Revisiting Noise-adaptive Transpilation in Quantum Computing: How Much Impact Does it Have?

Yuqian Huo
Rice University, USA

Jinbiao Wei
Rice University, USA

Christopher Kverne
Florida International University (FIU), USA

Mayur Akewar
Florida International University (FIU), USA

Janki Bhimani
Florida International University (FIU), USA

Tirthak Patel
Rice University, USA

arXiv:2507.01195v2 [quant-ph] 2 Oct 2025

***Abstract*—Transpilation, particularly noise-aware optimization, is widely regarded as essential for maximizing the performance of quantum circuits on superconducting quantum computers. The common wisdom is that each circuit should be transpiled using up-to-date noise calibration data to optimize fidelity. In this work, we revisit the necessity of frequent noise-adaptive transpilation, conducting an in-depth empirical study across five IBM 127-qubit quantum computers and 16 diverse quantum algorithms. Our findings reveal novel and interesting insights: (1) noise-aware transpilation leads to a heavy concentration of workloads on a small subset of qubits, which increases output error variability; (2) using random mapping can mitigate this effect while maintaining comparable average fidelity; and (3) circuits compiled once with calibration data can be reliably reused across multiple calibration cycles and time periods without significant loss in fidelity. These results suggest that the classical overhead associated with daily, per-circuit noise-aware transpilation may not be justified. We propose lightweight alternatives that reduce this overhead without sacrificing fidelity – offering a path to more efficient and scalable quantum workflows.**

***Index Terms*—Quantum Computing, Quantum Transpilation, Quantum Characterization**

I. Introduction

Quantum computing has the potential to revolutionize fields like cryptography, optimization, and machine learning by leveraging quantum phenomena such as superposition, entanglement, and interference [3], [22]. However, near-term quantum hardware, such as superconducting qubits, remains highly susceptible to noise, resulting in errors that degrade computational accuracy [9], [24]. To mitigate these issues, quantum computing frameworks like IBM's Qiskit provide *transpilation* [1], a process that optimizes circuits by mapping qubits and gates to specific hardware configurations, taking into account noise and connectivity constraints.

Over the last few years, noise-aware transpilation has become a widely adopted strategy to improve quantum circuit fidelity on noisy quantum hardware [8], [12], [16], [18], [20], [25], [26], [28], [33]. Transpilation is also predicted to remain an important aspect of quantum circuit compilation, even in the early fault-tolerant quantum computing (FTQC) era, because of the qubit heterogeneity of quantum systems, which will require careful transpilation to find computer sections with low enough error rates to operate in the error-correction regime [11], [15], [28]. In addition, recent studies have analyzed noise effects and transpilation efficacy across various quantum computing systems [2], [4], [13], [19], [23], noting the reliance on constantly updated calibration data. These studies underscore the perceived importance of frequent re-transpilation to counter temporal drift in hardware noise characteristics.

However, these works were primarily conducted three to four years ago, using algorithms requiring fewer qubits (< 10) and less advanced quantum devices. Larger circuits face more complex error propagation patterns. Modern 127-qubit devices present different connectivity and noise profiles. At the same time, the classical overhead of transpilation – especially under full noise-aware optimization – has increased considerably, often adding seconds per circuit, which scales poorly for workloads requiring thousands of circuit runs, such as variational algorithms [8]. Yet it remains unclear if such frequent and costly recompilation is still necessary on today's hardware.

Our study aims to investigate if the assumed importance of transpilation still holds for the largest superconducting quantum computers for a diverse set of complex algorithms[1]. We empirically evaluate the necessity and impact of frequent noise-aware transpilation using a month-long study of 16 quantum algorithms (size of up to 31 qubits) on six 127-qubit IBM quantum computers, investigating whether the classical overhead associated with continual re-transpilation is justified in the context of the modern hardware landscape. Conducting such a study is non-trivial: quantum computers offer limited daily access, calibration data changes frequently, and empirical fidelity evaluation requires repeated executions over time. Nevertheless, our month-long open-source effort is designed to probe whether the conventional wisdom behind frequent noise-aware transpilation continues to hold in practice.

Our results reveal some novel and insightful findings:

1) Noise-adaptive transpilation often concentrates usage on a small subset of "good" qubits, resulting in uneven resource utilization and increased output error variability due to dynamic qubit wear and fluctuations.

[1]This work is published in the Proceedings of the International Conference on Computer-Aided Design (ICCAD), 2025.

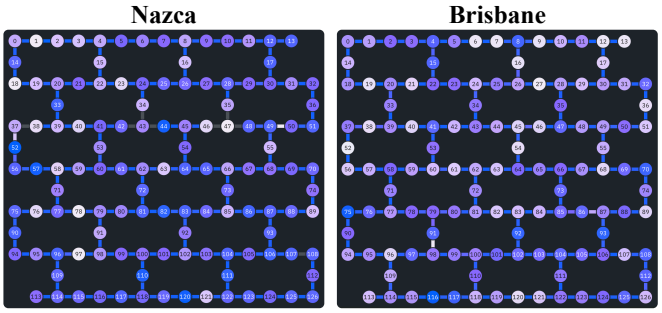

Fig. 1: Example 127-qubit IBM Nazca and Brisbane quantum computers. The circles represent the qubits (color-coded according to their respective T1 times), and the lines represent the qubit connections (color-coded as per ECR gate errors). Darker color indicates better values.

2) While circuit routing techniques improve the output fidelity, noise-adaptive mapping techniques are often not impactful – in fact, we find that simple random mapping achieves comparable average fidelity with significantly less output fluctuation, and also avoids qubit overuse.
3) Using lower transpiler optimization levels (e.g., levels 1 or 2) incurs far less compilation time and yields similar output fidelity compared to the most aggressive optimization level (level 3), especially for larger circuits.
4) We observe no statistically significant correlation between the time of execution and output fidelity, indicating that qubit drift over the course of a calibration cycle has minimal practical impact on algorithm results.
5) Circuits compiled once with calibration data from a single day maintain their fidelity when executed across multiple calibration cycles, challenging the conventional wisdom that frequent re-transpilation is necessary.

Based on these insights, we propose simple and lightweight methods to reduce transpilation overhead and mitigate output error, advancing more efficient approaches to executing quantum algorithms on superconducting devices. Our methods and data are open-sourced at:
https://github.com/positivetechnologylab/ICCAD25.

