NExt ApplicationS of Quantum Computing

<NE|AS|QC>

D3.6: Report on betatester results and quantum noise modelling (TNBS)

Document Properties

Contract Number	951821
Contractual Deadline	31/10/2024
Dissemination Level	Public
Nature	Report
Edited by :	Eviden Quantum Iberia
Authors	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN
Reviewers	Diego Andrade, UDC Andrés Gómez, CESGA
Date	31/10/2024
Keywords	Benchmark, Quantum computers, kernels; Quantum Noise, Qaptiva
Status	Final
Release	3.0

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951821

History of Changes

Release	Date	Author, Organization	Description of Changes
1.0	10/09/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	First version
2.0	28/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	Keywords
2.0	28/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	Executive Summary
2.0	28/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	Restructuration of the Benchmark cases
2.0	28/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	Capabilities of Qaptiva emulator
2.0	28/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	Restructuring of the description and conclusions of phase I
2.0	28/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	Modification of the results plot
2.0	28/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	Restructuring of the implemented methodology.
2.0	28/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	Restructuring of the description and conclusions of Phase II
3.0	30/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	Restructuration of the concluding remarks
3.0	31/10/2024	Andrés Bravo Montes, EVIDEN Miriam Bastante Chichón, EVIDEN	General review of the report

Table of Contents

List of Figures

List of tables

Executive Summary

This report presents the results of *The NEASQC Benchmark Suite* (TNBS) testing phase and the noisy emulation of some of the benchmark cases included in the TNBS. The main objective of this study is to evaluate the performance of Quantum Computing (QC) platforms using benchmark cases defined from the NEASQC project use-cases.

The report is divided into two main phases:

- **Phase I: Benchmark Validation.** In this phase, the benchmark cases based on the NEASQC project documentation were replicated, and their performance was validated. The main benchmark cases evaluated included:
    - **Probability Loading Algorithms:** Assessing algorithms' capability to map classical probability distributions onto quantum states for use in various quantum applications.
    - **Amplitude Estimation Algorithms:** Focused on estimating integrals using quantum circuits.
    - **Phase Estimation Algorithms:** It enables the determination of the phases of a quantum state, allowing for the calculation of eigenvalues of a specific unitary operator applied to that state.
    - **Parent Hamiltonian Benchmark:** This evaluated the system's energy to find the ground state using variational algorithms.
- **Phase II: Noise Implementation and Analysis.** In this phase, noise was introduced into the emulation of the benchmark execution, and its impact was studied using different models, such as Amplitude Damping, Pure Dephasing, Idle noise, Depolarizing Channel, and Depolarising Channel + Idle noise. The experiments used parameters from superconducting qubit technologies from IBM and trapped-ion qubits from Quantinuum. The report analyzed how noise affects the performance and the reliability of quantum algorithms, focusing on error metrics such as KS, KL, Energy and execution times.

The results of these phases demonstrated that TNBS can be effectively used to compare different quantum computers with varying architectures. Additionally, the knowledge gained about the impact of noise on this set of benchmarks will contribute to a better understanding of the robustness of quantum systems under real conditions, which is crucial for the advancement of quantum technologies.

This report concludes that TNBS provides valuable tools for evaluating the performance of QPUs, laying the foundation for future quantum applications and hardware optimizations.

1. Overview

The NEASQC Benchmark Suite (TNBS) is a subtask of task T3.3 of WP3 of NEASQC project. It aims to evaluate the performance of Quantum Computing (QC) platforms using benchmark cases defined from the NEASQC project use-cases. To achieve this, it uses accuracy metrics that are defined per-case, along with execution time measurements for both quantum and classical computations The main characteristics of TNBS are:

- It is application-oriented, meaning that the benchmark cases are drawn from various Quantum Computing use-cases. These benchmarks are defined at the mathematical level, allowing each case to be implemented using different algorithmic approaches and quantum programming tools or languages.
- It evaluates all the components of the stack simultaneously.
- Each case can be defined for different numbers of qubits.
- The accuracy metrics vary for each benchmark, while performance metrics remain constant across use-cases.
- The precision of the quantum implementation can be verified through classical computations.

Each benchmark case consists of 3 components:

1. **Kernel**: A high-level quantum subroutine, common to various algorithms and that can be implemented using different algorithmic approaches.
2. **Test Case**: A problem whose solution involves using the Kernel at least once. The result can be verified analytically or through classical methods.
3. **Metrics**: A parameter used to evaluate the performance of a quantum architecture in each benchmark case.

TNBS allows the evaluation of different components of the QC stack specifically:

- Algorithms.
- Programming languages/libraries.
- Compilers.
- Transpilers.
- Assemblers and transpilers.
- Underlying quantum hardware or emulator.

1.1. Objectives

The aim of this task is to verify the adequacy and completeness of the TNBS documentation. This is done by implementing all the benchmark cases using exclusively this documentation as a reference. The general objectives are as follows:

1. **Phase I**: In this phase, a review of the official TNBS documentation was conducted. Based on this document, all benchmark the cases have been replicated for subsequent validation.
    - **Objective 1**: Review of documentation.
    - **Objective 2**: Replicate each benchmark case using the project documentation.
    - **Objective 3**: Compare the obtained results with the project's results.
    - **Objective 4**: Document the results.
2. **Phase II**: In this phase, noise has been introduced into the emulation of the benchmark cases. Additionally, the impact of noise was evaluated based on the technology used, analyzing the obtained metrics.
    - **Objective 1**: Noise implementation.
    - **Objective 2**: Review the impact of noise on the results.
    - **Objective 3**: Document the results.

1.2. Benchmark cases

This section gives a brief description of the TNBS benchmarks cases considered in this study.

1.2.1. Benchmark for Probability Loading Algorithms

The Probability Loading (PL) kernel maps a classical probability distribution, whether discrete or continuous, onto a quantum state. This process involves preparing a quantum state in superposition, where each basis state has an amplitude proportional to the square root of the corresponding probabilities. The encoding of this distribution in a quantum circuit is used in quantum algorithms such as HHL (Harrow–Hassidim–Lloyd algorithm), quantum PCA (Principal Component Analysis), or amplitude estimation, as the correct preparation of the initial state is an important step for the performance and accuracy of these process. Since this process is often computationally costly, having an efficient implementation is key to leveraging the power of quantum systems in problems involving large data volumes or complex distributions.

The test case starts by generating a random normal distribution, from which arrays of values are created and subsequently normalized. A specific algorithm creates a quantum operator to load the probabilities and apply them to an initial quantum state. Among the available methods are brute force, multiplexer, and **KPTree**, with the latter being the one used in our implementation. The resulting quantum state is measured and compared to the ground-truth distribution using metrics such as **Kolmogorov-Smirnov (KS)** and **Kullback-Leibler (KL)** divergence. Additionally, the **elapsed time** is measured, and when possible, the **quantum time** dedicated specifically to the quantum part.

1.2.2. Benchmark for Amplitude Estimation Algorithms

The Amplitude Estimation (AE) kernel enables the estimation of the amplitude of a specific subset of basis states in a quantum superposition, generated by a unitary operator A applied to an initial state of n qubits. There are three main approaches to solving the amplitude estimation problem: the classical Monte Carlo solution, which scales as $1/\sqrt{N}$ with respect to the number of measurements; the canonical quantum solution with Quantum Phase Estimation (QPE), which offers a quadratic improvement, scaling as $1/N$ with the number of oracle calls; and the amplitude estimation algorithms without QPE, which provide an intermediate performance, being less precise than QPE but more efficient than classical methods. In this implementation, we select the canonical quantum solution with QPE due to its superior precision and efficiency. This technique leverages advanced subroutines like the quantum Fourier transform and Grover's algorithm to significantly enhance the estimation compared to classical methods. The applications of this kernel cover areas such as financial optimization, risk estimation, computational chemistry, and machine learning, in addition to general tasks like numerical integration.

The test case for this kernel is based on calculating the integral of the function $f(x) = \sin x$ over specific intervals. The goal is to use the AE algorithm to efficiently estimate the value of this integral. The main test is performed on the interval $\left[0, \frac{3\pi}{8}\right]$, whose exact integral value is 0.6173, and optionally on the interval $\left[\pi, \frac{5\pi}{4}\right]$. The algorithm must estimate these values and compare them with the exact integrals to evaluate its accuracy.

The metrics used to evaluate the quality of amplitude estimation include the **sum absolute error** between the AE estimator and the Riemann sum, expressed as $\epsilon = \left|\tilde{S}^{I}_{[a,b]} - S_{[a^I,b^I]}\right|$, and the total **number of oracle calls**, which measures the computational efficiency of the solution. In terms of temporal performance, we considered the total benchmark time, which spans from the discretization of the domain to the calculation of metrics; the **execution time**, which focuses specifically on the duration of the amplitude estimation (AE) algorithm; and, when possible, the **quantum time**, which isolates the duration of the purely quantum part of the algorithm.

1.2.3. Benchmark for Phase Estimation Algorithms

The Quantum Phase Estimation (QPE) kernel is designed to determine the phases associated with a specific unitary operator when applied on a quantum state, allowing for the calculation of the eigenvalues of an $n - dimensional$ unitary operator applied to that state. QPE relies on the use of the inverse quantum Fourier transform and phase control, facilitating the efficient extraction of phase information. This kernel is fundamental for a wide variety of applications, including integer factorization through Shor's algorithm, the simulation of quantum systems in chemistry, the solving of linear equation systems, and various quantum optimization problems.

The test case focuses on evaluating the performance of the QPE algorithm in estimating the phase introduced by an rotation operator applied to an state. This process involves preparing an initial state in superposition, applying the controlled version of such an operator, followed by the inverse quantum Fourier transform and measuring the phase register. In this context, a phase estimation routine is implemented specifically designed to calculate the eigenvalues of quantum operators. For these calculations, angles are generated using both exact and random methods, incorporating both approaches into the implementation.

The accuracy evaluation of the kernel uses the following metrics: for randomly generated angles, the **Kolmogorov-Smirnov (KS)** test is applied, while for exact angles **Fidelity** is used. Additionally, performance metrics are obtained, differentiating between **elapsed time**, **quantum time** dedicated specifically to the quantum part of the algorithm, and **classical time** required for the classical processing components of the algorithm.

1.2.4. Benchmark for Parent Hamiltonian

The Parent Hamiltonian (PH) kernel focuses on the construction and evaluation of a Hamiltonian designed to have a specific quantum state as its unique ground state. Its main function is to assess the quality of quantum computers by executing typical ansatz forms without relying on the optimizer used. This approach is particularly relevant for preparing quantum states and evaluating their minimum energy.

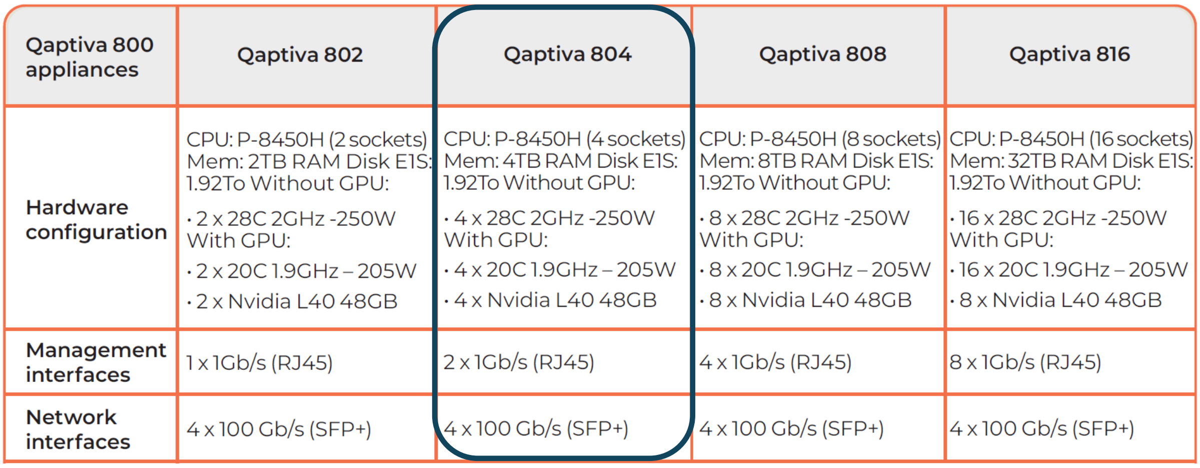
The applications of the Parent Hamiltonian are fundamental in the context of variational algorithms, being especially useful in the Variational Quantum Eigensolver (VQE). This kernel is applied in areas such as quantum chemistry and condensed matter physics, facilitating the simulation of complex quantum systems, the determination of ground states of molecules and materials, as well as optimizing quantum circuits for specific problems. Additionally, it allows for the evaluation of the performance of quantum hardware in tasks associated with the VQE.

The associated test case focuses on implementing a specific ansatz and evaluating its ability to represent the ground state of a given Hamiltonian. The process involves generating a random Hamiltonian with a known spectrum, constructing a parametric ansatz based on this Hamiltonian (in this case, ***ansatz_qlm_01***), and executing the corresponding quantum circuit. The effectiveness of the employed method and the quality of the quantum computation can be determined by comparing the obtained outputs with the theoretical values.

To evaluate the performance of this benchmark case, several metrics are employed. These include the **elapsed time**, the **quantum time** dedicated to the **quantum** part of the algorithm, and the **classical time** for the classical part of the algorithm. The accuracy is assessed by the difference between the **energy** estimated by the quantum circuit and the theoretical ground state value, which is expected to be zero, providing a clear indication of the solution's precision.

