Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Average Quantum Relative Entropy and Transpilation Depth as Indicators of Noise Robustness in Quantum Binary Classification

Aakash Ravindra Shinde

shinde.aakashravindra@helsinki.fi

University of Helsinki

Arianne Meijer - van de Griend

University of Helsinki

Jukka K. Nurminen

University of Helsinki

Research Article

Keywords: Quantum Machine Learning, Quantum Relative Entropy, Quantum Parameterized Binary Classification, Transpilation Depth, Noisy Quantum Hardware, Quantum Noise Channels

Posted Date: July 8th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10244202/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Average Quantum Relative Entropy and Transpilation Depth as Indicators of Noise Robustness in Quantum Binary Classification

Aakash Ravindra Shinde[1*], Arianne Meijer - van de Griend[1] and Jukka K. Nurminen[1]

[1*]Department of Computer Science, University of Helsinki, Finland.

*Corresponding author(s). E-mail(s): shinde.aakashravindra@helsinki.fi;
Contributing authors: arianne.vandegriend@helsinki.fi;
jukka.k.nurminen@helsinki.fi;

Abstract

Quantum machine learning models trained under ideal (noiseless) conditions often exhibit significant performance degradation when executed on noisy quantum systems. Ideal simulation accuracy or circuit depth alone does not explain why classifiers with similar structures can show markedly different robustness. A practical question is therefore what factors can explain this difference in accuracy, and whether it can be anticipated and classically verified before costly hardware execution. We propose combining the average quantum relative entropy between classes and backend-specific transpilation depth as a practical pre-hardware indicator of the discrepancy in accuracy between ideal and noisy execution in Quantum Parameterized Binary Classification(QPBC) models. Average quantum relative entropy captures the distinguishability of learned class states, while transpilation depth serves as a practical indicator for cumulative noise exposure following compilation. Building on distinguishability and the contraction of quantum relative entropy under noisy channels, we theoretically derived indicators for both noisy-execution accuracy and simulation-to-device accuracy loss. We support this analysis with controlled experiments across multiple QPBC architectures, ansatz families, encoding methods, measurement strategies, datasets, and noisy backends. Our results show that classifiers with similar ideal accuracy and comparable transpilation depth can nonetheless exhibit substantially different degradation under noise. In contrast, the combined behavior of transpilation depth and average quantum relative entropy provides a more consistent interpretation of these robustness differences than either quantity alone. These

findings support a practical pre-hardware assessment approach for screening trained quantum classifiers according to expected noisy-execution robustness.

Keywords: Quantum Machine Learning; Quantum Relative Entropy; Quantum Parameterized Binary Classification; Transpilation Depth; Noisy Quantum Hardware; Quantum Noise Channels

1 Introduction

Variational Quantum Algorithms (VQAs) have emerged as a promising approach for Noisy Intermediate-Scale Quantum (NISQ) devices due to their potential resilience to noise and adaptability across a wide range of applications (Sharma et al. 2020; Cerezo et al. 2021). In the domain of machine learning, particularly concerning classification tasks, VQAs can take the form of Variational or Parameterized Quantum Classifiers. In this work, we focus on the binary classification setting of this family, referred to here as Quantum Parameterized Binary Classification (QPBC). Because classification is inherently decision-based and has probabilistic certainty while classifying, moderate stochastic uncertainty in measurements does not necessarily affect predictive performance by a lot. This property can make such models comparatively more tolerant of measurement variability than some other quantum learning tasks. However, excessive uncertainty can still degrade class separability and lead to misclassifications.

In the current NISQ landscape, we have access to a variety of quantum devices that are not fully error-corrected. Noise significantly affects the overall implementation of quantum circuits and skews the results (Lidar and Brun 2013). Executing quantum algorithms on current NISQ hardware involves several challenges, including noisy systems, readout errors, limitations of native gate sets, qubit routing overhead, and calibration drifts. For VQAs, additional limitations arise from finite-shot sampling and the monetary cost of repeated circuit execution during training and evaluation (Phalak and Ghosh 2023). While extensive research has been conducted to enhance these approaches and their applications, these methods are often trained and tested in noise-free simulations on classical hardware. This raises concerns about the reproducibility of results on noisy quantum systems.

Existing explanations often emphasize circuit depth or the intuition that shallower circuits tend to be more resilient to noise (Sharma et al. 2020); however, such accounts remain incomplete. Empirical executions on NISQ devices nevertheless show that models with similar logical structures or comparable depths can exhibit substantially different noisy-execution behavior (Khanal and Rivas 2023). In practice, models with comparable circuit depths can still demonstrate significantly different behaviors during noisy execution. This variability may be attributed to differences in the distinguishability of their learned class states, the manner in which their compiled forms interact with backend-specific noise, and potential non-trivial overhead introduced during transpilation beyond the logical circuit design. Consequently, a more comprehensive understanding of noisy robustness is essential.

Restricting the analysis to QPBCs provides a controlled framework for examining the performance and reliability of trained variational classifiers under realistic noisy execution. This setting enables a more precise investigation of their robustness and operational behavior in practical scenarios. It is also well-suited for both theoretical analysis and experimental evaluation, allowing us to explore different implementation strategies and to compare model performance in ideal and noisy environments across a range of quantum platforms, including noisy simulators and hardware backends.

This work addresses the following research questions.

1. **RQ1:** Why do some QPBC circuits perform better than others at similar transpilation depth?
2. **RQ2:** What degradation trends could be observed when a trained QPBC model is executed on noisy quantum hardware?
3. **RQ3:** Can we obtain a classical pre-execution indication of this degradation before hardware execution?

These questions are motivated by a practical push for a train-classical, deploy-quantum (Recio-Armengol et al. 2026) setting in which access to hardware is limited and costly. Yet model selection requires anticipating whether strong ideal-simulation performance is likely to survive realistic noisy execution.

Our central hypothesis is that the noisy robustness of Quantum Parameterized Binary Classification (QPBC) models can be partially interpreted through two complementary factors. First, the quantum states learned for different classes must remain distinguishable under noise; otherwise, predictive performance will inevitably degrade. To effectively measure class-state distinguishability in an ideal simulation, we analyze the average quantum relative entropy between the learned quantum states associated with respective classes. Second, the degree of noise exposure is affected not only by the abstract design of the circuit but also by the backend-specific implementation; thus, transpilation depth serves as a practical indicator of the cumulative exposure to noisy channels, which includes routing and basis-gate overhead introduced by the target device. Accordingly, we propose that the combination of average quantum relative entropy and transpilation depth may act as a practical indicator of QPBC model performance under noisy execution.

The contributions of this article are as follows.

1. **C1:** We define an empirical average quantum relative entropy for trained QPBC models as a pre-measurement class-state distinguishability indicator.
2. **C2:** We derive distinguishability-motivated degradation indicators linking relative-entropy contraction, backend-specific transpilation depth, and noisy-execution robustness trends.
3. **C3:** We evaluate these indicators across multiple QPBC architectures, ansatz families, encodings, measurement rules, datasets, and noisy backend simulators, with selected real hardware consistency checks.
4. **C4:** We show through representative cases and backend-wise observations that ideal accuracy and transpilation depth alone are insufficient to interpret noisy-execution outcomes.

5. **C5:** We provide a classical pre-assessment methodology for screening trained QPBC models according to expected degradation trends before hardware execution.

The scope of this study is intentionally constrained. We focus exclusively on binary classification, use fixed 8-qubit experiments, and restrict our attention to selected trained QPBC models that demonstrate high accuracy in ideal simulations. This approach allows us to focus on post-training performance degradation rather than on failure to learn. We do not assert any quantum advantage over classical machine learning methods. Instead, we aim to understand under what conditions trained QPBC models are likely to remain reliable in the presence of realistic noise and when they may fail to do so. Additionally, while we incorporate hardware validation whenever possible, such validation is inherently limited by factors such as restricted access, queue times, and execution costs. Consequently, a significant portion of this study relies on noisy backend simulators, supplemented by select hardware experiments.

The remainder of this article is organized as follows. Section 2 reviews related work on variational quantum classifiers, noisy VQA execution, hardware-aware implementation, and quantum state distinguishability. Section 3 develops the theoretical relation between average quantum relative entropy, noisy-channel contraction, transpilation depth, and accuracy degradation. Sections 4 and 5 describe the experimental components and experimental procedure, respectively. Section 6 presents the empirical observations across noisy simulators and selected hardware executions. Finally, Sections 7 and 8 discuss the implications, limitations, future work, and conclusions of the study. Additional information and resources are provided in the Appendix and Code repository to substantiate the findings.

