study the impact of the number of ensembles on the effectiveness of EQUAL using a single benchmark problem. For a given trial budget of 100K trials, we choose two modes for EQUAL. In the first instance, we use 10 QMIs and run each of them for 10K trials each. In the second instance, we use 100 QMIs and run each of them for 1K trials each. Fig. 12 shows the ER for the baseline and these two instances of EQUAL. Note that we access the QA device through cloud services, and more rigorous sensitivity analysis in terms of QMIs and trials is challenging. We observe that executing more QMIs introduces more randomness and makes them vulnerable to sampling errors. EQUAL with 10 QMIs achieve a sweet spot between the baseline and EQUAL with a large number of ensembles such that we have both diversity as well as sufficient trials for each QMI to reduce sampling errors.

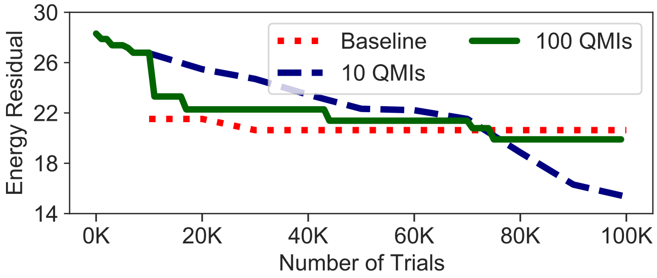

Fig. 12. Energy Residual for benchmark (B1). The baseline executes a single QMI for all 100K trials. EQUAL has 10 QMIs for 10K trials each or 100 QMIs executed for 1K trials each.

V. Combining EQUAL with Error-Mitigation

Ensembles are generated by only adding controlled perturbations to the problem Hamiltonian. Therefore, they have limited capability to shift the noisy distribution from a QA towards the ideal distribution even if a large number of ensembles are used. Alternately, large perturbations may significantly change the landscape of the problem. Instead, we take an orthogonal approach and explore existing error-mitigation schemes that introduce a shift in the energy histogram.

A. Primer on Error-Mitigation Schemes for QA

Error-mitigation schemes for QAs can be classified into: (1) preprocessing techniques that are applied at the casting or embedding steps; (2) hardware-based schemes that control the device-level parameters; and (3) postprocessing policies that apply modifications on the outcomes obtained from QAs. For our analysis, we choose spin reversal transform, longer inter-sample delay, and single-qubit correction (SQC) as representatives of preprocessing, hardware-based, and postprocessing techniques, respectively [42]. We perform characterization studies for these three error mitigation schemes and found that SQC is the most effective scheme in (1) eliminating the systematic bias on their own and (2) shifting the noisy distribution of the QA towards the ideal distribution. Therefore, we use SQC as the error mitigation scheme for our study.

B. Overview of SQC PostProcessing

SQC is analogous to the gradient descent scheme but only applicable to discrete optimization problems. Instead of computing the gradient for determining the direction of the move in every iteration, SQC uses a greedy approach and moves to a neighbor with the lowest energy value. Fig. 13 illustrates the overview of an iteration of SQC. For each candidate outcome generated by executing the QMI, SQC evaluates the one Hamming distance away neighbors and move to the candidate with the lowest energy value. The process is repeatedly executed until we cannot find any new neighbor that has better quality.

C. EQUAL+: Combining EQUAL and SQC

Fig. 14 shows an overview of EQUAL+. EQUAL+ applies SQC on the outcomes of each QMI and obtains the best outcome for each QMI. The process is performed for each QMI in parallel. Once applying SQC on each QMI converges, the final output of EQUAL+ is picked as the candidate with the lowest energy among all the individual best candidates from the QMIs. The time to converge depends on several factors, such as the size of the problem, number of outcomes, and quality of the outcomes. However, our evaluations show that EQUAL+ converges within a few seconds even for large benchmarks such as the ones used in our evaluations.