1.3. Qaptiva emulator

For executing this evaluation, we have used the EVIDEN Qaptiva 800 quantum emulator, specifically the Qaptiva 804 model. This appliance is an all-in-one solution for the development and optimization of quantum applications, standing out for its powerful combination of hardware and software.



Qaptiva 800 appliances	Qaptiva 802	Qaptiva 804	Qaptiva 808	Qaptiva 816
Hardware configuration	CPU: P-8450H (2 sockets) Mem: 2TB RAM Disk E1S: 1.92To Without GPU: • 2 x 28C 2GHz -250W With GPU: • 2 x 20C 1.9GHz – 205W • 2 x Nvidia L40 48GB	CPU: P-8450H (4 sockets) Mem: 4TB RAM Disk E1S: 1.92To Without GPU: • 4 x 28C 2GHz -250W With GPU: • 4 x 20C 1.9GHz – 205W • 4 x Nvidia L40 48GB	CPU: P-8450H (8 sockets) Mem: 8TB RAM Disk E1S: 1.92To Without GPU: • 8 x 28C 2GHz -250W With GPU: • 8 x 20C 1.9GHz – 205W • 8 x Nvidia L40 48GB	CPU: P-8450H (16 sockets) Mem: 32TB RAM Disk E1S: 1.92To Without GPU: • 16 x 28C 2GHz -250W With GPU: • 16 x 20C 1.9GHz – 205W • 8 x Nvidia L40 48GB
Management interfaces	1 x 1Gb/s (RJ45)	2 x 1Gb/s (RJ45)	4 x 1Gb/s (RJ45)	8 x 1Gb/s (RJ45)
Network interfaces	4 x 100 Gb/s (SFP+)	4 x 100 Gb/s (SFP+)	4 x 100 Gb/s (SFP+)	4 x 100 Gb/s (SFP+)

Figure 1: Technical specifications of the Qaptiva 800 models, highlighting in blue the Qaptiva 804 used in the TNBS implementation.

On the **hardware** side, the Qaptiva 800 offers scalable CPU configurations with up to 16 sockets using P-8450H processors and up to 32 TB of RAM, depending on the model as seen in Figure 1, enabling emulations of up to 41 qubits in quantum circuits. Additionally, the system includes an option for GPU acceleration, supporting up to 8 Nvidia L40 units with 48 GB of memory. It can also emulate noisy systems, adapting noise models to different quantum technologies, such as trapped ions and superconductors.

On the **software** side, Qaptiva 800 provides an agnostic quantum programming model that supports various computing paradigms, including gate-based computing, quantum annealing, and analog computing. Moreover, this environment is compatible with other Python-based frameworks, allowing for the import and export of code in languages like OpenQASM, Perceval, Cirq, Pyquil, ProjectQ, and Qiskit. This flexibility enables users to develop programs for diverse quantum hardware platforms without being constrained by a specific technology, while also simulating real noise conditions. The emulation environment includes several advanced simulators, such as Linalg, Feynman, Stabilizers (stabs), Matrix Product State (MPS), Advanced Matrix Product State (QPEG), Quantum Multi-valued Decision Diagrams, Noisy QPU, Simulated Quantum Annealing, and Analog QPU.

In the TNBS implementation, the **Qaptiva 804** model was used, equipped with a CPU featuring 4 sockets and 4 TB of RAM, achieving emulations of up to 35 qubits. Access to the quantum platform was managed via SSH, facilitating its remote operation in distributed environments.

2. Phase I

In this phase, a review of the document ***"D3.5 NEASQC Benchmark_Suite_v0.2.pdf"*** provided by the official project documentation has been carried out. From this document, all the benchmark cases described in Section 1.2 have been replicated. Subsequently, the results obtained have been verified by comparing them with those obtained by the team responsible for the development of these benchmarks.

2.1. Results

A check has been carried out for each benchmark case to ensure the correct functioning of the produced code and the results obtained have been compared with the validation team.

The key metrics analyzed in these benchmarks will be represented using boxplots. In these plots, the mean will be indicated by a dashed line, the median by a solid line, and outliers will be displayed as empty circles, as detailed in the legend. Each boxplot includes a box representing the interquartile range (IQR), which spans from the first quartile (Q1, or 25th percentile) to the third quartile (Q3, or 75th percentile), indicating the central 50% of the data. The whiskers extending from the box mark the minimum and maximum values within 1.5 times the IQR from Q1 and Q3, respectively. Any point outside this range is considered an outlier.

To evaluate the behavior across different configurations, the results for the 4-qubit and 6-qubit models will be compared in the PL, AE, and QPE benchmarks, while the PH will contrast the results of seven distinct configurations. These comparisons highlight the differences in mean values, dispersion, and the presence of outliers among the configurations, also demonstrating the ability to reliably replicate these benchmarks across various setups.

2.1.1. Benchmark case 1: Probability Loading Algorithms

This test was carried out using configurations of 4 and 6 qubits, within the range of 4 to 8 qubits recommended in the document "D3.5: The NEASQC Benchmark Suite (TNBS)". Configurations with a higher number of qubits were not chosen due to the elevated execution times, caused by the need for between 4000 and 6000 repetitions. For each implementation, two CSV files were generated: one containing all the benchmark data, including key metrics for evaluation, and another storing the average of these metrics. The number of shots used, which totaled 20020, was determined by the reference equation provided in the Probability Loading benchmark documentation.

Figure 2 shows the results of the **Kolmogorov-Smirnov (KS) metric** for a noiseless emulated QPU. KS is a non-parametric statistical test that evaluates the goodness of fit between two probability distributions. A value close to zero indicates a high degree of fit between the compared distributions, while higher values suggest significant differences between them. For **4 qubits**, the KS value is approximately $\mathbf{0.0048}$, with high variability and several outliers above the average. For **6 qubits**, the KS value decreases to around $\mathbf{0.0027}$, with lower variability, though outliers are still present. Despite these fluctuations, both cases exhibit KS values close to zero, indicating that the implementation was successful.

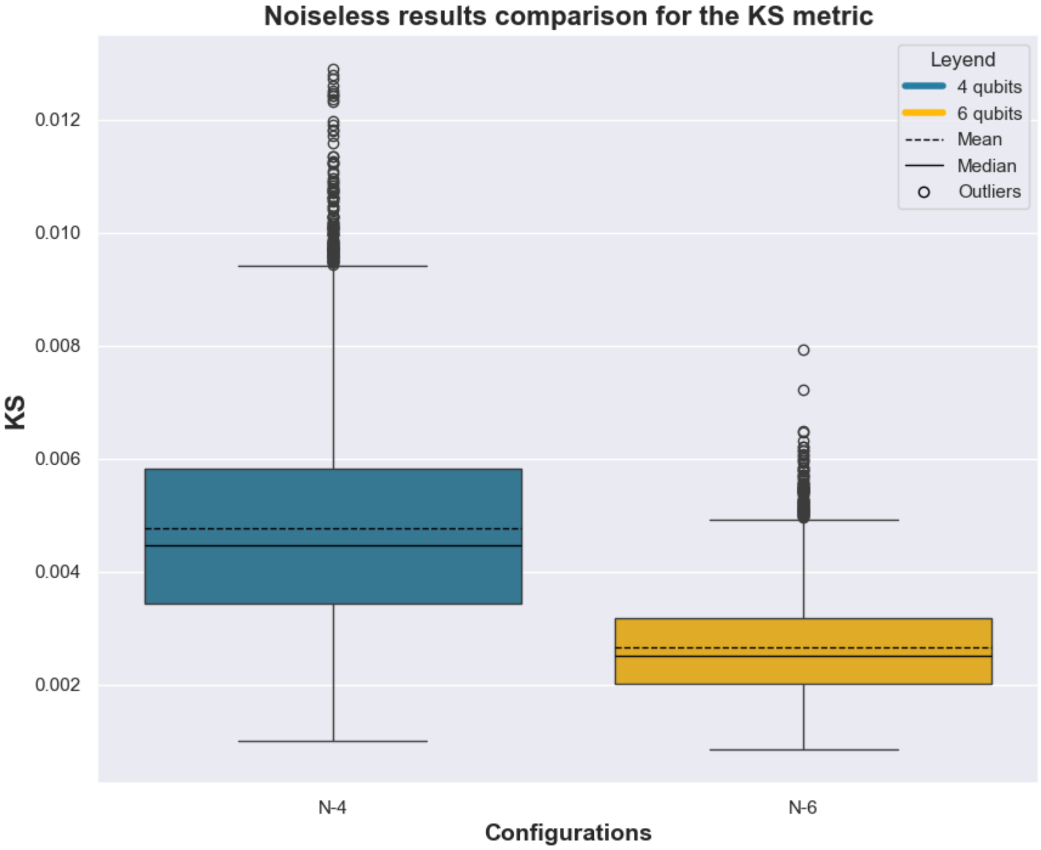


Figure 2: Comparison of the box plot of benchmark case PL noiseless using KS metric.

Figure 3 shows the results of the **Kullback-Leibler (KL) divergence metric** for a noiseless emulated QPU. KL is an asymmetric measure of similarity or difference between two probability distribution functions P and Q. This metric measures the expected number of additional bits required to convert distribution P into distribution Q. Its minimum value is zero, which is only achieved when both distributions are identical. For **4 qubits**, the KL value is approximately $\mathbf{0.00037}$, with high variability and many outliers above the mean. In the **6-qubits** case, the KL value is $\mathbf{0.00038}$, and although the variability is lower compared to the 4-qubit case, several outliers mainly above the average are still observed. Despite these variations, both cases have KL values close to 0, indicating that the implementation has been done correctly.

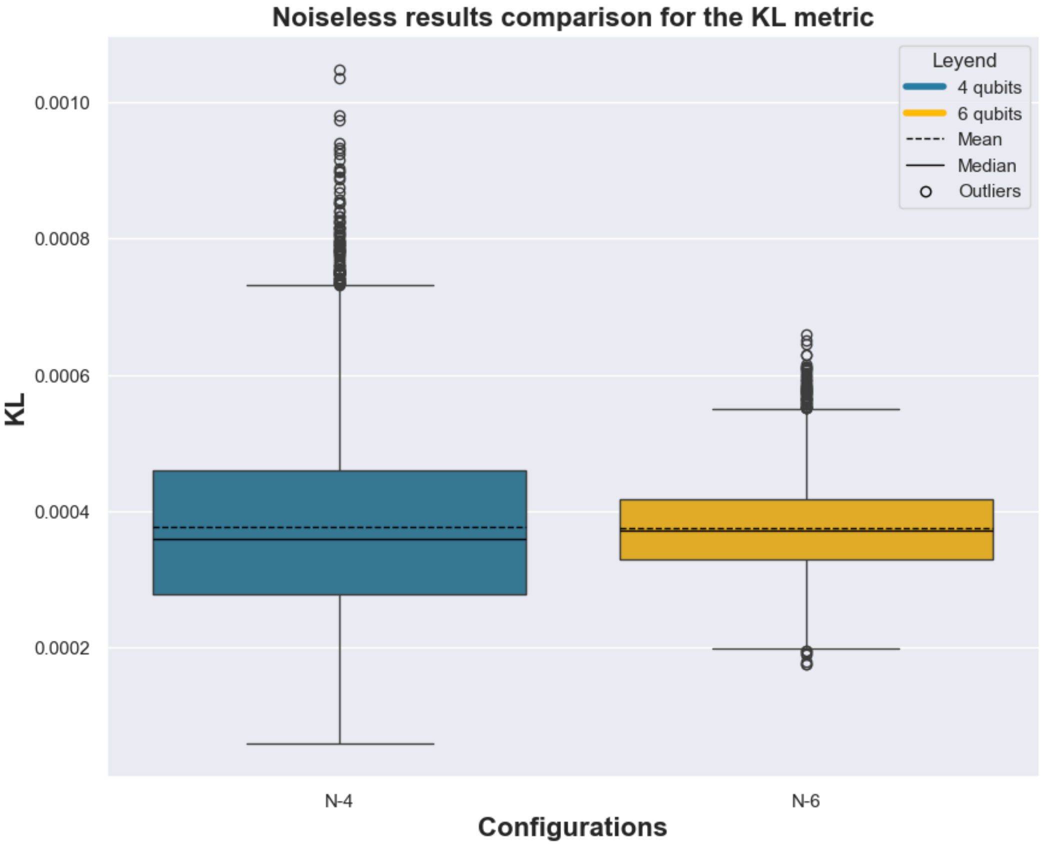


Figure 3: Comparison of the box plot of benchmark case PL noiseless using KL metric.

The observed variability in Figure 2 and Figure 3 can be attributed to several factors, such as the precision of the *Probability Loading* algorithm, the quality of the quantum circuit design, which used the **KPTree** algorithm from the *myQLM* library, or the inherent limitations of the hardware, even though the emulated QPU is noiseless. As the number of qubits increases, the Hilbert space expands exponentially, allowing for a better approximation of the desired probability distributions. Additionally, a higher number of qubits improves the fidelity of the probability loading operator implementation, contributing to a more accurate estimation of amplitudes. With more qubits, the quantum circuit gains a greater capacity to represent complex states, which can result in reduced variability in the obtained results.

2.1.2. Benchmark case 2: Amplitude Estimation Algorithms

This test was conducted using configurations of **4** and **6 qubits**, in line with the recommended range of 4 to 8 qubits outlined in the "D3.5: The NEASQC Benchmark Suite (TNBS)" document. The number of **auxiliary qubits** was kept constant at **10** to ensure high precision in the amplitude estimation. The choice of **1000 shots** was based on the fact that, while a higher number of shots reduces statistical uncertainty and improves estimation accuracy, it also significantly increases execution time. Since we are working with a noiseless emulated QPU, we determined that **1000 shots** strike a suitable balance between accuracy and computational efficiency.

