From Ideal to Noisy: Adapting Property-Based Testing for Real-World Noisy Quantum Computers

1st Gabriel Pontolillo
Department of Informatics
King's College London
London, United Kingdom
gabriel.pontolillo@kcl.ac.uk

2nd Asmar Muqeet
Simula Research Laboratory
and University of Oslo
Oslo, Norway
asmar@simula.no

3rd Shaukat Ali
Simula Research Laboratory
Oslo, Norway
shaukat@simula.no

4th Mohammad Reza Mousavi
Department of Informatics
King's College London
London, United Kingdom
mohammad.mousavi@kcl.ac.uk

***Abstract*—Quantum software testing is essential for the quality assurance of quantum programs. However, existing techniques face many challenges, such as constructing test oracles and distinguishing genuine faults from noise-induced errors. Property-based testing is a promising solution for dealing with the test oracle issue, which verifies general properties rather than requiring inputs and outputs. However, its effectiveness in noisy quantum environments has not been studied. To this end, we evaluate the feasibility of applying property-based testing in noisy quantum computers by integrating it with a state-of-the-art machine learning-based noise mitigation method called QOIN. Our results show that on average, QOIN mitigates noise for individual circuits, though it may introduce outliers. Crucially, this does not guarantee property preservation: we show that some properties align better with QOIN, while others remain closer under unmitigated noise. To address this, we apply a hybrid approach that selectively applies QOIN during property-based testing. We show that this hybrid approach improves alignment with the ideal execution and significantly reduces false positives in assertion outcomes across most executed mutants. This provides a solid foundation for applying property-based testing to noisy quantum systems.**

***Index Terms*—Quantum Computing, Quantum Software Testing, Property-Based Testing, Noise Mitigation, Machine Learning**

I. Introduction

Quantum computing offers the prospect of significantly advancing fields such as medicine and materials science by solving problems that are currently intractable on classical computers [1]. However, realizing this potential is hindered by fundamental obstacles such as quantum noise. In the current era of noisy quantum computing, hardware is inherently affected by noise, which can substantially degrade the reliability and accuracy of quantum computations. This presents a major obstacle to developing a reliable discipline for testing quantum software.

Quantum software testing aims to efficiently detect bugs in quantum programs to ensure a certain level of correctness [2]. However, testing quantum software is particularly challenging due to the fundamental principles of quantum mechanics, such as superposition and entanglement [3]. Noise further complicates this process, making it difficult to determine whether a test case failed due to a software fault or quantum noise despite the software being correct. Existing quantum software testing approaches primarily adapt classical testing techniques, including combinatorial testing [4], coverage-based test selection [5]–[7], search-based testing [8], [9], mutation testing [9]–[11], and metamorphic testing [12], [13]. However, these methods have two major limitations: (1) they typically assume an ideal, noise-free quantum simulator, which are inefficient for large real problems; lack of a proper treatment of noise makes these testing techniques unsuitable for real, noisy quantum computers, and (2) they rely on explicit test oracles to determine whether a test case passes or fails, which is difficult to construct due to the black-box nature of quantum programs.

To address the challenge of explicit test oracles, property-based testing (e.g., QSharpCheck [14] and QuCheck [15]) has recently been explored as a potential alternative to methods that require explicit oracles. Unlike testing methods that rely on predefined test oracles, property-based testing verifies whether a quantum program satisfies general properties across a range of automatically generated test cases, reducing the need for manually specified oracles. This approach mitigates the difficulty of constructing explicit test oracles by focusing on high-level properties rather than specific outputs. Prior studies have demonstrated its effectiveness in ideal, noise-free quantum simulators [15], [16]; however, its feasibility in the presence of quantum noise remains an open question.

This paper addresses whether property-based testing can be effectively applied to noisy quantum environments by integrating it with a noise mitigation technique. We utilize QOIN [17], a state-of-the-art machine learning-based noise mitigation technique. QOIN reduces the impact of noise on quantum program outputs and can be integrated with existing quantum software testing methods. It has demonstrated effectiveness in mitigating noise across various quantum programs executed on simulated noisy computers from IBM, Google, and Rigetti [17]. Therefore, we investigate whether combining property-based testing with QOIN enables its application in noisy, simulated quantum environments. By doing so, we aim to extend the applicability of property-based testing beyond idealized, noise-free conditions. The contribution of the paper is summarized below:

- We extend the state-of-the-art property-based testing method (QuCheck) to support noisy quantum computers by integrating a noise mitigation technique (QOIN).
- We empirically evaluate the extended QuCheck method

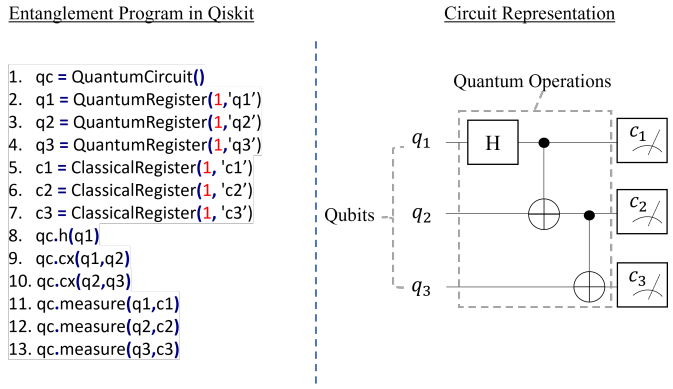

Fig. 1. A three-qubit entanglement quantum circuit written in Qiskit with its corresponding circuit representation.

on five real quantum programs using a noisy simulator of IBM's latest quantum computer, IBM_Fez, to demonstrate its effectiveness in noisy environments.

The remainder of this paper is organized as follows: Section II provides necessary background and reviews prior research on property-based testing for quantum programs, machine learning for noise mitigation, and the general effects of noise on quantum computation. Section III outlines the design and execution of our experiments. Section IV presents the outcomes of our study, followed by an interpretation and discussion of these findings. Finally, Section VI summarizes our contributions and suggests directions for future research.

II. Background and Related Work

A. Quantum Computing Basics and Quantum Circuit

In classical computers, the information is stored and processed using bits that exist exclusively in one of two states: 0 or 1. In contrast, quantum computers operate with quantum bits, or *qubits*, which can exist in a superposition of the states $|0\rangle$ and $|1\rangle$ with associated complex amplitudes. In polar form, each amplitude has both a magnitude and a phase. Mathematically, a qubit is represented in Dirac notation as:$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$, where α_0 and α_1 are the amplitudes corresponding to states $|0\rangle$ and $|1\rangle$, respectively. The probability of measuring the qubit in a particular state is given by the square of the magnitude of the corresponding amplitude, and these probabilities sum to 1.

Quantum gates are modeled as unitary operators that act on the quantum state of qubits, applying transformations defined by unitary matrices [18]. For example, the Hadamard gate creates a superposition by transforming a qubit from a basis state into an equal combination of $|0\rangle$ and $|1\rangle$. Quantum computers are programmed by constructing *quantum circuits*, where the logic of the program is implemented as a sequence of quantum gates applied to qubits. Throughout this paper, we will use the term *circuit* to refer to a quantum program for improved readability.

Figure 1 shows a three-qubit *entanglement* circuit in Qiskit [19] with its corresponding circuit representation.

TABLE I
The ideal and noisy outputs of a three-qubit quantum circuit after execution on an ideal and a noisy simulator. The *Probability* column shows the likelihood of obtaining each specific output.

	Probability							
Output States	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Ideal Simulator	0.5	-	-	-	-	-	-	0.5
Noisy Simulator	0.443	0.012	0.016	0.008	0.007	0.019	0.021	0.476

The circuit creates entanglement among three qubits, meaning that the measurement outcomes are correlated and cannot be described independently. In lines 1-7, the circuit is initialized with three qubits (*q1*, *q2*, *q3*) in the state $|000\rangle$, and three classical registers (*c1*, *c2*, *c3*) to store the final measurement results. At line 8, a Hadamard gate is applied to *q1*, placing it into a superposition of $|0\rangle$ and $|1\rangle$. Then, at line 9, a controlled-NOT (CX) gate entangles *q1* and *q2*; since *q1* is already in superposition, the application of CX results in the circuit being in a superposition of the two-qubit states $|00\rangle$ and $|11\rangle$. At line 10, another CX gate entangles *q2* with *q3*, extending the entanglement across all three qubits. At this stage, measuring the qubits will yield either $|000\rangle$ or $|111\rangle$, each with a 50% probability. Finally, lines 11-13 apply measurement operations to all three qubits, storing the results in the corresponding classical registers (*c1* to *c3*).