Using this greedy approach helps locate neighbors from current outcomes that were not produced by the QA originally. With each neighbor located, EQUAL+ shifts the outcome distribution towards the ideal solution (global optima). Note that although SQC is effective on its own, the diversity of EQUAL+ is essential to improve its search space. The capability of SQC alone to introduce new outcomes is limited by the quality of outcomes from the QMI. In EQUAL+, the ensembles enable us to explore a much larger neighborhood compared to applying SQC alone. In the end, EQUAL+ may discover a solution from one of the weakest outcomes corresponding to one of the weakest QMIs (sub-optimal outcome that did not correspond to the best solution in any of the QMIs).

Note that EQUAL+ is versatile, and any other postprocessing candidate that introduces the desired shifting property in the energy distribution may be used. We use SQC for its performance and low time complexity.

D. Analysis of Overheads

We discuss the overheads for both EQUAL and EQUAL+. EQUAL generates the ensemble of QMIs prior to execution on the QA. As the perturbed Hamiltonian only adjusts the coefficients of the original problem QMI, the ensemble does not need to re-perform the casting or embedding step. Although embedding can take up to several hours and may fail for certain Hamiltonians, this overhead and limitation is entirely avoided by EQUAL. EQUAL also requires the programmer to estimate the precision of the hardware using a set of profiling experiments. However, profiling need not be done for each application. As the precision supported is only device-specific, profiling once for each QA hardware is enough, and the same information can be re-used for multiple applications. For execution on the QA, EQUAL requires the same number of trials as the baseline and therefore, does not incur any overhead of additional trials.

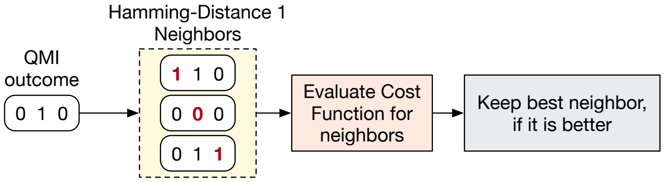
Hamming-Distance 1
Neighbors
QMI
outcome
0 1 0
1 1 0
0 0 0
0 1 1
Evaluate Cost Function for neighbors
Keep best neighbor, if it is better

Fig. 13. Single Qubit Correction PostProcessing [42].

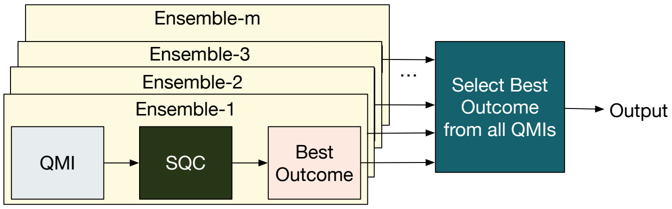

Fig. 14. Overview of EQUAL+ design. It applies the SQC postprocessing algorithm to the outcomes from each QMI in parallel. Finally, it selects the best outcome from all the QMIs as the output solution.

EQUAL+ incurs some additional overheads for the postprocessing step as it applies the SQC heuristic algorithm to all the outcomes obtained from all the QMIs. The space complexity of the postprocessing phase in EQUAL+ is linear with the number of qubits [42]. As SQC is iteratively applied to every outcome of a QMI, the time complexity depends on the number of outcomes which is equal to the number of trials in the worst case (assuming each trial generates a unique outcome). The postprocessing for each QMI is done in parallel. Our studies show that EQUAL+ converges within a few iterations and the postprocessing step for EQUAL+ only takes a few seconds. Therefore, the overheads are acceptable.

E. Case-Study: How EQUAL+ reduces Systematic Bias

Fig. 15 shows the histograms of energy values of benchmark B1 for EQUAL and EQUAL+. We observe that the optimal solution is at a distance d_1 from the ground state energy in EQUAL. EQUAL+ exploits the shifting property of SQC to obtain a solution at distance d_2 and is closer to the ground state energy ($d_2 < d_1$). Note that EQUAL+ shifts the overall histogram towards the ideal solution and achieves the intended goal. As EQUAL+ applies postprocessing on the outcomes from the QMIs, the introduced shift in the histogram does not alter the original problem Hamiltonian.

F. Results for Energy Residual

Fig. 16 shows the Energy Residual of EQUAL+ relative to the baseline. We also compare against EQUAL and SQC standalone. We observe that EQUAL+ improves the ER by 0.45 compared to the baseline on average and by up to 0.32. In other words, EQUAL+ improves the quality of solutions by 55% on average and up to 68%.

