Ensemble of Diverse Mappings:
Improving Reliability of Quantum Computers by Orchestrating Dissimilar Mistakes

Swamit S. Tannu
swamit@gatech.edu
Georgia Institute Technology

Moinuddin Qureshi
moin@gatech.edu
Georgia Institute Technology

ABSTRACT

Near-term quantum computers do not have the ability to perform error correction. Such *Noisy Intermediate Scale Quantum (NISQ)* computers can produce incorrect output as the computation is subjected to errors. The applications on a NISQ machine try to infer the correct output by running the same program thousands of times and logging the output. If the error rates are low and the errors are not correlated, then the correct answer can be inferred as the one appearing with the highest frequency. Unfortunately, quantum computers are subjected to correlated errors, which can cause an incorrect answer to appear more frequently than the correct answer.

We observe that recent work on qubit mapping (including the recent work on variation-aware mapping) tries to obtain the best possible qubit allocation and uses it for all the trials. This approach significantly increases the vulnerability to correlated errors – if the mapping becomes susceptible to a particular form of error, then all the trials will get subjected to the same error, which can cause the same wrong answer to appear as the output for a significant fraction of the trials. To mitigate the vulnerability to such correlated errors, this paper leverages the concept of diversity and proposes an *Ensemble of Diverse Mappings (EDM)*. EDM uses diversity in qubit allocation to run copies of an input program with a diverse set of mappings, thus steering the trials towards making different mistakes. By combining the output probability distributions of the diverse ensemble, EDM amplifies the correct answer by suppressing the incorrect answers. Our experiments with `ibmq-melbourne` (14-qubit) machine shows that EDM improves the inference quality by 2.3x compared to the current state-of-the-art mapping algorithms.

CCS CONCEPTS

• **Hardware: Quantum technologies**;

KEYWORDS

Quantum Compilers, Correlated Errors, NISQ

ACM Reference Format:
Swamit S. Tannu and Moinuddin Qureshi. 2019. "Ensemble of Diverse Mappings:Improving Reliability of Quantum Computers by Orchestrating Dissimilar Mistakes" . In *The 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-52), October 12–16, 2019, Columbus, OH, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3352460.3358257

MICRO-52, October 12–16, 2019, Columbus, OH, USA

ACM ISBN 978-1-4503-6938-1/19/10…$15.00
https://doi.org/10.1145/3352460.3358257

1 INTRODUCTION

Near-term quantum computers face significant reliability challenges as the qubits are extremely fickle and error-prone. Furthermore, with a limited number of qubits, implementing quantum error correction (QEC) may not be possible as QEC require 20 to 50 physical qubit devices to build a single fault-tolerant qubit. Therefore, fault-tolerant quantum computing is likely to become viable only when we have a system with thousands of qubits. In the meanwhile, the near-term quantum computes with several dozens of qubits are expected to operate in a noisy environment without any error correction using a model of computation called as *Noisy Intermediate Scale Quantum (NISQ) Computing* [35].

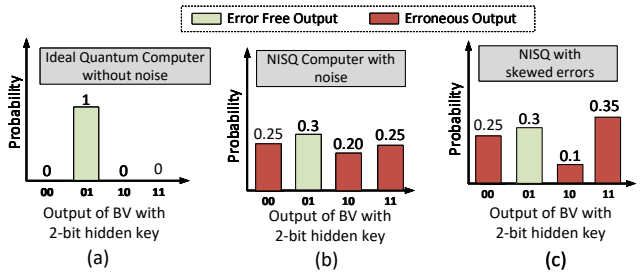

Figure 1: Output of Bernstein-Vazirani with 2-bit key on (a) Ideal machine (b) NISQ machine providing a correct answer (c) NISQ machine providing a wrong answer

The NISQ machines can produce an incorrect output as the computation is subjected to errors. Therefore, to infer the correct answer, the program is run thousands of the times on the NISQ machine to produce a probability distribution of the possible output states. This distribution is analyzed to infer the correct answer, for example, by selecting the most frequently occurring output. Consider the Bernstein-Vazirani (BV) algorithm that allows the program to infer the hidden key in a single shot. On an idealized machine, this program will provide the correct answer with a probability of 1, as shown in Figure 1(a). However, if we execute BV on a NISQ machine, then we will get the correct answer for some trials and wrong answer for others. Figure 1(b) shows the output distribution for BV, where the correct answer occurs with 30% probability and the most dominant incorrect answer occurs with 25% probability. The correct answer can be inferred by selecting the most frequent output. Unfortunately, the NISQ machine can have correlated errors that cause the same incorrect answer to appear with a high frequency. Inferring the correct answer can become challenging in such scenarios. For example, consider Figure 1(c), where the correct answer still occurs with 30% probability, but one of the incorrect answers occurs with 35% probability. We observe

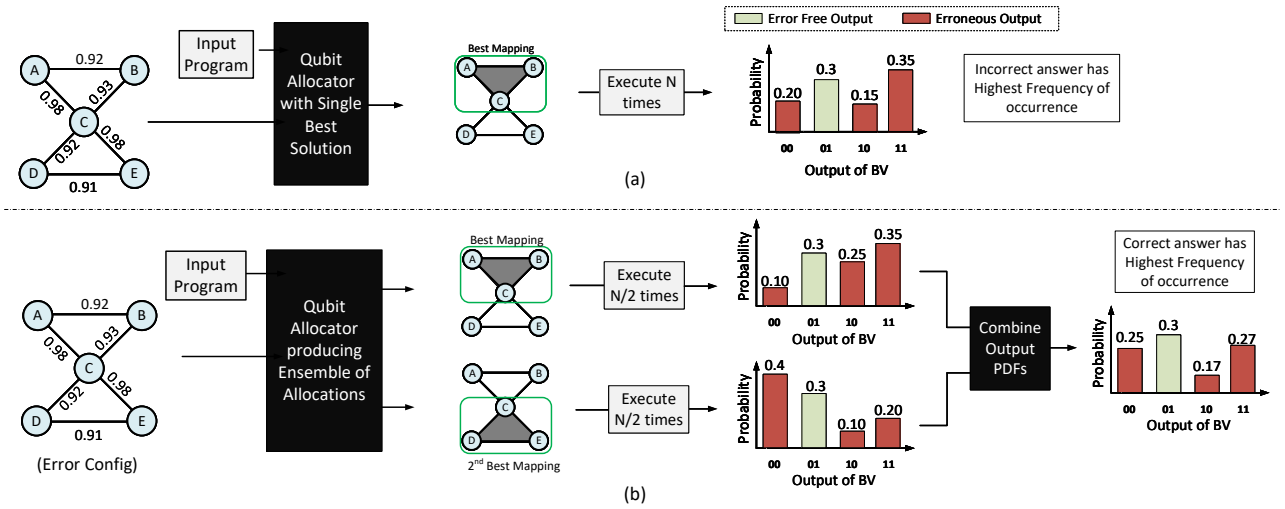

Figure 2: Bernstein-Vazirani using (a) Single best mapping (b) Ensemble of Diverse Mappings (EDM), running two allocations and merging the outputs (EDM infers correct answer even if both mappings have a dominant incorrect answer).

that the task of inferring the correct answer can be achieved via two means: increasing the probability of the correct answer or by reducing the probability of the dominant wrong answer. Recent work on qubit allocation policies (swap minimizing or variation-aware) have focused on the former, whereas, in this paper, we focus on the latter.

Qubit allocation policies deal with the problem of assigning the program qubits to the physical qubits (qubit assignment) and moving the qubit from source to destination for performing two-qubit operations (qubit routing). Qubit allocation policies have a significant impact on the reliability of the NISQ machine as these policies can determine the number of operations required to execute a given program. Routing of the qubit from source to destination is typically accomplished by inserting additional SWAP instruction that can swap two neighboring qubits. Recently proposed qubit mapping policies try to minimize the number of SWAP instructions. Recent studies have also investigated variation-aware qubit mapping policies that try to use the strongest qubits and links (the ones with lowest error rates) to perform the computation. All of the prior proposals on intelligent qubit mapping (both SWAP minimizing and variation-aware) try to determine the best mapping and use that mapping for running all of the trials on the NISQ machine. Unfortunately, such an approach also makes the application vulnerable to correlated errors – if the computation is subjected to a particular error, the computation for all of the trials will continue to be performed on the same set of qubits and links, causing the same erroneous output to occur for a large number of trials.

To mitigate the vulnerability to such correlated errors, this paper leverages the concept of diversity,[1] and proposes *Ensemble of Diverse Mappings (EDM)*. EDM is based on the insight that rather than having all the trials be subjected to the same sources of errors, split the trials into multiple groups, and have a different mapping for each group so that the trials in each group get subjected to different sources or errors and hence different incorrect outputs. For example, consider the scenario is shown in Figure 2(a) where the baseline performs N trials using the best mapping and still obtains an incorrect output. EDM splits the N trials into two groups and uses a different mapping (best and the second-best) for these groups. Even though both of these groups individually produce an incorrect answer with the highest probability, these incorrect answers are different – so when we merge the output distributions, the incorrect outputs end up getting attenuated, and the correct answer ends up getting accentuated. Even though the two groups individually failed to produce the correct answer, the diversity in EDM allows the ensemble to infer the correct answer. While we explain EDM with two mappings, EDM can be implemented with more than two mappings. For our studies, we use EDM with four mappings, with each mapping used for one-quarter of the trials.