2 Related Work

Variational Quantum Algorithms (VQAs) are classical-quantum hybrid methods that have been an extensively researched paradigm for NISQ devices due to their applicability to several problems, ease of implementation, and their ability to keep circuits shallow and tolerate certain noise sources to some degree (Cerezo et al. 2021). Within Quantum Machine Learning (QML), variational or parameterized quantum classifiers combine data-encoding circuits, trainable quantum transformations, classical optimizers, and measurements to map classical inputs to predicted labels (Hur et al. 2022; Hubregtsen et al. 2021; Pérez-Salinas et al. 2020). This work primarily focuses on the binary classification setting, which we refer to as Quantum Parameterized Binary Classification (QPBC), as a specific instance of the broader family of variational quantum classifiers.

Prior work on variational classifiers has examined model performance from several perspectives, including circuit expressibility (Du et al. 2020), classical simulability (Bermejo et al. 2024), entangling capability, data encoding (Zang et al. 2025), trainability (Larocca et al. 2025), benchmarking practices (Bowles et al. 2024), and the effect of hardware noise (Buonaiuto et al. 2024). Despite this extensive research, it remains largely unexplained why trained models with comparable transpilation depth can exhibit different levels of accuracy degradation under noisy execution. This section

reviews the literature relevant to this gap and positions average quantum relative entropy between classes as a complementary indicator of noisy robustness.

Substantial research has been conducted to investigate the effect of noise on variational quantum algorithms (VQAs), with particular attention to whether such algorithms can be executed reliably on NISQ devices. One line of research suggests that some variational procedures can tolerate specific noise processes. Sharma et al. (2020), for instance, showed that the VQAs can exhibit resilience to measurement and depolarizing noise. Fontana et al. (2021) and Khanal and Rivas (2023) provide empirical analysis of observed noise resilience in shallow (low-depth) variational or parameterized quantum circuits.

The association between noise accumulation and circuit depth is widely recognized (Cerezo et al. 2021), but it provides limited guidance for determining what depth remains acceptable for a specific VQA circuit and target backend. This also raises the question of what is meant by 'depth', as there is a significant difference in depth observed at a higher algorithmic level and the circuit depth observed on the device-level implementation, also known as transpilation depth. Transpilation depth depends on hardware connectivity, native gate sets, routing, and basis-gate decomposition, all of which can influence fidelity and output variability (Huo et al. 2025).

In contrast, other studies emphasize the limitations imposed by noise. Noise-induced barren plateau results show that hardware noise can lead to vanishing gradients and thereby obstruct VQA training, as addressed by the works of Wang et al. (2021) and reviewed by Larocca et al. (2025). Similarly, Singkanipa and Lidar (2025) analyzed the effects of the noise-induced barren plateau phenomenon in VQA training under general completely positive trace-preserving (CPTP) maps. Bärligea et al. (2025) presented research involving VQE and QAOA algorithms under stochastic noise and raised concerns regarding the scalability of VQE algorithms under such conditions.

Together, these perspectives provide important insights into how VQAs behave under noise, both during execution and training. However, they do not fully explain why trained classifiers in ideal simulations with comparable backend-specific circuit depths can exhibit different levels of degradation under noisy execution. This distinction is essential to train-classical, deploy-quantum settings (Recio-Armengol et al. 2026). The present work therefore focuses on post-training robustness and studies how expected degradation can be interpreted through the combined effect of transpilation depth and average quantum relative entropy for state distinguishability. In particular, quantum relative entropy is contractive under quantum channels (De Palma et al. 2023; Wilde 2014), i.e.

$$\mathcal{D}(\Phi(\rho)\|\Phi(\sigma)) \leq \mathcal{D}(\rho\|\sigma) \tag{1}$$

This property is particularly relevant for noisy execution, where a transpiled circuit can be interpreted as a sequence of noisy operations whose length is related to the backend-specific transpilation depth. Since different datapoints induce different initial quantum states, we use the average quantum relative entropy between states from opposite classes as an indicator of expected post-training robustness.

The scope of this work is intentionally separate from areas such as error mitigation, error correction, noise-aware compilation, and noisy-training strategies. We do not propose a method for suppressing hardware noise, altering the transpilation process, enhancing the training landscape, or demonstrating quantum advantage over classical machine learning methods. Instead, our focus lies in the post-training deployment context: specifically, we examine why the performance of QPBC models—having achieved high accuracy under ideal simulation—may differ during noisy execution, and we assess whether the anticipated degradation can be evaluated classically before hardware execution.

3 Theoretical Analysis

In this section, we provide a theoretical analysis to establish a qualitative relationship among the average quantum relative entropy between classes, transpilation depth, and the possible accuracy difference observed between ideal simulation and noisy execution for a quantum parameterized binary classification (QPBC) model. We also provide a theoretical indicator for the error that can be observed in a given noise channel, along with an indicative measure of the perceived difference in accuracy under similar constraints.

3.1 Quantum Parameterized Binary Classification (QPBC) model definition

A broad class of parameterized binary quantum classification models could be defined as follows. Let the initial state be $\rho_{init} = |0\rangle^{\otimes n} \langle 0|^{\otimes n}$ and let the dataset be $\{x_i, y_i\}_{i=1}^{k}$ where $y_i \in \{0, 1\}$ for probabilistic measurement or $y_i \in \{-1, 1\}$ for expectation value measurement in case of binary classification problems. Hence, data encoding procedure converts the initial state ρ_{init} to $\rho(x)$ by applying encoding unitary $U(x)$ as $\rho(x) = U(x)\rho U^{\dagger}(x)$. Further, another set of unitaries $U(\theta)$ are trained where θ is a set of trainable parameters(rotational values) based on loss function defined as $\mathcal{L}(\theta, X, y) = \frac{1}{k}\sum_i l(\theta, x_i, y_i)$, resulting state $\rho(x_i, \theta) = U(\theta)\rho(x_i)U^{\dagger}(\theta)$. Finally, the prediction value $\hat{y}_i$ is obtained from the measurement operator $\{M\}$. Hence, $\hat{y} = Tr[M\rho(x_i, \theta)]$ provides the predicted output for the input state $\rho(x_i)$ (Bowles et al. 2024). The classification model intends to best distinguish between state $\rho(x_i, \theta)$ and $\rho(x_j, \theta)$ where $y_i = 0\ \forall i$ and $y_j = 1\ \forall j$. Let ρ_0 and ρ_1 denote representative pre-measurement states for the two classes. If the $U(\theta)$ is a good classifier, the states ρ_0 and ρ_1 are easy to distinguish.

3.2 Maximum achievable accuracy

Post-measurement for a fixed measurement (POVM) operation $\{M\}$, these values from the classification model for ρ_0 and ρ_1 can be considered classical values; hence, their probability distribution can be given as $\{p_0, p_1\}$ for each class. Therefore, the corresponding decision error can be treated as a classical discrimination error. By Bayes error rates rule, for balanced binary classes with equal priors, the minimum error for distinguishing two classical output probability distributions is (P_{err}^{T}), which can

theoretically be given as,

$$P_{err}^{T} = \frac{1}{2}(1 - d_{tv}(p_0, p_1)), \tag{2}$$

where total variational distance $d_{tv}(p, q)$ is defined as $d_{tv}(p, q) = \frac{1}{2}\int |p(x) - q(x)|dx$ (Nielsen 2014). The maximum possible accuracy for a QPBC model can be further defined as $1 - P_{err}^{T}$. Therefore, the accuracy equation can be written as,

$$Acc^{T} = \frac{1}{2}(1 + d_{tv}(p_0, p_1)), \tag{3}$$

From Nielsen and Chuang (2000), we can establish the correlation between trace distance between ρ_0 and ρ_1 and total variational distance of their probability distributions p_0 and p_1 as $d_{tv}(p_0, p_1) \leq T(\rho_0, \rho_1)$ where $T(\rho_0, \rho_1) = \frac{1}{2}\|\rho_0 - \rho_1\|$. Hence, we can express an achievable-accuracy envelope in terms of trace distance as,

$$Acc^{T} \leq \frac{1}{2}(1 + T(\rho_0, \rho_1)), \tag{4}$$

The trace-distance expression should be interpreted as an achievable-accuracy envelope associated with optimal distinguishability, whereas the implemented classifier uses a fixed measurement rule. This gap is later represented phenomenologically through the coefficient c.