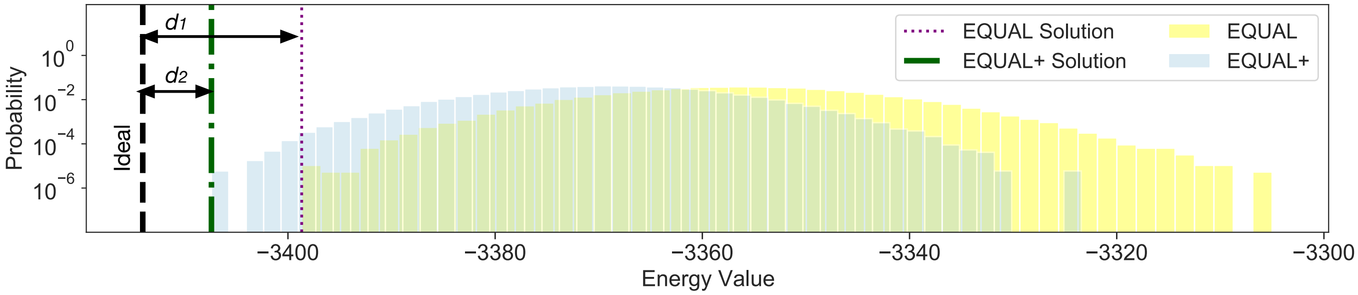

Fig. 15. Histogram of energy values from the outcomes on the QA for benchmark B1 using EQUAL and EQUAL+.

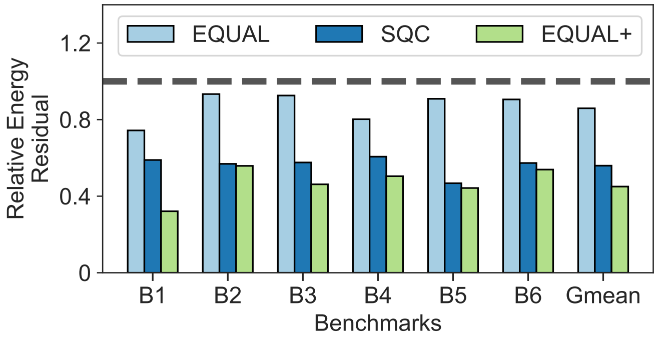

Fig. 16. Energy Residual using EQUAL+ relative to the baseline. We also compare with EQUAL and SQC standalone.

VI. Related Work

Both gate-model quantum computers and QAs promise significant advantages for a wide range of applications [11], [21], [26]–[41], [67], [72], [73]. Thus, developing error-mitigation policies is an active area of research for both QAs and gate-model quantum computers. We discuss prior works and compare them against schemes that use ensembles.

A. *Priors works using Ensembles*

The potential of ensembles has been explored for both gate-model quantum computers and QAs.

Ensemble policies for Gate-model quantum computers: Systematic bias in QAs is similar to correlated errors on gate-based quantum computers. To tackle these errors on gate-model quantum computers, recent studies propose the use of an ensemble of diverse mappings [52] or different machines [53]. Leveraging a similar approach for EQUAL is nontrivial due to the complexities involved in the embedding process, particularly for problems at scale. Instead, EQUAL uses an ensemble of QMIs by introducing controlled perturbations while minimizing the alterations in the functionality of the original problem Hamiltonian.

Ensemble policies for QAs: Using ensembles in QAs have been investigated at the casting level for two different applications [21], [37]. However, as each application uses its own casting algorithm, this approach cannot be generalized. On the other hand, EQUAL avoids such application-specific assumptions and is applicable irrespective of the problem at hand.