For each qubit configuration, two CSV files were generated: one containing detailed benchmark data, including key metrics for evaluation, and another summarizing the average of those metrics across different executions.

Figure 4 illustrates the results of the absolute error metric in a noiseless emulated QPU. The sum absolute error is calculated as the difference between the Amplitude Estimation (AE) estimator and the Riemann sum, following the equation $\epsilon = \left|\tilde{S}^I_{[a,b]} - S_{[a^I,b^I]}\right|$. For the **4-qubit** configuration, the absolute error is approximately $\mathbf{0.0007}$, with low data variability and a mean slightly higher than the median. In the case of **6 qubits**, the absolute error increases to around $\mathbf{0.0008}$, showing greater variability compared to the 4-qubit configuration, with a median that is higher than the mean. Despite the differences in variability and data behavior, both configurations show absolute error values close to zero, indicating that the implementation was performed correctly and that the amplitude estimation algorithm functioned properly under the emulated conditions.

The greater variability observed with 6 qubits compared to 4 qubits can be attributed to the increased complexity of the quantum circuit, particularly due to the higher number of Grover-like operators. Even though the QPU is noiseless, the exponential growth of the Hilbert space increases computational load and, consequently, the likelihood of errors in amplitude estimation. This is reflected in the greater data dispersion observed with 6 qubits compared to the 4 qubits configuration.

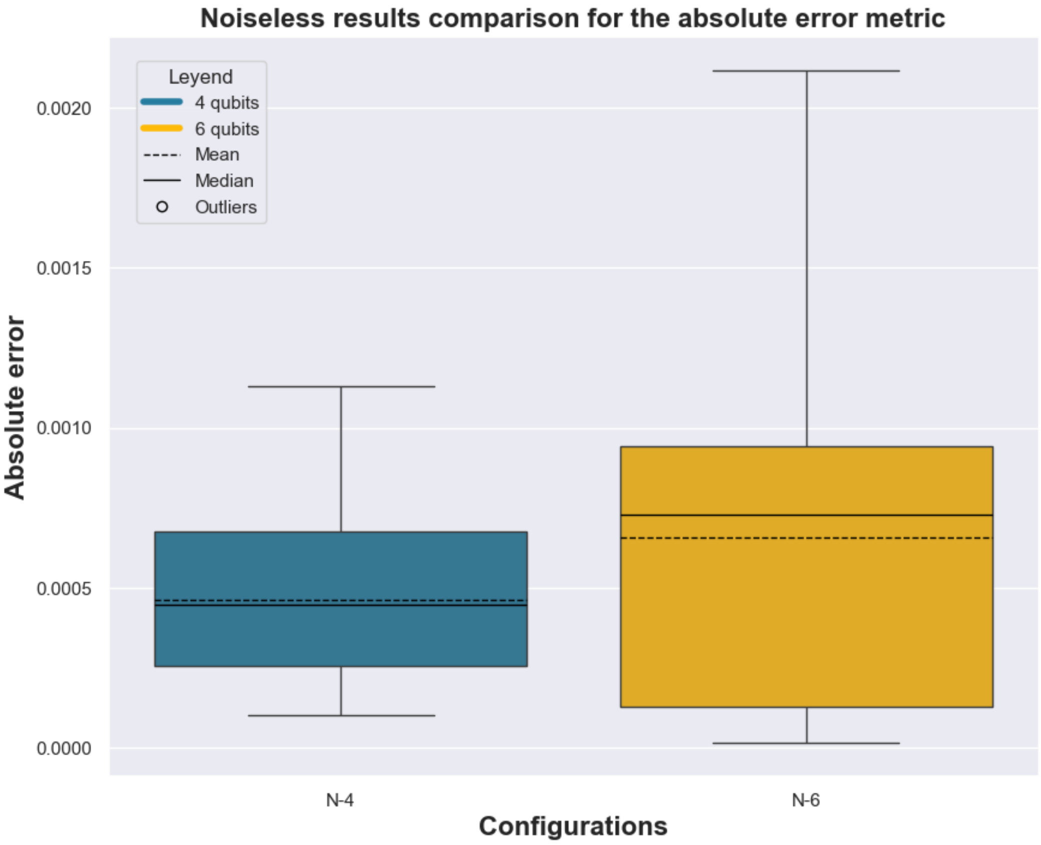


Figure 4: Comparison of the box plot of benchmark case AE noiseless using the absolute error metric.

2.1.3. Benchmark case 3: Phase Estimation Algorithms

This test was conducted using two configurations: 4 qubits with 4 auxiliary qubits and **4 qubits with 8 auxiliary qubits**, following the recommendations of the document "D3.5: The NEASQC Benchmark Suite (TNBS)." This document specifies that the number of auxiliary qubits must be equal to or greater than the number of primary qubits, increasing in steps of 2. In this test, the number of primary qubits remained constant while varying the number of auxiliary qubits to assess their impact on the accuracy of the metrics used. We chose **1000 shots** as a balance between precision and execution time, based on the logic used in the Amplitude Estimation benchmark. While a higher number of shots could reduce statistical uncertainty and improve accuracy, it would also significantly increase execution time.

Two CSV files were generated for each qubit configuration: one containing all the detailed benchmark data, including key metrics, and another summarizing the average of these metrics. The most important metrics were represented using box plots, with the mean indicated by a dashed line, the median by a solid line, and outliers shown as empty circles. Different colors were used for each configuration: blue for 4 auxiliary qubits and yellow for 8 auxiliary qubits, allowing for a clear comparison of accuracy and variability between the configurations. Additionally, two methods for angle selection were evaluated: random and exact for each configuration.

Figure 5 shows the results of the **KS** metric for the random angle selection method on a noiseless emulated QPU. With **4 auxiliary qubits**, the KS value is approximately $\mathbf{0.09}$, showing notable variability in the data and a mean slightly higher than the median. In the case of **8 auxiliary qubits**, the KS value decreases to approximately $\mathbf{0.06}$, with lower variability compared to the 4-qubit configuration, and with a median close to the mean. Both cases show KS values close to zero, indicating a correct implementation.

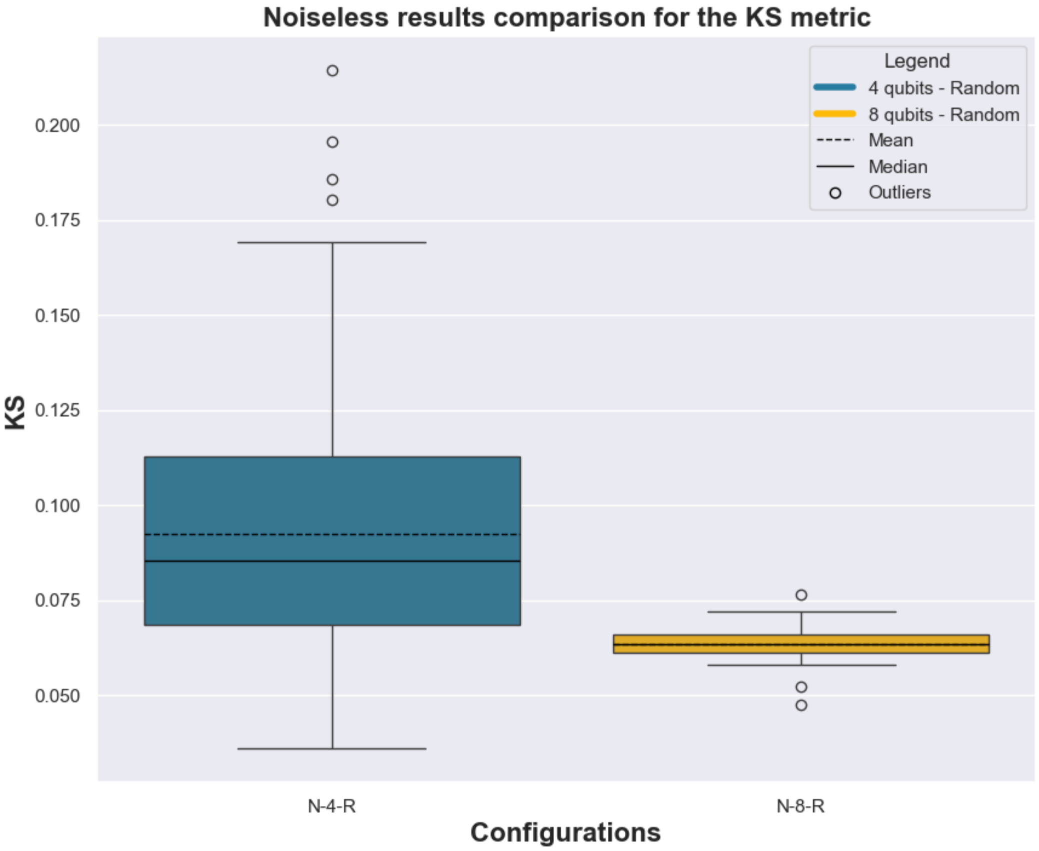


Figure 5: Comparison of the boxplot of benchmark case QPE noiseless using KS metric.

Figure 6 presents the results of the **Fidelity** metric, calculated for the exact angle selection method on a noiseless emulated QPU. For **4 auxiliary qubits**, the fidelity value is approximately $\mathbf{0.9997}$, with low variability in the data and a mean below the median. In the case of **8 auxiliary qubits**, fidelity slightly increases to $\mathbf{0.9998}$, with lower variability and a median above the mean. In both cases, the fidelity values are close to 1, indicating a successful implementation. In the random angle selection method, the difference between both configurations is more pronounced, as randomness introduces greater uncertainty. Conversely, in the exact method, where accuracy directly depends on the number of auxiliary qubits, the difference in variability is smaller.

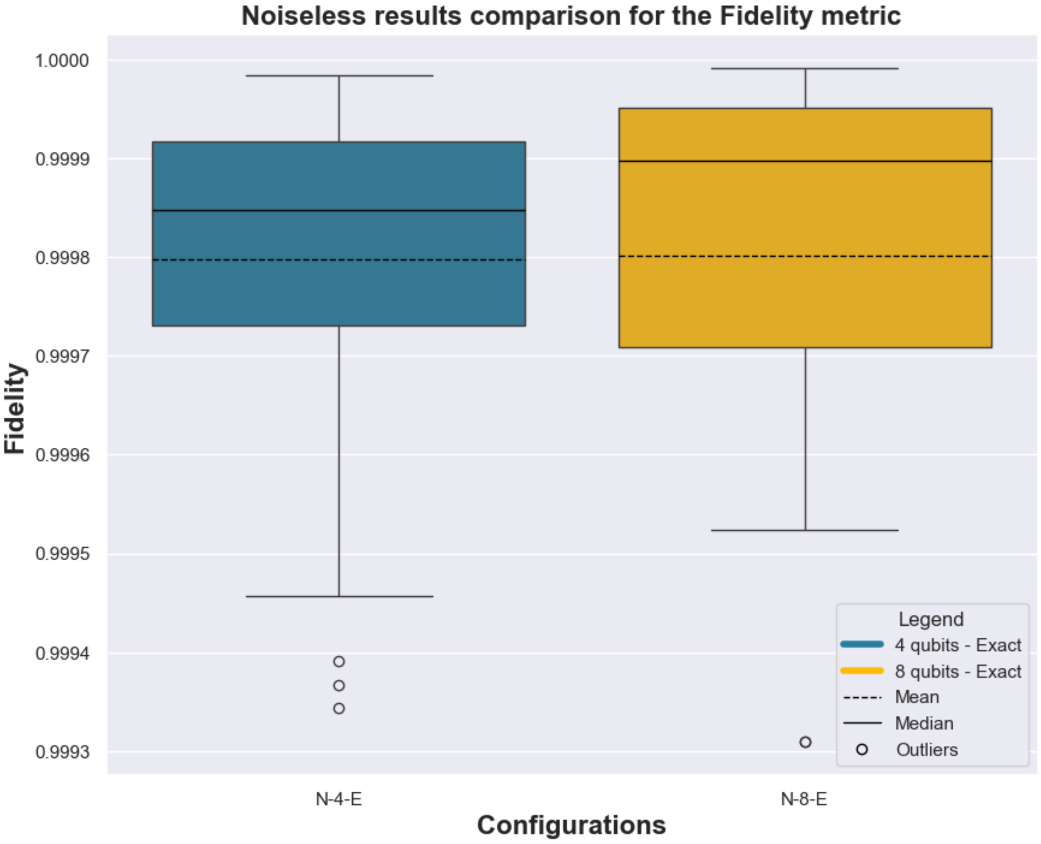


Figure 6: Comparison of the box plot of benchmark case QPE noiseless using the fidelity metric.

In both angle selection methods, greater variability was observed in the results with 4 auxiliary qubits compared to 8 auxiliary qubits. This is because a higher number of auxiliary qubits allows for improved accuracy in phase estimation. In the case of random angle selection, the difference between both configurations is more evident, while in the exact method, variability is smaller since angle selection directly depends on the number of auxiliary qubits.

2.1.4. Benchmark case 4: Parent Hamiltonian

In this benchmark case, the local Hamiltonian method was used to analyze different configurations of qubits and layers in a noiseless emulated environment. Configurations of **3, 4, 15, 20, 22**, and **23 qubits**, all with **1 layer**, were evaluated, as well as an additional configuration of **8 qubits with 4 layers**. The parameters used for each layer, θ_{x_n} and θ_{z_n} where n represents the layer number, refer to the parameterized gates R_x. and R_z, respectively. These parameters and the Pauli coefficients were taken from the TNBS documentation. **10,000 shots** were used in each configuration, as specified in the document "D3.5: The NEASQC Benchmark Suite (TNBS)." The single-layer configurations were chosen due to their low computational demand, allowing for a quick analysis of the impact of the number of qubits on the algorithm. The 4-layer configuration was included to assess how increasing the number of layers affects the results, though it was applied to only one configuration due to the significant increase in execution time.