B. The Impact of Noise on Quantum Circuit Testing

In the current noisy quantum computing era, quantum devices are constrained by a limited number of qubits, which are vulnerable to multiple sources of noise. First, environmental factors such as magnetic fields and radiation can disturb qubit states, leading to decoherence, where interactions with the environment cause information loss and state disturbances [3], [20]. Second, even in the absence of environmental noise, qubits can unintentionally interact with one another during gate execution, a phenomenon known as crosstalk noise [21], which results in the creation of erroneous quantum states that affect computations. Third, imprecise calibrations of quantum gates contribute to noise. Although quantum gate calibration is essential for optimizing performance and minimizing errors, even minor inaccuracies can induce small phase shifts, amplitude variations, and other discrepancies. These subtle errors might not immediately destroy a quantum state but can accumulate over a series of gate operations, ultimately leading to unintended outcomes [21]. Noise interferes with the quantum state and propagates across the circuit during execution, ultimately leading to unreliable measurement outcomes.

In the context of testing quantum circuits, noise presents a significant challenge, as it can cause a circuit to produce incorrect outputs. For example, consider a three-qubit quantum circuit designed to ideally output two states, $|\mathbf{000}\rangle$ and $|\mathbf{111}\rangle$ as in Figure 1 with equal probability (see Table I). However, when executed on a real quantum computer, the circuit's output is distributed over all eight possible output states, rather than just the two expected ones. Noise can cause incorrect output states to appear or shift the probabilities of correct states,

making it difficult to determine whether a test failure is due to a genuine bug in the quantum circuit or the effects of noise. This complicates the testing and validation of quantum circuits.

C. Property-based Testing of Quantum Circuits

A major challenge in testing quantum circuits is the oracle problem. Since quantum circuits are designed to solve complex problems that are intractable for classical computers, determining the expected output for a given test case is often difficult. In this context, classical black-box testing methods, such as property-based testing, offer a more practical approach. Property-based testing has been successfully applied to quantum circuits using frameworks like Q# [14] and Qiskit [15], [16]. Prior research has also integrated property-based test oracles within delta debugging to analyze the outcomes of sub-circuits constructed from applying subsets of failure-inducing changes [16]. In these approaches, properties define expected behaviors, and the testing framework verifies these behaviors against randomized inputs. However, existing studies have focused solely on ideal simulators. This paper investigates whether property-based testing can be extended to noisy quantum environments, evaluating its effectiveness in the presence of quantum noise.

In property-based testing, the invariants of a program that hold for a range of inputs are verified instead of specific test cases. To specify a property, four components are necessary: input generation, preconditions, operation, and postcondition. Inputs are automatically generated during property-based testing to find counterexamples, an input generator is required for this, such as `random_state(num_qubits)`, which generates a random statevector for a given number of qubits. Furthermore, for a property to be valid, restrictions may need to be placed on the generated inputs; this is achieved through a precondition. Operations describe the sequence of steps that need to be performed; this involves utilizing the generated inputs and calling the function to be tested. This is followed by the verification of the postcondition, which checks whether the invariant still holds. Huang and Martonosi propose *statistical assertions* [22] for verifying characteristics of quantum states, such as superposition and entanglement, using chi-square tests. More recent work [23] extends this concept by introducing an assertion framework that automatically determines where to insert measurements, computes the number of shots required to guarantee statistical confidence, and accounts for device noise by adjusting the expected outcome distribution during assertion evaluation. QuCheck offers predefined assertions that can be inserted within a properties' operation function, such as `AssertEqual` and `AssertDifferent`, which compare the marginal probability distributions of outputs in pairs of circuits, and are exclusively evaluated in Section IV-B. Similarly, QuCheck automatically inserts the necessary measurements for each assertion. However, it does not compute the number of measurement shots required to achieve a statistical confidence level. In this work, we apply a noise mitigation technique to reduce the impact of noise on the observed marginal distribution, rather than adjusting the expected outcomes. An example of a property for the GHZ circuit can be seen in Figure 2.

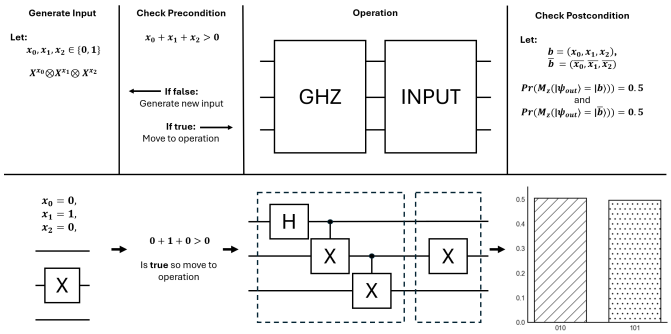

Fig. 2. Example GHZ circuit property. *Input*: A generator that constructs a three qubit circuit by randomly applying a Pauli-X gate to each register ($X^{x_0} \otimes X^{x_1} \otimes X^{x_2}$, where $x_0, x_1, x_2 \in \{0, 1\}$). *Precondition*: At least one Pauli-X gate must be applied ($x_0 + x_1 + x_2 > 0$). *Operation*: Apply the GHZ circuit to the $|000\rangle$ state, then apply the generated Pauli-X circuit, yielding the state $|\psi_{out}\rangle$. *Postcondition*: Measuring the computational basis yields two bitstrings $b = (x_0, x_1, x_2)$ and $\bar{b} = (\bar{x}_0, \bar{x}_1, \bar{x}_2)$ with equal probability: $Pr(M_z(|\psi_{out}\rangle) = |b\rangle) = 0.5$, and $Pr(M_z(|\psi_{out}\rangle) = |\bar{b}\rangle) = 0.5$, where M_z is a measurement in the computational basis, and Pr denotes the probability of obtaining the specified outcome.

D. Machine Learning to Mitigate Quantum Noise

Recent advancements in noise mitigation for quantum circuits have increasingly leveraged machine learning techniques. Notable approaches include Clifford Data Regression (CDR) [24], Learning-based Probabilistic Error Cancellation (L-PEC) [25], ML-QEM [26], QRAFT [27], QLEAR [28], and QOIN [17]. Among these, QOIN represents the state-of-the-art, employing a machine learning model to map noisy measurement results to their ideal, noise-free counterparts [17].

Previous research has demonstrated QOIN's effectiveness in significantly reducing noise across various quantum circuits and noisy simulators that emulate the noise characteristics of IBM, Google, and Rigetti's quantum hardware. In this work, we integrate QOIN into the property-based testing framework by applying it to measurement outcomes before verifying the defined properties. This integration aims to assess whether property-based testing remains effective in noisy quantum environments when combined with machine learning-based noise mitigation.

III. Methodology and Experiment Design

We performed a comparative analysis across three scenarios (described below) to extend the applicability of property-based testing to noisy quantum computers. For reproducibility, the repository is available at [29].

a) Scenario 1 - Noisy Simulator Without Error Mitigation: In the first scenario, we analyzed property-based testing on a noisy simulator to assess the impact of quantum noise on property evaluations. To establish a baseline, we conducted a controlled experiment using only a noisy simulator without applying QOIN for error mitigation. This allowed us to

determine the extent to which quantum noise leads to excessive assertion failures and whether noise mitigation techniques like QOIN are necessary for property-based testing to remain effective in a noisy environment.

b) Scenario 2 - Noisy Simulator with QOIN Integration: In the second scenario we integrated the QOIN noise error mitigation method to assess whether the failures observed in the first scenario, which used only a noisy simulator, could be corrected through noise mitigation. This scenario will help us understand the effectiveness and limitations of combining noise mitigation techniques with property-based testing in noisy quantum environments.

c) Scenario 3 - Hybrid Approach: In the third scenario, we evaluated a hybrid strategy that selectively applies QOIN error mitigation. Property-based testing is first performed on a noisy simulator without QOIN, reflecting the observation from Section IV-B that unmitigated noisy executions can sometimes better approximate noiseless outcomes. When tests fail under these conditions, the failing cases are re-evaluated with QOIN applied. This approach aims to distinguish between failures caused by noise and those stemming from actual faults. By selectively applying QOIN only when necessary, this scenario investigates whether test robustness can be improved while reducing computational overhead.

Throughout the results section, scenario 1 will be referred to as the "noisy execution" and scenario 2 as the "QOIN execution" for brevity.