B. *Software error mitigation policies*

These techniques are either applied prior to the execution of the QMI (preprocessing) or after the QMI is executed (postprocessing). Preprocessing schemes transform the problem QMI at the casting or embedding level such that it is less vulnerable to errors during execution time [21], [42], [74]. Preprocessing schemes are analogous to compiler-level optimizations on gate-model quantum computers [22]–[24], [52], [53], [75]–[80], [80]–[90]. Postprocessing schemes for QAs exploit the fact that even if a QA cannot generate the solution with the lowest energy, it quickly locates the neighborhood where the optimal solution might reside. By modifying the outcome obtained from the QA using classical heuristic algorithms, postprocessing schemes can significantly improve the quality of solutions [42], [45], [46]. We use Multi-Qubit Correction (MQC) [42], [69] to obtain the best-known estimate of the ground state energy in our evaluations. However, this algorithm requires significant classical computational overheads and may take up to days to obtain a better quality solution. Nonetheless, the performance of MQC depends on the quality of the outcomes obtained from the QA and both EQUAL and EQUAL+ can benefit from it. Postprocessing algorithms can significantly improve the application fidelity even for gate-based quantum computers [53], [91]–[94].

VII. Conclusion

This paper proposes EQUAL—Ensemble Quantum Annealing—a software framework that creates multiple perturbed copies of an input problem by injecting controlled perturbations to the original problem Hamiltonian. By executing an ensemble of quantum machine instructions (QMIs), EQUAL projects the program to different noise profiles and therefore, different biases. Our evaluations using the 2041-qubit D-Wave QA show that EQUAL bridges the difference between the baseline and the ideal by an average of 14% (and up to 26%), without requiring any additional trials. We also propose EQUAL+, which exploits the properties of existing error mitigation schemes for enhanced performance. EQUAL+ bridges the difference between the baseline and the ideal by an average of 55% (and up to 68%).

References

[1] Peter W Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, 1999.

[2] Lov K Grover, "A fast quantum mechanical algorithm for database search," in *STOC 1996*, 1996.

[3] RP Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6, 1982.

[4] Seth Lloyd, "Universal quantum simulators," *Science*, 1996.

[5] Michael A Nielsen and Isaac L Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.

[6] Tameem Albash and Daniel A Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, 2018.

[7] International Business Machines Corporation, "Universal Quantum Computer Development at IBM:," http://research.ibm.com/ibm-q/research/, 2021, [Online; accessed 22-July-2021].

[8] Google, "Google Quantum AI," https://quantumai.google/, 2022, [Online; accessed 22-July-2021].

[9] Honeywell, "Honeywell Quantum Solutions," https://www.honeywell.com/us/en/company/quantum, 2021, [Online; accessed 22-July-2021].

[10] D-Wave Systems Inc., "The first and only quantum computer built for business," https://www.dwavesys.com/, 2022, [Online; accessed 22-July-2021].

[11] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al., "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, 2019.

[12] Benjamin Villalonga, Dmitry Lyakh, Sergio Boixo, Hartmut Neven, Travis S Humble, Rupak Biswas, Eleanor G Rieffel, Alan Ho, and Salvatore Mandrà, "Establishing the quantum supremacy frontier with a 281 pflop/s simulation," *Quantum Science and Technology*, vol. 5, no. 3, 2020.

[13] Andrew D King, Jack Raymond, Trevor Lanting, Sergei V Isakov, Masoud Mohseni, Gabriel Poulin-Lamarre, Sara Ejtemaee, William Bernoudy, Isil Ozfidan, Anatoly Yu Smirnov, et al., "Scaling advantage over path-integral monte carlo in quantum simulation of geometrically frustrated magnets," *Nature communications*, vol. 12, no. 1, 2021.

[14] Yulin Wu, Wan-Su Bao, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, et al., "Strong quantum computational advantage using a superconducting quantum processor," *arXiv:2106.14734*, 2021.

[15] Amazon, " Amazon Braket - Explore and experiment with quantum computing:," https://aws.amazon.com/braket/, 2022, [Online; accessed 22-July-2021].

[16] Microsoft, "Azure Quantum - Quantum Service — Microsoft Azure," https://azure.microsoft.com/en-us/services/quantum/#product-overview, 2022, [Online; accessed 22-July-2021].

[17] John Preskill, "Quantum computing in the nisq era and beyond," *arXiv:1801.00862*, 2018.

[18] Arnab Das and Bikas K Chakrabarti, "Colloquium: Quantum annealing and analog quantum computation," *Reviews of Modern Physics*, vol. 80, no. 3, 2008.