3.3 Quantum Relative Entropy

Quantum relative entropy can be interpreted as a measure of distinguishability between two quantum states (Wilde 2014). Quantum relative entropy(Von Neumann relative entropy) between a density operator $\rho \in \mathcal{D}(\mathcal{H})$ and a positive semi-definite operator $\sigma \in \mathcal{L}(\mathcal{H})$ is defined as (Wilde 2014),

$$\mathcal{D}(\mu||\nu) \equiv Tr[\mu(\log \mu - \log \nu)], \tag{5}$$

with $\mathcal{D}(\mu||\nu) = +\infty$ when $supp(\mu) \not\subseteq supp(\nu)$. Hence, in our case, the quantum relative entropy between the two quantum states ρ_0 and ρ_1 for classification can be denoted as $\mathcal{D}(\rho_0||\rho_1)$. If $\mathcal{D}(\rho_0||\rho_1) = 0$, then $\rho_0 = \rho_1$, meaning that the two states are indistinguishable.

3.4 Noise Channel Definition, Effective Depth and Observable Error Bound

We now analyze how much error can be observed in a noise channel of depth L. We first define the noise channel. For analytical tractability, we assume a local Markovian channel $\mathcal{N}$ that can be written as a perturbation over L layers. Applying the noisy channel $\mathcal{N}_L$ to the quantum state σ would result in the quantum state σ_{noise}. Let $\mathcal{U}$ be the ideal channel resulting in quantum state σ_{ideal} when applied to quantum state σ and $\mathcal{E}$ be the error-induced channel post each layer.

In this subsection, $\mathcal{N}_L$ denotes the effective number of noisy operational layers in the channel representation. Each layer is treated as an ideal unitary operation followed by a noise contribution. At this stage, $\mathcal{N}_L$ is an abstract channel depth rather than a directly measured circuit property. This distinction is important because the logical circuit depth before compilation may differ substantially from the physical depth after routing, basis-gate decomposition, and hardware-specific compilation. Therefore,

$$\mathcal{N}_L = \mathcal{E}_L \circ \mathcal{U}_L \ldots \mathcal{E}_2 \circ \mathcal{U}_2 \circ \mathcal{E}_1 \circ \mathcal{U}_1 \tag{6}$$

As mentioned by Lidar and Brun (2013), we can further define $\|\mathcal{N}_L - \mathcal{U}\| \leq \sum_{i=0}^{L} \delta_i = \delta$ where δ_i is the overall fault contribution at each layer. Following Lidar and Brun (2013), noise strength of full channel $\mathcal{N}_L$ is $\epsilon = \underset{j}{max} \|\mathcal{N}_j - \mathcal{I}\|_\diamond$, under the assumptions stated above.

We can further define the distance between the σ_{noisy} and σ_{ideal} post measurement for outcome $\{\mu\}$ as

$$\delta = \sum |Tr[M_\mu(\sigma_{noisy} - \sigma_{ideal})]|, \tag{7}$$

where M_μ represents the final projection measurement. Under this formulation, it follows that

$$\delta \leq L\epsilon \tag{8}$$

This equation provides us with bounds over the maximum difference observed in a measurement of σ_{noisy} and σ_{ideal} in a noisy channel of depth L (Lidar and Brun 2013). But it doesn't address the maximum difference one can observe while trying to differentiate two different states on a noisy and ideal system. Similarly, it doesn't explain the performance difference between models based on prediction, especially for models with similar depth and structure.

3.5 Quantum Relative Entropy in Quantum Channel of Depth (L)

We now examine how the quantum relative entropy between ρ_0 and ρ_1 would behave in a noisy channel as mentioned in Eq. (6). Quantum relative entropy can either decrease or remain unchanged when a quantum channel is applied to it (Wilde 2014). From the Strong Data Processing Inequality (SDPI), we can define the SDPI constant η_Φ as the change in quantum relative entropy post-acting of channel Φ.

$$\eta_\Phi = \underset{\nu \neq \mu}{sup} \frac{\mathcal{D}(\Phi(\nu)||\Phi(\mu))}{\mathcal{D}(\nu||\mu)} \tag{9}$$

The coefficient η_Φ is the strong data-processing contraction coefficient of the channel Φ. It quantifies the largest possible fraction of quantum relative entropy that can remain after the channel acts on two input states. For noisy channels that reduce distinguishability, $0 \leq \eta_\Phi \leq 1$. A value close to one indicates that the channel approximately preserves relative entropy, while a smaller value indicates stronger contraction of distinguishability.

Eq. (9) can be defined as follows in the context of the noisy quantum channel and quantum relative entropy,

$$\mathcal{D}(\mathcal{N}_k(\rho_0)||\mathcal{N}_k(\rho_1)) \leq \eta_k \cdot \mathcal{D}(\rho_0||\rho_1), \tag{10}$$

In our setting for the loss in quantum relative entropy between ρ_0 and ρ_1 when acted upon by the noise channel $\mathcal{N}_k$ at depth k. It can be further illustrated that,

$$\mathcal{D}(\mathcal{N}_2(\rho_0)||\mathcal{N}_2(\rho_1)) \leq \eta_1 \cdot \mathcal{D}(\mathcal{N}_1(\rho_0)||\mathcal{N}_1(\rho_1)), \tag{11}$$

Therefore, for a quantum channel of depth L, the entropy reduction over L layers can be denoted as,

$$\mathcal{D}(\mathcal{N}_L(\rho_0)||\mathcal{N}_L(\rho_1)) \leq \prod_{i=1}^{L} \eta_i \cdot \mathcal{D}(\rho_0||\rho_1), \tag{12}$$

The product of SDPI coefficients $\prod_{i=1}^{L} \eta_i$ represents the cumulative distinguishability-retention factor over the full noisy evolution. If every layer preserves distinguishability well, this product remains close to one. If many layers introduce appreciable contraction, the product decreases, indicating that the class states become less distinguishable after noisy execution.

If it is assumed that each noise channel acting on the system has approximately the same amount of information loss, i.e., $\eta_j = \eta_k, \forall\{j,k\} \in \{1, L\}, j \neq k$, then the equation can be modified as,

$$\mathcal{D}(\mathcal{N}_L(\rho_0)||\mathcal{N}_L(\rho_1)) \leq \eta^L \cdot \mathcal{D}(\rho_0||\rho_1), \tag{13}$$

This provides us with the factor at which quantum relative entropy degrades in a noisy quantum channel. We further presume that the depth of the quantum channel can be associated with the transpilation depth. As for Eq. (6), a quantum channel is defined as a set of noisy channel perturbations over L layers.

We approximate the effective noisy-channel depth by the backend-specific transpilation depth. The noisy unitary is subdivided based on the number of parallel operations in the transpiled circuit, which is the depth of the transpiled circuit. Hence, in the following calculation, we associate a circuit with a transpilation depth L_{tr} with a quantum channel decomposition consisting of L steps, where each circuit layer is represented by a CPTP map $\mathcal{U}'_k$, and the overall channel is $\mathcal{U}'_{L_{tr}} = \mathcal{U}'_L \circ \mathcal{U}'_{L-1} \circ \cdots \circ \mathcal{U}'_2 \circ \mathcal{U}'_1$. For all the following expressions, $L = L_{tr}$.

3.6 Relating Relative Entropy, Fidelity, Trace Distance, and Accuracy Degradation

To establish a relation between the difference in prediction accuracy between ideal simulation and noisy execution and the quantum relative entropy in a noise channel, let's first define the relation among trace distance, quantum relative entropy, and fidelity.