Two CSV files were generated for each qubit configuration: one containing all detailed benchmark data, including key metrics, and another summarizing the averages of these metrics. Different colors were used for the various configurations, as described in the legend, allowing for a clear comparison of differences in accuracy and variability across the configurations.

Figure 7 shows the **energy metric** results for the noiseless emulated QPU. The system's Hamiltonian defines the energy associated with different configurations of the quantum system. Therefore, the goal is to minimize the energy value to ensure that the quantum state obtained is as close as possible to the ground state or minimum energy state, in which the system is most stable. In the lower complexity models, such as **N-3-1** (3 qubits, 1 layer) and **N-4-1** (4 qubits, 1 layer), there is less dispersion in the energy results, with no outliers, indicating greater consistency in the emulations. In contrast, models with a higher number of qubits and layers, such as **N-8-4** (8 qubits, 4 layers), **N-22-1** (22 qubits, 1 layer), and **N-23-1** (23 qubits, 1 layer), exhibit greater variability in their results. The inclusion of more layers in **N-8-4** seems to increase the fluctuations in the energy results, reflecting the added complexity introduced by increasing the number of layers in the quantum circuit.

Regarding the **means** and **medians**, it is observed that in most models, these are close to zero, suggesting a fairly uniform distribution of energy around this value. This implies that, despite the variability present in the more complex models, on average, the energies tend to be close to zero, indicating that the algorithm's implementation was successful under the emulated conditions. Although outliers are few, they are present in some models, such as N-20-1, suggesting that even in configurations with lower variability, unusual results can still occur.

Overall, the number of qubits and layers has a direct impact on the variability of energy results, with more complex models showing greater dispersion, while simpler models tend to offer more consistent results. This provides a clear view of how the complexity of the quantum system affects the stability of the results in this context.

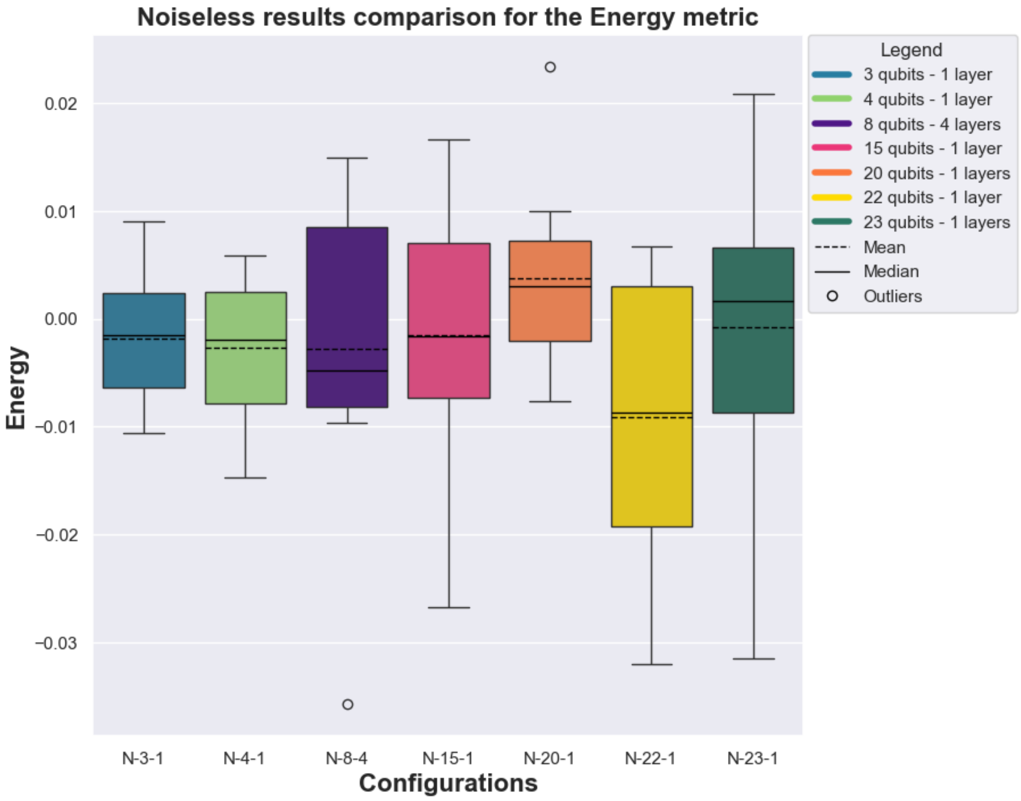


Figure 7: Comparative boxplot of use case PH noiseless using the energy metric.

2.2. Conclusions

The first phase of the project has been completed, with all benchmark cases defined in the project being accurately replicated: Probability Loading, Amplitude Estimation, Phase Estimation, and the Parent Hamiltonian.

Furthermore, the accuracy of the replicated benchmark cases was validated by the creators of the benchmark, who confirmed that the results met the standards and expectations set at the beginning of the project.

This achievement not only demonstrates the technical feasibility of the chosen approach but also establishes a robust foundation for the upcoming phases. The correct implementation and validation of the benchmark cases ensures confidence in the continuation of the project and the reliability of future developments.

3. Phase II

In this second phase, two of the benchmark cases (Probability Loading and Parent Hamiltonian) described in Section 1.2, were emulated using a noisy QPU. For this analysis, the noise model uses calibration data from the superconducting qubits of IBM's Brisbane backend and the trapped ion-based qubits of Quantinuum's H1 quantum computer.

3.1. Noise implementation

The methodology illustrated in Figure 8 was proposed for the implementation of noise in reference use cases. This methodology consists of three modules, each of which allows different configurations to be applied to build a customized hardware model.

The following is a brief description of the configurations available in this methodology as well as those selected in the present implementation:

1. Topology (Topology): default/custom. In this experiment, the custom option was selected, and the Brisbane (IBM) and H1 (Quantinuum) topologies were developed and implemented.
2. Native gate decomposition: default/custom. In this module, we also opted for a custom configuration, implementing a native gate decomposition using the Qaptiva transportation libraries for IBM and IONS systems.
3. Noise Models: default/custom. For this configuration, the custom option was applied because the calibration parameters used in the construction of the noise models were obtained from the Brisbane and H1 backends of the IBM and Quantinuum, respectively.

Finally, Figure 8 presents the hardware model building flow with the aforementioned configurations, the results of which were obtained for further analysis and evaluation.

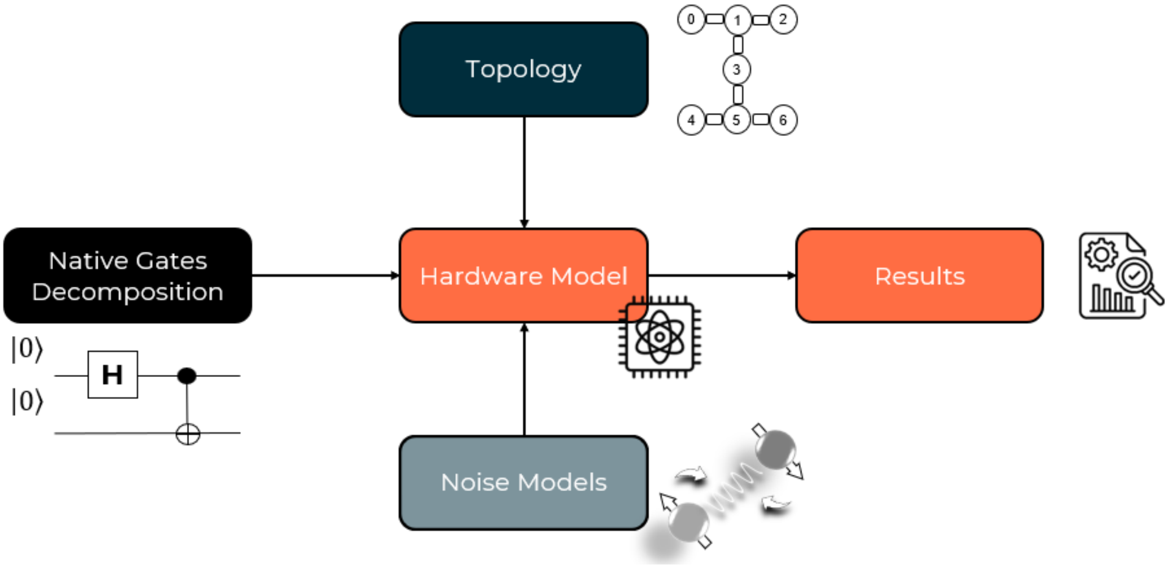


Figure 8: Methodology of noise implementation in a quantum circuit.

3.1.1. Qaptiva emulation models

There are two emulation modes available in the Qaptiva emulator to emulate QPUs with noise:

1. **Deterministic mode**: This is based on the representation of the density matrix, which contains all the information about the quantum state, including noise effects. However, its use is limited to systems of around 20 qubits due to its high memory consumption, as the size of the density matrix grows exponentially with the number of qubits. In this case, the main computational challenge is memory consumption, not execution time.
2. **Stochastic mode**: This uses a state vector based simulation, interpreting the noisy system as a probabilistic distribution over pure states, rather than directly storing the density matrix. This allows it to emulate larger qubit systems more efficiently in terms of memory, but it requires a large number of samples to obtain statistically reliable results. Consequently, while it can emulate more qubits, the execution time increases considerably due to the processing required for the necessary samples to approximate the system's behavior.

The deterministic method is faster than the stochastic one because it directly calculates the system's behavior by manipulating the density matrix, avoiding the need for repeated sampling and averaging. However, its main limitation is high memory consumption, which makes it impractical for large quantum systems. Despite this drawback, the deterministic method was chosen for this evaluation due to its higher accuracy, speed and simplicity, as it does not require sampling parameter optimization, as in the stochastic method.

3.1.2. Types of noise implemented

Qaptiva supports several noise models:

- **Amplitude Damping (AD):** It simulates energy dissipation in a quantum system. It describes the gradual loss of energy from a qubit, typically due to its interaction with the environment, eventually returning it to its ground state. This process models phenomena such as spontaneous emission.
- **Pure Dephasing (PD):** It addresses the loss of quantum coherence without any energy dissipation. It is considered a purely quantum form of noise, as it affects only the phase information of the quantum state without altering the energy levels of the qubit. It is especially relevant in systems where maintaining coherence is important, such as Grover's algorithm.
- **Idle Noise:** It combines amplitude damping and pure dephasing to represent the noise experienced by qubits when they are idle but still affected by environmental decoherence. It provides a more comprehensive view of the noise affecting idle qubits, though it may lose specificity in capturing details of individual noise sources.
- **Depolarizing Channel**: It simulates a general noise process, similar to "white noise", that randomizes the state of the qubit. It describes the depolarization of qubits with a probability p, where the qubit is replaced by a completely mixed state ($I/2$), and with a probability $1-p$, the qubit remains unaffected by the noise. This model is frequently used to represent gate errors, making it particularly useful in gate-based quantum computing.
- **Depolarizing Channel + Idle Noise**: It combines the effects of depolarizing noise with idle noise. It provides a more complete representation of the noise affecting both idle qubits and those undergoing gate operations. However, its added complexity can increase computational overhead without necessarily improving accuracy in simpler scenarios.

The Amplitude Damping, Pure Dephasing, and Idle Noise models are used to describe the noise that occurs when qubits remain idle, reflecting their interaction with the environment. On the other hand, the Depolarizing Channel model is associated with logic operations on the qubits, simulating the impact of errors during quantum gate executions.

3.2. Noise results

In all the analyzed benchmark cases, the five implemented noise models (Amplitude Damping, Pure Dephasing, Idle Noise, Depolarising Channel and Depolarising Channel + Idle Noise) are compared with the noiseless emulated QPU. The comparisons will be represented by box plots, in which the mean will be indicated by a dashed line, the median by a solid line, and outliers will be shown as unfilled circles, as detailed in the legend of the plots. The calibration parameters used for both IBM and Quantinuum are detailed in Appendix A. Appendix B presents the gate counts for each circuit, differentiating the various decompositions into native gates according to the employed technologies.

For the graphs shown in this section, the CSV files containing all the data from each benchmark case have been used, while the mean values, execution times, and respective standard deviations for the most relevant metrics of these reference cases are detailed in Section 4 of this document. Because the evaluation was performed using an emulator, the elapsed time is not an indication of the performance of a QPU. In fact, it is related to the performance of the emulator on the specific hardware. Because we are not testing the emulator and its configuration was not calibrated, we do not report the elapsed time for any benchmark.

3.2.1. Benchmark case 1: Probability Loading Algorithms

Now, we present figures similar to those of section 2.1.1 but using noisy emulations instead of noiseless emulated platforms. Each plot corresponds to a different metric of this benchmark case: KS test, KL divergence. These diagrams include the results of emulations on a noisy platform using different noise models.