II. Background and Motivation

Quantum computing harnesses the principles of quantum mechanics to process information in ways fundamentally different from classical computers. Unlike classical bits, which can only be in one of two states (0 or 1), quantum bits, or qubits, can exist in a superposition of both 0 and 1 simultaneously. This property, combined with quantum entanglement and interference, enables quantum computers to perform certain calculations (gates and operations) far more efficiently than their classical counterparts.

A. Superconducting Qubits and Noise Effects

Superconducting qubits, the most advanced qubit realization currently, are fabricated using materials that exhibit zero electrical resistance when cooled to extremely low temperatures, enabling stable quantum states [9], [31]. These qubits are manipulated using microwave pulses to implement quantum gates. However, due to the inherent fragility of quantum states, superconducting qubits are highly susceptible to noise, which can introduce errors into quantum computations [4], [23]. Noise manifests in several forms, each characterized by distinct error metrics as described below.

- **Relaxation Time (T_1):** The T_1 time is the time it takes for an excited qubit to decay to its ground state.
- **Dephasing Time (T_2):** The T_2 time measures the duration over which a qubit maintains its phase coherence.
- **One- and Two-Qubit Gate Errors:** Gate operations are prone to control inaccuracies, which lead to gate errors. These errors accumulate as gate operations are applied within a circuit, impacting overall fidelity.
- **Readout Errors:** Quantum computations are finalized by measuring qubit states, but this measurement process can introduce errors. Readout errors occur when a qubit in state 0 or 1 is incorrectly measured as the opposite state.

The noise characteristics of superconducting qubits vary not only across different qubits (shown for IBM quantum computers in Fig. 1) but also over time due to changes in the physical environment and device qubit drift. IBM quantum computers, for example, are calibrated daily to minimize these noise effects. Consequently, the noise parameters (T_1, T_2, gate error, readout error, and CX error) are subject to temporal variations, which complicates the task of achieving reliable quantum computations on such hardware [13], [19].

B. Transpilation and Optimization Levels in Qiskit

Quantum circuits designed for an ideal, noise-free model must be adapted to the constraints of real quantum hardware through a process known as *transpilation*. Transpilation maps an abstract quantum circuit onto the physical qubits and gates available on a quantum processor, taking into account the connectivity and noise characteristics of the device. The transpiler in Qiskit [1], IBM's widely-used quantum computing framework, provides three levels of optimizer passes implemented from prior, state-of-the-art research [11], [29]:

- **Level 1: Minimal Optimization** *(Basic Routing and Error Reduction)* - At this level, the transpiler applies only basic routing transformations to ensure that all required qubit connections are physically feasible (e.g. use swap gates to move qubits adjacent to each other when needed). Level 1 prioritizes speed.
- **Level 2: Moderate Optimization** *(Balanced Error Mitigation)* - In addition to basic routing, Level 2 introduces more aggressive error-mitigation techniques. This level performs further gate cancellation and circuit rewriting, balancing the trade-off between transpilation time and circuit fidelity.

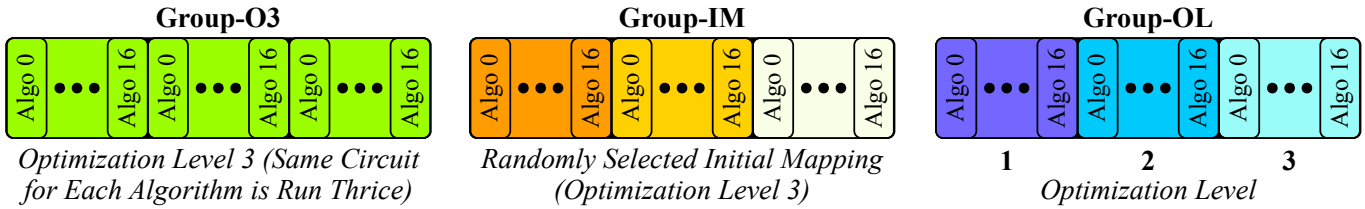

Fig. 2: We run three groups of experiments (all three groups are run daily), where each group is assembled into one job with $16 \times 3 = 48$ circuits. Here, 16 is the number of algorithms run, and we run each cluster of algorithms thrice within each group. All groups are submitted in parallel on each available computer daily. If a group's job does not finish within the calibration cycle for which it is optimized, the job is canceled or disregarded. Groups may end up running in any order. Each color in the visualization represents a specific transpiled circuit variation according to the mapping methodology or optimization level.

- **Level 3: Full Optimization** *(Advanced Noise-Aware Optimization)* - This level provides the most thorough optimization, considering noise characteristics and targeting specific low-error qubits for placing critical circuit paths whenever possible. Level 3 is the most time-intensive.

C. Motivation for our Study

While noise-aware transpilation is widely accepted as a best practice in today's quantum computing workflows, this assumption has not been thoroughly reevaluated in the context of modern, large-scale quantum hardware. Much of the foundational work in this space was conducted on earlier systems with tens of qubits, higher baseline noise, and simpler circuit structures. Today's 127-qubit superconducting machines are substantially different – featuring better average fidelities, more complex qubit layouts, and greater heterogeneity across devices. As quantum applications scale and the number of required circuit executions increases, especially in variational and hybrid algorithms, the classical overhead of frequent transpilation becomes non-trivial [8]. The cost of compiling thousands of circuit instances daily using Level 3 optimization – each taking seconds of time to transpile, which is equivalent to or more than the actual execution time on the quantum computer [23] – can quickly dominate the overall workflow.

These evolving realities motivate a careful re-examination of the assumptions underlying noise-adaptive transpilation. In this work, we study how transpilation strategies affect fidelity, performance variability, and classical overhead across a diverse set of algorithms executed on modern IBM quantum computers. Our goal is to understand not just how to optimize for short-term error reduction, but how to achieve performance stability and resource efficiency – particularly as quantum algorithms scale beyond prototype problems.

III. Experimental Methodology

Experimental Setup and Software: We performed our experiments using IBM's quantum computing cloud services. Our implementation used IBM's open-source quantum computing framework Qiskit [1] version 1.2.1, with Python 3.11.9 scripts for managing cloud job submissions and result retrieval. Bash shell scripts are used for automated job submissions, scheduled via crontab for timely execution. The experiments were conducted on six IBM quantum processors: *ibm_cusco*, *ibm_kyoto*, *ibm_nazca*, *ibmq_kyiv*, *ibmq_brisbane*, and *ibmq_sherbrooke*. Each quantum computer features an identical 127-qubit architecture and topography. All temporal data in our analysis references the UTC-5 (EST) time zone.