A. Research Questions

To evaluate whether the property-based testing method (QuCheck) can be effectively applied to noisy quantum computers by integrating the noise mitigation technique QOIN, we formulate the following research questions.

1) **RQ1:** To what extent does QOIN mitigate the impact of noise on circuit outputs executed in noisy quantum environments?
   In RQ1, we aim to examine how well QOIN mitigates noise within our selected noisy quantum simulator. While QOIN has been evaluated in prior work with other noise backends, it has not been evaluated for the more recent IBM_Fez. Moreover, we aim to understand how QOIN interacts with the circuits generated during the property-based testing of our case studies that include our mutants, which may yield different results.
2) **RQ2:** How does noise impact the reliability of post-condition evaluations in QuCheck, and how effectively does QOIN mitigate this impact?
   In RQ2, we aim to determine the extent to which noise impacts the properties defined in the QuCheck property-based testing approach. Additionally, we assess whether applying QOIN can mitigate noise while preserving these properties, ensuring the validity of test evaluations despite the presence of quantum noise.
3) **RQ3:** How does the *effectiveness* of the hybrid approach compare to the ideal approach and the noisy simulation with and without mitigation?

TABLE II
QUANTUM PROGRAMS USED IN THE EXPERIMENTS, AND THEIR DETAILS

Quantum Program	Width	Depth	Transpiled Depth
Quantum Teleportation (QT)	3	4 - 8	16 - 32
Quantum Fourier Transform (QFT)	2 - 5	4 - 42	10 - 5530
Grover's Algorithm (GR)	3 - 5	9 - 55	65 - 4306
Deutsch-Jozsa (DJ)	2 - 5	3 - 18	4 - 45
Quantum Phase Estimation (QPE)	2 - 4	3 - 21	4 - 459

In RQ3, we aim to assess the effectiveness of the hybrid approach, where only the failing test cases are re-evaluated using the noisy simulator with QOIN applied. Where *effectiveness* is measured through the difference in the number of failed assertions to the ideal execution. This evaluation will help determine whether the selective application of QOIN to failed assertions provides a computationally efficient yet reliable method for property-based testing in noisy quantum environments.

B. Benchmarks

In this study, we employ the same quantum programs as in the original QuCheck study [15]. These selected quantum circuits are: Quantum Fourier Transform, Quantum Phase Estimation, Quantum Teleportation, Deutsch-Jozsa, and Grover's Algorithm. These circuits vary in terms of gate depth, as shown in Table II, and contain different quantum subroutines, such as phase estimation and amplitude amplification. This provides a modest, yet diverse set of test cases for evaluating the robustness of our methodology.

Using the QuCheck property-based testing framework, we defined three properties for each quantum program, labeled A, B, and C in Section IV. To evaluate these properties under noisy conditions, we used 200 randomly generated test inputs and performed 3000 measurement shots for each test case. In our previous QuCheck study [15], we observed diminishing returns in mutation score at approximately 3000 measurement shots. However, we increased the number of inputs tested from 50 to 200, as we did not observe similar diminishing returns with respect to input counts. This increase allowed a greater number of assertions to be executed and evaluated, enabling a more thorough comparison between each different scenario.

C. Quantum Circuit Execution

To execute quantum circuits in our experiment, we utilized IBM's Qiskit Aer noiseless simulator [19] as the ideal baseline. For noisy quantum circuit execution, we configured the Aer simulator with the noise parameters of the IBM_Fez quantum computer. All circuits were transpiled using Qiskit's `transpile` function, targeting the backend's native gate set (`['id', 'sx', 'x', 'cz', 'rz']`) with optimization level zero to avoid simplifying equivalent mutants. These parameters, obtained from IBM's calibration data or real quantum hardware [30], include thermal relaxation errors, gate errors, and measurement errors, enabling the Aer simulator to replicate the noise effects present in IBM's quantum computers.

TABLE III
QOIN MODEL TRAINING METRICS

Quantum Program	Unique Circuits (N)	Storage (s)	Execution (s)	Training (s)
QT	3645	26.0	315.9	61.1
QFT	4608	465.6	2752.9	86.7
GR	990	54.9	501.1	19.0
DJ	1605	7.3	115.2	57.7
QPE	4977	155.4	1035.9	50.5

D. QOIN model Training

QOIN is a machine learning-based approach that requires training a model for each noisy quantum computer. To achieve this, we followed a similar training procedure outlined in the original study [17]. We generated and stored circuits through the same process as when running property-based tests, meaning the same 15 mutants, 3 properties, but with 20 random inputs per property and 10,240 total shots for both the ideal simulator and `IBM_Fez` noise model. These values were selected as a practical balance between recording sufficient data for model training and keeping the computational overhead manageable, as indicated by our preliminary experiments. The number of unique circuits (N) executed varied by quantum programs due to: optimizations performed by QuCheck, some properties requiring the execution of pairs of circuits, and different basis measurements (i.e., some properties only required Z basis measurement, thus needed fewer circuits to evaluate). We then extract three features, POS, ODR, and POF, using the circuit results, according to [17]. These are used to train a two-stage model: a LightGBM classifier identifies which basis states are expected to appear in the ideal simulation, and a linear regression model estimates their ideal amplitudes. The data is split into a stratified 70/30 train-test partition. The trained model is then used during evaluation to reconstruct the ideal output distribution by filtering and correcting the noisy results.

The number of distinct circuits executed, time taken for circuit generation and storage, training circuit execution, and model training are listed in Table III. All experiments were performed on a Windows 11 desktop equipped with an AMD Ryzen 7 5700X CPU and an NVIDIA RTX 3060 Ti GPU.

E. Metrics

To answer our research questions, we employed mutation analysis, a technique that involves introducing syntactic modifications (mutations) to a program, running tests on these mutated versions, and measuring the proportion of cases where the tests fail (mutation score). In particular, we included both equivalent and non-equivalent mutants to evaluate distinct characteristics of our testing approach: testing with equivalent mutants enables us to measure the false-positive rate of property-based testing in isolation, while non-equivalent mutants help assess its fault-detection effectiveness. For our experiments, we used the same mutants from the original QuCheck study [15], comprising five equivalent mutants and ten additional mutants generated using QMutPy [31] (hereby referred to as *generated mutants*). QMutPy applies different mutation types, such as gate deletions, substitutions, and insertions, though it does not guarantee semantic differences in the resulting circuits. QMutPy was not used for the creation of equivalent mutants because it does not provide an option to guarantee their creation. Instead, equivalent mutants were generated by inserting gate identities into the base quantum circuit at random locations, as this feature was not explicitly available in QMutPy.

Although generated mutants are designed to measure the fault-detection capability, an increase in the mutation score under noisy conditions does not necessarily indicate improved detection. It may reflect a rise in false positives caused by noise, which we distinguish by comparing noisy results to an ideal (noiseless) execution.

For RQ1 and RQ2, we use the Hellinger distance as a metric to quantify the difference between ideal execution results, noisy execution results, and QOIN-mitigated results. The Hellinger distance is a well-established measure for comparing quantum circuit execution outcomes in noisy environments [28], [32], [33]. The Hellinger distance (H) between two probability distributions (P, Q) is a value that ranges between 0 and 1, where a distance of 0 indicates that the probability distributions are equivalent.

$$H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{n}(\sqrt{p_i} - \sqrt{q_i})^2},$$

where $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$ are discrete probability distributions.

In RQ2, we specifically consider `AssertEqual` and `AssertDifferent` assertions in the QuCheck property-based testing framework, which evaluate qubit measurement distributions between circuit pairs, in order to determine whether QOIN preserves circuit properties while mitigating noise. Concretely, we compute the Hellinger distance *between circuit pairs* in the QOIN-mitigated execution, using both ideal and noisy executions as reference baselines. For RQ3, we measure overall effectiveness by the number of failed assertions in QuCheck.

IV. Results and Discussion

A. RQ1: Effectiveness of QOIN Noise Mitigation

To assess the effectiveness of QOIN in mitigating noise, we analyze the distribution of Hellinger distances between the measurement outcomes of quantum circuits executed under different simulation conditions. Specifically, we compare the measurement distributions obtained from (i) an ideal, noiseless simulator, (ii) a noisy simulator execution without mitigation, and (iii) a noisy simulator with QOIN mitigation applied. By contrasting these distributions, we aim to quantify the extent to which QOIN reduces the divergence introduced by noise.