We note that while EDM increases the likelihood that the system is able to infer the correct answer, it does so by reducing the amplitude of the dominant incorrect answers and not by proactively increasing the amplitude of the correct answer. To analyze the impact of our solution on the ability of the NISQ machine to infer the correct answer, we introduce a metric termed as the *Inference Strength (IST)*, which is the ratio of the frequency of the correct answer to the frequency of the most frequently occurring wrong answer. When IST exceeds 1, the system can infer the correct answer but not otherwise, and this is true regardless of the probability of the correct output. Thus, IST is an intuitive metric to reason about inference. Our evaluations on `ibmq-melbourne`, show that IEDM improves the IST by up to 1.6x.

Our EDM implementation gives equal weights to the output produced by each of the mappings. We observe that further robustness against correlated errors can be obtained by weighing each of the output distribution differently, depending on the measure of

[1] We note that when a team is formed with members of very similar skills and backgrounds, then all the members may share the same *blind-spot* and the team overall becomes vulnerable to that blind-spot. Whereas, when teams are formed with members of a diverse set of skills and backgrounds, then each member may have a different blind-spot, which may not be present in the other team members, making the overall group more resilient to such blind-spots.

divergence (diversity) of the output that it produces compared to the other outputs in EDM. We propose *Weighted EDM (WEDM)* that is based on this insight of non-uniform weights. Our evaluations on `ibmq-melbourne` show that WEDM increase the IST by up to 2.3x. Overall, our paper makes the following contributions:

(1) We observe that current quantum machines can have correlated errors and mitigating correlated errors can improve the ability of the NISQ machine to infer the correct answer.
(2) We propose *Ensemble of Diverse Mappings (EDM)* to tolerate correlated errors. Rather than using a single mapping for all the trials, EDM splits the trials into groups and applies a diverse mapping to each group.
(3) We propose *Weighted Ensemble of Diverse Mappings (WEDM)* that places different weights to the output produced by each mapping, depending on the divergence. WEDM further improves reliability compared to EDM.

2 BACKGROUND

The last five years represent a significant milestone in the history of quantum computing, where the field has moved from theoretical ideas and single-qubit demonstrations to having several systems with dozen or more qubits. Several industry labs have announced blueprints for quantum computers with few dozens of qubits [2, 20, 21]. The available quantum computers provide a unique opportunity to understand the exact types of errors and behaviors that happen on a real quantum device and enable efficient solutions that are based on exploiting this understanding of errors. Quantum computation is based on two key principles: superposition and entanglement. Quantum algorithms leverage these principles to perform operations and can solve problems that are intractable on conventional machines. Unfortunately, qubits are susceptible to errors.

2.1 Errors in Quantum Computers

Qubits can encounter Coherence-errors, Gate-errors, and State Preparation and Measurement (SPAM) errors.

Coherence Errors: Coherence errors result from a natural tendency of qubit devices to attain the lowest possible energy state. Coherence errors are analogous to retention errors in conventional systems. However, conventional computers are only subjected to bit-flip errors, whereas, quantum computers can experience both bit-flip and phase flip errors [8]. Coherence times for current quantum computers is quite small. For example, on IBM quantum computers, T1 coherence time (that affects the probability of bit-flips) is about 50μS. Whereas, T2 coherence time (that affects the probability of phase-flip error) is about 30μS.

Gate Errors: Quantum operations or gates manipulate the state of a qubit. Unfortunately, quantum gates are not perfect as performing operations on qubits can result in undesired state changes. For example on an IBM quantum-computer, single qubit gate that is used to manipulate the state of an individual qubit can encounter an error with a probability of 0.1% such that there is about one in thousand chance that single qubit gate operation would produce an undesired state change. Whereas, a two-qubit gate that entangles the state of two quantum bits, show an average error rate of 4% on IBM quantum computers. The two-qubit operational errors are one of the most dominant forms of errors on quantum computers as they limit the number of operations we can perform before a program encounters an error.

SPAM Errors: Current quantum computers are susceptible to State Preparation and Measurement (SPAM) Errors. For instance, on IBM machine, all qubits are initialized to "$|0\rangle$" state at the beginning of the program. Unfortunately, there is small chance that a qubit may not be correctly initialized. This is known as state preparation error. Similarly reading the state of a qubit can be erroneous. Qubit is a superposition of two basis states:$|0\rangle$ and $|1\rangle$. When measured, qubit produces a binary output: either 1 or 0 depending on the degree of superposition. Unfortunately, the process of measurement is erroneous as sensing the state of the qubit is challenging due to the extremely low energy associated with the qubit. On IBM machines, average qubit measurement error rate is 8%, whereas the worst case measurement error rate can be up to 30%.

2.2 NISQ Model for Quantum Computing

Near-term quantum computers with few hundreds of qubits can not leverage error correction even for an application requiring few dozens of logical qubits. However, there is hope that some important class of applications (such as discrete optimization and quantum chemistry simulations) can still be viable with *Noisy and Intermediate-Scale Quantum (NISQ)* [27] model of computing.

In NISQ model, a program can produce incorrect output as there are no formal guarantees of fault-tolerance. To produce correct output, a program must be executed for multiple trials, and the output for each trial is logged. The output log for a NISQ program is a collection of both correct and incorrect answers. We can infer the correct results by analyzing the output log. If errors on NISQ computers are independent and occur with a low error rate, then the correct answer will appear with the highest frequency. Unfortunately, existing NISQ machines are susceptible to correlated errors that produce few incorrect answers more frequently than the correct answer.

2.3 Qubit Allocation Problem

The computational power of quantum computers (QC) stem from the ability to produce entangled qubit states. Qubits can be entangled using a two-qubit gate such as CNOT. However, for superconducting qubits, such as IBM quantum computers, we can perform two-qubit gate if the qubits are physically connected via a coupling resonator. Unfortunately, due to complex design and large area required for coupling resonators, we can not build a solid-state quantum computer with all to all connectivity. Existing QC uses limited connectivity such that only neighboring qubits are connected using coupling resonators.

It is possible to entangle physical qubits without a direct connection by moving qubit data from one device to another. For example, we can use SWAP-gate that move qubit state from one physical qubit to another physical qubit. By using the sequence of SWAP gates, we can facilitate an entanglement between any two qubits on a quantum computer with limited connectivity. Unfortunately, SWAP operations are unreliable (with average error rate of 8% to 11% on IBM machines) and inserting extra SWAP operations can

degrade the reliability of an application. To mitigate this problem, prior works [38, 48, 49] have developed qubit allocation algorithms that search for program qubit to physical qubit mapping that reduce the number of SWAP operations.

2.4 Variation-Aware Qubit Mapping

The different qubits and links of a NISQ machines can have widely varying error rates as not all qubits have the same level of vulnerability to errors, and this variation in error rates has a large impact on the reliability of NISQ applications [13, 28, 31, 40]. For example, if we can map a program on the most reliable qubit, then the probability of errors can be reduced significantly (up to 10x). Especially for a class of NISQ programs that use less than available physical qubits, a programmer can choose the most reliable qubits to improve the reliability. Moreover, we can extend the idea of variation-aware allocation to qubit movement. For example, SWAP operations are unreliable and show significant variation in reliability (up to 20x on IBM-Q14), by using quantum links with high reliability and avoiding links with low reliability the system can reduce the impact of noise on NISQ machines. This makes the overall system reliability be dictated less by the worst-case qubits and links, and more by the average-case qubits and links.

To enable variation-aware techniques, we need error characterization data that describe the error rates for all the qubits and the links on a quantum computer. Fortunately, the error rates can be evaluated using randomized bench-marking and gate tomography. For IBM machines the error rates are evaluated after every calibration cycle, and the error characterization data is available to the programmer using IBM's qiskit API. However, the estimated error rates are not constant as qubit are non-linear devices that can have time-varying deviations due to drift and changing operating conditions. Our experimental evaluations show the relative reliability of collection of qubits and quantum links to largely have repeatable behavior. To estimate the reliability of the circuit in a variation-aware manner, prior works have used the *Estimated Probability of Success (ESP)* metric [31]. ESP for an executable can be computed by taking a product of all the gate success rates (g^s) and measurement success rates (m^s). The gate success rate is the probability of performing a gate without any error, which is calculated using the gate error rate (g^e). The measurement success rate (m^s) captures the probability of performing all measurements without any error. ESP is given by the equation below. Variation-aware mapping scheme tries to find the mapping that has the highest ESP. We use a variation-aware mapping policy as our baseline.

$$ESP = \prod_{i=1}^{N_{gates}} g_i^s * \prod_{j=0}^{N_{meas}} m_i^s$$

$$g_i^s = (1 - g_i^e) \quad m_i^s = (1 - m_i^e)$$

2.5 The Inference Problem for NISQ

A NISQ machine is subjected to errors. Therefore, to infer the right answer, the given program is run for thousands of trials, and the output of each trial is logged. In the end, we get an output probability distribution that is influenced by both correct and incorrect answers. The task of inferring the correct answer becomes challenging at high error rates. For example, if the error rate is small, then the correct answer would appear with the highest frequency. As qubit error rate increases the likelihood of correct answer decreases significantly such that the incorrect answers may be produced as frequently as correct answers.