Fidelity is another metric that defines the closeness of two density operators and is defined as $\mathcal{F}(\rho,\sigma) = (Tr\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}})$. Min-relative entropy, described as the worst-case measure of distinguishability between a density operator ρ and another density operator σ (Datta 2009). From Nuradha and Wilde (2024), min-relative entropy can be established in terms of fidelity as $\mathcal{D}_{min}(\rho||\sigma) := -\log_2(\mathcal{F}(\rho,\sigma))$, also from our perspective $\mathcal{D}_{min}(\rho_0||\rho_1)) \leq \mathcal{D}(\rho_0||\rho_1))$ and hence we can define fidelity in terms of quantum relative entropy as,

$$\mathcal{F}(\rho_0,\rho_1) \geq 2^{-\mathcal{D}(\rho_0||\rho_1)} \tag{14}$$

From the Fuchs–van de Graaf inequalities (Nielsen and Chuang 2000), we get the upper-bound relation between Trace distance and fidelity as $T(\rho_0,\rho_1) \leq \sqrt{1-\mathcal{F}(\rho_0,\rho_1)}$. Substituting this inequality and Eq. (14) in (4), we get the subsequent equation for the accuracy and relative entropy relation in an ideal simulation,

$$Acc^T(\rho_0,\rho_1) \leq \frac{1}{2}\left(1+\sqrt{1-2^{-\mathcal{D}(\rho_0||\rho_1)}}\right), \tag{15}$$

This equation motivates the interpretation that a larger quantum relative entropy can support a higher achievable classification accuracy based on distinguishability in an ideal simulation. However, because quantum relative entropy can, in principle, diverge while accuracy is bounded by 1, the relation is not directly informative for optimizing accuracy alone. But this equation can be further modified to get the noisy execution accuracy and quantum relative entropy relation in a noise channel with the help of Eq. (12) as,

$$Acc^T(\mathcal{N}_L(\rho_0),\mathcal{N}_L(\rho_1)) \leq \frac{1}{2}\left(1+\sqrt{1-2^{-\left(\prod_{i=1}^{L}\eta_i\cdot\mathcal{D}(\rho_0||\rho_1)\right)}}\right), \tag{16}$$

Therefore ΔAcc^T could be defined as the difference between the $Acc^T(\rho_0,\rho_1)$ and $Acc^T(\mathcal{N}_L(\rho_0),\mathcal{N}_L(\rho_1))$ which is also the accuracy difference of the models in a ideal simulation execution and noisy channel execution,

$$\Delta Acc^T = Acc^T(\rho_0,\rho_1) - Acc^T(\mathcal{N}_L(\rho_0),\mathcal{N}_L(\rho_1)), \tag{17}$$

Therefore, by re-substituting values from Eq. (16) and Eq. (15) into the above Eq. (17), we obtain a corresponding distinguishability-based expression that captures the change in the achievable accuracy envelope between ideal and noisy settings.

$$\Delta Acc^T \propto \frac{1}{2}\left(\sqrt{1-2^{-\mathcal{D}(\rho_0||\rho_1)}} - \sqrt{1-2^{-\left(\prod_{i=1}^{L}\eta_i\cdot\mathcal{D}(\rho_0||\rho_1)\right)}}\right) \tag{18}$$

It is crucial to understand that Eq. (18) should not be construed as a definitive upper limit on the observed accuracy difference. The expressions for both ideal and noisy

accuracy are themselves upper bounds derived from distinguishability arguments. Therefore, the difference between the two upper bounds does not typically provide a valid upper limit on the difference of the underlying quantities. Consequently, Eq. (18) should be regarded as an indicator of degradation based on distinguishability. It reflects the expected monotonic relationship between accuracy degradation and the initial class-state distinguishability, as well as its reduction under the influence of noise.

3.7 Practicalities of Quantum Parameterized Binary Classification (QPBC) models

3.7.1 Average Quantum Relative Entropy between Classes

In the practical implementation of any Quantum parameterized binary classification algorithm for each data value, $\{x_i, y_i\}_{i=1}^{k}$ where $y_i \in \{0, 1\}$, there is a difference in creating the initial state for classification, resulting in pre-measurement states approximately equal to ρ_0 and ρ_1 i.e. $\rho(x_i, \theta) \approx \rho_0$ and $\rho(x_j, \theta) \approx \rho_1$ where $y_i = 0$ and $y_j = 1$. This further creates approximations in their respective probability distributions. Therefore, comparing the relative entropy of only one pair of data points from different classes is insufficient. Hence, we propose the average quantum relative entropy between classes using the test dataset, which yields,

$$\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1}) := \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathcal{D}(\rho_i||\rho_j), \tag{19}$$

where, $y_i = 0, y_j = 1$ and $i \in \{0, m\}$ and $j \in \{0, n\}$. and ρ_i, ρ_j are pre-measurement states for respective data values of said classes, while notations $\hat{\rho_0}$ and $\hat{\rho_1}$ denotes the empirical class ensembles rather than single averaged density matrices. Thus, $\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})$ is an empirical pairwise cross-class average. It measures how distinguishable the learned pre-measurement states of one class are, on average, from the learned pre-measurement states of the other class. A larger value indicates stronger learned class separation before measurement.

Quantum relative entropy is asymmetric in general i.e., $\mathcal{D}(\rho_i||\rho_j) \neq \mathcal{D}(\rho_j||\rho_i)$. However, the quantity in Eq. 19 is the average of the quantum relative entropies across classes. If we calculate the difference in the bidirectional average quantum relative entropy as defined in Eq. (19),

$$\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1}) - \mathcal{D}_{avg}(\hat{\rho_1}||\hat{\rho_0}) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[\mathcal{D}(\rho_i||\rho_j) - \mathcal{D}(\rho_j||\rho_i)], \tag{20}$$

Hence, the bidirectional average can be close even when individual pairwise quantum relative entropies are asymmetric. In our experimental setting of the QPBC evaluations, the dataset is balanced, and both classes are processed by the same trained parameterized circuits, detailed observations for which are presented in Appendix Table B3. If the learned class-state ensembles share comparable spectral structures

and dispersions, their antisymmetric pairwise contributions may cancel out when averaged. Therefore, $\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1}) \approx \mathcal{D}_{avg}(\hat{\rho_1}||\hat{\rho_0})$ observed experimentally does not imply that the averaged quantum relative entropy itself is symmetric. Rather, it should be interpreted as an empirical ensemble-level property of the learned class representation. Therefore, in the main experimental analysis we report one directional average, $\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})$, while verifying separately that the opposite directional average gives nearly identical ensemble-level values for the evaluated models.

This modifies Eq. (16) and Eq. (18) as follows,

$$Acc^T(\mathcal{N}_L(\rho_0), \mathcal{N}_L(\rho_1)) \leq \frac{1}{2}\left(1 + \sqrt{1 - 2^{-\left(\prod_{i=1}^{L} \eta_i \cdot \mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})\right)}}\right), \tag{21}$$

$$\Delta Acc^T \propto \frac{1}{2}\left(\sqrt{1 - 2^{-\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})}} - \sqrt{1 - 2^{-\left(\prod_{i=1}^{L} \eta_i \cdot \mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})\right)}}\right) \tag{22}$$

This provides us with equations for determining the accuracy expected in a noisy channel execution and the accuracy difference observed between the simulated model and the noisy channel execution when tested over the same test dataset.

3.7.2 Post Measurement Threshold Approximations

In binary classification, the continuous model output is converted into a class label using a decision threshold. This decision threshold would be 0.5 for probabilistic measurement between classes '0' and '1', while in the case of expectation value-based models, the threshold would be 0 for values between '-1' and '1'.

The distinguishability quantity $\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})$ is computed from pre-measurement density matrices, whereas the classifier output is obtained only after measurement and classical thresholding. Measurement is a CPTP map and can reduce distinguishability. In addition, the conversion of measured probabilities or expectation values into class labels introduces a thresholding effect. Two models with similar pre-measurement distinguishability may therefore behave differently if their measured outputs lie at different distances from the decision boundary.

To account for this practical gap between pre-measurement distinguishability and post-measurement classification, we introduce a coefficient c. The coefficient c represents the effective fraction of the pre-measurement average quantum relative entropy that remains relevant for classification after measurement and thresholding under ideal simulation. It is not a fundamental constant; it is a practical calibration factor depending on the measurement rule, threshold, dataset, and trained model family.

Thus, we modify Eq. (15) as follows, providing us with the practical equation between the accuracy of a QPBC model and average quantum relative entropy,

$$Acc^P(\rho_0, \rho_1) \leq \frac{1}{2}\left(1 + \sqrt{1 - 2^{-(c \cdot \mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1}))}}\right), \tag{23}$$

Here, Acc^P denotes a practical post-measurement accuracy indicator. Unlike Acc^T in Eq. (15), Acc^P includes the empirical calibration coefficient c, which accounts for measurement and thresholding effects.