Figure 3 presents a box plot illustrating the Hellinger distances between the noiseless execution and the two noisy execution scenarios (with and without QOIN). The x-axis represents each quantum circuit evaluated in our benchmark, aggregating all circuits evaluated across the 15 mutants, and the three tested properties per circuit.

QOIN's noise mitigation showed significant improvements in reducing Hellinger distances to the ideal distribution across

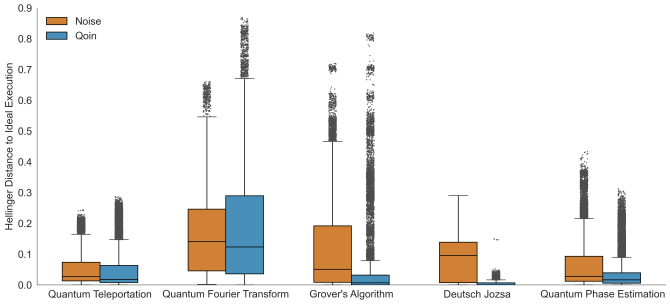

Fig. 3. Hellinger distance between the ideal (noiseless) circuit output distribution and those from (1) noisy (orange) and (2) noise-mitigated QOIN (blue) executions. Lower values for the QOIN execution indicate show the effectiveness of noise mitigation.

TABLE IV
MEAN, MEDIAN, AND STANDARD DEVIATION OF THE HELLINGER DISTANCE BETWEEN THE IDEAL EXECUTION AND NOISY/QOIN EXECUTION ACROSS DIFFERENT QUANTUM ALGORITHMS.

Algorithm	Execution	Mean	Median	Std. Dev.	Max	Min
Quantum Teleportation	Noise	0.0554	0.0267	**0.0607**	**0.2437**	0.0000
	QOIN	**0.0476**	**0.0178**	0.0629	0.2875	0.0000
Quantum Fourier Transform	Noise	**0.1585**	0.1405	**0.1233**	**0.6608**	0.0017
	QOIN	0.1699	**0.1235**	0.1563	0.8682	**0.0000**
Grover's Algorithm	Noise	0.1263	0.0507	0.1523	**0.7200**	0.0000
	QOIN	**0.0731**	**0.0066**	**0.1453**	0.8188	0.0000
Deutsch-Jozsa	Noise	0.0807	0.0951	0.0748	0.2908	0.0000
	QOIN	**0.0042**	**0.0000**	**0.0071**	**0.1503**	0.0000
Quantum Phase Estimation	Noise	0.0631	0.0277	0.0756	0.4329	0.0000
	QOIN	**0.0348**	**0.0166**	**0.0485**	**0.3123**	0.0000

the evaluated quantum circuits. Although, the extent of its effectiveness varied, forming three patterns based on the mean, median, and maximum distances observed.

The first group, consisting of the Deutsch-Jozsa (DJ) and Quantum Phase Estimation (QPE) algorithms, demonstrated the most consistent improvements. For DJ, the mean Hellinger distance decreased from 0.0807 in the noisy execution to 0.0042 with QOIN mitigation (Table IV). Notably, the median distance was reduced to 0, compared to 0.095 in the unmitigated scenario, while the standard deviation also decreased substantially from 0.0748 to 0.0071. QPE saw similar improvements in its mean and median distances, with the mean decreasing from 0.0631 to 0.0348, and median from 0.0277 to 0.0166. In both cases, the maximum Hellinger distance was reduced. A potential reason for QOIN's effectiveness in the DJ algorithm is the nature of the properties being measured. In DJ A, the lower register remains in the $|-\rangle$ state at the end of the computation, while DJ B and DJ C determine whether a function is balanced or constant. Specifically, for constant functions, the output should be $|0^{\otimes n}\rangle$, whereas for balanced functions, the output should not be entirely $|0^{\otimes n}\rangle$. These relationships are relatively simple, making them easier for QOIN's model to recognize and correct under noise. Furthermore, both algorithms, when transpiled, had a relatively low gate depth compared to the other algorithms.

The second group, including Grover's algorithm (GR) and Quantum Teleportation (QT), followed a slightly different pattern. Again, QOIN reduced the mean compared to the noisy execution (Grover: 0.1263 to 0.0731, QT: 0.0554 to 0.0476) and median distances (GR: 0.0507 to 0.0066, QT: 0.0267 to 0.0178). Coincidentally, these are the other two algorithms that contain a property that verify a static invariant: GR A checks the lower register is $|-\rangle$, QT B checks the upper register is in the $|++\rangle$ state after teleportation, which should always hold in the CZ and CX implementation. However, for GR, the maximum Hellinger distance rose from 0.7200 in the noisy execution to 0.8188 with QOIN, while for QT, it increased from 0.2437 to 0.2875. The increase in maximum Hellinger distances to ideal observed in the QOIN execution is much higher for GR (0.099) compared to QT's maximum increase (0.044), this may be due to the larger number of max gates (4306 vs. 32) after transpilation.

Finally, the Quantum Fourier Transform (QFT) followed its own, more negative trend. While QOIN lowered the median distance (0.1405 to 0.1235), it increased both the mean (0.1585 to 0.1699) and maximum distances (0.6608 to 0.8682), along with the standard deviation (0.1233 to 0.1563) to the ideal execution over the noisy scenario. Interestingly, the noisy execution for QFT was the only case that did not reach a minimum of zero Hellinger distance to the ideal execution (Table IV). The overall reduced performance may stem from the fact that the properties defined for testing QFT are heavily phase-dependent. This, combined with the circuits having the highest maximum gate count after transpilation, could explain the observed drop in performance.

RQ1: When applying QOIN across five quantum algorithms, we generally observed a reduction in the mean and median Hellinger distance to the ideal execution over the noisy execution, indicating less overall noise. However, the maximum distance sometimes increased, suggesting that while QOIN can lower average noise, it may also introduce greater variability in outlier runs.

B. RQ2: Property Preservation under Noise

The `AssertEqual` and `AssertDifferent` assertions, which utilize Fisher's exact test to compare the distributions of measurement results in pairs of circuits, were applied as postconditions in 13 of the 15 properties examined across the five case studies. To evaluate whether QOIN's noise mitigation preserves these properties, we analyzed its effect on the Hellinger distances between pairs of circuits (hereafter referred to as the pairwise distance), relative to the Hellinger distances from the circuits' ideal executions, further comparing them to the distances obtained from unmitigated noisy executions. The remaining two properties (GR B, GR C) used `AssertMostFrequent`, check the most frequent output for a given basis; this assertion does not compare pairs of circuits, and thus was excluded from the following figures.

Figures 4 to 7 illustrate the absolute difference in pairwise Hellinger distances between the ideal execution with the QOIN and noisy executions under the `AssertEqual`

TABLE V
EARTH MOVER'S DISTANCE (EMD) BETWEEN THE DISTRIBUTIONS OF HELLINGER DISTANCES COMPUTED FROM NOISY AND QOIN EXECUTIONS AND THOSE FROM THE IDEAL EXECUTION. RESULTS ARE SHOWN SEPARATELY FOR GENERATED AND EQUIVALENT MUTANTS, AND CORRESPOND TO THE GROUPED COMPARISONS SHOWN IN FIGURES 4 TO 7.

Property	Generated		Equivalent	
	Noise EMD	QOIN EMD	Noise EMD	QOIN EMD
QT A	**0.0168**	0.0281	**0.00662**	0.0531
QT B	0.0476	**0.0187**	0.0186	**0.00122**
QT C	**0.0434**	0.0750	0.00780	**0.00376**
QFT A	0.148	**0.0809**	**0.00268**	0.0406
QFT B	**0.0381**	0.0499	**0.0356**	0.0382
QFT C	**0.0895**	0.1300	**0.0268**	0.0753
GR A	0.0659	**0.0298**	0.0636	**0.0394**
DJ A	0.0127	**0.00270**	0.00430	**0.00114**
DJ B	0.107	**0.00212**	0.128	**0.000**
DJ C	0.0733	**0.00210**	0.0206	**0.000**
QPE A	**0.0254**	0.0366	**0.0253**	0.0367
QPE B	0.0792	**0.0291**	0.0294	**0.00414**
QPE C	0.0475	**0.0172**	0.0627	**0.0196**

and `AssertDifferent` properties. Smaller values indicate closer alignment with the ideal execution.