We can improve the inference quality of the NISQ machine by either increasing the frequency of the correct answer or by reducing the occurrence of the most common wrong answer. Existing mapping policies focus only on the first option and try to perform the computation using the strongest qubits and links. Therefore, they run all the trials using the mapping that maximizes the probability of getting the correct answer.

2.6 The Challenge: Correlated Errors

We observe that even with the mapping that maximizes the ESP, NISQ machines can fail to provide the correct answer as the most frequently occurring outcome. In such cases, a particular incorrect answer dominates the correct answer. The wrong answer occurring with a high frequency happens because the computation gets subjected to similar types of error repeatedly leading the same wrong outcome. Thus, quantum computers can have correlated errors. Current approach to performing all the trials with a single mapping policy makes the application vulnerable to correlated errors – if the computation is subjected to a particular error, the computation for all of the trials will continue to be performed on the same set of qubits and links, causing the same erroneous output to occur for large number of the trials. Correlated errors is a real problem on IBM quantum machines, for example, recent study reports the correlated nature of SPAM errors [39]. In this paper we develop solutions for addressing the correlation in the incorrect answer. We provide the characterization for correlated errors next.

3 CORRELATED ERRORS ON NISQ

In this section, we analyze the correlation in errors on IBM's fourteen qubit machine (IBMQ-14) and study how the correlated errors produce incorrect answers such that the frequency of some incorrect answers is more than the correct answers.

3.1 Impact of Noise on Application Reliability

IBMQ-14 suffers from high measurement and gate error-rates. To understand the nature of errors and the impact on the system reliability, we execute the Bernstein-Vazirani (BV) benchmark with a 6-bit secret key. We perform each experiment for 16 thousand trials. Figure 3 shows the probability distribution for the different outcomes, with the outcomes arranged from the highest frequency of occurrence to the lowest. Notice that due to high error rates, the probability of getting the correct answer is fairly low (2.8%) and the output log consists all 64 possible outcomes (63 incorrect answers plus one correct answer). Furthermore, some of the incorrect outputs occur with almost 1.5x the frequency of the correct answer. We observe that the relative strength of the correct answer (probability normalized to the most frequent incorrect answer) is only 68%, and therefore, inferring the correct answer is not straightforward. In the Appendix-A, we describe how correlated errors can degrade quality of inference in NISQ model.

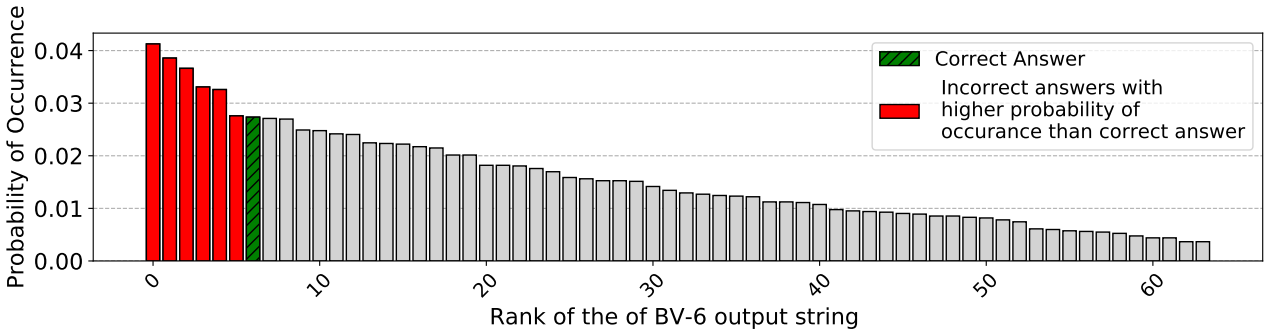

Figure 3: Output probability distribution for *Bernstein-Vazirani* (BV-6) with 6-bit hidden key executed on the IBM-Q14 machine (note that the states are sorted by the frequency of occurrence, from the highest to the lowest).

3.2 Correlation in Errors

To test if using the same set of qubits cause correlated errors, we execute two sets of experiments. The first set containing eight runs using the best mapping and the second set contains eight runs with different mappings (top-8 mappings).

BV-6 with Single Best Mapping: We run eight copies of BV-6 with single best mapping that maximizes the reliability. To understand if the trials with single mapping produce similar incorrect answers, we measure the divergence or dissimilarity between the output probability distribution using the *KL-divergence*. KL-divergence estimates the distance between the pair of probability distributions. If KL-divergence is close to zero, then the output distributions are similar. Figure 4 shows the heat map (darker shades are close to zero, indicating similarity) that illustrates the pairwise divergence between output probability distribution of BV-6 runs. All non-diagonal elements (d_{ij}) represent the divergence between the output of i^{th} and j^{th} run when BV-6 is executed with the single best mapping. Note that the pairwise KL-divergence between all the runs are close to zero. Thus with identical mapping, NISQ programs tend to produce similar incorrect outputs.

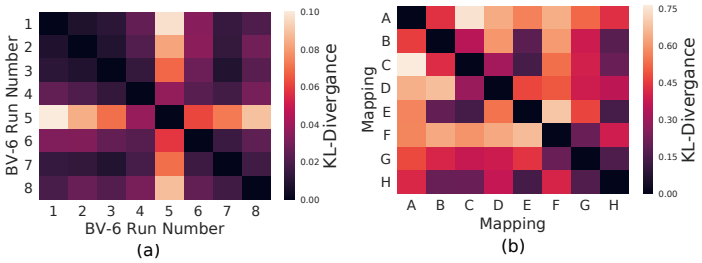

Figure 4: (a) Divergence between output of eight BV-6 runs with the strongest mapping. Dark squares indicate a value close to zero (indicating that the distributions are close to identical). (b) Pairwise divergence for the output of eight copies of BV-6 that are run with eight different mappings (light colors indicate divergent distributions).

BV-6 with Diverse Mappings: In the second experiment, we run BV-6 benchmark with eight completely different mappings and estimate the divergence between output probability distributions. Figure 4 shows a heat-map corresponding to the pairwise KL divergence for the eight copies of BV-6. We observe significant dissimilarity between all the eight copies of BV-6 such that average KL-divergence between two copies is 0.5, which is significantly higher as compared to eight runs of single best copy with average KL-divergence of 0.03. Furthermore, the most frequently occurring incorrect answers show a large variation across eight copies of BV-6. Thus by introducing diversity in the qubit mapping, we can enable diversity in the output probability distribution. Note that all the mappings used were within 10% of the ESP of best mapping and the executed identical number of gates.

4 EXPERIMENTAL METHODOLOGY

In this section, we briefly describe the benchmarks, system configuration, and the metrics used in our work.

4.1 Benchmarks

Existing quantum computers such as publicly available IBM fourteen qubit machine are severely limited due to noise. Due to low coherence and high gate error rates it can execute circuits with small number of qubits for short duration (low depth). Table 1 describe benchmarks and total number of single qubit gate operations (SG), CNOT operations (CX), and measurement operations (M) for the respective benchmarks.

Greycode Decoder: Grey code decoder decodes a binary string to a grey code string using a reversible circuit. For this benchmark, number of two qubit and measurement operations scale linearly with number of qubits. We use six bit circuit described in Rev-Lib [34]. The greycode benchmark is used to understand the effects of correlated errors on shallow circuit that measure qubits in standard basis. Moreover, greycode has identical number of measurement and two-qubit gates, which is useful to understand if the correlation in errors stem from measurement or two qubit operations.

Bernstein-Vazirani (BV): BV finds a n-bit binary secret encoded in the quantum oracle by querying the oracle once. On execution, BV outputs a binary string corresponding to the secret key. For BV, number of two qubit and single qubit gates scale linearly with number of qubits. BV is sensitive to phase and T2 errors as it measure qubits in Hadamard basis. We use two instances of BV to understand

if SWAPs can cause correlated errors as BV-7 has one additional SWAP operation compered to BV-6.

Quantum Approximate Optimization Algorithm: QAOA is a generalized algorithm that can be used to solve combinatorial optimization problems. We use QAOA to solve the max-cut problem, which tries to partition an input graph into two subsets (S_1,S_2) of nodes to maximize the number of edges between the first (S_1) and the second(S_2) subset. Note that QAOA-5, QAOA-6, QAOA-7 do not require any SWAP operations. For QAOA, number of two qubit gates scale super linearly with number of qubits. Whereas number of single qubit operations scale quadratically. QAOA is believed to be robust against certain class of two and single qubit errors.

Reversible circuits: We use three reversible circuits (Fredkin gate, two bit adder, and 2:4 decoder) to understand how correlated errors would affect the reliability of short width circuits. For instance, all reversible circuits use three to four qubits, but it contains more than 10 two-qubit gates. For these circuits, T1 decoherence might be the dominant error mechanism and such workloads can provide insights into how decoherence can cause correlated errors.