[19] Catherine C McGeoch, "Theory versus practice in annealing-based quantum computing," *Theoretical Computer Science*, 2020.

[20] Andrew Lucas, "Ising formulations of many np problems," *Frontiers in physics*, vol. 2, 2014.

[21] Ramin Ayanzadeh, Milton Halem, and Tim Finin, "Reinforcement quantum annealing: A hybrid quantum learning automata," *Scientific Reports*, vol. 10, no. 1, 2020.

[22] Alwin Zulehner, Alexandru Paler, and Robert Wille, "An efficient methodology for mapping quantum circuits to the ibm qx architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 7, 2018.

[23] Prakash Murali, Jonathan M Baker, Ali Javadi-Abhari, Frederic T Chong, and Margaret Martonosi, "Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers," in *ASPLOS 2019*, 2019.

[24] Swamit S Tannu and Moinuddin K Qureshi, "Not all qubits are created equal: a case for variability-aware policies for nisq-era quantum computers," in *ASPLOS 2019*, 2019.

[25] C McGeoch and P Farre, "The d-wave advantage system: An overview," Tech. Rep., Tech. Rep. (D-Wave Systems Inc, Burnaby, BC, Canada, 2020).

[26] Rupak Biswas, Zhang Jiang, Kostya Kechezhi, Sergey Knysh, Salvatore Mandra, Bryan O'Gorman, Alejandro Perdomo-Ortiz, Andre Petukhov, John Realpe-Gómez, Eleanor Rieffel, et al., "A nasa perspective on quantum computing: Opportunities and challenges," *Parallel Computing*, vol. 64, 2017.

[27] Eleanor G Rieffel, Davide Venturelli, Bryan O'Gorman, Minh B Do, Elicia M Prystay, and Vadim N Smelyanskiy, "A case study in programming a quantum annealer for hard operational planning problems," *Quantum Information Processing*, vol. 14, no. 1, 2015.

[28] Davide Venturelli, Dominic JJ Marchand, and Galo Rojo, "Quantum annealing implementation of job-shop scheduling," *arXiv:1506.08479*, 2015.

[29] Tony T Tran, Minh Do, Eleanor G Rieffel, Jeremy Frank, Zhihui Wang, Bryan O'Gorman, Davide Venturelli, and J Christopher Beck, "A hybrid quantum-classical approach to solving scheduling problems," in *SOCS 2016*, 2016.

[30] Zhengbing Bian, Fabian Chudak, Robert Brian Israel, Brad Lackey, William G Macready, and Aidan Roy, "Mapping constrained optimization problems to quantum annealing with application to fault diagnosis," *Frontiers in ICT*, vol. 3, 2016.

[31] Juexiao Su, Tianheng Tu, and Lei He, "A quantum annealing approach for boolean satisfiability problem," in *DAC 2016*. ACM, 2016.

[32] Daniel O'Malley, Velimir V Vesselinov, Boian S Alexandrov, and Ludmil B Alexandrov, "Nonnegative/binary matrix factorization with a d-wave quantum annealer," *PloS one*, vol. 13, no. 12, 2018.

[33] WangChun Peng, BaoNan Wang, Feng Hu, YunJiang Wang, XianJin Fang, XingYuan Chen, and Chao Wang, "Factoring larger integers with fewer qubits via quantum annealing with optimized parameters," *SCIENCE CHINA Physics, Mechanics & Astronomy*, vol. 62, no. 6, 2019.

[34] Feng Hu, Lucas Lamata, Mikel Sanz, Xi Chen, Xingyuan Chen, Chao Wang, and Enrique Solano, "Quantum computing cryptography: Finding cryptographic boolean functions with quantum annealing by a 2000 qubit d-wave quantum computer," *Physics Letters A*, vol. 384, no. 10, 2020.

[35] Alejandro Perdomo-Ortiz, Joseph Fluegemann, Sriram Narasimhan, Rupak Biswas, and Vadim N Smelyanskiy, "A quantum annealing approach for fault detection and diagnosis of graph-based systems," *The European Physical Journal Special Topics*, vol. 224, no. 1, 2015.