Although the figures visualize Hellinger distance differences between circuit pairs, the data is grouped by which execution, QOIN or noisy, achieved a lower Earth Mover's Distance (EMD; see Table V) to the ideal distribution. Specifically, Figures 4 and 5 show the generated mutants, while Figures 6 and 7 present the equivalent mutants. In both cases, Figures 4 and 6 depict properties where QOIN was closer to the ideal, and Figures 5 and 7 correspond to properties where the noisy execution was closer. Smaller Hellinger distance values indicate a closer alignment with the ideal execution for each pair of circuits. While the figures plot Hellinger distances on the axes, we discuss the results in terms of the Earth Mover's Distance, which quantifies how similar the distributions of these Hellinger distances are.

The Observed trends indicate that the effectiveness of QOIN is heavily influenced by the property under test, as different properties of the same quantum program exhibit very different mitigation results. For example, in Quantum Teleportation, the QOIN Hellinger distance distribution aligns more closely with the ideal for property QT B. However, for QT A and QT C, the noisy (unmitigated) execution is closer to the ideal than QOIN. Across all 13 properties, QOIN improved alignment with the ideal in 8 cases, regardless of mutation type.

For *generated mutants*, QOIN outperformed the noisy execution in QT B, achieving an earth mover's distance (EMD) of 0.019 compared to 0.048 for the noisy execution (values from Table V; visualized in Figure 4). Similarly, QOIN improved results for QFT A (0.081 vs. 0.15), GR A (0.030 vs. 0.066), QPE B (0.029 vs. 0.079), QPE C (0.017 vs. 0.047), and all DJ properties: A (0.0027 vs. 0.013), B (0.0021 vs. 0.11), and C (0.0021 vs. 0.073).

Conversely, the noisy execution performed better in five properties, specifically QT A (0.017 vs. 0.028), QT C (0.043 vs. 0.075), QFT B (0.038 vs. 0.050), QFT C (0.090 vs. 0.13), and QPE A (0.025 vs. 0.037) (Figure 5).

For the *equivalent mutants*, QOIN's distribution of Hellinger distances was closer to ideal than the noisy executions' distribution in several cases (Figure 6). Specifically, it reduced the earth mover's distance compared to the noisy execution for QT B (0.0066 vs. 0.053), QT C (0.0038 vs. 0.0078), GR A (0.039 vs. 0.064), QPE B (0.0041 vs. 0.029), and QPE C (0.020 vs. 0.063). Similarly, all Deutsch-Jozsa (DJ) properties benefited from QOIN mitigation, with properties A (0.0011 vs. 0.0043), B (0.00 vs. 0.13), and C (0.00 vs. 0.021) showing strong improvements.

Conversely, the noisy execution outperformed QOIN for QT A (0.0066 vs. 0.053) and QFT properties, including QFT A (0.0027 vs. 0.041), QFT B (0.036 vs. 0.038), QFT C (0.027 vs. 0.075), and QPE A (0.025 vs. 0.037) (Figure 7).

Overall, the performance trends for equivalent mutants mirrored those of the generated mutants. If QOIN was effective in reducing noise for a particular property in the generated mutant set, it remained effective for the equivalent mutants as well. However, two exceptions were observed: QT C, which was closer to the ideal execution under QOIN in this set, and QFT A, where the noisy execution without mitigation performed better than QOIN. These results suggest that QOIN's effectiveness is not significantly affected by the semantic equivalence of mutants but may be influenced by the underlying circuit structure and the type of property being evaluated.

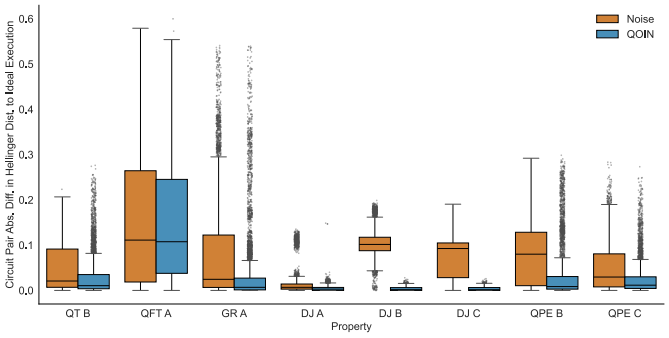

Fig. 4. Absolute difference in Hellinger distance between circuit pairs, for *generated mutants* where the **QOIN execution** achieved a lower Earth Mover's Distance (EMD) to the ideal.

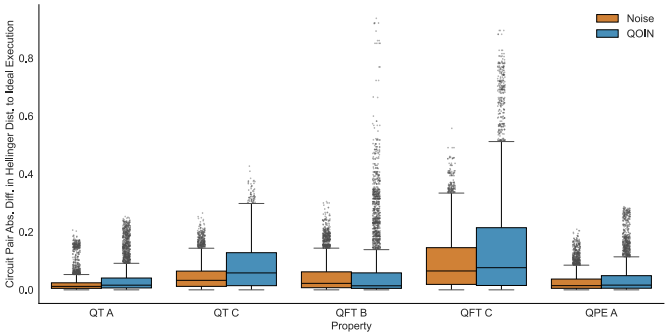

Fig. 5. Absolute difference in Hellinger distance between circuit pairs, for *generated mutants* where the **noisy execution** achieved a lower Earth Mover's Distance (EMD) to the ideal.

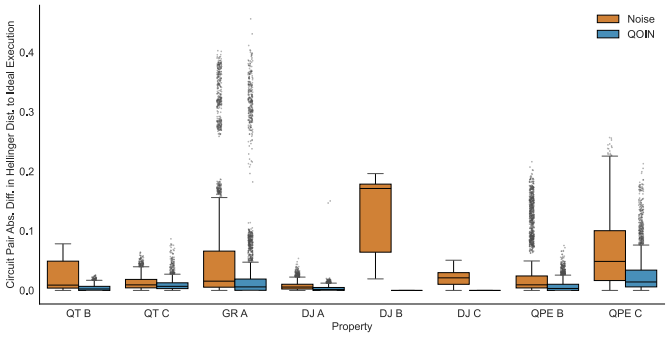

Fig. 6. Absolute difference in Hellinger distance between circuit pairs, for *equivalent mutants* where the **QOIN execution** achieved a lower Earth Mover's Distance (EMD) to the ideal.

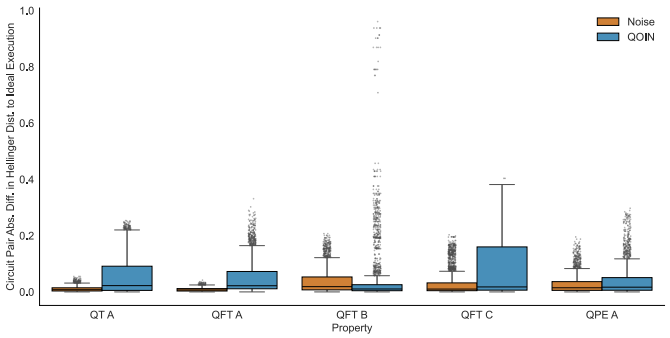

Fig. 7. Absolute difference in Hellinger distance between circuit pairs, for *equivalent mutants* where the **noisy execution** achieved a lower Earth Mover's Distance (EMD) to the ideal.

To provide a more comprehensive view of the variability and extremities in pairwise Hellinger distances, we report the 1% and 0.1% highest and lowest values for each test scenario in Tables VI and VII. These metrics allow us to examine the range of deviations observed in the measurement distributions between pairs of circuits, which may disproportionately skew statistical test outcomes. By analysing these extremes, we assess whether the QOIN distribution is able to approach the ideal distribution, and provide an improvement over performing no mitigation.

Across *generated mutants* (Table VI), QOIN demonstrated closer upper-bound distances to ideal (evaluating both 1% and 0.1% high values) than the noisy execution in 20 out of 26 cases, whereas noisy was closer in 6 out of 26 cases, and tied in no cases. This total of 26 cases arises from assessing 13 distinct properties at two threshold levels (1% and 0.1%). As for the lower bound cases (1% and 0.1% low values), QOIN demonstrated closer upper-bound distances to ideal than the noisy execution in 16 out of 26 cases, whereas noisy was closer in 2 of 26 cases, and tied in 8 of 26 cases.

In particular, under QOIN mitigation, all Deutsch-Jozsa's properties were very close across the high and low extremes (dist. < 0.006 to ideal). However, in a few cases, such as QFT A and QPE A, the noisy execution was comparable to or was closer to ideal than the QOIN execution.