3.2.1.1. Noisy Brisbane emulation: 4 qubits

Figure 9 provides a detailed comparison of the **KS metric** for 4 qubits for various noise models implemented using the calibration parameters of IBM's Brisbane QPU. In the case of the **noiseless QPU emulation** (N), an average of approximately $\mathbf{0.005}$ is observed, with relatively low data dispersion compared to noisy models, although a few outliers exceed $\mathbf{0.01}$. The **Amplitude Damping Noise model** (AD) shows a higher mean, around $\mathbf{0.02}$, with moderate data dispersion and outliers reaching $\mathbf{0.03}$, indicating a greater impact on the KS metric than the noiseless model. The **Pure Dephasing Noise model** (PD) behaves similarly to the noiseless model, with an average of about $\mathbf{0.005}$ and low data dispersion, though it does display some outliers above $\mathbf{0.01}$. This is because the *PD* model only affects phase, while this benchmark primarily evaluates amplitudes, not phases. The **Idle Noise model** (I) exhibits characteristics similar to the *AD* model, with an average of around $\mathbf{0.02}$, moderate dispersion, and outliers near $\mathbf{0.03}$. The **Depolarizing Channel model** (D) shows a more significant impact, with an average of around $\mathbf{0.025}$ and higher dispersion than previous models and multiple outliers. Finally, the **Depolarizing Channel and Idle Noise** (D+I) combined model produces the most substantial impact, with an average of around $\mathbf{0.04}$, dispersion similar to the *AD* and *I* models, and outliers reaching up to $\mathbf{0.05}$. Characteristically, the mean and median are very close, suggesting a relatively symmetrical distribution in the data.

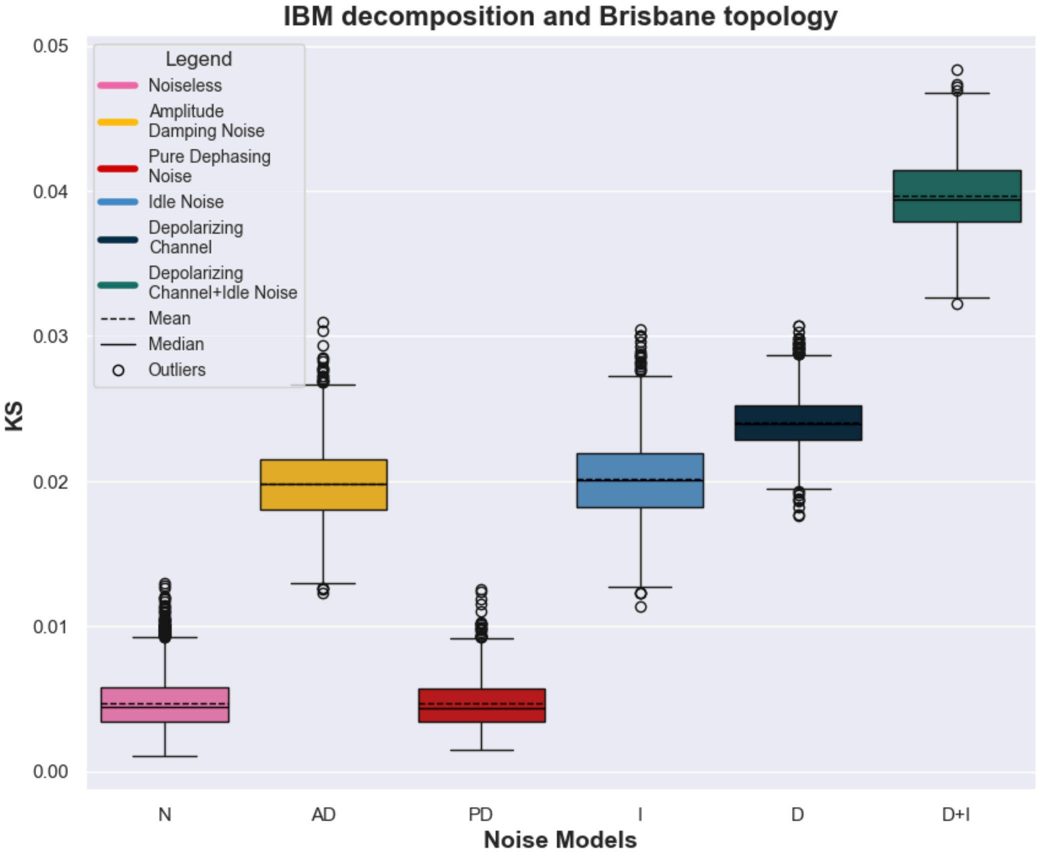


Figure 9: Comparative boxplot for benchmark case PL with noise and 4 qubits of the KS metric using IBM Brisbane QPU parameters.

Figure 10 provides a detailed comparison of the **KL metric** for 4 qubits across various noise models implemented using the calibration parameters of IBM's Brisbane QPU. The **noiseless QPU emulation** (N) exhibits the lowest values, with an average of around $\mathbf{0.0004}$ and minimal dispersion, showing only a few outliers slightly above $\mathbf{0.001}$. The **Amplitude Damping Noise model** (AD) shows a notable increase, with an average of around $\mathbf{0.003}$ and moderate dispersion, and outliers reaching up to $\mathbf{0.005}$. The **Pure Dephasing Noise model** (PD) behaves similarly to the noiseless model, as this noise model affects only phase, while this benchmark primarily evaluates amplitudes. The **Idle Noise model** (I) exhibits characteristics similar to the *AD* model, with an average of around $\mathbf{0.003}$ and moderate dispersion, including outliers up to $\mathbf{0.005}$. The **Depolarizing Channel model** (D) shows a significantly higher impact, with an average of approximately $\mathbf{0.009}$, substantial dispersion, and several outliers reaching $\mathbf{0.012}$. The combined **Depolarizing Channel and Idle Noise model** (D+I) demonstrates the greatest impact, with an average of approximately $\mathbf{0.015}$, the highest observed dispersion, and outliers reaching up to $\mathbf{0.018}$. A clear correlation is observed between the mean magnitude and data dispersion, with models showing higher averages also presenting a greater variability in results.

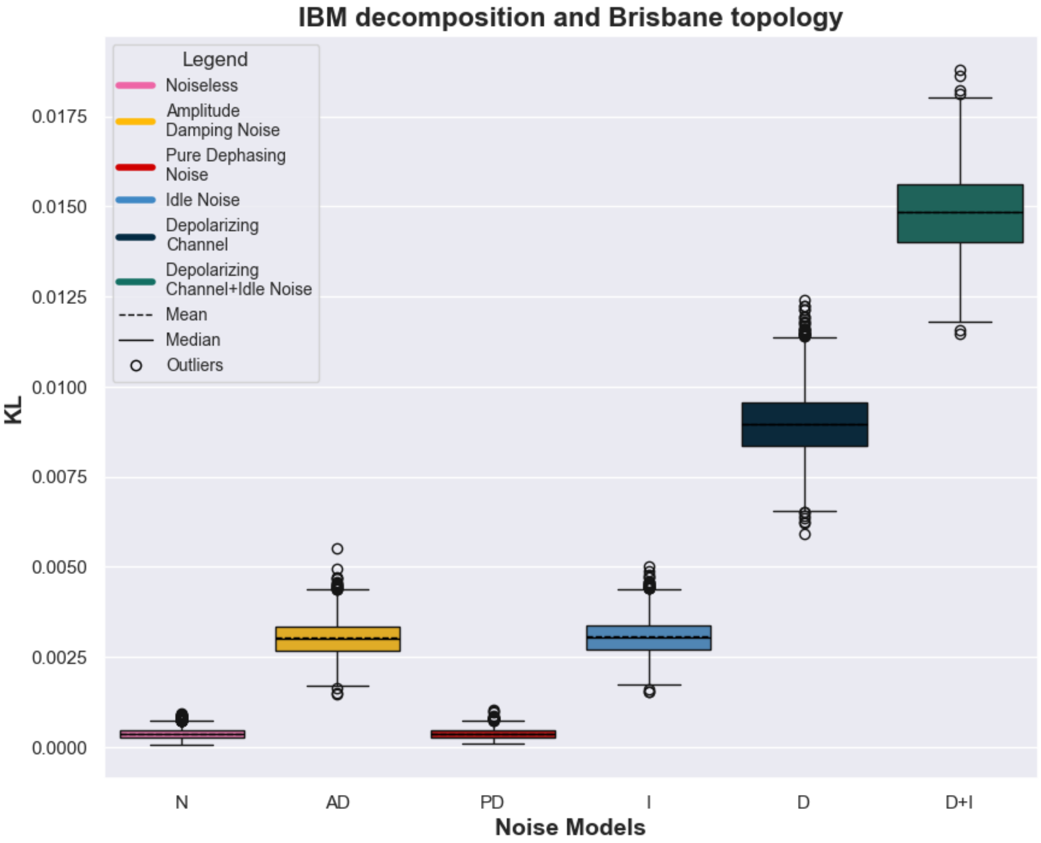


Figure 10: Comparative boxplot for benchmark case PL with noise and 4 qubits of the KL metric using IBM Brisbane QPU parameters.

In both metrics KS and KL, *AD* and *D* are the models with the most significant impact. The *PD* model has a minimal effect due to this benchmark primarily assessing amplitudes, because *PD* noise only affects phase. Furthermore, the *D+I* model, which closely approximates the behavior of real noise in a QPU, exhibits the greatest impact, significantly deviating from the near-zero values obtained in a noiseless QPU.

3.2.1.2. Noisy Quantinuum emulation: 4 qubits

Figure 11 provides a detailed comparison of the **KS metric** for 4 qubits across different noise models implemented using the calibration parameters of Quantinuum's H1 QPU. In the **noiseless QPU emulation** (N), the average is approximately $\mathbf{0.0045}$ with moderate dispersion and several outliers extending up to $\mathbf{0.013}$. The **Amplitude Damping Noise** (AD), **Pure Dephasing Noise** (PD), and **Idle Noise** (I) models show very similar characteristics in both mean and data dispersion. This similarity may be attributed to the T_1 and T_2 parameters, which are of the same order and extremely large, around 10^9 ns. Consequently, there is minimal impact on this metric for these three noise models. More significant changes are observed in the **Depolarizing Channel model** (D), which shows a slight increase in the mean, approximately 0.006, and slightly higher dispersion than the previous models, with outliers extending up to $\mathbf{0.014}$. Finally, the combined **Depolarizing Channel and Idle Noise model** (D+I) has the highest mean, close to $\mathbf{0.006}$, and the largest dispersion among all models, with outliers reaching $\mathbf{0.015}$. Notably, unlike other implementations, the first four models (N, AD, PD, I) exhibit very similar behavior on Quantinuum's H1 topology, with minimal differences in their statistics. Only the *D* and *D+I* models display a distinguishable increase in both mean and dispersion. Considerable variability can be seen in the results with quite a few outliers above the mean. This causes the mean to be slightly above the median for all noise models implemented

The Quantinuum trapped-ion technology has more constant values for this metric than the IBM superconducting qubits (Figure 9) for all models, although a slight increase is seen in the models that include the Depolarizing Channel.

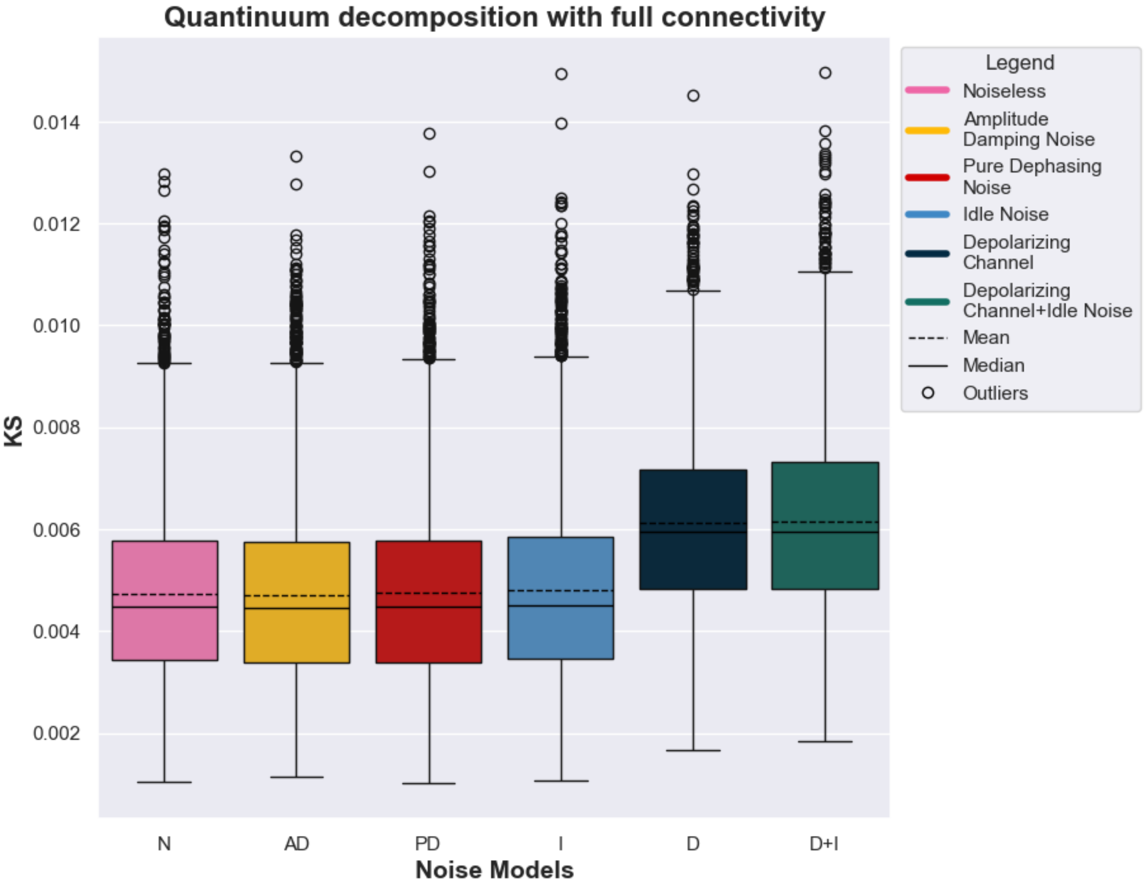


Figure 11: Comparative boxplot for benchmark case PL with noise and 4 qubits of the KS metric using Quantinuum H1 QPU parameters.