TABLE I: QASMBench [10] algorithms used in our study.

Algo.	Description	#Qubits	#Gates
VQE	Variational Quantum Eigensolver	6	2282
HHL	Harrow–Hassidim–Lloyd Algorithm	7	689
QPE1	Quantum Phase Estimation (Small)	9	123
ADD1	Quantum Adder (Small)	10	142
SAT	Satisfiability Circuit	11	679
SECA	Shor's Error Correction Algorithm	11	216
GCM	Generator Coordinate Method	13	3148
MULT	Quantum Multiplier	13	98
QPE2	Quantum Phase Estimation (Large)	15	311
DNN	Quantum Deep Neural Network	16	2016
QEC	Quantum Error Correction Encoding	17	53
ADD2	Quantum Adder (Large)	18	284
SQRT	Square Root Circuit	18	2300
QRAM	Quantum Random Access Memory	20	92
KNN	Quantum k-Nearest Neighbors (Small)	25	230
SWAP	SWAP Test Circuit	25	230

Experiments Conducted: We divide experiments into three groups, shown in Fig.2. The first group (Group-O3) reflects the conventional approach used in most quantum computing workflows: full noise-aware transpilation using optimization level 3 on the latest calibration data [2], [15], [28]. This baseline is widely adopted in practice and assumed to yield the highest fidelity. The second group (Group-IM) evaluates whether randomizing the initial mapping (while retaining all routing optimizations) can yield comparable fidelity with reduced variability by avoiding repeated use of the same "good" qubits. This tests the hypothesis that uniform qubit usage may improve performance stability. The third group (Group-OL) investigates whether lower optimization levels (1 or 2) can match the fidelity of level 3 while significantly reducing classical transpilation time. Together, these three groups allow us to isolate the impact of mapping strategy and transpilation complexity on fidelity and performance variability. Each group was structured to decouple distinct factors: standard transpilation, daily noise adaptation, and stochastic remapping.

This separation enables a controlled comparison of fidelity, variability, and runtime overhead across strategies.

Note: the three clusters in Group-O3 are all the same color in the figure as they run the exact same transpilation of the circuits thrice for statistical meaningfulness. The other groups can have randomization among the clusters due to random initial mapping and varying optimization levels. Hence, they have differently-color-coded clusters. We detail how each group supports specific analysis in Sec. IV.

Algorithms Evaluated: We evaluate 16 algorithms listed in Table I. These algorithms are taken from the QASMBench [10] benchmark suite. We selected these algorithms to represent a variety of diverse characteristics in terms of width and depth. Note: We cap the circuit width at 25 qubits because beyond that, the circuit produced uninterpretable results due to the noise on current quantum computers. Further, evaluating 6-25-qubit circuits on 127-qubit hardware allows us to investigate how transpilation strategies interact with hardware heterogeneity and qubit selection policies, especially in the context of large and sparsely connected topologies.

Quantum Computers Used: We ran our experiments on six 127-qubit IBM quantum computers: Brisbane, Cusco, Kyiv, Kyoto, Nazca, and Sherbrooke (limited use). All computers feature the Eagle r3 processor architecture. Experiments were run for all three groups on all computers, not necessarily at the same time, due to varying computer availabilities. At least 3-8 days of experiments were run on each computer. A total of 4958 circuits were run across all six computers over an experimentation period spanning approximately one month.

Statistical Metrics used for Analysis: To gauge the output fidelity of quantum runs, we must compare the ideal quantum simulation output and the noisy output when the circuits are run on real computers. Various metrics could be used to capture this output difference. Total Variation Distance (TVD), Hellinger Distance (HLD), Jensen-Shannon Distance (JSD), and Kullback-Leibler Divergence (KLD) are common measures [6], [21], [26]. These measures are all closely related; for instance, HLD bounds TVD and JSD bounds KLD [7], [17]. *Thus, we select KLD for our study as it is frequently chosen for its interpretability in terms of information loss in the noisy output, making it a comprehensive metric for comparing quantum distributions.* For two probability distributions P and D, their KLD is calculated as $D_{\text{KL}} = \sum_{i \in S} P(i) \log \frac{P(i)}{Q(i)}$, over output states $i \in S$. We also refer to KLD as "output error" and its inverse as "output fidelity" [30].

We also use the Spearman correlation to measure the monotonic relationship between two variables using their rank orders, making it robust to outliers and non-linearities. It ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no monotonic relationship. It is given by $\rho = 1 - \frac{6 \sum d_i^2}{m(m^2-1)}$, where d_i is the rank difference for each sample pair, and m is the number of samples [32].

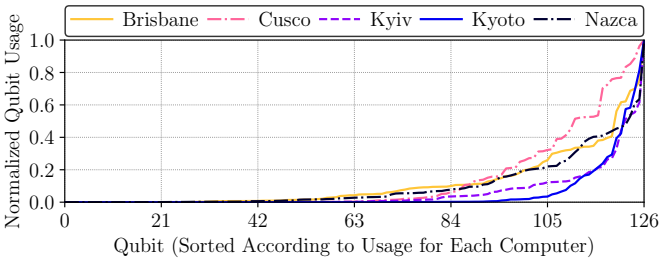

Fig. 3: Our analysis indicates that across all quantum computers, a small set of qubits gets used more frequently.

TABLE II: Spearman correlation between qubit usage frequency and different qubit properties.

Property	Correlation	p-value
Mean T1 Time	+0.08	6.73×10^{-6}
Mean T2 Time	+0.18	1.25×10^{-27}
Mean Qubit Frequency	-0.02	0.14
Mean One-Qubit Gate Error	-0.19	1.27×10^{-31}
Mean Two-Qubit Gate Error	-0.28	4.10×10^{-66}
Mean Measurement Error	-0.18	4.12×10^{-29}

IV. Observations and Analysis

We organize our analysis into two categories: spatial effects, which capture how frequently and where circuits are mapped, and temporal effects, which consider when in time they are executed. Together, these analyses probe the core assumptions behind noise-adaptive transpilation, such as the need to always use "good" qubits and the necessity of continual re-transpilation over time.