[36] Ramin Ayanzadeh, Seyedahmad Mousavi, Milton Halem, and Tim Finin, "Quantum annealing based binary compressive sensing with matrix uncertainty," *arXiv:1901.00088*, 2019.

[37] Ramin Ayanzadeh, Milton Halem, and Tim Finin, "An ensemble approach for compressive sensing with quantum annealers," in *IGARSS 2020*. IEEE, 2020.

[38] Daisuke Inoue, Akihisa Okada, Tadayoshi Matsumori, Kazuyuki Aihara, and Hiroaki Yoshida, "Traffic signal optimization on a square lattice with quantum annealing," *Scientific reports*, vol. 11, no. 1, 2021.

[39] Nada Elsokkary, Faisal Shah Khan, Davide La Torre, Travis S Humble, and Joel Gottlieb, "Financial portfolio management using d-wave quantum optimizer: The case of abu dhabi securities exchange," Tech. Rep., Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2017.

[40] Koki Kitai, Jiang Guo, Shenghong Ju, Shu Tanaka, Koji Tsuda, Junichiro Shiomi, and Ryo Tamura, "Designing metamaterials with quantum annealing and factorization machines," *Physical Review Research*, vol. 2, no. 1, 2020.

[41] Vikram Khipple Mulligan, Hans Melo, Haley Irene Merritt, Stewart Slocum, Brian D Weitzner, Andrew M Watkins, P Douglas Renfrew, Craig Pelissier, Paramjit S Arora, and Richard Bonneau, "Designing peptides on a quantum computer," *bioRxiv*, 2020.

[42] Ramin Ayanzadeh, John Dorband, Milton Halem, and Tim Finin, "Multi-qubit correction for quantum annealers," *Scientific Reports*, vol. 11, 2021.

[43] Jun Cai, William G Macready, and Aidan Roy, "A practical heuristic for finding graph minors," *arXiv:1406.2741*, 2014.

[44] Prasanna Date, Robert Patton, Catherine Schuman, and Thomas Potok, "Efficiently embedding qubo problems on adiabatic quantum computers," *Quantum Information Processing*, vol. 18, no. 4, 2019.

[45] John K Golden and Daniel O'Malley, "Pre-and post-processing in quantum-computational hydrologic inverse analysis," *Quantum Information Processing*, vol. 20, no. 5, 2021.

[46] Ajinkya Borle and Josh McCarter, "On post-processing the results of quantum optimizers," in *TPNC 2019*. Springer, 2019.

[47] Timothy D Goodrich, Blair D Sullivan, and Travis S Humble, “Optimizing adiabatic quantum program compilation using a graph-theoretic framework,” *Quantum Information Processing*, vol. 17, no. 5, 2018.

[48] Shuntaro Okada, Masayuki Ohzeki, Masayoshi Terabe, and Shinichiro Taguchi, “Improving solutions by embedding larger subproblems in a d-wave quantum annealer,” *Scientific reports*, vol. 9, no. 1, 2019.

[49] D-Wave Systems Inc., “D-wave ocean software documentation,” https://docs.ocean.dwavesys.com/en/stable/, 2022, [Online; accessed 22-July-2021].

[50] Walter Vinci, Tameem Albash, and Daniel A Lidar, “Nested quantum annealing correction,” *npj Quantum Information*, vol. 2, no. 1, 2016.

[51] Hidetoshi Nishimori and Kabuki Takada, “Exponential enhancement of the efficiency of quantum annealing by non-stoquastic hamiltonians,” *Frontiers in ICT*, vol. 4, 2017.

[52] Swamit S Tannu and Moinuddin Qureshi, “Ensemble of diverse mappings: Improving reliability of quantum computers by orchestrating dissimilar mistakes,” in *MICRO 2019*, 2019.

[53] Tirthak Patel and Devesh Tiwari, “Veritas: accurately estimating the correct output on noisy intermediate-scale quantum computers,” in *SC20*. IEEE, 2020.

[54] Patricia Amara, D Hsu, and John E Straub, “Global energy minimum searches using an approximate solution of the imaginary time schrödinger equation,” *The Journal of Physical Chemistry*, vol. 97, no. 25, 1993.