Under noisy finite-shot execution, the effective decision threshold can become less reliable. Gate noise, readout noise, finite sampling, and backend fluctuations can shift measured probabilities or expectation values toward the decision boundary or across it. We model this additional loss by introducing a nonnegative noisy-threshold error term err_n. The noisy post-measurement coefficient is then defined as $c_n = c - err_n$, where $0 \leq c_n \leq c$ in the typical case. The coefficient c_n represents the effective classification-relevant distinguishability after measurement, thresholding, finite-shot effects, and noisy execution. Hence, for the practical equation between the accuracy of a QPBC model and average quantum relative entropy for a model in a noisy channel of depth L, we update Eq. (16) as,

$$Acc^P(\mathcal{N}_L(\rho_0), \mathcal{N}_L(\rho_1)) \leq \frac{1}{2}\left(1 + \sqrt{1 - 2^{-\left(\prod_{i=1}^{L} \eta_i \cdot c_n \cdot \mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})\right)}}\right), \tag{24}$$

Hence, similarly, we can compute the expected accuracy difference for practical execution by updating Eq. (22) with Eq. (23) and Eq.(24), resulting in the following equation.

$$\Delta Acc^P \propto \frac{1}{2}\left(\sqrt{1 - 2^{-(c \cdot \mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1}))}} - \sqrt{1 - 2^{-\left(\prod_{i=1}^{L} \eta_i \cdot c_n \cdot \mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})\right)}}\right) \tag{25}$$

Eq. (25) should be interpreted as a practical degradation indicator. It indicates that accuracy degradation increases when the noisy circuit strongly contracts class-state distinguishability, when the transpilation depth is large, or when finite-shot and threshold effects reduce c_n. Conversely, if $\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})$ is large and L is small, the model is expected to retain more of its ideal-simulation performance under noisy execution.

3.7.3 Hardware-dependent Residual Effects and the Drift Coefficient

The preceding formulation accounts for three main effects: learned class-state distinguishability through $\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})$, cumulative noisy-channel contraction through $\prod_{i=1}^{L} \eta_i$, and post-measurement threshold effects through c and c_n. However, real quantum devices and their noise simulators can still differ from this formulated prediction because of calibration drift, imperfect noise-model estimation, qubit-dependent readout fidelity, crosstalk, temporal fluctuations, and other backend-specific effects. We collect these unmodeled backend-dependent residual effects into a residual drift coefficient δ_u defined as,

$$\delta_u := Acc^P_{form} - Acc^P_{obs} \tag{26}$$

Here, Acc^P_{form} denotes the formulated practical noisy-accuracy indicator obtained from Eq. (24), while Acc^P_{obs} denotes the accuracy observed from a noisy simulator or hardware execution. The observed practical noisy accuracy can therefore be written as the formulated noisy-accuracy indicator corrected by the residual drift term:

$$Acc^P_{obs}(\mathcal{N}_L(\rho_0), \mathcal{N}_L(\rho_1)) := Acc^P_{form}(\mathcal{N}_L(\rho_0), \mathcal{N}_L(\rho_1)) - \delta_u \tag{27}$$

Substituting Eq. (24) into this residual-corrected expression gives the following observed noisy-accuracy indicator:

$$Acc^P_{obs}(\mathcal{N}_L(\rho_0), \mathcal{N}_L(\rho_1)) \leq \frac{1}{2}\left(1 + \sqrt{1 - 2^{-\left(\prod_{i=1}^{L} \eta_i \cdot c_n \cdot \mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})\right)}}\right) - \delta_u \tag{28}$$

In most cases, δ_u is expected to be nonnegative, but finite-shot fluctuations, simulator-hardware mismatch, or noise-induced changes near the threshold can occasionally cause the observed noisy accuracy to exceed the formulated indicator. However, it is not always consistent.

We can further update the observed drift for the accuracy difference between the simulation and noisy execution as

$$\Delta Acc^P_{obs} := Acc^P_{sim}(\rho_0, \rho_1) - (Acc^P(\mathcal{N}_L(\rho_0), \mathcal{N}_L(\rho_1) - \delta_u) \tag{29}$$

Substituting the practical distinguishability expressions into the preceding equation gives the following residual-corrected degradation indicator:

$$\Delta Acc^P_{obs} \propto \frac{1}{2}\left(\sqrt{1 - 2^{-(c \cdot \mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1}))}} - \sqrt{1 - 2^{-\left(\prod_{i=1}^{L} \eta_i \cdot c_n \cdot \mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})\right)}}\right) + \delta_u \tag{30}$$

In Eq. 30, $\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})$ is computed classically from pre-measurement density matrices of test-set states belonging to opposite classes. L denotes the effective noisy-channel depth and is approximated experimentally by the backend-specific transpilation depth L_{tr}. The coefficients η_i describe layer-wise contraction of quantum relative entropy, and $\prod_{i=1}^{L} \eta_i$ represents the cumulative retention of class distinguishability after noisy execution. The coefficient c accounts for the reduction from pre-measurement distinguishability to post-measurement classification in ideal simulation, while c_n is the corresponding noisy-execution coefficient after finite-shot and noise-induced threshold effects. Finally, δ_u captures residual backend-dependent effects not explicitly represented by the entropy-depth formulation. These coefficients are practical calibration factors rather than universal constants.

Overall, the theoretical analysis links noisy robustness in QPBC models to two complementary quantities: the distinguishability of the learned class states, captured by $\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})$, and the cumulative exposure of the compiled circuit to backend-specific noise, approximated by the transpilation depth L_{tr}. Since quantum relative entropy contracts under noisy channels, class-state distinguishability is expected to

decrease during noisy execution, and this reduction becomes more relevant as the effective transpilation depth of circuits increases. The practical coefficients c, c_n, $\prod_{i=1}^{L} \eta_i$, and δ_u account for measurement thresholding, finite-shot effects, channel contraction, calibration drift, and other backend-dependent deviations. Therefore, Eq. (28) and Eq. (30) should be interpreted as degradation indicators rather than exact accuracy predictions or strict bounds. The main implication is that, among trained QPBC models with comparable ideal-simulation accuracy, models with larger average quantum relative entropy and lower backend-specific transpilation depth are expected to retain performance more reliably under noisy execution.

The following experimental section evaluates this expected relationship across different architectures, ansatzes, encodings, measurement strategies, datasets, and noisy backends.

4 Experimental Design and Components

Variational quantum classifiers can be effectively described by the theoretical framework outlined in Section 3. However, from an experimental perspective, the implementation of these mathematical representations involves a range of diverse architectures, circuit substructures (ansatzes), data encoding methods, and measurement techniques. Each of these factors impacts the achievable results. In addition to the theoretical constraints affecting the QPBC models, the dataset utilized for training these models plays a significant role in determining the outcomes, along with the specific training conditions. The theory presented in Section 3 establishes a qualitative relationship based on a general noise channel. Therefore, to generalize our findings, we needed to take into account different noisy quantum systems.

In this section, we present a comprehensive overview of the components involved in designing the experiments and conducting an empirical analysis of the theoretical qualitative relationship introduced in Section 3. We examine several major variables that may influence the overall accuracy of the models in both noisy and ideal systems to reduce the likelihood that the observed trend is caused by a single architecture, encoding, dataset, or backend. This approach enabled us to establish a generalizable qualitative relationship between the average quantum relative entropy between classes, as detailed in Eq. (19), transpilation depth, and the performance differences observed in a noisy system. Table 1 provides a summary of variables considered during the experimental procedure and the reasoning behind choosing the variations. Detailed definition and description for each of the experimental components are provided in Appendix A.

4.1 System Size

The experiments conducted on QPBC models were limited to an 8-qubit system size due to various constraints. This restriction allowed for a meaningful comparison of models with the same qubit size and expressibility within a similarly sized Hilbert space. Additionally, limiting the models to 8 qubits facilitated the evaluation of density matrices of comparable size for the computation of average quantum relative entropy, while also ensuring that the noisy channel size encountered by the models remained

Experimental factor	Values used	Purpose
System size	8 qubits	Limits the nominal Hilbert-space dimension and enables comparison across models with the same system size.
Architectures	PQC/VQC, QNN-style circuits, Data reuploading circuits	Tests whether the proposed relation holds across common QPBC model families with different expressivity and structural depth.
Ansatz library	More than 20 parameterized circuits for each architecture	Avoids conclusions based on a single hand-designed circuit and provides diversity in trainable gate structure.
Data encoding	Angle Encoding using X, Y, and Z rotations; Amplitude encoding; IQP encoding	Tests the effect of feature-map choice, entangling data maps, and state-preparation overhead on robustness.
Measurement	Probabilistic readouts and Expectation-value readouts	Tests whether the observed trends persist under different label-extraction mechanisms.
Dataset family	Make-classification linear dataset (Scikit-learn (Pedregosa et al. 2011)), Synthetic Non-linear dataset (Capital One (Barr et al. 2020)), Hidden manifold, Hyperplanes, and Two-curves datasets (Bowles et al. 2024)	Evaluates the proposed indicators across binary classification problems with different data properties.
Noisy backends	IBM, IQM Device Noise Simulators and Hardware Executions; IonQ Device Noise Simulator.	Tests whether robustness trends persist across different backend families and noise channels.