Across *equivalent mutants* (Table VII), the QOIN and noisy execution each produced distances closer to the ideal in 13 out of 26 instances for the upper-bound deviations (1% and 0.1% high values). However, for the lower-bound deviations (1% and 0.1% low values), QOIN demonstrated closer distances to the ideal in 18 out of 26 cases, while the noisy execution was never closer, and ties occurred in 8 cases.

A notable trend emerged in QOIN's behavior: extreme values were very close to the ideal (within a distance of 0.01) when evaluating invariants such as lower register minus (DJ A, GR A) and lower register plus (QT B). An exception was observed in the equivalent mutant execution of GR A, which exhibited a substantial deviation from the ideal.

The shift from equivalent to generated mutants had minimal impact on the overall trends in the central tendency graphs (Figures 4 to 7), affecting only two properties. However, this shift had a much more pronounced effect on the extremities of the pairwise Hellinger distance distribution (Tables VI and VII). A clear example is the QOIN execution of GR A, where the results changed from being identical to ideal in Table VI, to a difference of at least 0.34 across the 1% and 0.1% highs in Table VII.

RQ2: QOIN preserves many quantum properties but not all. While it generally reduces the overall distance to the ideal execution, these improvements do not consistently extend to pairwise similarity in central tendencies or distribution extremes. In some cases, the unmitigated noisy approach better preserves certain properties, suggesting that a hybrid strategy which combines QOIN-mitigated and noisy executions may be more robust for property-based testing.

C. RQ3: Effectiveness of the hybrid approach

In Section IV-A, we observed discrepancies between the QOIN and noisy simulator executions. While QOIN sometimes outperformed the unmitigated noisy execution, there were instances where the latter better preserved properties. To leverage the strengths of both scenarios, we applied a hybrid strategy that, as will be shown in this section, outperforms both noisy execution and QOIN mitigation individually. The hybrid approach first executes circuits using the noisy simulator. If an assertion fails, the circuit is executed again with QOIN mitigation. If the assertion still fails, it is considered a true failure; otherwise, it is classified as passing.

Figure 8 shows the *absolute difference* in the number of failed assertions compared to the ideal execution for the generated mutants, while Figure 9 shows the net difference for the equivalent mutants. This distinction is necessary because, for the equivalent mutant set, running with a noisy simulator only increased (or tied) the number of failed assertions. Whereas for some cases in the generated set, we surprisingly observed more assertion failures under the ideal execution, indicating

TABLE VI
GENERATED MUTANTS' PAIRWISE HELLINGER DISTANCE: 0.1% LOW, 1% LOW, 1% HIGH, 0.1% HIGH

Property	Qoin				Ideal				Noise			
	0.1% low	1% low	1% high	0.1% high	0.1% low	1% low	1% high	0.1% high	0.1% low	1% low	1% high	0.1% high
QT A	**0.0**	0.0	**0.859**	**1.0**	0.0	0.000275	0.783	0.94	**0.0**	**0.000476**	0.671	0.766
QT B	**0.0**	**0.0**	**1.0**	**1.0**	0.0	0.0	1.0	1.0	0.00331	0.00546	0.811	0.819
QT C	**0.0**	**0.00173**	1.0	**1.0**	0.000239	0.00287	0.829	0.943	0.0105	0.0144	**0.662**	0.744
QFT A	0.00217	**0.00726**	0.86	0.937	0.00362	0.00754	0.573	0.636	**0.00398**	0.0101	**0.397**	**0.504**
QFT B	**0.0**	**0.0**	**1.0**	1.0	0.0	0.0	0.881	0.924	0.00167	0.00614	0.698	**0.857**
QFT C	**0.0**	**0.0**	**1.0**	**1.0**	0.0	0.0	0.882	0.988	0.00354	0.00735	0.609	0.767
GR A	**0.0**	**0.0**	**1.0**	**1.0**	0.0	0.0	1.0	1.0	**0.0**	0.000471	0.786	0.797
DJ A	**0.0**	**0.0**	**0.781**	**0.787**	0.0	0.0	0.789	0.797	**0.0**	**0.0**	0.68	0.686
DJ B	**0.0**	**0.0**	**1.0**	**1.0**	0.0	0.0	1.0	1.0	0.000788	0.00612	0.966	0.99
DJ C	**0.0**	**0.0**	**1.0**	**1.0**	0.0	0.0	1.0	1.0	**0.0**	0.00161	0.998	0.999
QPE A	**0.0**	**0.0**	0.269	0.285	0.0	0.0	0.0237	0.03	**0.0**	0.000257	**0.16**	**0.2**
QPE B	**0.0**	**0.0**	**1.0**	**1.0**	0.0	0.0	1.0	1.0	**0.0**	0.000472	0.768	0.9
QPE C	**0.0**	**0.0**	**0.545**	**0.62**	0.0	0.0	0.513	0.542	**0.0**	0.0006	0.361	0.386

TABLE VII
EQUIVALENT MUTANTS' PAIRWISE HELLINGER DISTANCE: 0.1% LOW, 1% LOW, 1% HIGH, 0.1% HIGH

Property	Qoin				Ideal				Noise			
	0.1% low	1% low	1% high	0.1% high	0.1% low	1% low	1% high	0.1% high	0.1% low	1% low	1% high	0.1% high
QT A	**0.0**	**0.0**	0.242	0.255	0.0	0.0	0.0233	0.0282	**0.0**	0.000237	**0.048**	**0.057**
QT B	**0.0**	**0.0**	**0.0238**	**0.0289**	0.0	0.0	0.0278	0.0339	**0.0**	0.00172	0.0714	0.077
QT C	**0.0**	**0.0**	0.0727	0.0993	0.0	0.00024	0.0388	0.0427	0.00806	0.0112	**0.0546**	**0.0672**
QFT A	**0.00124**	**0.0053**	0.277	0.321	0.0023	0.00518	0.0622	0.0683	0.00426	0.00727	**0.0621**	**0.0675**
QFT B	**0.0**	**0.0**	0.461	0.989	0.0	0.0	0.0599	0.0691	0.00467	0.00498	**0.204**	**0.232**
QFT C	**0.0**	**0.0**	0.399	0.416	0.0	0.0	0.0632	0.0672	0.00293	0.00621	**0.17**	**0.196**
GR A	**0.0**	**0.0**	**0.37**	0.404	0.0	0.0	0.0222	0.0264	**0.0**	0.000471	0.38	**0.396**
DJ A	**0.0**	**0.0**	**0.013**	**0.0172**	0.0	0.0	0.0184	0.0229	**0.0**	**0.0**	0.0316	0.0438
DJ B	**1.0**	**1.0**	**1.0**	**1.0**	1.0	1.0	1.0	1.0	0.804	0.809	0.968	0.975
DJ C	**0.0**	**0.0**	**0.0**	**0.0**	0.0	0.0	0.0	0.0	**0.0**	**0.0**	0.046	0.0505
QPE A	**0.0**	**0.0**	0.262	0.286	0.0	0.0	0.0232	0.0283	**0.0**	0.000361	**0.153**	**0.187**
QPE B	**0.0**	**0.0**	**0.0642**	**0.0841**	0.0	0.0	0.0347	0.0437	0.0	0.000471	0.179	0.202
QPE C	**0.0**	**0.000707**	**0.571**	**0.592**	0.0	0.000236	0.526	0.544	0.0	0.000943	0.365	0.379

that noise may mask the effect of circuit mutations, causing them to erroneously pass. Different hatching patterns are used to indicate the specific properties associated with each failed assertion. In both figures, the X-axis plots individual mutants, grouped by testing scenario, and the Y-axis shows the absolute difference in the number of failed assertions relative to the ideal execution. Lower values indicate closer alignment with the ideal simulator.

For the generated mutants (Figure 8), we can see the impact of the variable effectiveness of QOIN (blue) as shown in Section IV-B, with properties failing at different rates depending on the individual mutant and execution scenario. The hybrid approach (purple) performed similarly to the better execution between QOIN and noisy for each mutant. However, there were notable improvements with the hybrid approach in: QT mutants 8 and 9, QFT mutants 4 and 9, and QPE 2 and 8, where it provided a significant advantage.