[55] AB Finnila, MA Gomez, C Sebenik, C Stenson, and JD Doll, “Quantum annealing: a new method for minimizing multidimensional functions,” *Chemical physics letters*, vol. 219, no. 5-6, 1994.

[56] Tadashi Kadowaki and Hidetoshi Nishimori, “Quantum annealing in the transverse ising model,” *Physical Review E*, vol. 58, no. 5, 1998.

[57] Masayuki Ohzeki and Hidetoshi Nishimori, “Quantum annealing: An introduction and new developments,” *Journal of Computational and Theoretical Nanoscience*, vol. 8, no. 6, 2011.

[58] David Sherrington and Scott Kirkpatrick, “Solvable model of a spin-glass,” *Physical review letters*, vol. 35, no. 26, 1975.

[59] Matthew P Harrigan, Kevin J Sung, Matthew Neeley, Kevin J Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C Bardin, Rami Barends, Sergio Boixo, et al., “Quantum approximate optimization of non-planar graph problems on a planar superconducting processor,” *Nature Physics*, vol. 17, no. 3, 2021.

[60] Quantum AI team and collaborators, “Recirq,” Oct. 2020.

[61] Google AI Quantum and Collaborators, “Sycamore qaoa experimental data,” 7 2020.

[62] John E Dorband, “Extending the d-wave with support for higher precision coefficients,” *arXiv:1807.05244*, 2018.

[63] Kristen L Pudenz, Tameem Albash, and Daniel A Lidar, “Quantum annealing correction for random ising problems,” *Physical Review A*, vol. 91, no. 4, 2015.

[64] Tomas Boothby, Andrew D King, and Aidan Roy, “Fast clique minor generation in chimera qubit connectivity graphs,” *Quantum Information Processing*, vol. 15, no. 1, 2016.

[65] Hamed Karimi, Gili Rosenberg, and Helmut G Katzgraber, “Effective optimization using sample persistence: A case study on quantum annealers and various monte carlo optimization methods,” *Physical Review E*, vol. 96, no. 4, 2017.

[66] Hamed Karimi and Gili Rosenberg, “Boosting quantum annealer performance via sample persistence,” *Quantum Information Processing*, vol. 16, no. 7, 2017.

[67] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann, “A quantum approximate optimization algorithm,” *arXiv:1411.4028*, 2014.

[68] Ramin Ayanzadeh, Milton Halem, John Dorband, and Tim Finin, “Quantum-assisted greedy algorithms,” *arXiv:1912.02362*, 2019.

[69] Ramin Ayanzadeh, John Dorband, Milton Halem, and Tim Finin, “Multi qubit correction (mqc) for quantum annealers,” 2021, Python implementation of MQC.

[70] Ravindra K Ahuja, Claudio B Cunha, and Güvenç Şahin, “Network models in railroad planning and scheduling,” in *Emerging theory, methods, and applications*. INFORMS, 2005.

[71] Brian Carlson, Yonghong Chen, Mingguo Hong, Roy Jones, Kevin Larson, Xingwang Ma, Peter Nieuwesteeg, Haili Song, Kimberly Sperry, Matthew Tackett, et al., “Miso unlocks billions in savings through the application of operations research for energy and ancillary services markets,” *Interfaces*, vol. 42, no. 1, 2012.

[72] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alan Aspuru-Guzik, and Jeremy L O’brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature communications*, vol. 5, no. 1, 2014.

[73] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta, “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets,” *Nature*, vol. 549, no. 7671, 2017.

[74] Elijah Pelofske, Georg Hahn, and Hristo Djidjev, “Optimizing the spin reversal transform on the d-wave 2000q,” *arXiv:1906.10955*, 2019.

[75] Wei Tang, Teague Tomesh, Martin Suchara, Jeffrey Larson, and Margaret Martonosi, “Cutqc: using small quantum computers for large quantum circuit evaluations,” in *ASPLOS 2021*, 2021.

[76] Prakash Murali, Norbert Matthias Linke, Margaret Martonosi, Ali Javadi Abhari, Nhung Hong Nguyen, and Cinthia Huerta Alderete, “Full-stack, real-system quantum computer studies: Architectural comparisons and design insights,” in *ISCA 2019*. IEEE, 2019.