A. Spatial Observations and Analysis

The first research question (RQ) we pose is about the frequency of qubit usage across the five quantum computers to examine how transpilers distribute computational load. This question is central to our broader investigation: if noise-aware transpilation always prioritizes qubits with the best noise properties [19], [28], it may repeatedly map circuits to the same physical qubits. Such behavior not only reflects transpiler bias, but also has potential downstream effects – especially as workloads scale up and those qubits become overutilized.

RQ 1: Do some qubits get used more frequently than other qubits? What characteristics affect this frequency?

Our analysis confirms that a small subset of qubits indeed gets used more frequently across all five quantum computers, as illustrated in Fig. 3. Following the Pareto Principle, about 80% of the qubits get used less than 20% of the time. This aligns with our hypothesis that the transpiler's noise-aware optimizations tend to favor qubits with better noise characteristics, prioritizing those with lower error rates and more stable coherence times. Across the experiments, we observed that the same core subset of qubits repeatedly appeared in the transpiled circuits, regardless of the specific algorithms used.

To quantify this observation, we computed the correlation between qubit usage frequency and several qubit properties, including T1 and T2 times, gate error rates, and measurement

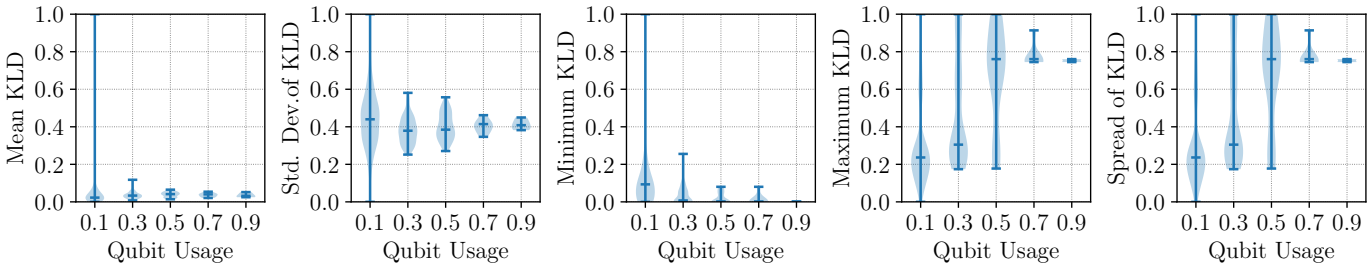

Fig. 4: Relationship between qubit usage and different KLD metrics (exponentiated) demonstrates the strong correlation.

TABLE III: Spearman correlation between the qubit usage frequency and KLD properties of algorithms run on them.

Metric	Correlation	p-value
Mean KLD	+0.39	6.28×10^{-18}
Standard Deviation of KLD	+0.41	8.96×10^{-20}
Maximum KLD	+0.72	7.59×10^{-74}
Minimum KLD	-0.71	7.78×10^{-69}
Spread of KLD	+0.79	5.49×10^{-97}

error rates, as shown in Table II. The analysis reveals a weak correlation with individual qubit properties – for example, the negative correlation between the gate/measurement error rates and qubit usage is as expected because qubits with lower error rates are used more often. Similarly, qubits with higher coherence times are used more often; hence, that correlation is positive. The weakest correlation is with qubit frequency (a quantum property), which is also as expected, as frequency is not factored into making transpilation decisions. Note that the correlations with individual properties are weak because the optimization passes take into account multiple properties that could be conflicting (e.g., a qubit has high coherence times but also high gate errors). Nonetheless, the expected correlation signs confirm that the transpiler maps circuits to the highest-performing qubits available on each device.

Observation 1

Our results show that noise-aware transpilation heavily concentrates circuit mapping onto a limited set of qubits, resulting in a small fraction of qubits handling the majority of computational tasks. This constrained resource utilization reflects a load-imbalance bias toward minimal error rates, leaving most qubits unutilized.

This result confirms that state-of-the-art transpilation policies prioritize fidelity by exploiting qubit heterogeneity. However, it also highlights an unintended consequence: qubits with slightly better metrics bear disproportionate workload, while most others remain underutilized. Whether this bias is justified – or whether it limits robustness and scalability – becomes a key question in our subsequent analysis.

RQ 2: Does the frequency of qubit usage affect the fidelity of the algorithms that are run on those qubits?

We now investigate whether the concentrated use of a small subset of qubits, as shown in RQ1, leads to observable effects on algorithm fidelity. Our core hypothesis is that while noise-aware transpilation may improve short-term fidelity by selecting the "best" qubits, this strategy could result in volatility or instability as those qubits are overused, particularly in workloads requiring repeated executions. To evaluate this, we analyze the correlation between qubit usage frequency and several fidelity metrics derived from the KLD. Table III reports the Spearman correlations between usage frequency and the mean, standard deviation, maximum, minimum, and spread of the KLD for circuits using those qubits.

Our findings, presented in Table III, show a small positive correlation (0.39) between qubit usage frequency and the mean KLD, indicating that circuits mapped to frequently used qubits may experience higher average output divergence from ideal results. More significantly and surprisingly, the maximum KLD exhibits a strong positive correlation (0.72) with qubit usage, and the minimum KLD exhibits a strong negative correlation (-0.71) with qubit usage, suggesting that frequently used qubits are more prone to worst-case fidelity degradation. The correlation between qubit usage frequency and the KLD spread (maximum KLD - minimum KLD) is similarly high (0.79), which suggests that the fidelity variability, or the fluctuations around the mean, increase with higher usage.

Fig. 4 further illustrates these relationships, with violin charts showing that as qubit usage increases, we observe higher KLD metrics, particularly in terms of maximum and spread. It is important to clarify what we are not claiming: this result does not prove that high-usage qubits undergo long-term physical degradation (e.g., from aging or permanent loss of coherence). In fact, our temporal analysis in later sections shows that compiled circuits can maintain fidelity across multiple calibration cycles. Instead, the takeaway is that aggressive reuse of select qubits leads to larger output variability in the short term. By output variability, we refer to the inconsistency in output distributions across repeated executions of the same circuit. Increased variability implies less predictable and less reliable quantum outputs. This is likely a result of dynamic cross-talk, thermal fluctuations, and other correlated noise sources that affect the "best" qubits more often simply because they are more used.