Table 1 shows the characteristics of the benchmarks used in our study. The terms "SG", "CX" and "M" respectively denote the number of single-qubit, two-qubit, and measurement operations in the workload. Workload evaluation on existing quantum computers is severely limited due to high error rates, which limits the length of the programs that can be run reliably on the current machines. Therefore, similar to prior studies [28, 29, 31, 40] we perform our experiments on small benchmarks.

Table 1: Benchmark Characteristics

Benchmark Name	Benchmark Description	Output	Number of Gates
Greycode	Greycode decoder	output: 001000	SG: 13, CX: 5, M: 6
bv-6	Bernstein-Vazirani	key: 110011	SG: 13, CX: 7, M: 5
bv-7	Bernstein-Vazirani	key: 1101011	SG: 13, CX: 11, M: 6
qaoa-5	max-cut 5 node graph	cut: 10101	SG: 24, CX: 8, M: 5
qaoa-6	max-cut 6 node graph	cut: 101010	SG: 30, CX: 10, M: 6
qaoa-7	max-cut 8 node graph	cut: 10101010	SG: 36, CX: 12, M:7
Fredkin	Fredkin gate	output:110	SG: 26, CX: 13, M:3
adder	1bit adder	output:011	SG: 12, CX: 15, M:3
Decode-24	2:4 Decoder	output: 100000	SG:119, CX:71, M:6

4.2 System Configuration

For all our evaluations, we use publicly available IBM quantum computer with fourteen qubits *ibmq-16-melbourne* [6]. For clarity we refer *ibmq-16-melbourne* as IBMQ-14. Moreover, for all the evaluations, we use a variation-aware mapping policy [40] as the baseline. As the error-characteristics of the NISQ machine can change dramatically between two calibrations, to guarantee statistical significance, we always execute baseline and the proposed policy for 16 thousand trials within a short succession of each other in each round. We repeat 10 such rounds and report the improvement for the median round.

4.3 Figure-of-Merit for Reliability

The goal of running the workload on a NISQ machine is to be able to infer the correct answer. This can be achieved by either increasing the probability of the correct answer or by suppressing the strongest sets of wrong answer or both. We need reliability metrics that account for both effects and has an intuitive implication on what it would mean to the ability to infer the correct answer on the NISQ machine.

The metric commonly used to indicate the reliability of a NISQ machine is the *Probability of Successful Trial (PST)*. PST is calculated by computing the ratio of a number of error-free trials to the total number of trials. PST is a good metric to compare two design points, for example comparing an ion-trap machine with a superconducting machine [23]. Moreover, recent papers on noise adaptive and variation-aware qubit mapping polices also use similar metrics to capture the reliability of applications [28, 31, 40].

$$PST = \frac{\text{Number of Error Free Trials}}{\text{Total Number of Trials}}$$

Unfortunately, PST does not always indicate the ability to infer the output of a NISQ machine correctly. For example, with PST=0.2 we can have reliable inference if all incorrect answers occur with less than 0.2 probability. However, another system with PST=0.2 will be unable to infer the correct output if one of the wrong answers is more dominant, say, for example, it occurs with 30% probability. To account for the magnitude of both the correct and the incorrect answers, we define a metric, *Inference Strength (IST)*. IST is a ratio of the frequency of correct output to the frequency of the most commonly occurring erroneous output.

$$IST = \frac{\text{Pr (Error free output)}}{\text{Pr (Erroneous output with highest frequency)}}$$

If IST exceeds 1, the system will be able to correctly infer the output, whereas if IST is significantly lower than 1, then the wrong answer(s) would mask out the correct answer. As our objective is to improve the ability to infer the correct answer on a NISQ machine, we use IST as the primary figure of merit in our evaluations.

4.4 Need for Real System Evaluations

As we perform our evaluations on a real-system, we are limited in our evaluations to workloads that have a non-negligible probability of being successfully executed on current machines. Alternatively, some studies have also used simulation-based models of quantum machines for for estimating the PST for their proposed technique. Unfortunately, existing simulators are based on independent-and-identically distributed (IID) model of errors and do not take into account the correlation of errors, therefore they may be useful for tracking PST but are unable to give reliable estimates for IST. To understand the gap between simulation and real devices, we executed all the workloads on IBM simulator and IBMQ-14 (real device) and observed significant difference in the Inference Strength (IST) of the simulator and the real system. Therefore, we do all our evaluations on a real machine instead of using a simulator.

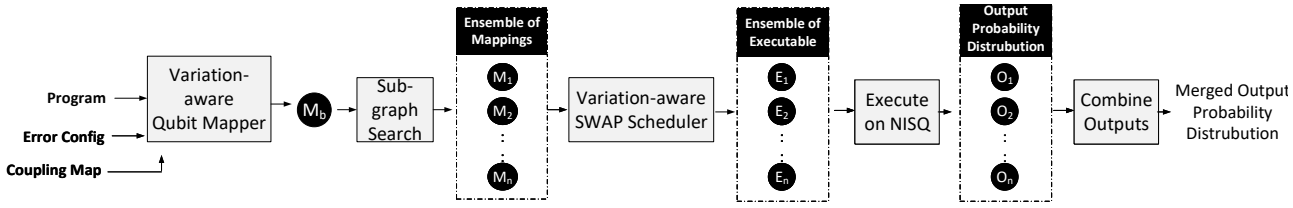

Figure 5: Overview of EDM: Design contains four steps (1) get top "K" mappings (2) generate K executable (3) Run the trials for each executable (4) merge the output probability distributions to create the combined output for the ensemble.

5 ENSEMBLE OF DIVERSE MAPPINGS

To mitigate the correlated errors on NISQ machines, we propose *Ensemble of Diverse Mapping (EDM)*. In this section, we will discuss design and reliability improvement provided by the EDM.

5.1 Motivation

Variation-aware qubit allocation improves the reliability of NISQ machines [28, 31, 40, 42]. However, running a NISQ application with just one mapping can increase its vulnerability to correlated errors. Running the program with single mapping multiple times produces incorrect outcomes with correlated errors. To mitigate the correlation, we need to introduce diversity in the program. One way to introduce variety in the program is by running the input program using a diverse set of qubit devices rather than being restricted to always using the same program assignment for all of the trials.

To test if the diverse mappings provide better reliability, we use BV-6 benchmark. Similar to the previous experiment, we use eight different logical to physical mappings (`A,B,C,D,E,F,G,H`) to run BV-6 on IBMQ-14 each for 16,384 trials. Figure 6 shows the *IST* for BV-6 with different mappings. IST captures the relative strength of the correct answer compared to the incorrect answer. When we use different mappings, we can expect variation in the reliability of individual qubit assignments. For example, Mapping C produces the output probability distribution with highest IST as compared to the other mappings. However, no single mapping has the IST exceeding 1. IST greater than one means the correct answer occurs with the highest frequency. To test if an ensemble of mapping can improve the IST, We execute the BV-6 for 4096 trials with mappings A, B, C, and D and merge the output probability distributions to generate EDM. We use 4096 trials each to match the number of trials in the baseline that runs with the single best solution.

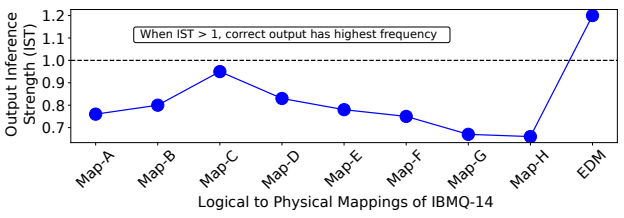

Figure 6: IST for BV-6 executed with the eight different mappings (A-H) on IBMQ-14 and the Ensemble of Mappings (EDM: A+B+C+D). Note that, none of the individual mappings have an IST ≥ 1, but the EDM has IST of 1.2.

Figure 6 shows the IST of 1.2 for BV-6 when executed with an ensemble of qubit assignments. The Ensemble of mapping improves the IST as incorrect answers get average out when we merge output probability distributions that are not similar. Use of Ensembles is one of the proven machine learning techniques that can improve the accuracy and robustness of classification tasks [9].

EDM is inspired by the principle of maximum entropy that suggests the probability distribution, which best represents the current state of knowledge is the one with largest entropy [17]. By using diverse mappings, EDM tries to avoid the repeated incorrect answers such that the incorrect results are spread across multiple outcomes.

5.2 EDM: Overview and Design

Our proposed *Ensemble of Diverse Mapping (EDM)* enables diversity in the output distributions by using an ensemble of qubit mappings. Figure 5 provides an overview of EDM. EDM contains four steps. In the first step, a compiler generates the best initial mapping and SWAP schedule for a given input program using coupling map (network topology) of a quantum computer and the error rate characterization data. In the second step, we use the initial mapping, and find all the isomorphic sub-graphs for the given quantum computer, and rank the sub-graphs as per the Estimated Success Probability (ESP). EDM picks the top "k" sub-graphs based on the ESP. In the third step, we re-compile the program by using the ensemble of initial mappings ($M_1, M_2, ..., M_n$) to produce an ensemble of executable ($E_1, E_2, ..., E_n$), and run all executable on a NISQ machine as shown in the Figure 5, to produce set of output probability distributions ($O_1, O_2, ...O_n$). Finally, we merge the probability distributions of all the members in the Ensemble to generate the final result.