[77] Tirthak Patel, Baolin Li, Rohan Basu Roy, and Devesh Tiwari, “{UREQA}: Leveraging operation-aware error rates for effective quantum circuit mapping on nisq-era quantum computers,” in *USENIX ATC ’20*, 2020.

[78] Junde Li, Mahabubul Alam, and Swaroop Ghosh, “Large-scale quantum approximate optimization via divide-and-conquer,” *arXiv:2102.13288*, 2021.

[79] Mahabubul Alam, Abdullah Ash-Saki, and Swaroop Ghosh, “An efficient circuit compilation flow for quantum approximate optimization algorithm,” in *DAC 2020*. IEEE, 2020.

[80] Prakash Murali, Norbert M Linke, Margaret Martonosi, Ali Javadi Abhari, Nhung Hong Nguyen, and Cinthia Huerta Alderete, “Architecting noisy intermediate-scale quantum computers: A real-system study,” *IEEE Micro*, vol. 40, no. 3, 2020.

[81] Tirthak Patel and Devesh Tiwari, “Disq: a novel quantum output state classification method on ibm quantum computers using openpulse,” in *ICCAD 2020*, 2020.

[82] Yunong Shi, Nelson Leung, Pranav Gokhale, Zane Rossi, David I Schuster, Henry Hoffmann, and Frederic T Chong, “Optimized compilation of aggregated instructions for realistic quantum computers,” in *ASPLOS 2019*, 2019.

[83] George S Barron and Christopher J Wood, “Measurement error mitigation for variational quantum algorithms,” *arXiv:2010.08520*, 2020.

[84] Hyeokjea Kwon and Joonwoo Bae, “A hybrid quantum-classical approach to mitigating measurement errors,” *arXiv:2003.12314*, 2020.

[85] Pranav Gokhale, Yongshan Ding, Thomas Propson, Christopher Winkler, Nelson Leung, Yunong Shi, David I Schuster, Henry Hoffmann, and Frederic T Chong, “Partial Compilation of Variational Algorithms for Noisy Intermediate-Scale Quantum Machines,” in *MICRO 2019*. ACM, 2019.

[86] Mahabubul Alam, Abdullah Ash-Saki, and Swaroop Ghosh, “Circuit compilation methodologies for quantum approximate optimization algorithm,” in *MICRO 2020*. IEEE, 2020.

[87] Prakash Murali, David C McKay, Margaret Martonosi, and Ali Javadi-Abhari, “Software Mitigation of Crosstalk on Noisy Intermediate-Scale Quantum Computers,” *arXiv:2001.02826*, 2020.

[88] Swamit S Tannu and Moinuddin K Qureshi, “Mitigating measurement errors in quantum computers by exploiting state-dependent bias,” in *MICRO 2019*, 2019.

[89] Gushu Li, Yufei Ding, and Yuan Xie, “Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices,” *arXiv:1809.02573*, 2018.

[90] Pranav Gokhale, Ali Javadi-Abhari, Nathan Earnest, Yunong Shi, and Frederic T Chong, “Optimized Quantum Compilation for Near-Term Algorithms with OpenPulse,” *arXiv:2004.11205*, 2020.

[91] IBM, “Measurement Error Mitigation,” https://qiskit.org/textbook/ch-quantum-hardware/measurement-error-mitigation.html, 2010, [Online; accessed 26-July-2020].

[92] Sergey Bravyi, Sarah Sheldon, Abhinav Kandala, David C Mckay, and Jay M Gambetta, “Mitigating measurement errors in multi-qubit experiments,” *arXiv:2006.14044*, 2020.

[93] Tirthak Patel and Devesh Tiwari, “Qraft: reverse your quantum circuit and know the correct program output,” in *ASPLOS 2021*, 2021.

[94] Swamit S Tannu, Poulami Das, Ramin Ayanzadeh, and Moinuddin K Qureshi, “Hammer: Boosting fidelity of noisy quantum circuits by exploiting hamming behavior of erroneous outcomes,” in *ASPLOS 2022*, 2022.