TABLE IV: Spearman correlation between the qubit usage frequency and KLD properties of algorithms run on them when algorithms are mapped to random qubits.

Metric	Correlation	p-value
Mean KLD	+0.27	6.41×10^{-12}
Standard Deviation of KLD	-0.27	2.54×10^{-12}
Maximum KLD	+0.45	7.30×10^{-33}
Minimum KLD	-0.43	1.99×10^{-30}
Spread of KLD	+0.47	4.28×10^{-36}

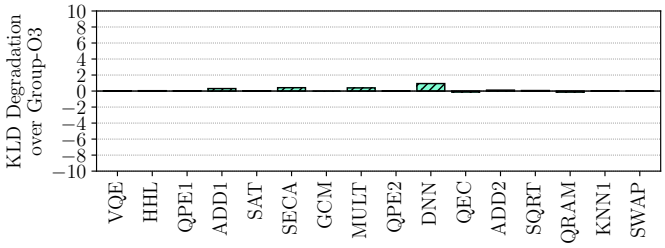

Fig. 5: The average KLD difference for each algorithm when run with the optimal mapping vs. random mapping is low.

Observation 2

Our analysis shows that, as opposed to conventional belief [15], [19], [26], [28], noise-adaptive mapping that focuses on the "best" qubits may actually degrade fidelity. Frequently used qubits yield higher worst-case error and larger variability in fidelity. While noise-aware transpilation selects these qubits to minimize immediate error, our results suggest that this over-reliance may increase unpredictability and reduce robustness, especially for workloads sensitive to fidelity fluctuations.

These findings make a compelling case for revisiting the goals of noise-aware transpilation. Rather than minimizing average error, an alternative objective might be to minimize variance or worst-case performance, especially for batch-executed or iterative quantum programs. This result thus sets the stage for RQ3, where we test whether evenly distributing circuit mappings across all qubits, regardless of their noise profiles, can mitigate variability while maintaining fidelity.

RQ 3: If the algorithms are distributed more evenly across different qubits, will we observe more stable results? Will the quality of the result decrease?

To address RQ3, we tested whether distributing algorithms more evenly across different qubits (using uniform random mapping) rather than concentrating them on the "best" qubits would yield more stable fidelity without compromising accuracy. We analyze this using Group-IM, which applies the same transpilation optimizations as Group-O3 but introduces randomness in the initial qubit mapping.

Table IV shows that the correlations between qubit usage frequency and KLD metrics drop consistently compared to Group-O3 (Table III). Notably, the correlation with maximum KLD drops from 0.72 (Group-O3) to 0.45, and with KLD spread from 0.79 to 0.47. These substantial reductions indicate that a more balanced use of qubits significantly improves fidelity consistency. Even more importantly, the mean KLD remains largely unchanged (0.27 versus 0.39 in Group-O3), suggesting that the average fidelity does not suffer despite the lack of noise-aware mapping. This answers a key concern: while random mapping might appear naive, it still retains comparable average fidelity while offering improved robustness.

Why is the mean KLD similar despite omitting noise-aware optimization? A likely reason is that while noise-aware transpilation optimizes for the average case, it often crowds computation onto a small patch of the device. As a result, dynamic qubit wear may negate the theoretical gains from selecting low-error qubits. Random mapping, by contrast, spreads risk across the chip and may avoid correlated noise effects. Fig. 5 confirms that the average fidelity of each algorithm under random mapping (Group-IM) is comparable to that achieved with noise-aware mapping (Group-O3) (KLD difference is ≤ 1, which is considered small), suggesting that a more even workload distribution can maintain accuracy while reducing extreme fluctuations and worst-case errors. An additional advantage is that this also eliminates the classical computing overhead related to noise-adaptive mapping.

Observation 3

Our findings suggest that a simple random mapping strategy can maintain similar average fidelity to noise-adaptive transpilation while significantly reducing output variability and worst-case errors. This approach not only stabilizes performance but also removes the classical overhead associated with frequent noise-adaptive mapping for each device calibration (e.g., once daily). This insight also opens up the opportunity to co-locate multiple programs on the same computer (not currently practiced due to the limited number of "good" qubits), as they do not have to compete for the best qubits and can be assigned an arbitrary random section of the chip.

These results also suggest new design principles for future transpilation strategies. Instead of always prioritizing lowest-error qubits, transpilers could include a diversity-aware term in their objective, trading small increases in expected KLD for reduced variance or more even qubit wear. For example, transpilers might randomize initial mappings within a subset of "acceptable" qubits or rotate mappings across calibrations to distribute usage over time. While our experiments use naïve random mapping, they motivate future work to formalize this trade-off between stability and short-term optimality.

B. Temporal Observations and Analysis

We now study the temporal effects of transpilation. The first natural question to ask is whether the classical overhead of transpilation is worth the effort to reduce the output error.

RQ 4: Is the classical computation time spent on routing optimizations well worth the effort?

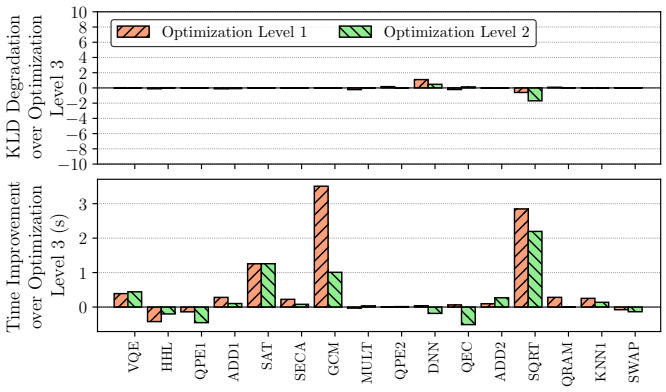

Fig. 6: Optimization levels 1 and 2 achieve similar KLD values as optimization level 3, while achieving much lower transpilation time than optimization level 3.