For the first step, EDM can use any variation-aware quantum compiler. In this paper, we use variation-aware qubit mapper that uses A* search with reliability-aware heuristics proposed by [40, 48]. Furthermore, we use ESP as a cost function to select the strongest mapping on IBMQ-14 [31]. ESP incorporates measurement and single qubit gate errors. We also use benchmark specific heuristics to ensure optimal mapping. For example, a path graph satisfies the CNOT constraints for QAOA such that no SWAPs are required to perform QAOA. We verify the cost of all the mappings by using a brute force search to check the optimally of the mapping. For BV and QAOA, our compiler produces an optimal mapping.

To generate the Ensemble of initial qubit assignments, we need to ensure that the selected mapping has high reliability. When assigning program qubits to physical qubits, two major factors impact the output reliability: measurement errors and two-qubit gate errors. To leverage the mapping produced by the variation-aware mapper, we use graph isomerism to transfer the mapping

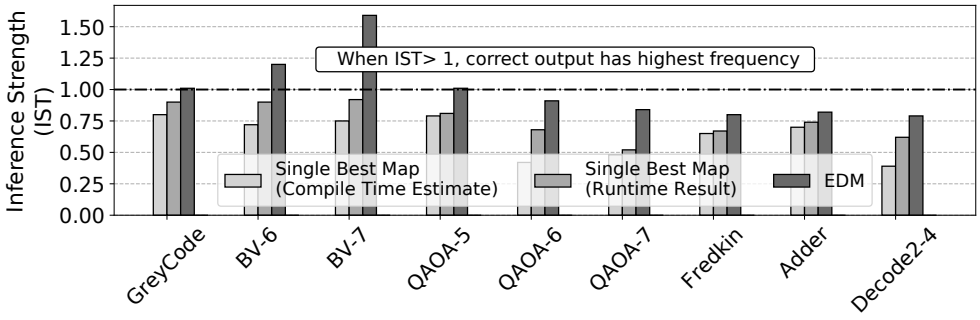

Figure 7: Improvement in IST with EDM, compared to single-best mapping. EDM has significantly higher IST compared to the best single mapping (both: that is estimated at compile time and the one that is observed at runtime)

from one set of qubits to another set of qubits. We search for all isomorphic sub-graphs, on the IBMQ-14 coupling graph using VF2 algorithm [5]. Once we have the list of all isomorphic graphs, we compute ESP and select the sub-graphs with highest ESP.

For the final step of producing the combined probability distribution, we use a simple average to merge the probability distributions of all of the members in the Ensemble.

5.3 Why Select the Top-K Mappings?

Variation aware allocation policies use compile-time information to estimate reliability (ESP). However, maximizing the ESP at compile time may not always result in maximizing the PST at runtime, as the behavior of the devices can change unpredictably at runtime. similar. Figure 8 shows ESP and the corresponding PST after evaluation for eight maps used for BV-6. There is a good correlation between ESP and PST. However, this correlation is not perfect. For example, Map-A is estimated to be the best mapping at compile-time; yet, at runtime, Map-C has the highest PST. Moreover, picking mapping with highest ESP cannot guarantee the highest IST. As error calibration data used to estimate ESP is not perfect due to temporal variations in qubit reliability and error-rates can change substantially due to cross-talk. Nonetheless, there is a good correlation between mappings that are good at compile time with the mappings that produce the highest PST at the runtime. Hence, we use the top K mappings to generate our Ensemble for EDM.

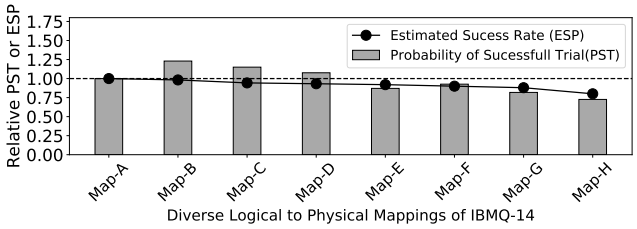

Figure 8: Comparing estimated reliability (ESP) at compile-time and observed reliability (PST) at run-time for BV-6 with eight different qubit mappings.

Our evaluations also show a weak correlation between PST and IST. For example, a slight improvement in PST for a given mapping does not result in increase in IST as the probability of the wrong answer can increase as well. Our analysis encountered several cases where a mapping with the highest ESP had lower IST compared to other mappings. We could form an ensemble of mappings that is estimated to produce the highest IST, however, to keep the design simple, we select the top K mappings that are deemed to have the highest PST for forming EDM. [2]

5.4 Impact of EDM on Inference Strength

EDM is designed to mitigate the correlation in errors and improve the IST such that the frequency of individual incorrect answer reduces by spreading the mistakes. Figure 7 shows the improvement in IST for QAOA and BV. We compare EDM against two different mappings: *single best mapping at compile time* that is estimated using ESP and single best mapping post-execution which is evaluated after running an ensemble of mappings. For example, as shown in the Figure 8, we estimated Map-A as the most reliable mapping based on its ESP. However, after running BV-6 with other mappings, we may realize that Map-C has the highest PST. To understand if the benefits of using Ensemble are due to diversity in the mappings or because of uncertainty in ESP, we also compare EDM with another baseline, *single best mapping post execution*, which represents the best mapping encountered at runtime. For example, this would be Map-C, as shown in the Figure 8.

The ensemble of mappings not only outperforms the best-estimated mapping at the compile time but also beats the best-single mapping encountered at runtime. This suggests that uncertainty in ESP is not a key reason behind the success of EDM. As for QAOA-5, the estimated best mapping at compile time is identical to the mapping at runtime, and even then EDM outperforms the baselines. EDM increases the entropy of output distribution such that, for the resulting output probability distributions, errors are spread across multiple possible incorrect answers.

[2] In extreme cases, the noise profile of the machine can change quickly, and cause the output distribution to be close to uniform. We can identify such cases by computing the relative standard deviation (σ/μ) of the probability distribution, comparing it with that of the uniform distribution, and discarding the results if the distance is quite small. We found such a strategy to be quite useful under such cases of extreme noise.

5.5 Impact of Ensemble Size

There is an inherent trade-off in ensembles selection. By increasing the size of Ensemble, we can introduce more diversity, but at the same time, we expose the program to relatively unreliable qubits. Finding the right size of an ensemble is especially crucial for the IBM machine, as it shows high variability in error rates. Our default implementation of EDM uses four mappings in the Ensemble. The number of ensembles is dictated by our ability to find the initial mapping that has similar SWAP cost and the ESP. EDM finds the graphs that are isomorphic to the initial mapping produced by the baseline. For IBMQ-14 due to limited connectivity, and high variability[3] in error rate, we observe that number of strong ensembles are limited two to four.

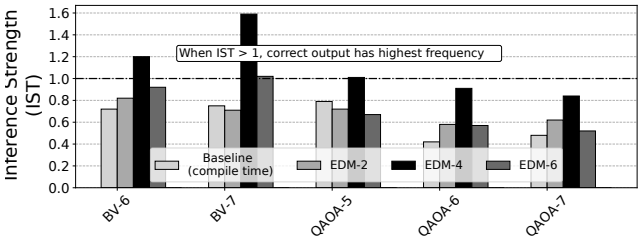

Figure 9: Sensitivity of EDM to the number of members in the Ensemble. With increasing ensemble, computation gets mapped to weaker qubits. Hence the benefit of EDM with larger ensemble size starts to reduce.

We evaluate the sensitivity of EDM to the number of members in the Ensemble. We form Ensemble with two mappings (EDM-2), four mappings (EDM-4, default), and six mappings (EDM-6) and run the workloads with the differently sized ensembles. Figure 9 shows the IST of the EDM with varying ensemble sizes. We observe that with only two members in the Ensemble, we do not add enough diversity, and in fact, the other copy can reduce the overall PST slightly for some cases and reduce the IST compared to even the baseline (BV-7 and QAOA-5). When the Ensemble contains four members, there is a good balance between the increase in diversity and the loss of PST. Overall we see significant improvement in IST. When the Ensemble contains six members, the mapping is forced to choose qubits that may have significantly lower reliability than the best qubits, and the overall degradation of PST is significantly greater than the gain from combining the diverse outputs. Therefore, in our experiments, we use a default size of 4 members in the Ensemble to balance both the increase is diversity and the pitfall of being forced to use more unreliable qubits for computation.

Note that the best number of ensembles will depend on the machine and the correlation in errors on that machine. So, there is no single best number of members in EDM that will always work well across variety of machines. We would recommend that users of EDM perform sensitivity while deciding the ensemble size.