Table 1: Summary of experimental components used to evaluate QPBC robustness under noisy execution. The listed factors define the model families, datasets, measurement rules, and backend environments considered in the empirical analysis.

within the 8-qubit limit. From a training perspective, models of this smaller size were less susceptible to becoming trapped in barren plateaus and made the initial classical simulation process feasible on a classical system.

4.2 QPBC Architecture

The QPBC implementations considered in this work are constructed from several variational circuit architectures, which differ in how ansatz blocks, data-encoding operations, and measurements are arranged. There are multiple methods for defining these architectures, and a consistent naming convention is lacking. We have chosen to focus on three primary classifications based on their popularity: Parameterized or

Variational Quantum Circuits (PQC/VQC), Quantum Neural Networks (QNN), and Data Reupload models. Each of these model types has its distinct advantages and disadvantages.

PQC/VQC structures consist of models characterized by multiple training layers without consecutive reductions in complexity. In contrast, QNN structures demonstrate a reduction in the total number of operations with each successive training layer. Data Reupload models, on the other hand, are defined by their practice of reintroducing classical data encoding between consecutive training layers, whereas the other models apply classical data encoding only once at the beginning of the training process. More details of these structures are provided in Appendix A.

4.3 Ansatz

An integral component of any Variational Quantum Circuit (VQC) architecture is the ansatz, which consists of a combination of parameterized single-qubit and multi-qubit operations. Designing these ansatzes is not an intuitive task; it often involves a process of trial and error. To avoid limiting our exploration to a single type of ansatz, we experimented with over 20 distinct quantum ansatz structures, available in the accompanying repository. We applied these configurations across multiple architectures, which were discussed earlier. These ansatzes were inspired by various works presented in Hubregtsen et al. (2021), Hur et al. (2022), and Sim et al. (2019).

4.4 Classical to Quantum Data Encoding

The data encoding procedure serves as a crucial bridge, enabling the translation of classical information into its quantum representation. Because there is no universally optimal data-encoding strategy for all datasets and variational architectures, suitable encodings are typically selected empirically. In our experiments, we explored three primary encoding techniques: Angle(X, Y, Z), Amplitude, and Instantaneous Quantum Polynomial (IQP), leading to five distinct approaches for encoding information. While these represent the most prominent methods for encoding classical information, it is worth noting that there are more specialized encoding techniques beyond the scope of our investigation, as we are conducting a generalizable experiment.

4.5 Measurement

Measurement in a quantum system entails the retrieval of information (Uotila et al. 2025). Within the framework of a classification Quantum Machine Learning (QML) system, the emphasis is on measuring quantum circuits to extract information derived from quantum processing of classical information (Cerezo et al. 2021). This approach enables us to determine the class associated with a specific set of labels.

In the practical application, measurements performed for QPBC models are either probabilistic measurements or expectation values. For probabilistic measurements, the classes are represented by the labels '0' and '1', with a classification threshold set at '0.5'. When the estimated probability over n shots exceeds 0.5, the sample is assigned to class 1; otherwise, it is assigned to class 0. (Phalak and Ghosh 2023). Alternatively, for expectation value-based measurements, the labels correspond to the eigenvalues

'-1' and '1'. In this case, values greater than the threshold of '0' are classified as '1', while values at or below are classified as '-1' (Seo 2026).

4.6 Datasets

All datasets used in the evaluations were synthetically generated. Each generation method had its own characteristic, allowing us to evaluate models under a more generalized setting rather than tailoring conclusions to a specific dataset. Another advantage of having a generated dataset is the controlled class balance and feature dimensionality. This assisted us in evaluating data dynamics for noisy simulations of backends and their real hardware backends under a smaller validation dataset to describe the overall dataset.

To generate data samples, we selected five methods: Make Classification, Capital One's Synthetic Data, Hidden Manifold, Hyperplane, and Two Curve. The Make Classification dataset from Scikit-learn, as referenced in (Pedregosa et al. 2011), serves as a generator for linear-Gaussian style datasets. Capital One's synthetic non-linear Gaussian dataset was specifically designed for deep learning models (Barr et al. 2020). The Hidden Manifold, Hyperplanes, and Two Curves dataset generators were chosen based on the evaluation criteria proposed by (Bowles et al. 2024) for QML model evaluations.

We generated a total of 5,000 normalized data points, ensuring an equal distribution among the classification classes for balanced representation across all dataset generations. Each dataset comprised 8 features, with the importance of these features guided by the chosen data generation methods. Furthermore, the data points were divided into an 80-20 split, wherein 80% were designated for training and the remaining 20% for testing.

4.7 Machine Learning Training Parameters

Another important design choice in any machine-learning task is the selection of cost functions and optimizers. We have explored various cost functions and optimizers to ensure that our models are not overly tailored to any particular training procedure, thus maintaining a more generalized analysis. Our experiments included a range of optimizers, such as Nesterov Momentum, Adam, Gradient Descent, and AdaGrad, as well as several cost functions, including mean absolute error, hinge loss, cross-entropy, and squared loss. All the models were trained on an equal number of iterations or epochs.

4.8 Noisy Quantum Systems

In Section 3, we defined an arbitrary noise channel and are conducting noisy simulations across various device noise simulators to support a noise-agnostic approach. We validate the results of these device noise simulators against their corresponding hardware to ensure accuracy. To strengthen our findings, we utilized multiple noisy systems for simulation and also executed a few experiments on actual hardware from different vendors.

Given the extensive computational cost of evaluating theoretical results, we focused on assessing our models based on the noise profiles of available devices. We later

compared selected simulator results with executions on actual noisy quantum hardware using representative validation subsets. Additionally, we frequently updated noise simulation values to incorporate recent calibration data from all available hardware.

From IBM, we utilized simulated quantum devices including 'Sherbrooke,' 'Prague,' 'SydneyV2,' 'JohannesburgV2,' 'MelbourneV2,' 'BoeblingenV2,' 'GuadalupeV2,' 'AlmadenV2,' 'SingaporeV2,' 'Marrakesh,' 'Fez,' and 'Torino.' We confirmed our results using available IBM real quantum hardware, specifically Marrakesh, Fez, and Torino, all of which are currently accessible via the IBM cloud.

From IQM, we conducted simulations with quantum devices such as 'Apollo' and 'Aphrodite', both of which feature noise models and architectures that closely resemble the noise models of actual quantum hardware, 'Garnet' and 'Emerald', respectively. Additionally, 'Aphrodite' has a structure similar to that of 'Q50' quantum hardware offered at VTT, which we also used for verification. 'Q50' values were modulated to the 'Aphrodite' noise model based on the daily calibration values to verify on hardware.

From IonQ, we utilized the 'Aria 1' Noisy Quantum Simulator; however, we were unable to verify it against the actual 'Aria 1' device due to access restrictions. Future experiments and options are being explored to verify results from IonQ systems.

Noise model simulations for each of the quantum architectures were conducted on Puhti and Lumi, the high-performance computers (HPCs) for IBM, IQM, and VTT devices. For IonQ's 'Aria 1', we obtained the noisy simulations by accessing the online simulator for the hardware provided by IonQ (IonQ, Inc. 2026). While the hardware runs were conducted through direct API calls to the respective hardware providers. Although several backend models were considered during development, the main statistical analysis reported here uses the selected backend set summarized in Section 6 and Appendix B.

4.9 Backend-specific Transpilation Depth Computation

Usually, the depth of a circuit is defined by the total number of parallel operations conducted during the execution of a quantum circuit. While this can be easily observed for a predefined quantum circuit architecture, due to the positioning of the qubits and the native gate sets, the transpiled circuit may have a different depth value.

Each of the devices provided by the vendors seen prior have their individual constraints, such as connectivity, compilation algorithm, and local universal gatesets for implementing any quantum circuit catering to specific hardware. Considering the circuit based on its initial definition is not enough to evaluate how it proceeds through a noise channel. Hence, for each of the QPBC models, we considered transpilation depth according to the circuit compiler for each of the noisy architectures to establish the noisy channel interactions.

After transpilation to the target backend, the circuit depth was computed using Qiskit's QuantumCircuit.depth() method (Javadi-Abhari et al. 2024). This resulted in getting information regarding the physical qubits and gate decomposition selected by the backend compiler to implement our original QPBC circuit, along with their gate combinations. We were also able to store information regarding the measurement error, T1-T2 error values, and the total number of different gate operations required to implement a given QPBC model across several quantum architectures.