Figure 12 provides a detailed comparison of the **KL metric** for 4 qubits across various noise models implemented using the calibration parameters of Quantinuum's H1 QPU. The **noiseless QPU emulation** (N) shows an average of approximately $\mathbf{0.00035}$ with moderate dispersion and outliers extending up to $\mathbf{0.0009}$. The **Amplitude Damping Noise** (AD), **Pure Dephasing Noise** (PD), and **Idle Noise** (I) models exhibit characteristics very similar to the noiseless QPU, both in terms of mean, median and data dispersion. The most notable changes are observed in the **Depolarizing Channel model** (D), which displays a significant increase in mean, approximately $\mathbf{0.0006}$, and substantially greater dispersion than the previous models, with outliers reaching up to $\mathbf{0.0014}$. The combined **Depolarizing Channel and Idle Noise model** (D+I) shows characteristics similar to the D model, with a slightly higher mean of around $\mathbf{0.0006}$ and comparable dispersion, with outliers extending up to $\mathbf{0.00175}$. Notably, in this H1 topology from Quantinuum, the first four models (N, AD, PD, I) display almost identical behavior, while the *D* and *D+I* models are clearly distinguishable by higher means and greater variability. All models present significant outliers, particularly pronounced in the *D* and *D+I* models, suggesting that these types of noise introduce a more irregular and potentially more severe impact on the system.

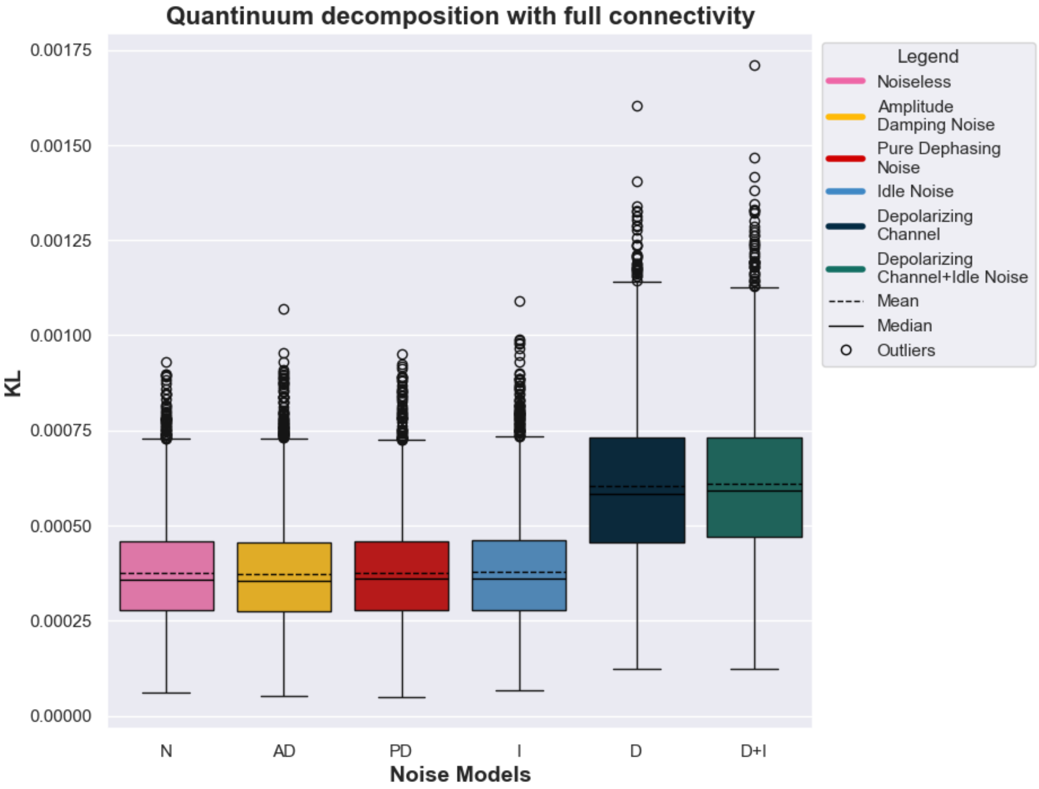


Figure 12: Comparative boxplot for benchmark case PL with noise and 4 qubits of the KL metric using Quantinuum H1 QPU parameters.

3.2.1.3. Comparison Between Different Technologies

To evaluate the effectiveness of the benchmark suite in assessing the precision of various metrics across different quantum computing platforms, the most realistic noise model in this implementation was selected: the Idle+Depolarizing Channel model, which includes both gate and environmental noise. Figure 13 and Figure 14 provide a boxplot comparison of the emulation of two QPUs from different technologies, IBM's Brisbane and Quantinuum's H1.

In the 4-qubit case, the IBM platform demonstrates greater accuracy compared to Quantinuum for both evaluated metrics. IBM’s results show higher dispersion, reflected in longer whiskers and wider interquartile ranges, whereas Quantinuum’s platform exhibits a more reduced dispersion, indicating greater stability and accuracy. This is possibly due to the full connectivity of Quantinuum's qubits compared to Brisbane’s topology, which requires additional SWAP gates.

These results highlight the performance differences between platforms under noisy conditions and confirm that the benchmark suite is effective in accurately evaluating discrepancies across different QPU technologies.

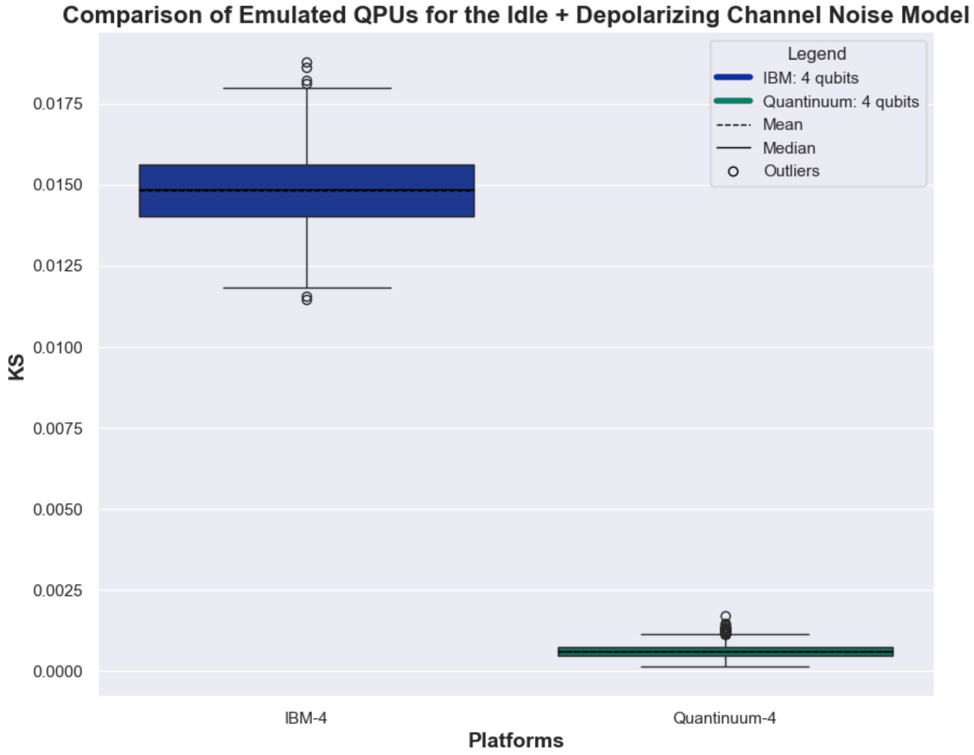


Figure 13: Comparative boxplot for benchmark case PL under the Idle + Depolarizing Channel noise model across different emulated QPUs using the KS metric.

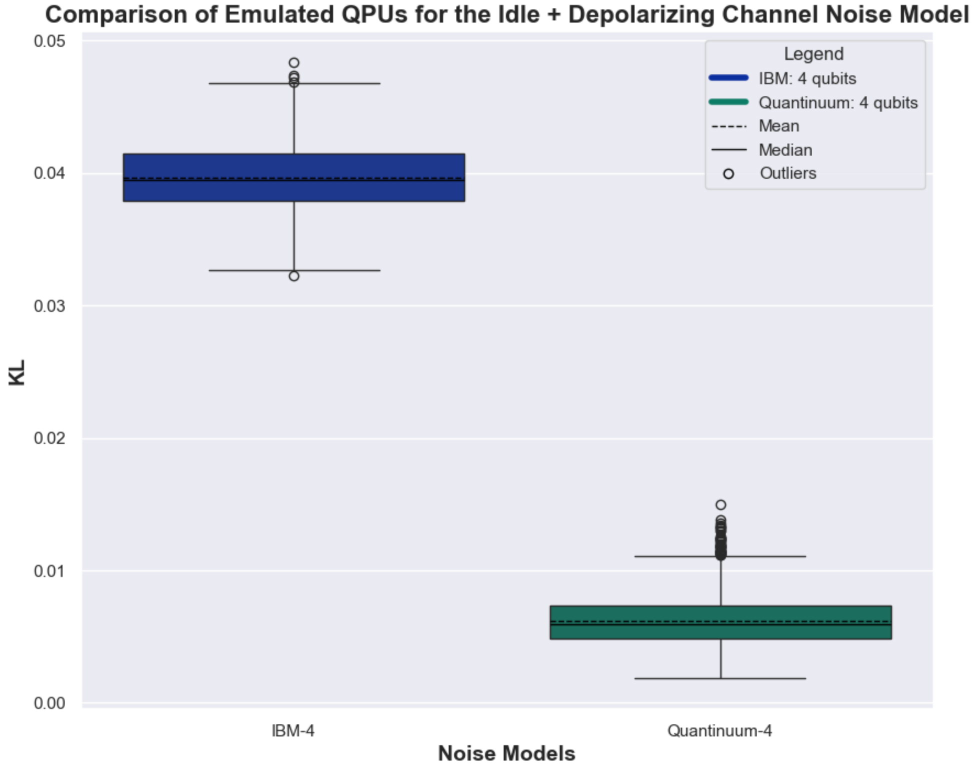


Figure 14: Comparative boxplot for benchmark case PL under the Idle + Depolarizing Channel noise model across different emulated QPUs using the KL metric.

3.2.2. Benchmark case 4: Parent Hamiltonian

Now, we present the noisy emulation of the Parent Hamiltonian benchmark case. Box plots comparing the different noise models implemented are shown below. Each graph corresponds to an important metric in this benchmark case: energy. These diagrams visualize and compare the performance of the models as a function of each specific metric.

3.2.2.1. Noisy Brisbane emulation: 4 qubits

Figure 15 shows a detailed comparison of the **Energy metric** for 4 qubits and 1 layer across different noise models, using the calibration parameters of IBM's Brisbane QPU. In the **noiseless QPU emulation** (N), values are very close to zero with minimal dispersion, although a few outliers are observed, including a positive one near $\mathbf{0.015}$ and a negative one around $-\mathbf{0.008}$. The **Amplitude Damping** (AD), **Pure Dephasing** (PD), and **Idle Noise** (I) models display similar distributions centered around zero, with moderate dispersion. The *PD* model has a negative outlier near $-\mathbf{0.015}$ and a positive one close to $\mathbf{0.02}$. The **Depolarizing Channel model** (D) and its combination with idle noise (D+I) show significantly higher energy levels, with means around $\mathbf{0.10}$, and exhibit similar dispersions, with no notable outliers. It is evident that depolarization effects lead to a marked increase in system energy, approximately an order of magnitude higher than other noise types, which remain near zero.

A not very large variability can be seen in the results. The mean is slightly above the median for most of the noise models implemented, with the exception of the *PD* and *I* models.

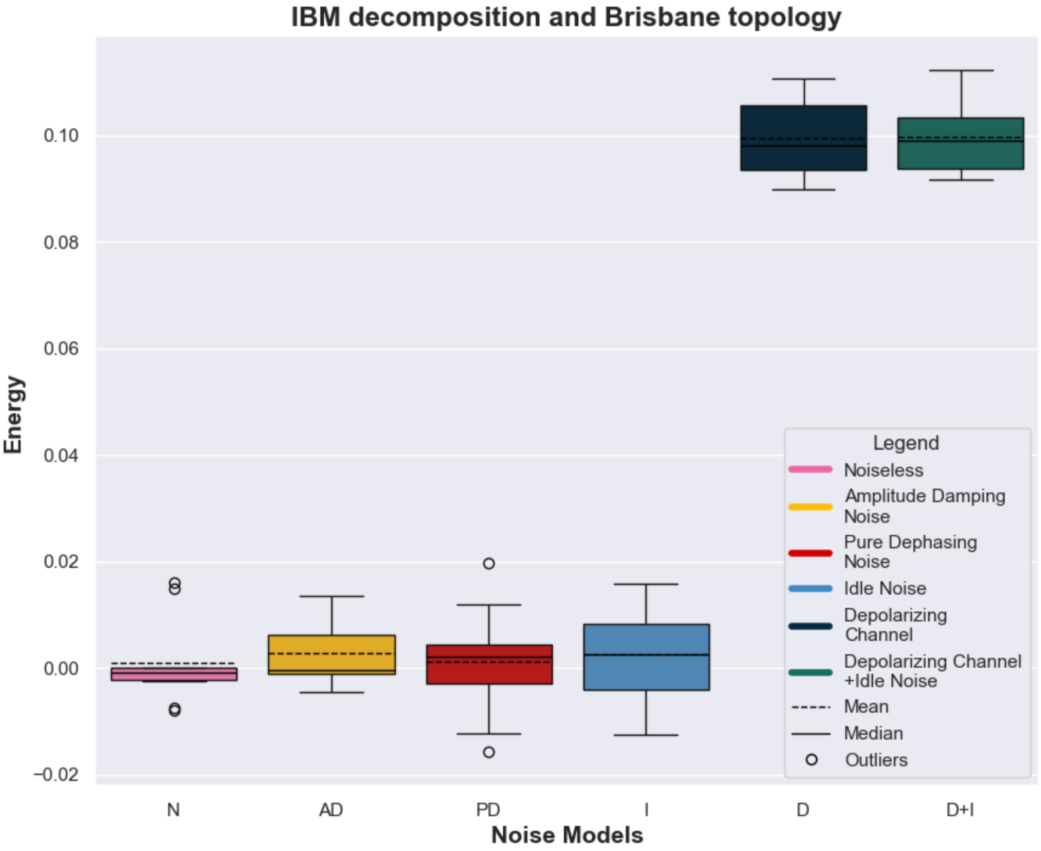


Figure 15: Comparative boxplot for benchmark case PH with noise and 4 qubits of the energy metric using IBM Brisbane QPU parameters.