[3]IBM-Q14 machine has two significantly noisy qubits Q12 and Q11 with readout error-rates up to 30%, we avoid using these qubits, which puts more constraints on finding a right isomorphic subgraph

6 WEIGHTED EDM

One of the limitations of our proposed implementation of EDM is that the members of the ensemble are based on prioritizing the maximization of ESP rather than maximizing the diversity in forming the ensemble. Therefore, outputs of some of the mappings can have a similar output probability distribution if mappings in an ensemble have a common set of qubits. Unfortunately, on existing IBM machine, due to a large variation in error rates, and a small number of qubits finding two sub-graphs that use a completely different set of qubits but have comparable ESPs is challenging. The effectiveness of EDM stems from the diverse set of outputs, and even a few unique qubit mappings can produce diverse incorrect answers. For example, in the case of BV-6, all the eight mappings had two to three common qubits. However, the diversity of the output was significant as illustrated by the Figure 4(b). Moreover, for all eight mappings, the common qubits are the strongest qubits that are less likely to produce errors. It might be possible to have enough diversity with few common qubits between two mappings in an ensemble.

6.1 Design of Weighted EDM

To maximize the diversity without deteriorating the reliability, we propose *Weighted Ensemble of Diverse Mappings (WEDM)*. Weighted EDM uses runtime information to maximize the diversity in the output probability distributions. In essence, it is risky to improve diversity at compile time by picking mapping with lower ESP. Whereas, we can solve this problem more efficiently at runtime. For instance, we can evaluate the diversity in the probability distributions and then perform scaling operation to increase the diversity. In contrast to EDM where we merge output probability distributions with identical weights, WEDM uses weighted average such that the weight is proportional to the cumulative mutual entropy of the output as shown in the Figure 10.

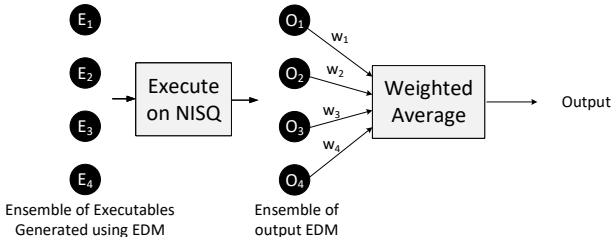

Figure 10: Design of Weighted EDM (WEDM)

The cumulative entropy of the output represents the uniqueness of the output probability distribution. For example, if we have four member ensemble with A, B, C, and D, we would calculate the KL-divergence of A with B, C, and D respectively and average these three values. The output of A will receive this weight before getting merged with the aggregated output distribution. A similar process will be repeated for B, C, and D. In the Appendix-B, we describe how to calculate weights in WEDM.

6.2 Impact of WEDM on Inference Strength

Figure 11 shows the improvements in IST with EDM and WEDM. WEDM improves the IST by up to 2.3x over the estimated single best mapping such that the correct answer has 1.73x higher likelihood compared to the incorrect answer. Both WEDM and EDM not only outperformed the estimated best mapping at compile time, but also showed improvements even over the single best mapping that we would have picked if we knew the behavior at runtime. With WEDM, all the workloads enter a regime where the correct answer has the highest frequency of occurrence. Thus achieving our goal of having higher confidence in the inference for NISQ.

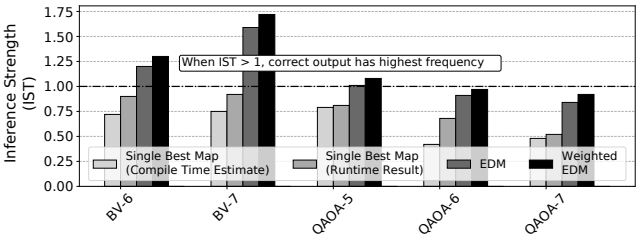

Figure 11: IST improvement with EDM and Weighted EDM (WEDM) over the baseline which uses the single best allocation for all of the trials. EDM and WEDM provide significant improvement in system reliability.

Using ensembles, we improve the IST but we can degrade the PST slightly as we use mappings that are not the most optimal when running EDM and WEDM. In both EDM and WEDM, at the end of the execution, we combine output probability distributions such that each entry in the distribution is averaged. The PST of an ensemble is bounded by the best and worst mapping in an ensemble. As we scale the workload, small improvements in PST or degradation does not change the effectiveness of NISQ applications. Whereas, improving the IST can correlate with the ability of the NISQ machine to infer the correct answer.

7 RELATED WORK

Early papers on qubit allocation focused on compilation techniques that eliminate redundant gates and minimize the number of SWAPs on solid state quantum computers [1, 14, 22, 25, 32, 33, 36, 38, 45, 48]. In recent papers, however, the focus has shifted to machine specific challenges such as variations in gate and coherence error rate [13, 28, 31, 40]. The philosophy behind the variation-aware and noise adaptive qubit allocation is that the distribution of errors across qubits is unequal with some qubits being more susceptible to errors than others, so application reliability can be improved by performing computation on the strongest set of qubits and links.

To tolerate noise, researchers are developing and benchmarking algorithms that are inherently resilient to noise, and require less number of resources [7, 12, 47]. Theorists have proposed application specific techniques [11, 16, 18, 19, 43, 44] for error mitigation. Another promising area to mitigate errors is by the use of low cost detection codes [15]. Prior works study the characteristics of the IBM machines to understand the fault-mechanisms [37].

In our concurrent work [41], we highlight the problem of data dependent bias in measurement errors. Our evaluations on IBM hardware show directional bias in measurement errors such that when measuring a qubit that is in state "1" we are more likely to encounter an error as opposed to measuring a qubit in state "0". Unfortunately, the bias can significantly degrade both the PST and IST. In particular, IST is affected because the incorrect answers with lower Hamming weight can occur more frequently than correct answer. To mitigate the measurement bias, we propose *Invert-and-Measure*: that transforms the weak state ("111...1") to strong state ("000...0") by performing qubit inversion right before performing measurement operation. Similar to EDM, the trails are split into groups that perform measruement on a diverse set of basis states.

In near future, we can expect an experimental evidence of quantum advantage [46]. However, developing practical applications using quantum computers is still an open problem [30]. Access to NISQ hardware via commercial cloud services has invigorated the development of practical applications [3, 10, 26]. This gives us a perfect opportunity to build compiler solutions, programming languages, and systems tools to enhance reliability, performance and usability of quantum computers [4, 24].

8 CONCLUSION

The arrival of quantum computers with dozens of qubits will enable a better understanding of the impact of qubit errors on applications. This can help us in developing efficient solutions to mitigate errors. In NISQ computing model, the program is run thousands of times, and the output log is used to infer the outcome. The ability to infer the correct outcome depends on both the probability of the correct outcome and the probability of the most-frequently occurring incorrect outcome. In this paper, we focus on the latter to improve the ability to infer the correct answer on NISQ machines.

Existing qubit allocation schemes search for one best mapping, and this mapping is used for all the trials. Unfortunately, such a method is vulnerable to correlated errors. The correlation in errors causes a few wrong answers to repeat for a large number of trials. To mitigate correlated errors, we leverage the principle of diversity, and propose an *Ensemble of Diverse Mappings (EDM)*. With EDM, the total number of trials are divided into multiple groups and a different mapping is applied to each group. To keep the implementation of EDM simple, we use the top-4 mappings produced by the underlying mapping policy. We show that with EDM, the magnitude of the dominant wrong answer decreases and the reliability of the NISQ system increases by up to 1.6x.

EDM merges the probability distributions generated by each of the mappings using an equal weight. We make an observation that the runs that have similar output have less information than the runs that have different outputs. Based on this insight, we propose *Weighted Ensemble of Diverse Mappings (WEDM)* that scales the output distributions generated by each of the mappings with a diversity score. WEDM improves reliability by up to 2.3x.

The key idea in our paper is to have multiple versions of the same quantum program, each tailored for a diverse set of mistakes. In this paper, we specifically use mapping policies to create such diverse programs. However, there are other sources of program transformations that can provide diversity as well. Exploring such diversification of quantum programs using alternative program transformations is a part of our future work.

ACKNOWLEDGMENTS

We thank Amruta Vidwans, Gururaj Shaileshwar, and Sanjay Kariyappa for feedback. This work was supported by a gift from Microsoft.

APPENDIX-A: ANALYZING CORRELATION IN ERRORS VIA BUCKETS-AND-BALLS ANALYSIS

To understand the impact of correlated errors on the inference quality of a NISQ machine, we use buckets and balls analysis.

A.1 Execution on NISQ as Buckets-and-Balls

The output of NISQ programs can be analyzed as buckets and ball problem. On NISQ machines, running a program that outputs m-bit string for N trials is equivalent to throwing N balls at the M buckets where $M = 2^m$. In this experiment, we have two types of buckets: green bucket that represents the correct answer and red buckets that represent all possible incorrect answers. We don't know the green bucket, but we can guess it by throwing a large number of balls and tracking the bucket with the most number of balls.