Fig. 6 shows the impact of optimization levels on KLD and transpilation time. The upper plot displays the KLD difference between circuits compiled at optimization levels 1 and 2 versus level 3. The lower plot shows the corresponding time difference between level 3 and the lower levels. Our analysis reveals that the choice of optimization level has minimal impact on circuit performance. Circuits compiled with optimization levels 1 and 2 have comparable performance to those using optimization level 3, with the majority showing KLD degradation of zero and none exceeding a degradation of 2. Interestingly, in some cases, like the *SQRT* result, lower optimization levels outperform optimization level 3. While the *DNN* benchmark shows a slight advantage when using optimization level 3, this is an exception rather than the norm. These results suggest that the aggressive optimization strategies of level 3 are not always necessary to maintain or improve fidelity. We hypothesize that this trend is due to diminishing returns from fine-tuned qubit selection on newer devices, where hardware variability is lower and coherence times are more stable. Moreover, because optimization level 3 is sensitive to calibration noise at a single point in time, it may, in some cases, overfit to transient characteristics rather than improving long-term performance.

In parallel with our fidelity analysis, we also wanted to see how much transpilation time was sacrificed when using higher optimization levels. The lower plot of Fig. 6 reveals that the majority of benchmarks (11 out of 16) demonstrate increased transpilation times when using optimization level 3 compared to levels 1 and 2. Furthermore, we observed that transpilation time overhead increases with algorithm size, making optimization level selection particularly crucial for larger circuits. Notably, certain algorithms, such as *SAT*, *GCM*, and *SQRT*, exhibited significant temporal improvements, reducing transpilation time by up to three seconds per execution. While this seems modest in isolation, these delays quickly compound at scale. For instance, running 10,000 circuits in a variational algorithm could lead to hours of additional compile time. We further observed that transpilation time tends to scale superlinearly with algorithm size, suggesting that this cost will grow even more pronounced for future circuits approaching the full 127-qubit capacity of current devices.

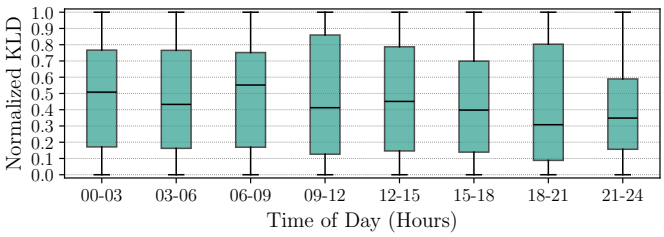

Fig. 7: Temporal variability of KLD values across all algorithms shows no monotonic trend based on time of day.

Observation 4

Circuits compiled with lower optimization levels (1 or 2) achieve fidelity comparable to level 3, but with significantly lower transpilation time, especially as circuit size increases. Therefore, full noise-aware optimization should not be used indiscriminately; instead, lightweight strategies may be preferable for large or repeatedly executed workloads, such as variational quantum algorithms.

The above insight propelled us to investigate the impact of the temporal characteristics of transpilation. Especially as IBM computers are calibrated once daily due to qubit drift [14] (around 12 am EST), we asked if it makes a difference if an algorithm is executed right after calibration or up to 24 hours after calibration. The implicit assumption is that the qubits may drift more significantly as more time passes after calibration, thus increasing the KLD. We test this assumption.

RQ 5: Does the distance of the circuit execution time from the time of machine calibration impact the output fidelity?

To investigate how the time of execution during the day affects quantum circuit results, we employed Group-O3 (same circuit each time) to measure the relationship between time and KLD performance. Treating each (Algorithm, Computer) pair as an independent group, we normalized the KLD values within each group to a range of 0 to 1. The data was then grouped into three-hour bins and visualized using box plots, as shown in Fig. 7. Our analysis shows normalized KLD medians consistently ranging between 0.3 and 0.55 across all time periods. While we observed slightly higher variability during 18:00-21:00 and more stable performance during 21:00-24:00 (evidenced by a smaller interquartile range), these variations are minimal. The results suggest no significant correlation between execution time and output fidelity, indicating that quantum circuit performance remains stable through the day.

We also conducted a detailed correlation analysis for individual algorithms as shown in Table V. It presents the Spearman correlation coefficients between execution time and KLD values for each algorithm across all computers. The strongest correlations were found in VQE (0.36) and HHL (-

TABLE V: Spearman correlation between time of day and KLD values for each algorithm shows weak correlation.

Algorithm	Corr.	p-value	Algorithm	Corr.	p-value
VQE	+0.36	8.78×10^{-4}	QPE2	+0.11	3.21×10^{-1}
HHL	-0.36	9.04×10^{-4}	DNN	-0.09	4.24×10^{-1}
QPE1	+0.34	1.84×10^{-3}	QEC	-0.08	4.78×10^{-1}
ADD1	-0.34	2.13×10^{-3}	ADD2	+0.05	6.36×10^{-1}
SAT	+0.29	9.25×10^{-3}	SQRT	-0.03	7.71×10^{-1}
SECA	-0.28	1.22×10^{-2}	QRAM	+0.01	9.57×10^{-1}
GCM	-0.25	2.33×10^{-2}	KNN	+0.14	2.11×10^{-1}
MULT	-0.20	6.87×10^{-2}	SWAP	-0.14	2.13×10^{-1}

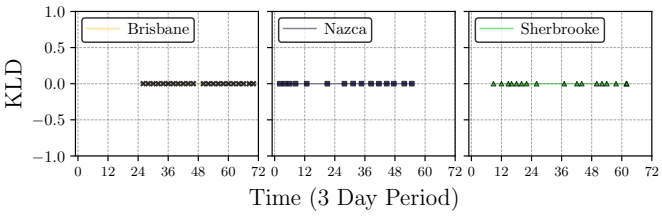

Fig. 8: Results for the GCM algorithm on three computers over a duration of 72 hours show little change in KLD.

0.36), both with p-values less than 0.001, yet these correlations are still relatively weak. Most algorithms showed even weaker correlations, with several (QRAM, SQRT, ADD2) showing correlations near zero. Interestingly, our analysis confirms that time-of-day effects remain insignificant not just on average but also at the individual algorithm level.

Importantly, this finding complements our earlier results: not only does random mapping yield more stable fidelity, but even noise-aware circuits compiled at the beginning of the day remain valid hours later. Together, these observations cast doubt on the assumption that circuit performance decays meaningfully with time since calibration. From a practical perspective, this opens the door to dramatically reducing transpilation frequency without sacrificing performance.

Observation 5

Our empirical findings indicate that the time elapsed since calibration – whether minutes or up to 24 hours – has minimal impact on output fidelity, i.e., there is no statistically significant relationship. This suggests that concerns about qubit drift post-calibration [14] may be overstated. This finding supports a shift toward fewer daily transpilation passes in production quantum workflows.