This backend-specific transpilation depth is the experimental value used for L_{tr} in the theoretical indicators introduced in Section 3. In other words, when plotting or comparing the degradation indicators, we use $L = L_{\text{tr}}$, where L_{tr} is obtained after compiling the trained QPBC circuit to the target backend.

4.10 Average Quantum Relative Entropy Computation

Average quantum relative entropy is the measure proposed in this work as a contributing factor to the robustness of some QPBC models under shallow noisy execution. The average quantum relative entropy as defined in Eq. (19), between the classification classes is computed based on their test dataset. These values were computed using PennyLane's qml.math.relative_entropy (Bergholm et al.), and density matrices were obtained using qml.density_matrix (Bergholm et al.). All of these procedures were conducted on a classical device in ideal conditions. In practical computations, we checked for NaN values and excluded numerically undefined relative-entropy evaluations. NaN values can arise from numerical support mismatch or near-zero eigenvalues in density-matrix logarithms. We removed such cases only when the relative entropy calculation was numerically undefined.

The number of removed cases was small relative to the total evaluated set and did not alter the qualitative trends reported here. All other average quantum relative entropy values observed were finite. The computed $\mathcal{D}_{avg}(\hat{\rho_0}||\hat{\rho_1})$ values, therefore, represent pre-measurement class-state distinguishability under ideal simulation. They do not include measurement thresholding, shot noise, or backend noise directly; those effects are represented separately in the theoretical formulation through c, c_n, $\prod_i \eta_i$, and δ_u.

All these components were used in the experimentation to establish a qualitative relationship among the average quantum relative entropy between classes, transpilation depth, and the performance difference observed between ideal and noisy quantum systems for a quantum parameterized binary classification (QPBC) model.

5 Experimental Procedure

This section describes how the components introduced in Section 4 are combined in the experimental workflow. This Section also provides insight into the scale of the experimental evaluation. The following Fig. 1 provides a diagrammatic representation of the QPBC model training procedure and an overview of where components mentioned previously are used during the experimentation.

The experimental procedures can be systematically categorized into three distinct phases. The first phase involved model training and selection. The second phase focused on computing the average quantum relative entropy and the transpilation depth. Lastly, the third phase encompassed the execution of models on noisy backend simulations and the associated hardware for those backends.

As in most machine-learning workflows, our procedure for the first phase begins with data generation. We used the data generation methods described in Section 4.6 to generate 5000 normalized data points with 8 feature dimensions and a balanced class distribution. We further split the data into an 80-20 train-test split, following common

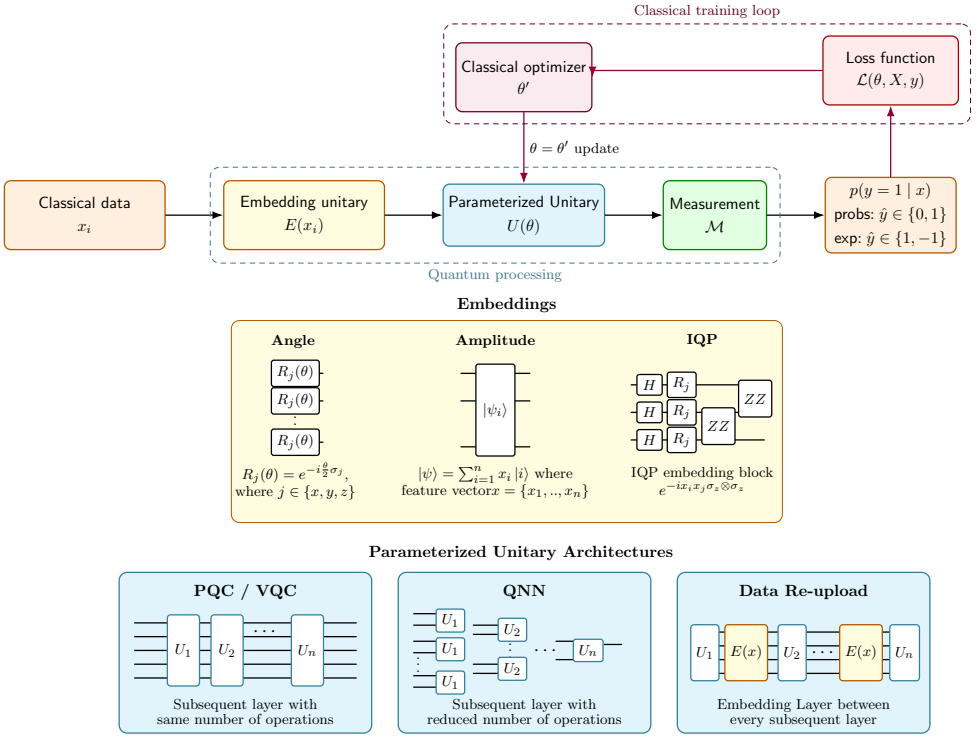

Fig. 1: Schematic overview of the QPBC training workflow and architectural variants considered in this study. The figure illustrates representative encoding, parameterized unitary, and measurement components rather than a single fixed circuit.

ML protocol. The randomization seed was kept constant for the generation of data points and the train-test split. This data was further used for training the models on an ideal simulator. All the training and simulation procedures were conducted using the Pennylane quantum programming framework (Bergholm et al. 2018). The Puhti and Lumi, the high-performance computers (HPCs), were used for training these QPBC models, considering the volume of data generation and processing involved.

In general, for Variational Quantum Algorithms, parameters are initialized randomly unless using a warm start. Since we were not exploring specific Variational Quantum Algorithm training techniques, we used values initialized randomly. Hence, training multiple models under similar configurations allowed us to explore which datasets were best suited to specific quantum circuit structures. We trained five models under ideal simulation conditions for each combination of components (such as Architecture, Ansatz, Encodings, Measurement, etc.) for each dataset as presented in Section 4, and later tested on the noisy systems mentioned in Section 4.8.

All of these models were trained under a similar number of iterations or epochs. We explored several different optimizers to attain good accuracy. Based on our early observation, one can achieve a lower cost value when using multiple optimizers in tandem. Hence, such measures were in place to use one optimizer for half the total

number of epochs and another for the remaining half. This reduced the number of iterations required to reach comparable accuracy relative to using a single optimizer. Other training components, such as batch size and learning rate, were kept constant.

We trained over 2,750 models for the VQC-QNN circuits utilizing shot-biased probabilistic measurement, as well as an additional 2,750 models for expectation-value measurements. Furthermore, we developed 3,000 models for data reupload circuits, incorporating both measurement types. From this pool of trained QPBC models, we selected only those that achieved an accuracy exceeding 85% on the test dataset for further analysis. This selection yielded 413 models for the VQC-QNN circuits with probabilistic measurement, 362 models for expectation value measurements, and 400 models for the data reupload circuits with both types of measurements. All the values used for training the models and parameters obtained from trained models were stored in a parquet file format for redundant storage and usage in further procedures.

We established the 85% accuracy threshold for model selection because the focus of the study is on degradation after learning, rather than failure to learn. For binary classification problems, considering models below this threshold would not have provided significant insights. Models with poor accuracy could confuse trainability with noise robustness, potentially contaminating the data.

Models that were selected for further evaluation in the second phase were examined using the parameters obtained from the training procedure. We calculated the average quantum relative entropy, as defined in Eq. (19) and described in Section 4.10. To compute the average quantum relative entropy between classes, we began by constructing density matrices for each data point in the test set using the trained model and its parameters. Next, we applied relative entropy calculation methods to measure the relative entropy of each density matrix in relation to the opposing class's density matrix. Finally, we averaged these values across all data points.

To calculate the transpilation depth, employing the methods as outlined in Section 4.9, by mapping the entire circuit to the backend-specific topology in its native gates of the noisy quantum system, up to the measurement stage. We approximate the effective number of noisy operational layers by the backend-specific transpilation depth. These values can differ from one device to another and between different vendors. Additionally, the qubit count of the transpiled circuit may vary due to the topological constraints of the implementation. These values for average quantum relative entropy and transpilation depth for each model for different noisy systems were also stored in parquet format.

The third phase involves running the pretrained models with the same test dataset and the same trained parameters on the noisy quantum systems. We executed most of the results on a backend-specific noise-model simulator, i.e., a noise simulator for the expected noise on actual quantum hardware. This allowed us to have an extensive amount of evaluations to support the theory and kept the cost (time and monetary) of execution on hardware low. Later, we validated these results of the noise simulator execution with the actual available hardware using a representative validation subset selected from the test set using K-medoids clustering. The validation subset size and selection details are reported in the accompanying repository. For the real hardware comparison with the noisy simulation, we calculated the mean difference in the measurement values of the validation dataset. These hardware executions are used

as subset-level consistency checks for selected backends, not as exhaustive hardware validation of all simulator observations.