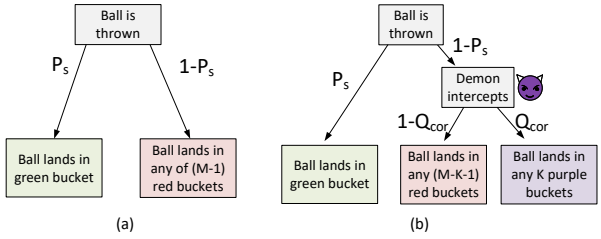

Figure 12: Buckets and Ball Model for NISQ (a) uncorrelated errors (b) correlated errors

A.2 Analytical Model for Uncorrelated Errors

For N balls and M buckets, if P_s is the probability of the ball landing in a green bucket then $(1 - P_s)$ is the probability of the ball landing in the any of the $M - 1$ red buckets as shown in Figure 12(a). With no correlation, the likelihood of ball landing in any of the $M - 1$ red buckets would be identical. For large N, number of balls in the green bucket (correct answer) would approach expected value of a Bernoulli trial: NP_s whereas number of balls in red bucket that has highest occupancy would be at the most $NP_e + 2 * \sqrt{(N * P_e * (1 - P_e))}$ (with 95% confidence), where $P_e = \frac{1-P_s}{M-1}$.

We use an analytical model (confirmed with Monte Carlo simulator) to understand how IST changes with P_s and M. For instance, Figure 13 describes the relationship between IST and P_s for M=64 buckets. The uncorrelated error model suggests that even with P_s=2%, we can distinguish the green bucket from rest as IST>1. Unfortunately, on real quantum computers, this model does not hold. Figure 13 show experimental P_s and IST data (blue dots) for three 6-bit applications (QAOA-6, BV-7, Grey-code) for 120 experiments executed on IBM-Q14 quantum computer. The experimental data show significantly smaller IST compared to the uncorrelated model for an identical P_s. To understand the mismatch, let's change our model and account for correlated errors.

A.3 Analytical Model for Correlated Errors

Correlated errors break the assumption that all incorrect answers are equally likely. To account for correlated errors, let's introduce a Demon in our model. This Demon biases errors such that k outcomes out of $2^m - 1$ incorrect outputs are more likely than the rest of the $2^m - k - 1$ outputs. These k more likely incorrect answers can be represented as purple buckets.

As shown in Figure 12(b) correlation-factor (Q_{cor}) determines what fraction of balls land in the k purple buckets after demon intercepts. The probability of balls hitting in the purple buckets is $(1-P_s)*(Q_{cor})$ and probability of ball hitting in any of the k purple bucket is $\frac{(1-P_s)*(Q_{cor})}{k}$. Figure 13 shows the result of the Monte Carlo simulation displaying the relationship between IST and P_s for $M = 64$, and $k = log(M) = 6$ and range of Q_{cor}. For simplicity, we assume that k scales with $O(log(M))$ as the correlation among errors tend to be local.

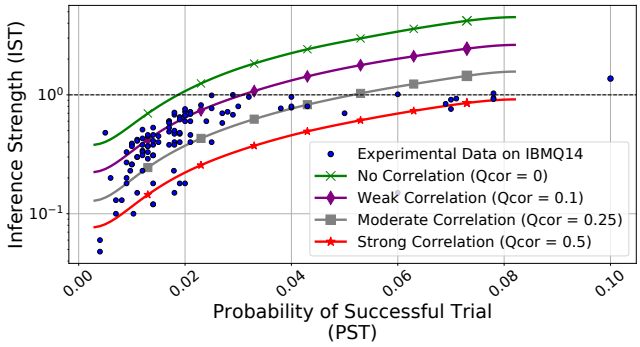

Figure 13: Inference Strength (IST) vs Probability of Successful Trial (PST) for Buckets and ball model and experimental data for 120 runs (each run was evaluated with 8192 trials) of QAOA-6, BV-6, and greycode-decoder on IBMQ-14 machine.

To understand the impact of correlated errors on reliability, we compute PST frontier Using Monte Carlo simulations. PST Frontier is the minimum PST required to infer the correct answer from given output distribution (PST at which IST=1). For the model with no correlation, PST frontier is at 1.8%, that means for 6-bit application with PST>1.8%, we can always deduce the correct answer. The PST Frontier shifts right to 3.6% with correlated errors that have weak correlation (Qcor=10%). Moreover, it shifts even further at 8% for strong correlation model (Qcor=50%).

Unfortunately, there is no simple way of deducing the correlation-factor on the real machine as it depends on the device characteristics and the type of application that we are running. High PST frontier degrades the effectiveness of NISQ applications like QAOA. For example, our experiments show that QAOA-6 with baseline policy consistently fails to meet PST Frontier criteria as it has a median PST of 2.5% and IST of 0.78 on IBMQ-14 for 30 experimental runs.

With EDM, we reduce the correlation in the incorrect answers. Therefore, the likelihood of the same wrong answer occurring with a high frequency gets reduced (by a factor based on the size of the ensemble) which can allow the machine to infer the right answer even at a lower PST than the baseline.

APPENDIX-B: A PRIMER ON KL-DIVERGENCE

NISQ machines can produce a probability distribution over all the possible outputs. For our study, we are interested in measuring the similarity (or dissimilarity) of two probability distributions. The Kullback-Leibler divergence (or KL-divergence) is a measure of how one probability distribution is different from another probability distribution. We use the KL-divergence to analyze the diversity in output distributions generated by different mappings. We also used symmetric KL-divergence to estimate the weights for merging the outputs of different mappings in the WEDM design. In this Appendix, we will discuss a few illustrative examples. For example, if we have two discrete probability distributions P and Q defined over a state of N values, the KL divergence between P and Q, denoted as $D_{KL}(P||Q)$, is shown by Equation 1.

$$D_{KL}(P||Q) = \sum_{i=1}^{N} P_i \log \frac{P_i}{Q_i} \tag{1}$$

For example, consider the two distributions P and Q over four values (0-3), as shown in Table 2.

Table 2: Example Probability Distributions

Distribution	0	1	2	3
P(x)	0.2	0.3	0.4	0.1
Q(x)	0.25	0.25	0.25	0.25

Then, $D_{KL}(P||Q)$ and $D_{KL}(Q||P)$ can be calculated as follows:

$$\begin{aligned} D_{KL}(P||Q) = 0.2 \cdot ln(\frac{0.2}{0.25}) + 0.3 \cdot ln(\frac{0.3}{0.25}) \\ +0.4 \cdot ln(\frac{0.4}{0.25}) + 0.1 \cdot ln(\frac{0.1}{0.25}) = 0.046 \end{aligned} \tag{2}$$

$$\begin{aligned} D_{KL}(Q||P) = 0.25 \cdot ln(\frac{0.25}{0.2}) + 0.25 \cdot ln(\frac{0.25}{0.3}) \\ +0.25 \cdot ln(\frac{0.25}{0.4}) + 0.25 \cdot ln(\frac{0.25}{0.1}) = 0.052 \end{aligned} \tag{3}$$

Thus, KL divergence may not be symmetric and can not qualify as a distance metric. However, it can be symmetrised to enable symmetric KL divergence (SD_{KL}) such that $SD_{KL}(P,Q) = SD_{KL}(Q,P)$.

$$SD_{KL}(P,Q) = D_{KL}(Q||P) + D_{KL}(P||Q) \tag{4}$$

For weighted EDM (WEDM), we use symmetric KL divergence (SD_{KL}) to compute resultant output probability distribution (O_{WEDM}) that is a weighted sum of ensemble output probability distributions (O_i). For N ensembles, the output probability distribution (O_{WEDM}) and normalized ensemble weights ($\overline{W}$) are evaluated as follows:

$$O_{WEDM} = \sum_{i=0}^{i=N} \overline{W_i} * O_i \tag{5}$$

$$W_i = \sum_{j=0}^{j=N} SD_{KL}(O_i, O_j) \quad \& \quad \overline{W_i} = \frac{W_i}{\sum_{i=0}^{i=N} W_i} \tag{6}$$

REFERENCES

[1] Kyle EC Booth, Minh Do, J Christopher Beck, Eleanor Rieffel, Davide Venturelli, and Jeremy Frank. 2018. Comparing and Integrating Constraint Programming and Temporal Planning for Quantum Circuit Compilation. *arXiv preprint arXiv:1803.06775* (2018).

[2] Davide Castelvecchi. 2017. IBM's quantum cloud computer goes commercial. *Nature News* 543, 7644 (2017), 159.

[3] Davide Castelvecchi. 2017. IBM's quantum cloud computer goes commercial. *Nature News* 543, 7644 (2017), 159.

[4] Frederic T Chong, Diana Franklin, and Margaret Martonosi. 2017. Programming languages and compiler design for realistic quantum hardware. *Nature* 549, 7671 (2017), 180.

[5] Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. 2004. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence* 26, 10 (2004), 1367–1372.

[6] International Business Machines Corporation. 2017. Universal Quantum Computer Development at IBM:. http://research.ibm.com/ibm-q/research/. [Online; accessed 3-April-2017].

[7] Gavin E Crooks. 2018. Performance of the quantum approximate optimization algorithm on the maximum cut problem. *arXiv preprint arXiv:1811.08419* (2018).