3.2.2.2. Noisy Brisbane emulation: 6 qubits

Figure 16 provides a detailed comparison of the Energy metric for 6 qubits and 2 layers across different noise models, using the calibration parameters of IBM's Brisbane QPU. To improve the clarity of the graphs given the differences among the models, a cut has been applied to the y-axis. The **noiseless QPU emulation** (N) shows an energy level very close to zero, with significant data variability, especially below the mean. The **Amplitude Damping** (AD), **Pure Dephasing** (PD), and **Idle Noise** (I) models have similar zero-centered distributions but show greater dispersion than the noiseless QPU emulation. The *AD* model, in particular, exhibits a significant outlier below the mean, around $-\mathbf{0.025}$. The **Depolarizing Channel model** (D) and combined **Depolarizing Channel with Idle Noise** (D+I) models display significantly higher energy levels, with means around $\mathbf{0.41}$, and similar dispersion levels between them. These last two models show no notable outliers.

Interestingly, the first four models (N, AD, PD, I) have energy levels close to zero, while the latter two (D and D+I) show a clear increase in system energy, indicating a significant impact of depolarization effects on system energy. Environmental noise from phase or amplitude changes increases the spread of values but maintains an average energy near zero.

If we compare these graphs with those obtained for 4 qubits and 1 layer represented in Figure 15, we can see that the increase in gates due to the increase in the number of qubits and layers, mainly affects the models that include Depolarizing Channel.

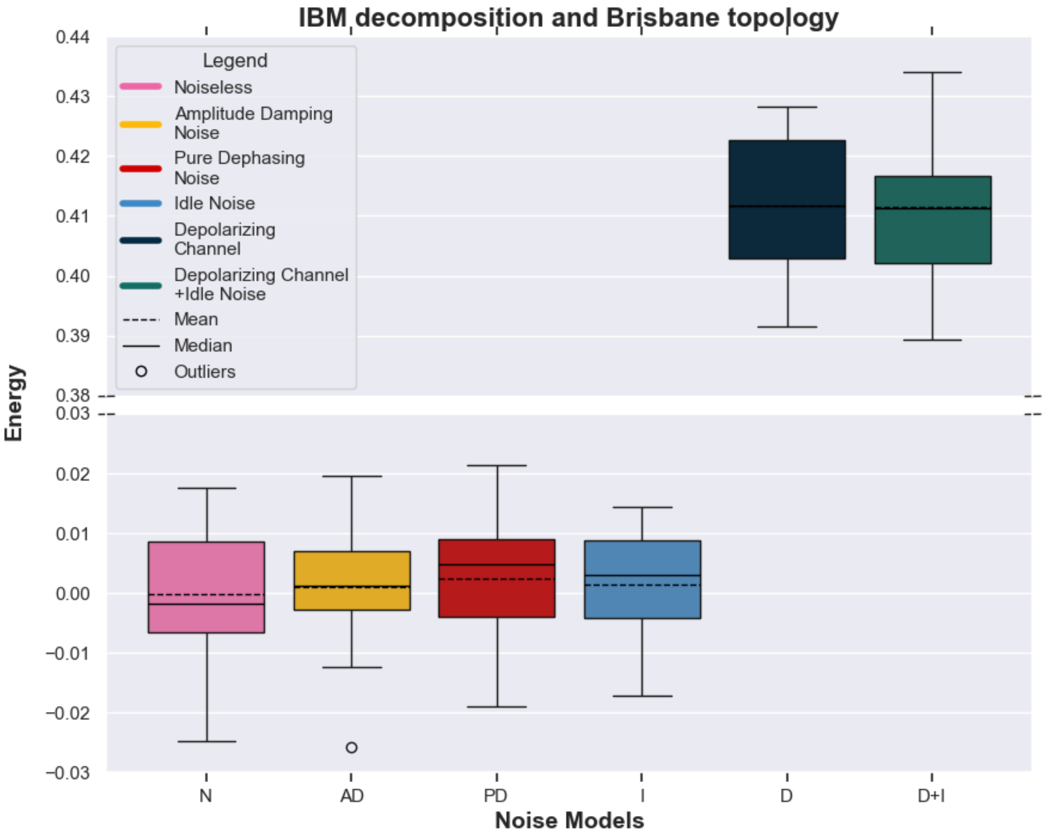


Figure 16: Comparative boxplot for benchmark case PH with noise and 6 qubits of the energy metric using IBM Brisbane QPU parameters.

3.2.2.3. Noisy Quantinuum emulation: 4 qubits

Figure 17 provides a detailed comparison of the **Energy metric** for 4 qubits and 1 layer across different noise models, using the calibration parameters of Quantinuum's H1 QPU. The **noiseless QPU emulation** (N) exhibits the lowest energy level, with a mean close to zero, and includes some outliers both above and below the mean. The **Amplitude Damping model** (AD) shows considerable dispersion, with a mean around $\mathbf{0.01}$ and no notable outliers. The **Pure Dephasing model** (PD) presents the most compact distribution among the environmental noise models, with a mean similar to *AD* but with lower variability. The **Idle Noise model** (I) has similar dispersion to *AD*, with a mean near $\mathbf{0.008}$ and a single negative outlier below -0.01. The **Depolarizing Channel** (D) and combined **Depolarizing Channel with Idle Noise** (D+I) models show higher energy levels, with means around 0.03. The *D* model exhibits two closely spaced outliers near 0.015, while the *D+I* model displays a wider dispersion without outliers.

It is notable how depolarization effects significantly increase energy levels compared to the idle qubit noise models.

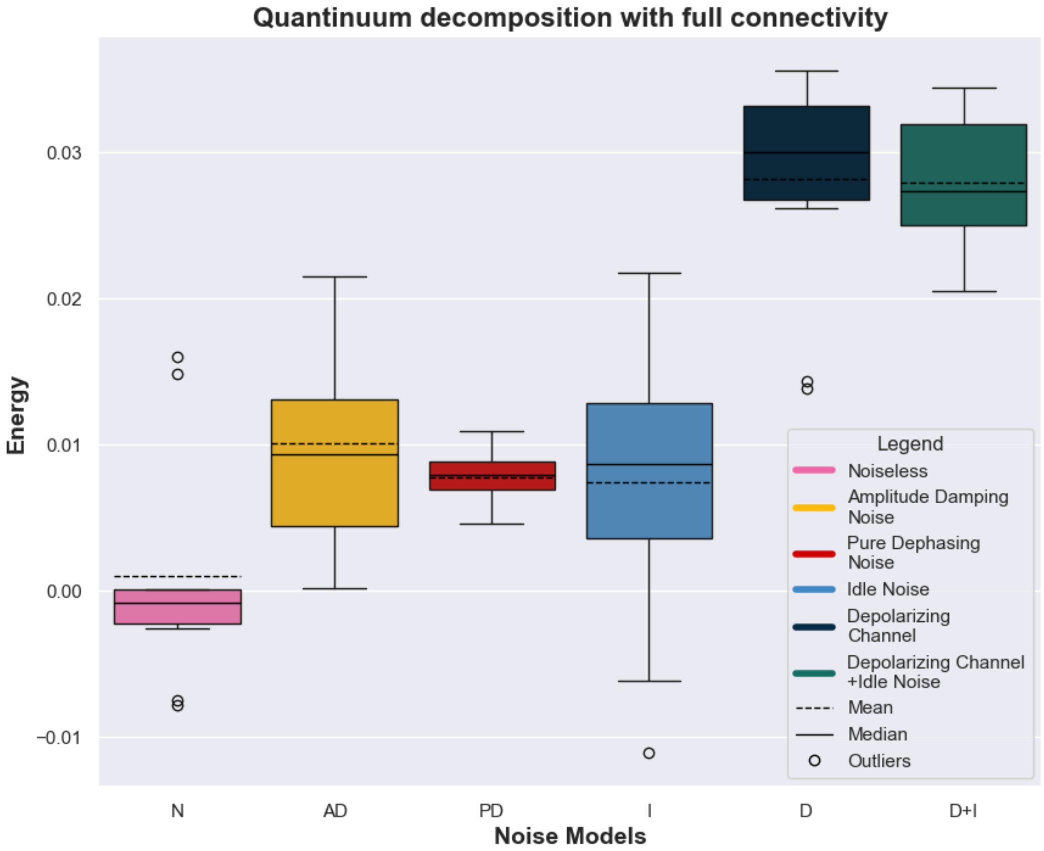


Figure 17: Comparative boxplot for benchmark case PH with noise and 4 qubits of the energy metric using Quantinuum H1 QPU parameters.

3.2.2.4. Noisy Quantinuum emulation: 6 qubits

Figure 18 shows a detailed comparison of the **Energy metric** for 6 qubits and 2 layers across different noise models implemented using the calibration parameters of Quantinuum's H1 QPU. To facilitate the visualization of two distinct energy ranges, a cut has been included on the vertical axis.

In the lower range, the case of **emulating a noiseless QPU** (N) shows values close to zero, with notable data dispersion. In the upper energy range (approximately between 5.45 and 5.50), the models exhibit differentiated behaviors. The **Amplitude Damping** (AD), **Pure Dephasing** (PD), and **Idle Noise** (I) models present similar energy values, with averages around 5.48, with *AD* showing slightly greater dispersion. The models that include depolarization (D and D+I) exhibit slightly lower energy values, with averages around 5.46 and comparable dispersion to the previous models.

An intriguing observation is that, unlike the other graphs, this figure reveals that the effects of depolarization tend to decrease the system's energy in comparison to other types of environmental noise. This behavior is unexpected for two reasons: first, the increase in energy compared to the 4-qubit case (represented in Figure 17) is substantial, and second, environmental noise leads to higher energy values than those observed in the depolarization model. These results suggest a potential anomaly in the algorithm's operation under noisy conditions in the 6-qubit implementation. However, in the absence of noise, the values remain within the expected error margin, indicating that the noiseless implementation operates in accordance with theoretical expectations. This variability allows us to understand both the limitations and strengths of the benchmarking methodology. By analyzing these discrepancies, we can gain valuable insights into the performance of quantum platforms under various conditions, which is fundamental for assessing their accuracy and robustness.

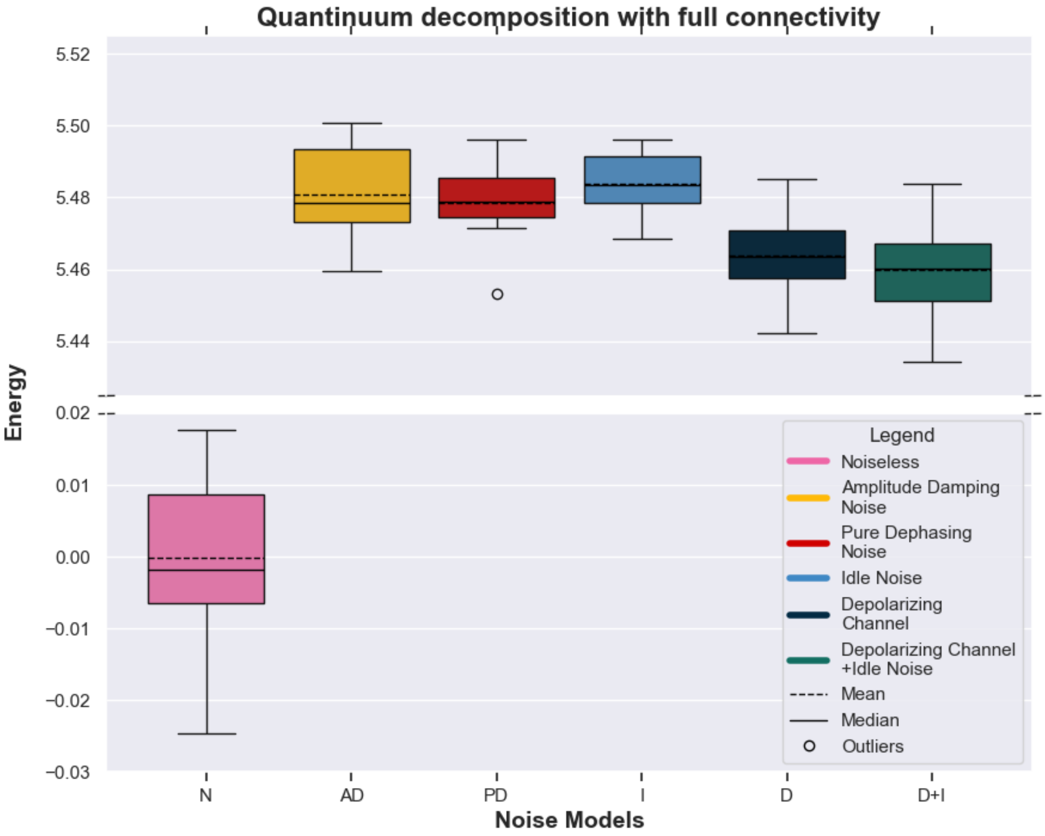


Figure 18: Comparative boxplot for benchmark case PH with noise and 6 qubits of the energy metric using Quantinuum H1 QPU parameters.