Figure 9 further highlights the rationale for the hybrid approach. While some properties align more closely with ideal under the noisy execution, others align more closely with the QOIN execution. By first running the noisy simulation and then applying QOIN mitigation only for failed assertions, the hybrid approach can achieve a closer alignment with the ideal. In some cases, it even outperforms both the noisy and QOIN executions when used individually. This is particularly visible in the equivalent mutants of the QT algorithm. With the QOIN execution, two properties failed (QT B and QT C), whereas in the unmitigated noisy execution, QT A and QT B failed. Since QT A does not fail when testing the property with QOIN, it is filtered out when re-tested under QOIN by the hybrid approach. Similarly, since QT C did not fail in the initial noisy execution, it is also eliminated from the hybrid output. Finally, the shared failing property (QT B) showed improvement because of fewer assertions failing when repeated with QOIN mitigation.

RQ3: The hybrid approach demonstrated the ability to improve alignment with the ideal execution for the equivalent mutants, often outperforming both the QOIN and noisy individual scenarios. Notably, it significantly reduced false positives during property-based testing, particularly in the Quantum Teleportation, Grover's Algorithm, and Quantum Phase Estimation algorithms. A comparable, but lesser effect was seen in the generated mutant set, where the hybrid approach performed similarly to the better option between the QOIN and noisy executions. However, it less frequently significantly outperformed both individual approaches, although still showcasing exceptions with notable improvements over both individual strategies.

V. THREATS TO VALIDITY

There are various threats to the validity of this study that may affect the generalisability of the results:

- **Quantum program subset:** The quantum programs used as case studies may not be representative of the wider set of all quantum programs. To improve coverage, future work could include a broader and more diverse set of

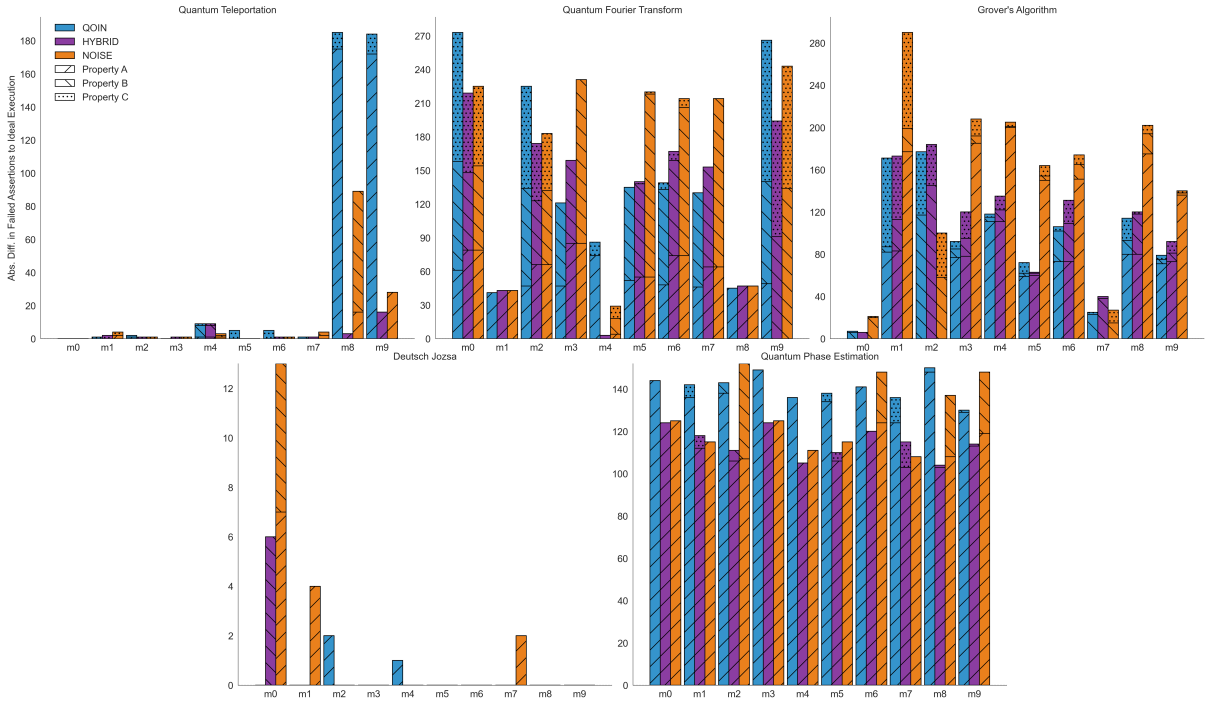

Fig. 8. Absolute Difference in Failed assertions relative to the Ideal execution's Number of Failed Assertions for *generated mutants*

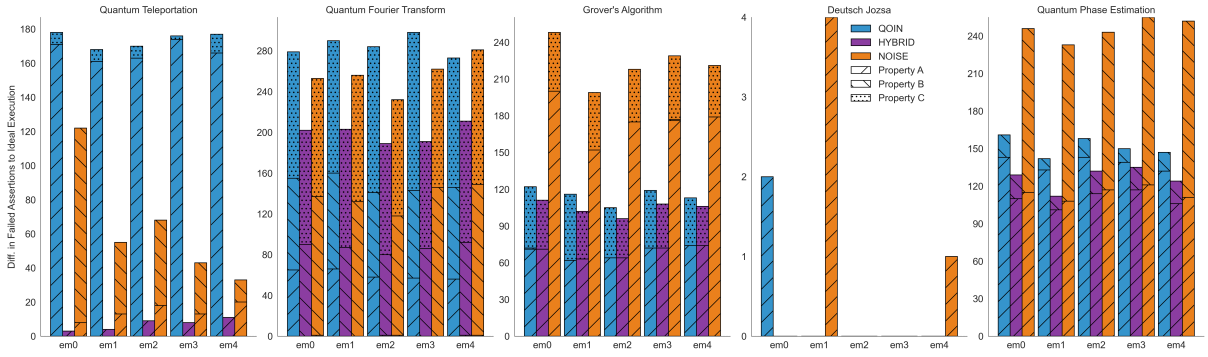

Fig. 9. Difference in Failed assertions relative to the Ideal execution's Number of Failed Assertions for *equivalent mutants*

quantum programs with further varying circuit structures or qualities.

- **Program size:** Certain programs, such as Grover's algorithm, do not have a fixed size and thus require an oracle as input. Because our experiments were conducted using a simulator, the program width and depth were limited to keep the circuits feasible. Scaling these programs up will help determine whether the observed trends hold at larger scales and potentially reveal new insights that smaller examples may not capture.
- **Noise model selection:** Quantum programs may perform differently according to the noise conditions that they are exposed to. Only one noise model was used for all our experiments. Testing multiple noise models, each with different levels or types of noise, could help determine how consistently the observed trends hold under different realistic conditions.
- **Use of simulator:** A noisy simulator was used due to limited access to a real quantum computer, which may not fully capture the behavior of an actual computer. This study used a static noise snapshot to train the QOIN model. However, real quantum computers exhibit noise fluctuations, so applying QOIN in such settings would likely require periodic retraining to maintain its performance. Assessing the sensitivity of the model to these fluctuations in noise is an important direction for future work. Repeating these experiments on physical quantum computers would provide a more accurate picture of how the tested programs behave under real-world conditions, and how this affects property-based testing.
- **Property subset:** A limited set of three properties per quantum program were evaluated, these may not

represent the entire space of possible property-based tests. Evaluating additional properties would help determine whether the observed results are valid for a wider variety of property-based tests.

VI. Conclusion

This work examined whether property-based testing can be adapted to noisy quantum environments through machine-learning-based noise mitigation. By evaluating five base quantum programs under mutation testing and employing the QOIN approach, we found QOIN is able to reduce median and mean noise on a per-circuit basis, though with mixed results when evaluating pairs of circuits that are compared within assertions. These results inspired a hybrid approach, which selectively applies QOIN when a property fails during property-based testing. The hybrid approach was found to significantly reduce the false positive rate introduced by the noise model compared to each individual approach.

While the hybrid strategy takes a step forward in bridging the gap between ideal and real-world conditions, there is still a noticeable gap in assertion failures compared to the ideal execution when the properties are evaluated under noisy conditions. This work serves as a stepping stone for integrating targeted noise mitigation techniques within property-based testing for quantum programs, underscoring its potential effectiveness in the future with enhanced noise-mitigation techniques. Nevertheless, some limitations remain. We relied on a single simulated noise model and a limited set of quantum programs, potentially limiting the generalizability of our findings. Moreover, the performance overheads of QOIN for higher qubit counts warrant further investigation.

In future work, the properties could be implemented to better interact with machine-learning based noise mitigation. Rather than comparing separate circuits, one circuit could be constructed that contains the qubits to compare, which may mitigate the mixed results observed when comparing pairs of circuits evaluated within property-based tests. Furthermore, exploring different noise mitigation approaches that can be inserted into the property-based testing loop could provide a better perspective on the viability of this approach.