[8] Simon J. Devitt, Kae Nemoto, and William J. Munro. 2009. Quantum Error Correction for Beginners. Rep. Prog. Phys. 76 (2013) 076001. (2009). https://doi.org/10.1088/0034-4885/76/7/076001 arXiv:arXiv:0905.2794

[9] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*. Springer, 1–15.

[10] Eugene F Dumitrescu, Alex J McCaskey, Gaute Hagen, Gustav R Jansen, Titus D Morris, T Papenbrock, Raphael C Pooser, David Jarvis Dean, and Pavel Lougovski. 2018. Cloud quantum computing of an atomic nucleus. *Physical review letters* 120, 21 (2018), 210501.

[11] Suguru Endo, Simon C Benjamin, and Ying Li. 2018. Practical quantum error mitigation for near-future applications. *Physical Review X* 8, 3 (2018), 031027.

[12] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Hartmut Neven. [n. d.]. Quantum Algorithms for Fixed Qubit Architectures. 2017. *arXiv preprint arXiv:1703.06199* ([n. d.]).

[13] Will Finigan, Michael Cubeddu, Thomas Lively, Johannes Flick, and Prineha Narang. 2018. Qubit Allocation for Noisy Intermediate-Scale Quantum Computers. *arXiv preprint arXiv:1810.08291* (2018).

[14] Gian Giacomo Guerreschi and Jongsoo Park. 2018. Two-step approach to scheduling quantum circuits. *Quantum Science and Technology* (2018).

[15] Robin Harper and Steven Flammia. 2018. Fault tolerance in the IBM Q Experience. *arXiv preprint arXiv:1806.02359* (2018).

[16] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. 2019. Supervised learning with quantum-enhanced feature spaces. *Nature* 567, 7747 (2019), 209.

[17] Edwin T Jaynes. 1967. Foundations of probability theory and statistical mechanics. In *Delaware seminar in the foundations of physics*. Springer, 77–101.

[18] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. 2017. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* 549, 7671 (2017), 242.

[19] Abhinav Kandala, Kristan Temme, Antonio D Córcoles, Antonio Mezzacapo, Jerry M Chow, and Jay M Gambetta. 2019. Error mitigation extends the computational reach of a noisy quantum processor. *Nature* 567, 7749 (2019), 491.

[20] Julian Kelly, Zijun Chen, Ben Chiaro, Brooks Foxen, and John Martinis. 2019. Operating and Characterizing of a 72 Superconducting Qubit Processor âĂIJBristlecone‚ÂÏ: Part 1. *Bulletin of the American Physical Society* (2019).

[21] Bjoern Lekitsch, Sebastian Weidt, Austin G Fowler, Klaus Mølmer, Simon J Devitt, Christof Wunderlich, and Winfried K Hensinger. 2017. Blueprint for a microwave trapped ion quantum computer. *Science Advances* 3, 2 (2017), e1601540.

[22] Gushu Li, Yufei Ding, and Yuan Xie. 2018. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. *arXiv preprint arXiv:1809.02573* (2018).

[23] Norbert M Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A Landsman, Kenneth Wright, and Christopher Monroe. 2017. Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences* 114, 13 (2017), 3305–3310.

[24] Margaret Martonosi and Martin Roetteler. 2019. Next Steps in Quantum Computing: Computer Science's Role. *arXiv preprint arXiv:1903.10541* (2019).

[25] Dmitri Maslov, Sean M Falconer, and Michele Mosca. 2007. Quantum circuit placement: optimizing qubit-to-qubit interactions through mapping quantum circuits into a physical experiment. In *Proceedings of the 44th annual Design Automation Conference*. ACM, 962–965.

[26] Masoud Mohseni, Peter Read, Hartmut Neven, Sergio Boixo, Vasil Denchev, Ryan Babbush, Austin Fowler, Vadim Smelyanskiy, and John Martinis. 2017. Commercialize quantum technologies in five years. *Nature News* 543, 7644 (2017), 171.

[27] Andrea Morello and David Reilly. 2018. What would you do with 1000 qubits? *Quantum Science and Technology* 3, 3 (2018), 030201.

[28] Prakash Murali, Jonathan M Baker, Ali Javadi Abhari, Frederic T Chong, and Margaret Martonosi. 2019. Noise-Adaptive Compiler Mappings for Noisy Intermediate-Scale Quantum Computers. *arXiv preprint arXiv:1901.11054* (2019).

[29] Prakash Murali, Norbert Matthias Linke, Margaret Martonosi, Ali Javadi Abhari, Nhung Hong Nguyen, and Cinthia Huerta Alderete. 2019. Full-stack, real-system quantum computer studies: architectural comparisons and design insights. In *Proceedings of the 46th International Symposium on Computer Architecture*. ACM, 527–540.

[30] Engineering National Academies of Sciences and Medicine. 2019. *Quantum Computing: Progress and Prospects*. The National Academies Press, Washington, DC. https://doi.org/10.17226/25196

[31] Shin Nishio, Yulu Pan, Takahiko Satoh, Hideharu Amano, and Rodney Van Meter. 2019. Extracting Success from IBM's 20-Qubit Machines Using Error-Aware Compilation. *arXiv preprint arXiv:1903.10963* (2019).

[32] Alexandru Paler. 2019. On the Influence of Initial Qubit Placement During NISQ Circuit Compilation. In *International Workshop on Quantum Technology and Optimization Problems*. Springer, 207–217.

[33] Alexandru Paler, Alwin Zulehner, and Robert Wille. 2018. NISQ circuit compilers: search space structure and heuristics. *arXiv preprint arXiv:1806.07241* (2018).

[34] Alex Parent, Martin Roetteler, Krysta M. Svore, and Krysta M. Svore. 2017. REVS: A tool for space-optimized reversible synthesis. In *Proceedings of the 9th International Conference on Reversible Computation (RC 2017)*. 90–101.

[35] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *arXiv preprint arXiv:1801.00862* (2018).

[36] Alireza Shafaei, Mehdi Saeedi, and Massoud Pedram. 2013. Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures. In *Proceedings of the 50th Annual Design Automation Conference*. ACM, 41.

[37] Abhishek Shukla, Mitali Sisodia, and Anirban Pathak. 2018. Complete characterization of the single-qubit quantum gates used in the IBM quantum processors. *arXiv preprint arXiv:1805.07185* (2018).

[38] Marcos Siraichi, Vinicius Fernandes Dos Santos, Sylvain Collange, and Fernando Magno Quintão Pereira. 2018. Qubit Allocation. In *CGO 2018-IEEE/ACM International Symposium on Code Generation and Optimization*. 1–12.

[39] Mingyu Sun and Michael R. Geller. 2019. Efficient characterization of correlated SPAM errors. arXiv:arXiv:1810.10523

[40] Swamit S Tannu and Moinuddin K Qureshi. 2018. A Case for Variability-Aware Policies for NISQ-Era Quantum Computers. *arXiv preprint arXiv:1805.10224* (2018).

[41] Swamit S Tannu and Moinuddin K Qureshi. 2019. Mitigating Measurement Errors in Quantum Computers by Exploiting State-Dependent Bias. *The 52nd Annual IEEE/ACM International Symposium on Microarchitecture* (Oct 2019). https://doi.org/10.1145/3352460.3358265

[42] Swamit S Tannu and Moinuddin K Qureshi. 2019. Not all qubits are created equal: a case for variability-aware policies for NISQ-era quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 987–999.

[43] Kristan Temme, Sergey Bravyi, and Jay M Gambetta. 2017. Error mitigation for short-depth quantum circuits. *Physical review letters* 119, 18 (2017), 180509.

[44] Takahiro Tsunoda, Andrew Patterson, Xiao Yuan, Suguru Endo, Joseph Rahamim, Peter Spring, Martina Esposito, Salha Jebari, Kitti Ratter, Sophia Sosnina, et al. 2019. Implementing the Variational Quantum Eigensolver with native 2-qubit interaction and error mitigation. *Bulletin of the American Physical Society* (2019).

[45] Davide Venturelli, Minh Do, Eleanor Rieffel, and Jeremy Frank. 2018. Compiling quantum circuits to realistic hardware architectures using temporal planners. *Quantum Science and Technology* 3, 2 (2018), 025004.

[46] Benjamin Villalonga, Dmitry Lyakh, Sergio Boixo, Hartmut Neven, Travis S Humble, Rupak Biswas, Eleanor G Rieffel, Alan Ho, and Salvatore Mandrà. 2019. Establishing the Quantum Supremacy Frontier with a 281 Pflop/s Simulation. *arXiv preprint arXiv:1905.00444* (2019).

[47] Jonathan Ward, Johannes Otterbach, Gavin Crooks, Nicholas Rubin, and Marcus da Silva. 2018. QAOA Performance Benchmarks using Max-Cut. In *APS Meeting Abstracts*.

[48] Alwin Zulehner, Alexandru Paler, and Robert Wille. 2017. Efficient Mapping of Quantum Circuits to the IBM QX Architectures. *arXiv preprint arXiv:1712.04722* (2017).

[49] A. Zulehner, A. Paler, and R. Wille. 2018. Efficient mapping of quantum circuits to the IBM QX architectures. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. https://doi.org/10.23919/DATE.2018.8342181