3.2.2.5. Comparison Between Different Technologies

Figure 19 presents a boxplot comparing the performance of two quantum computing platforms, IBM and Quantinuum, under the most realistic noise model used in this implementation: the Idle + Depolarizing Channel model, which accounts for both gate and environmental noise. For the case of **4-qubit and 1-layer**, IBM shows higher energy values than Quantinuum, likely due to the specific topology of the Brisbane backend, which requires additional SWAP gates compared to the fully connected qubit architecture of Quantinuum. In the **6-qubit and 2-layer** configuration, IBM exhibits higher energy than in the 4-qubit and 1-layer case, as expected due to the increased circuit complexity, but lower than Quantinuum, which reaches energy levels around $\sim\mathbf{5.46}$. This discrepancy could be attributed to Quantinuum's longer coherence and gate times, potentially introducing inaccuracies in the emulation of this specific test case, resulting in deviations from theoretical expectations. These results highlight performance differences between the platforms under noisy conditions, affirming the benchmark suite’s capability to accurately assess precision and variations across different QPU technologies.

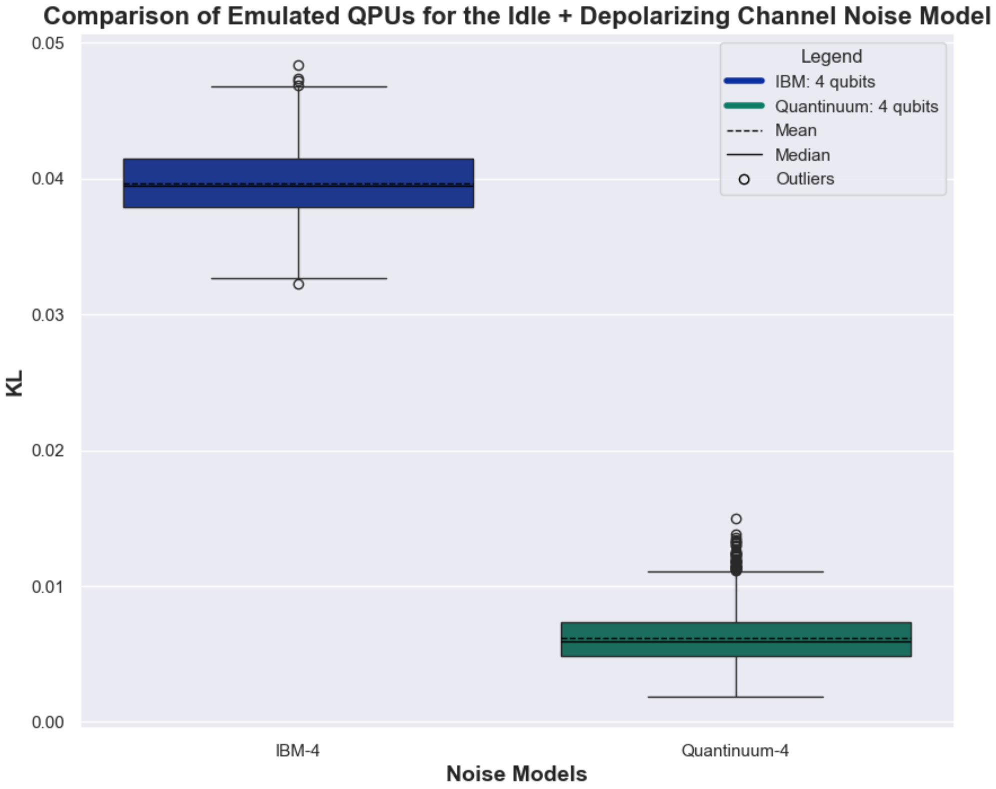


Figure 19: Comparative boxplot for benchmark case PH under the Idle + Depolarizing Channel noise model across different emulated QPUs using the Energy metric.

3.3. Conclusions

The task second phase has been successfully completed following the implementation of native gate decomposition and the integration of the topology of two QPUs with different quantum technologies: superconducting qubits in IBM's Brisbane and trapped ion qubits in IonQ's H1. Additionally, the following noise models have been successfully incorporated: Amplitude Damping, Pure Dephasing, Idle Noise, Depolarizing Channel, and Depolarizing Channel + Idle Noise.

It is important to note that the application of these noise models was limited to benchmark cases PL and PH due to the greater complexity of the algorithms in benchmark cases AE and QPE. The structure of these algorithms, as well as the type of quantum gates they employ, prevented the integration of the noise models in those scenarios.

It is observed that both benchmark cases exhibit similar behavior for both 4 and 6 qubits, reflecting the increase in noise associated with a greater number of qubits and the resulting complexity of the implemented circuit. However, a more pronounced difference emerges when changing the quantum technology, as each technology has notably different calibration parameters and a distinct decomposition into native gates.

4. Concluding remarks

The following tables summarize the results obtained from the implementation of phases one and two of this experiment. Based on this evaluation, we conclude that the first phase allowed us to validate and refine the benchmarking methodology documentation, ensuring the accurate replication of test cases as specified by the TNBS documentation. In the second phase, the effectiveness of the benchmark suite in comparing the accuracy of different quantum platforms was validated, as it successfully captured differences among them through the execution of various test cases. The inclusion of five noise models across two quantum technologies (IBM and Quantinuum) facilitated the assessment of noise impact on quantum circuits, confirming that TNBS is suitable both for comparing different architectures and for analyzing the effects of various noise types.

Quantum Technology	Model	N qubits	Noise Model	KS	KS std	KL	KL std	Elapsed time	Elapsed time std
-	*Noiseless*	4	–	0.0048	0.0018	0.00037	0.00014	0.09	0.06
-	*Noiseless*	6	–	0.0027	0.0009	0.00038	0.00007	0.38	0.14
IBM	*Noise + Decomposition*	4	*Idle*	0.020	0.003	0.0031	0.0005	0.66	0.11
	Noise + Decomposition	4	*PD*	0.0047	0.0018	0.00038	0.00014	0.66	0.13
	Noise + Decomposition	4	*AD*	0.020	0.003	0.0030	0.0005	0.69	0.19
	Noise + Decomposition	4	*Depolarizing Channel*	0.0241	0.0018	0.0090	0.0009	0.71	0.13
	Noise + Decomposition	4	*Depolarizing + Idle*	0.040	0.003	0.0148	0.0012	0.82	0.22
Quantinuum	*Noise + Decomposition*	4	*Idle*	0.0048	0.0018	0.00038	0.00014	0.66	0.14
	Noise + Decomposition	4	*PD*	0.0048	0.0018	0.00038	0.00013	0.71	0.14
	Noise + Decomposition	4	*AD*	0.0047	0.0018	0.00037	0.00014	0.72	0.14
	Noise + Decomposition	4	*Depolarizing Channel*	0.0061	0.0018	0.0006	0.0002	0.67	0.10
	Noise + Decomposition	4	*Depolarizing + Idle*	0.0062	0.0018	0.0006	0.0002	0.77	0.13

Table 1: Summary results of benchmark case PL.

Model	N qubits	Absolute Error	Absolute Error std	Elapsed time	Elapsed time std
Noiseless	4	0.0007	0.0006	43	3
Noiseless	6	0.0007	0.0005	88	8

Table 2: Summary results of benchmark case AE.

Model	N qubits	N aux qubits	Angle Method	KS	KS std	Fidelity	Fidelity std	Elapsed time	Elapsed time std
Noiseless	4	4	*Exact*	–	–	0.99979	0.00016	0.11	0.06
Noiseless	4	4	*Random*	0.09	0.03	–	–	0.16	0.04
Noiseless	4	8	*Exact*	–	–	0.9998	0.0002	0.29	0.04
Noiseless	4	8	*Random*	0.063	0.005	–	–	0.33	0.03

Table 3: Summary results of benchmark case QPE.

Quantum Technology	Model	Nº qubits	Layers	Noise models	Energy	Energy std	Elapsed time	Elapsed time std
—	*Noiseless*	3	1	—	−0.001	0.006	0.97	0.09
—	*Noiseless*	4	1	—	−0.003	0.007	4.95	0.07
—	*Noiseless*	6	2	—	−0.002	0.012	17.9	0.2
—	*Noiseless*	8	4	—	−0.003	0.014	495	70
—	*Noiseless*	15	1	—	−0.001	0.010	6.3	0.5
—	*Noiseless*	20	1	—	0.003	0.008	18.2	1.0
—	*Noiseless*	22	1	—	−0.009	0.013	960	30
—	*Noiseless*	23	1	—	−0.0008	0.014	2060	90
IBM	*Noise + Decomposition*	4	1	*Idle*	0.003	0.009	22.5	0.3
	Noise + Decomposition	4	1	*PD*	0.001	0.009	20.9	0.5
	Noise + Decomposition	4	1	*AD*	0.003	0.006	22.1	1.8
	Noise + Decomposition	4	1	*Depolarizing Channel*	0.099	0.007	29.27	0.16
	Noise + Decomposition	4	1	*Depolarizing Channel + Idle*	0.099	0.007	28.9	0.4
	Noise + Decomposition	6	2	*Idle*	0.002	0.010	485	13
	Noise + Decomposition	6	2	*PD*	0.002	0.011	472	13
	Noise + Decomposition	6	2	*AD*	0.001	0.011	480	20
	Noise + Decomposition	6	2	*Depolarizing Channel*	0.412	0.012	510	20
	Noise + Decomposition	6	2	*Depolarizing Channel + Idle*	0.411	0.012	502	18
Quantinuum	*Noise + Decomposition*	4	1	*Idle*	0.007	0.010	26	2
	Noise + Decomposition	4	1	*PD*	0.0078	0.0017	21.8	0.3
	Noise + Decomposition	4	1	*AD*	0.010	0.007	25	2
	Noise + Decomposition	4	1	*Depolarizing Channel*	0.028	0.007	28	3
	Noise + Decomposition	4	1	*Depolarizing Channel + Idle*	0.028	0.004	24.11	0.29
	Noise + Decomposition	6	2	*Idle*	5.484	0.009	495	10
	Noise + Decomposition	6	2	*PD*	5.478	0.011	470	20
	Noise + Decomposition	6	2	*AD*	5.481	0.012	443	13
	Noise + Decomposition	6	2	*Depolarizing Channel*	5.464	0.011	490	20
	Noise + Decomposition	6	2	*Depolarizing Channel + Idle*	5.460	0.011	524	14

Table 4: Summary results of benchmark case PH.

Appendices

A. Noise settings tables

To implement the noise models, it is necessary to have specific calibration parameters for each of the platforms. Table 5 and Table 6 for the IBM-Brisbane and Quantinuum-H1 technologies, including their associated values in terms of decoherence times, gate times, gate errors, and SPAM matrices.

	Parameter	Value
Times	$T1$	$2.7423 \times 10^5\ ns$
	$T2$	$1.7195 \times 10^5\ ns$
Gate times	$1\ qubit$	$6 \times 10^1\ ns$
	$2\ qubits$	$6.6\ \times 10^2\ ns$
	$state\_prep$	$4 \times 10^3\ ns$
	$measurement$	$4 \times 10^3\ ns$
Error Gates	$1\ qubit$	2.2452×10^{-4}
	$2\ qubits$	6.8449×10^{-3}
SPAM	$state\_prep$	$\begin{pmatrix} 0.977 & 0 \\ 0 & 0.023 \end{pmatrix}$
	$measurement$	$\begin{pmatrix} 0.0227 & 0 \\ 0 & 0.9773 \end{pmatrix}$

Table 5: Calibration parameters used for the IBM Brisbane backend.

	Parameter	Value
Times	$T1$	$2 \times 10^9\ ns$
	$T2$	$2 \times 10^9\ ns$
Gate times	$1\ qubit$	$5\ \times 10^3\ ns$
	$2\ qubits$	$2.5\ \times 10^4\ ns$
	$state\_prep$	$5\ \times 10^3\ ns$
	$measurement$	$5\ \times 10^3\ ns$
Error Gates	$1\ qubit$	4×10^{-5}
	$2\ qubits$	2×10^{-3}
SPAM	$state\_prep$	$\begin{pmatrix} 0.997 & 0 \\ 0 & 0.003 \end{pmatrix}$
	$measurement$	$\begin{pmatrix} 0.003 & 0 \\ 0 & 0.0997 \end{pmatrix}$

Table 6: Calibration parameters used for the H1 Quantinuum backend.

B. Summary tables of number of gates

In this section, the tables that show the number of gates in the circuits implemented for each benchmark case are presented as a summary and comparison, highlighting how they vary according to the decomposition into native gates of IBM and Quantinuum technologies. Table 7 and Table 8 describe the number of quantum gates in benchmark case PL and PH.

Circuit	Qubits	Parametrized gates	2 qubit gates	Total gates
Inicial	4	15	14	29
IBM	4	15	14	29
Quantinuum	4	55	14	69
Inicial	6	64	63	127
IBM	6	64	63	127
Quantinuum	6	200	53	253

Table 7: Benchmark case PL gates comparison.

Circuit	Qubits	Parametrized gates	2 qubit gates	Total gates
Inicial	4	8	4	12
IBM	4	8	8	16
Quantinuum	4	40	4	44
Inicial	6	24	12	36
IBM	6	24	12	36
Quantinuum	6	18	24	42

Table 8: Benchmark case PH gates comparison.