Building on our hour-of-the-day analysis, we now ask whether transpiled circuits can be used across multiple days, even as the calibration data changes. This question probes the heart of current transpilation workflows, which assume that circuits must be recompiled every day using the most recent calibration. If this assumption does not hold, the implications for system efficiency and large-scale workloads are substantial.

RQ 6: Does a circuit mapped, routed, and optimized for one calibration cycle perform just as effectively for other calibration cycles without needing to be transpiled again?

We examined the KLD values of the same circuit across three-day periods on three computers for a representative benchmark shown in Fig. 8. Due to varying queue times when accessing these quantum computers, the data points shown in the graphs have different times during which they end up being executed. The results show remarkably consistent KLD values near 0 across 72 hours for each computer. As shown in Fig. 8, the output fidelity, measured via KLD, remained remarkably stable across the 72-hour period. While minor fluctuations are expected due to queueing noise and machine load, there was no clear upward drift or consistent degradation trend. Practically, this opens the door to transpilation reuse — a simple optimization where a single transpilation output is cached and reused over time. This strategy could drastically reduce classical preprocessing cost, especially for variational algorithms that execute the same circuit thousands of times.

Observation 6

Circuits compiled once using calibration data from a single day retain stable fidelity across multiple calibration cycles, showing no observable performance degradation. This challenges the conventional wisdom that daily re-transpilation is necessary, and suggests that transpilation reuse may offer a practical path toward reducing compilation overhead for iterative quantum workloads [22].

Overall, our study redefines best practices in noise-adaptive transpilation, showing that frequent re-transpilation may not be essential to maintain fidelity on the latest superconducting quantum devices. This work advances efficient, scalable quantum workflows, especially for resource-intensive algorithms.

V. Related Work

Transpilation for noise-aware optimization has been a standard approach for improving quantum circuit performance on noisy quantum computing devices. As quantum circuits must be adapted to the error rates and hardware requirements of specific technologies and noise behaviors of different quantum computers, a large body of prior art has focused on developing transpilation strategies to improve fidelity, reduce error accumulation, and adapt to changing hardware conditions [8], [12], [16], [18], [20], [25], [26], [28], [33].

Early efforts in this space focused on noise-adaptive circuit mapping. Murali et al. [15] proposed a compilation approach that adapts qubit placement to minimize the impact of noise by routing operations through more reliable qubits. Similarly, Tannu and Qureshi [27] developed dynamic strategies to mitigate qubit-specific error rates, showing that careful qubit allocation and scheduling can significantly improve output fidelity. Li et al. [11] explored compilation techniques that strike a better balance between fidelity improvement and transpilation time, an issue that becomes critical as algorithm size increases. More recent work by Das et al. [5] extended

this direction by proposing adaptive transpilation strategies that respond to evolving noise profiles.

Beyond optimization methods, several works have sought to empirically characterize the behavior of quantum hardware and understand how noise impacts transpilation effectiveness [2], [4], [13], [19], [23]. For instance, Ash-Saki et al. [2] show that mitigating crosstalk through noise-aware transpilation can enhance fidelity, but it requires updated calibration data. Ravi et al. [23] emphasize the need for continual noise-aware transpilation to counteract dynamic noise changes, especially in cloud-based systems with high user contention. On the other hand, Dahlhauser et al. [4] model noisy circuits by decomposing them into smaller subcircuits for dynamic tuning, which reflects a granular approach to transpilation.

While these works underscore the importance of tailoring quantum circuits to hardware noise, they were primarily conducted during an earlier phase of quantum computing, typically using devices with fewer qubits and higher baseline noise levels. As quantum processors have grown and improved, the question arises whether these strategies continue to provide significant benefits. *Our work directly addresses this question through a large-scale empirical study on 127-qubit IBM quantum computers, re-evaluating the effectiveness and cost of noise-aware transpilation in light of modern hardware.*

VI. Conclusion

This work revisited a long-standing assumption in quantum computing: that noise-aware transpilation must be performed frequently to ensure high-fidelity execution. Through a large-scale empirical study, we found that this assumption does not always hold. Our results showed that noise-adaptive transpilation often concentrates execution on a small set of qubits, leading to instability and increased output variability. We demonstrated that random mapping strategies, though seemingly naive, can provide better load balancing and comparable fidelity while incurring lower compilation overhead.

Moreover, we found that circuit performance remains consistent not only throughout a 24-hour calibration cycle but also across multiple calibration cycles, enabling the reliable reuse of a single compiled circuit. These insights offer a new perspective on transpilation: it is not just about optimizing for transient noise characteristics, but about sustaining stable performance while minimizing classical costs. As quantum workloads scale up – particularly variational quantum algorithms – the ability to avoid frequent re-transpilation could dramatically reduce the classical overhead. We recommend a shift in best practices toward transpilation strategies that are both noise-aware and resource-efficient, especially as we approach the early fault-tolerant era of quantum computing.

Acknowledgement

We would like to thank the anonymous ICCAD reviewers, whose feedback helped improve this work. This work was supported by Rice University, the Rice University George R. Brown School of Engineering and Computing, and the Rice University Department of Computer Science. This work was supported by the DOE Quantum Testbed Finder Award DE-SC0024301, the Ken Kennedy Institute, and Rice Quantum Initiative, which is part of the Smalley-Curl Institute.

This work was also supported by the following NSF grants: CSR-2402328, CAREER-2338457, CSR-2406069, CSR-2323100, and HRD-2225201, as well as by the DOE Office of Science User Facility, supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, using NERSC award DDR-ERCAP0034716. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

References

[1] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, Antonio D. Córcoles-Gonzales, Abigail J. Cross, Andrew Cross, Juan Cruz-Benito, Chris Culver, Salvador De La Puente González, Enrique De La Torre, Delton Ding, Eugene Dumitrescu, Ivan Duran, Pieter Eendebak, Mark Everitt, Ismael Faro Sertage, Albert Frisch, Andreas Fuhrer, Jay Gambetta, Borja Godoy Gago, Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, Lukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Jose Martín-Fernández, Douglas T. McClure, David McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll, Diego Moreda Rodríguez, Giacomo Nannicini, Paul Nation, Pauline Ollitrault, Lee James O'Riordan, Hanhee Paik, Jesús Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia Rice, Abdón Rodríguez Davila, Raymond Harry Putra Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Silva, Yukio Siraichi, Seyon Sivarajah, John A. Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, Christopher Wood, Stephen Wood, Stefan Wörner, Ismail Yunus Akhalwaya, and Christa Zoufal. Qiskit: An Open-source Framework for Quantum Computing, January 2019.