Acknowledgments: Gabriel Pontolillo and Mohammad Reza Mousavi have been partially supported by the UKRI Trustworthy Autonomous Systems Node in Verifiability, Grant Award Reference EP/V026801/2, EPSRC project on Verified Simulation for Large Quantum Systems (VSL-Q), grant reference EP/Y005244/1 and the EPSRC project on Robust and Reliable Quantum Computing (RoaRQ), Investigation 009 Model-based monitoring and calibration of quantum computations (ModeMCQ), grant reference EP/W032635/1 and ITEA/InnovateUK projects GENIUS and GreenCode. Asmar Muqeet and Shaukat Ali have been supported by the Qu-Test project (Project #299827) funded by the Research Council of Norway.

References

[1] G. Jaeger, "Classical and quantum computing," *Quantum Information: An Overview*, pp. 203–217, 2007.

[2] J. M. Murillo, J. Garcia-Alonso, E. Moguel, J. Barzen, F. Leymann, S. Ali, T. Yue, P. Arcaini, R. Pérez-Castillo, I. García Rodríguez de Guzmán, M. Piattini, A. Ruiz-Cortés, A. Brogi, J. Zhao, A. Miranskyy, and M. Wimmer, "Quantum software engineering: Roadmap and challenges ahead," *ACM Trans. Softw. Eng. Methodol.*, Jan. 2025. [Online]. Available: https://doi.org/10.1145/3712002

[3] R. Alicki, "Decoherence and the Appearance of a Classical World in Quantum Theory," *Journal of Physics A: Mathematical and General*, vol. 37, no. 5, p. 1948, feb 2004.

[4] X. Wang, P. Arcaini, T. Yue, and S. Ali, "Application of Combinatorial Testing to Quantum Programs," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, 2021, pp. 179–188.

[5] S. Ali, P. Arcaini, X. Wang, and T. Yue, "Assessing the Effectiveness of Input and Output Coverage Criteria for Testing Quantum Programs," in *2021 IEEE 14th International Conference on Software Testing, Validation and Verification (ICST)*, 2021, pp. 13–23.

[6] P. Long and J. Zhao, "Testing multi-subroutine quantum programs: From unit testing to integration testing," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 6, jun 2024. [Online]. Available: https://doi.org/10.1145/3656339

[7] J. Ye, S. Xia, F. Zhang, P. Arcaini, L. Ma, J. Zhao, and F. Ishikawa, "QuraTest: Integrating quantum specific features in quantum program testing," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2023, pp. 1149–1161.

[8] X. Wang, P. Arcaini, T. Yue, and S. Ali, "QuSBT: Search-based testing of quantum programs," in *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*. New York, NY, USA: Association for Computing Machinery, 2022, pp. 173–177.

[9] X. Wang, T. Yu, P. Arcaini, T. Yue, and S. Ali, "Mutation-Based Test Generation for Quantum Programs with Multi-Objective Search," in *Proceedings of the Genetic and Evolutionary Computation Conference*. New York, NY, USA: Association for Computing Machinery, 2022, pp. 1345–1353.

[10] D. Fortunato, J. Campos, and R. Abreu, "Mutation Testing of Quantum Programs: A Case Study With Qiskit," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–17, 2022.

[11] E. Mendiluze, S. Ali, P. Arcaini, and T. Yue, "Muskit: A Mutation Analysis Tool for Quantum Software Testing," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2021, pp. 1266–1270.

[12] R. Abreu, J. P. Fernandes, L. Llana, and G. Tavares, "Metamorphic Testing of Oracle Quantum Programs," in *Proceedings of the 3rd International Workshop on Quantum Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2023, pp. 16–23.

[13] M. Paltenghi and M. Pradel, "MorphQ: Metamorphic testing of the Qiskit quantum computing platform," in *Proceedings of the 45th International Conference on Software Engineering*. IEEE Press, 2023, pp. 2413–2424.

[14] S. Honarvar, M. R. Mousavi, and R. Nagarajan, "Property-based testing of quantum programs in q#," in *ICSE '20: 42nd International Conference on Software Engineering, Workshops, Seoul, Republic of Korea, 27 June - 19 July, 2020*. ACM, 2020, pp. 430–435. [Online]. Available: https://doi.org/10.1145/3387940.3391459

[15] G. Pontolillo, M. R. Mousavi, and M. Grzesiuk, "Qucheck: A property-based testing framework for quantum programs in qiskit," 2025. [Online]. Available: https://arxiv.org/abs/2503.22641

[16] G. J. Pontolillo and M. R. Mousavi, "Delta debugging for property-based regression testing of quantum programs," in *Proceedings of the 5th ACM/IEEE International Workshop on Quantum Software Engineering, Q-SE 2024, Lisbon, Portugal, 16 April 2024*. ACM, 2024, pp. 1–8. [Online]. Available: https://doi.org/10.1145/3643667.3648219

[17] A. Muqeet, T. Yue, S. Ali, and P. Arcaini, "Mitigating noise in quantum software testing using machine learning," *IEEE Transactions on Software Engineering*, vol. 50, no. 11, pp. 2947–2961, 2024.

[18] D. P. DiVincenzo, "Quantum gates and circuits," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1969, pp. 261–276, 1998.

[19] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with Qiskit," 2024.

[20] S. Resch and U. R. Karpuzcu, "Benchmarking Quantum Computers and the Impact of Quantum Noise," *ACM Comput. Surv.*, vol. 54, no. 7, jul 2021.

[21] T. Ayral, F.-M. L. Régent, Z. Saleem, Y. Alexeev, and M. Suchara, "Quantum divide and compute: exploring the effect of different noise sources," *SN Computer Science*, vol. 2, no. 3, p. 132, 2021.

[22] Y. Huang and M. Martonosi, "Statistical assertions for validating patterns and finding bugs in quantum programs," in *Proceedings of the 46th International Symposium on Computer Architecture*, ser. ISCA '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 541–553. [Online]. Available: https://doi.org/10.1145/3307650.3322213

[23] D. Rovara, L. Burgholzer, and R. Wille, "A framework for the efficient evaluation of runtime assertions on quantum computers," 2025. [Online]. Available: https://arxiv.org/abs/2505.03885

[24] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, "Error mitigation with Clifford quantum-circuit data," *Quantum*, vol. 5, p. 592, Nov. 2021. [Online]. Available: https://doi.org/10.22331/q-2021-11-26-592

[25] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin, and Y. Li, "Learning-based quantum error mitigation," *PRX Quantum*, vol. 2, p. 040330, Nov 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PRXQuantum.2.040330

[26] H. Liao, D. S. Wang, I. Sitdikov, C. Salcedo, A. Seif, and Z. K. Minev, "Machine learning for practical quantum error mitigation," 2023.

[27] T. Patel and D. Tiwari, "Qraft: Reverse your quantum circuit and know the correct program output," in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 443–455. [Online]. Available: https://doi.org/10.1145/3445814.3446743

[28] A. Muqeet, S. Ali, T. Yue, and P. Arcaini, "A machine learning-based error mitigation approach for reliable software development on ibm's quantum computers," in *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, ser. FSE 2024. New York, NY, USA: Association for Computing Machinery, 2024, p. 80–91. [Online]. Available: https://doi.org/10.1145/3663529.3663830

[29] G. J. Pontolillo, A. Muqeet, S. Ali, and M. R. Mousavi, "Qucheck with qoin," 4 2025. [Online]. Available: https://figshare.com/articles/software/QuCheck_with_QOIN/28772231

[30] M. AbuGhanem, "Ibm quantum computers: Evolution, performance, and future directions," 2024. [Online]. Available: https://arxiv.org/abs/2410.00916

[31] D. Fortunato, J. Campos, and R. Abreu, "Qmutpy: a mutation testing tool for quantum algorithms and applications in qiskit," in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, ser. ISSTA 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 797–800. [Online]. Available: https://doi.org/10.1145/3533767.3543296

[32] S. Dasgupta and T. S. Humble, "Characterizing the reproducibility of noisy quantum circuits," *Entropy*, vol. 24, no. 2, 2022.

[33] R. Harper, S. T. Flammia, and J. J. Wallman, "Efficient learning of quantum noise," *Nature Physics*, vol. 16, no. 12, pp. 1184–1188, aug 2020.