[2] Abdullah Ash-Saki, Mahabubul Alam, and Swaroop Ghosh. Experimental characterization, modeling, and analysis of crosstalk in a quantum computer. *IEEE Transactions on Quantum Engineering*, 1:1–6, 2020.

[3] Davide Castelvecchi. The ai-quantum computing mash-up: will it revolutionize science? *Nature*, 2024.

[4] Megan L Dahlhauser and Travis S Humble. Modeling Noisy Quantum Circuits using Experimental Characterization. *Physical Review A*, 103(4):042603, 2021.

[5] Poulami Das, Swamit Tannu, Siddharth Dangwal, and Moinuddin Qureshi. Adapt: Mitigating idling errors in qubits via adaptive dynamical decoupling. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 950–962, 2021.

[6] Poulami Das, Swamit S Tannu, Prashant J Nair, and Moinuddin Qureshi. A Case for Multi-Programming Quantum Computers. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 291–303. ACM, 2019.

[7] Rui Ding and Andrew Mullhaupt. Empirical squared hellinger distance estimator and generalizations to a family of α-divergence estimators. *Entropy*, 25(4):612, 2023.

[8] Suguru Endo, Simon C. Benjamin, and Ying Li. Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan*, 90(3):032001, 2021.

[9] Morten Kjaergaard, Mollie E Schwartz, Jochen Braumüller, Philip Krantz, Joel I-J Wang, Simon Gustavsson, and William D Oliver. Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, 11(1):369–395, 2020.

[10] Ang Li, Samuel Stein, Sriram Krishnamoorthy, and James Ang. QASMBench: A Low-Level Quantum Benchmark Suite for NISQ Evaluation and Simulation. *ACM Transactions on Quantum Computing*, 4(2):1–26, 2023.

[11] Gushu Li, Yufei Ding, and Yuan Xie. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1001–1014. ACM, 2019.

[12] Ji Liu, Ed Younis, Mathias Weiden, Paul Hovland, John Kubiatowicz, and Costin Iancu. Tackling the qubit mapping problem with permutation-aware synthesis. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 1, pages 745–756. IEEE, 2023.

[13] Ji Liu and Huiyang Zhou. Reliability Modeling of NISQ-Era Quantum Computers. In *2020 IEEE International Symposium on Workload Characterization (IISWC)*, pages 94–105. IEEE, 2020.

[14] Yiding Liu, Zedong Li, Alan Robertson, Xin Fu, and Shuaiwen Leon Song. Enabling efficient real-time calibration on cloud quantum machines. *IEEE Transactions on Quantum Engineering*, 4:1–17, 2023.

[15] 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 *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1015–1029. ACM, 2019.

[16] Prakash Murali, David C McKay, Margaret Martonosi, and Ali Javadi-Abhari. Software Mitigation of Crosstalk on Noisy Intermediate-Scale Quantum Computers. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1001–1016, 2020.

[17] Kazuki Okamura. Metrization of powers of the jensen-shannon divergence. *arXiv preprint arXiv:2302.10070*, 2023.

[18] 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 *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pages 705–711, 2020.

[19] Tirthak Patel, Abhay Potharaju, Baolin Li, Rohan Roy, and Devesh Tiwari. Experimental Evaluation of NISQ Quantum Computers: Error Measurement, Characterization, and Implications. In *2020 SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 636–650. IEEE Computer Society, 2020.

[20] Tirthak Patel and Devesh Tiwari. VERITAS: Accurately Estimating the Correct Output on Noisy Intermediate-Scale Quantum Computers. In *2020 SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 188–203. IEEE Computer Society, 2020.

[21] Tirthak Patel, Ed Younis, Costin Iancu, Wibe de Jong, and Devesh Tiwari. QUEST: Systematically Approximating Quantum Circuits for Higher Output Fidelity. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 514–528, 2022.

[22] John Preskill. Quantum information and quantum computing. *American Journal of Physics*, 68(9):781–795, 2022.

[23] Gokul Subramanian Ravi, Kaitlin N Smith, Pranav Gokhale, and Frederic T Chong. Quantum computing in the cloud: Analyzing job and machine characteristics. In *2021 IEEE International Symposium on Workload Characterization (IISWC)*, pages 39–50. IEEE, 2021.

[24] Kaitlin N Smith, Gokul Subramanian Ravi, Jonathan M Baker, and Frederic T Chong. Scaling Superconducting Quantum Computers with Chiplet Architectures. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1092–1109. IEEE, 2022.

[25] Daniel Bochen Tan, Dolev Bluvstein, Mikhail D. Lukin, and Jason Cong. Compiling Quantum Circuits for Dynamically Field-Programmable Neutral Atoms Array Processors, 2023.

[26] Swamit S Tannu and Moinuddin Qureshi. Ensemble of Diverse Mappings: Improving Reliability of Quantum Computers by Orchestrating Dissimilar Mistakes. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 253–265. ACM, 2019.

[27] Swamit S Tannu and Moinuddin K Qureshi. Mitigating Measurement Errors in Quantum Computers by Exploiting State-Dependent Bias. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 279–290. ACM, 2019.

[28] 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 *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 987–999. ACM, 2019.

[29] Robert R Tucci. An introduction to Cartan's KAK decomposition for QC programmers. *arXiv preprint quant-ph/0507171*, 2005.

[30] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[31] Christopher D Wilen, S Abdullah, NA Kurinsky, C Stanford, L Cardani, G d'Imperio, C Tomei, L Faoro, LB Ioffe, CH Liu, et al. Correlated charge noise and relaxation errors in superconducting qubits. *Nature*, 594(7863):369–373, 2021.

[32] Clark Wissler. The spearman correlation formula. *Science*, 22(558):309–311, 1905.

[33] Alwin Zulehner and Robert Wille. Compiling SU (4) Quantum Circuits to IBM QX Architectures. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pages 185–190. ACM, 2019.