For all these simulations, we kept the total number of shots constant for all the noisy systems. The noisy simulations were conducted on the Puhti and Lumi HPCs as well, while the real hardware runs were conducted via the API access to the IBM and IQM devices. For VTT Q50, special Lumi access was used to access this device. We were unable to execute results for IonQ real hardware due to accessibility constraints; further actions are being taken to evaluate results on actual IonQ devices. Noise attributes for the noisy simulations were acquired from the available calibration data.

By executing the models on a noisy backend system, we obtain the noisy accuracy of the model specific to that system. We then compare these values with the accuracy obtained from the ideal simulation to determine the difference in accuracy (Δ Accuracy). Finally, we compare all these values obtained from the experimentation to support the theory proposed in Section 3. In the following section, we provide a detailed explanation of the dynamics observed in these values.

6 Experimental Observations

In this section, we present our observations derived from experimental results obtained using the components outlined in the previous section. Specifically, we focus on the data related to the noisy execution on the Qiskit Fez Backend. Fez is used as the primary illustrative backend. Additional backend-wise tables are provided in Appendix B, and additional backend-wise plots are provided in Appendix C. Here, we will discuss the qualitative relationship between transpilation depth (referred to as depth in this context), average quantum relative entropy (as defined in Eq. (19)), and the performance measures of the models, specifically noisy accuracy and accuracy difference.

During experimentation, we also examined several additional diagnostic quantities that could plausibly influence noisy-device performance, including the choice of classical-to-quantum data encoding, layer-wise relative-entropy behavior, dataset-generation family, measured-output probability distributions, within-class average quantum relative entropy, Quantum Rényi divergence, backend calibration attributes such as T1, T2, readout error, gate counts, and native gate-set characteristics. These quantities were useful for interpreting specific cases; however, within the representative comparisons and backend-wise trends examined here, we did not observe a consistent standalone qualitative trend that generalized across architectures, encodings, measurement strategies, datasets, and noisy backends.

In several cases, their effects are either indirectly reflected in the backend-specific transpilation depth, captured by the practical measurement-threshold and residual terms c, c_n, and δ_u, or strongly dependent on the specific circuit-to-backend mapping after transpilation. For example, measured probability distributions affect classification primarily through their distance from the decision threshold, while calibration quantities such as T1, T2, readout error, and gate errors depend on the physical

qubits, couplers, routing paths, and native-gate decompositions selected during compilation. Consequently, within the scope of this work, we focus on average cross-class quantum relative entropy and backend-specific transpilation depth as the most consistently interpretable pre-hardware indicators, while leaving a systematic multivariate analysis of these additional quantities for future work.

This observation should be interpreted qualitatively. The additional quantities were inspected as diagnostic variables through representative comparisons and backend-wise trends, not through an exhaustive multivariate statistical analysis. We therefore do not conclude that these factors are irrelevant; instead, we conclude only that they did not provide a consistently interpretable standalone indicator within the heterogeneous experimental setting considered here.

6.1 Lack of Direct Correlation and Limited Explanatory Power of Individual Quantities

The graphs presented in Fig. 2 and Fig. 3 are intended to illustrate the challenges of establishing relationships between factors that may appear to be easily correlated through equations presented in Section 3. They also provide crucial context for the significance of these experiments. These graphs aim to demonstrate the lack of correlation between aspects that have been contextually associated or could be mathematically related, as discussed in the previous sections.

Fig. 2 shows a wide range in the accuracy differences values when evaluating the same model on an identical test dataset using both an ideal simulator and Qiskit Fez backend noise simulator. Similar patterns can be identified in the other noisy quantum systems tested throughout these experiments. It is challenging to predict the performance of a quantum model solely based on its accuracy from a simulator and its transpilation depth (shallow circuits). Even with comparable depth values, there is a considerable distribution of accuracy differences, which tends to expand as depth increases. Conversely, the colored representation of average quantum relative entropy values between classified classes suggests that a larger average quantum relative entropy is generally associated with smaller degradation, although this is not a standalone linear relationship.

From Eq. (15) or Eq. (23), one might be inclined to correlate ideal simulation accuracy with quantum relative entropy. However, several caveats must be considered, as illustrated in Fig. 3. In this figure, it is clear that neither the average quantum relative entropy nor the transpilation depth of the models significantly impacts the accuracy of the ideal simulated model. The reasoning for this is quite straightforward: the transpilation depth does not affect ideal simulation. In the case of relative entropy, even if it can reach infinity, the accuracy remains capped at 1 or 100%. Additionally, the model's accuracy is more strongly influenced by the definition of the cost function, also known as the loss function, and the classification thresholds.

As such, these values have often been overlooked in the context of QPBC or general QML implementations. However, they reveal a meaningful relationship when viewed through the lens of noise, as will be further explored in the following subsection.

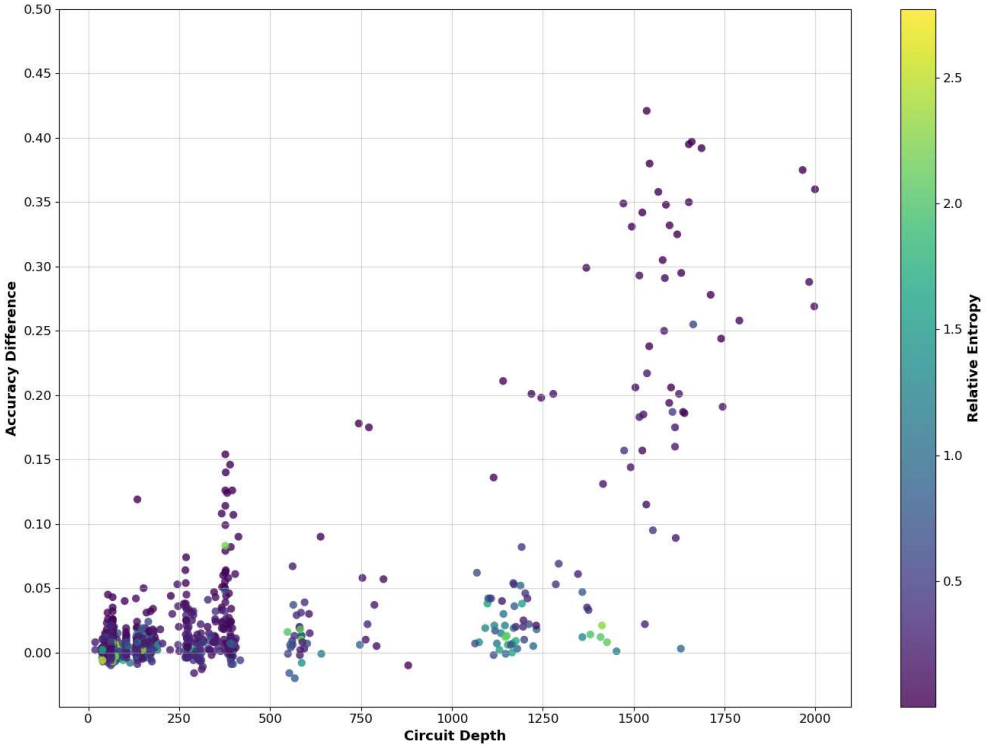

Fig. 2: Accuracy Difference ΔAcc (Acc. Simulated - Acc. Noise) and Transpilation Depth for independent QPBC models evaluated on the Qiskit Fake Fez backend. Each point on the graph shows the results for independent models. The color plot indicates average quantum relative entropy. The figure shows that transpilation depth alone does not fully explain noisy-execution degradation.

6.2 Statistical Support for the Entropy-Depth Indicator

The theoretical analysis motivates an entropy-depth indicator for noisy-execution degradation. However, representative plots and case studies alone do not quantify whether average quantum relative entropy contributes explanatory information beyond backend-specific transpilation depth. We therefore performed a statistical analysis to assess the empirical support for the central claim of this work: noisy degradation is more consistently interpreted through the joint behavior of backend-specific transpilation depth and average cross-class quantum relative entropy than through either quantity alone.

This analysis is not intended to replace the theoretical indicator or to fit Eq. (30) directly. Eq. (30) contains backend and measurement-dependent quantities, including the cumulative SDPI contraction factor, threshold-related coefficients, and residual drift terms, which are not independently available for every model-device pair. Instead, the statistical analysis evaluates the structural implication of the theory: degradation

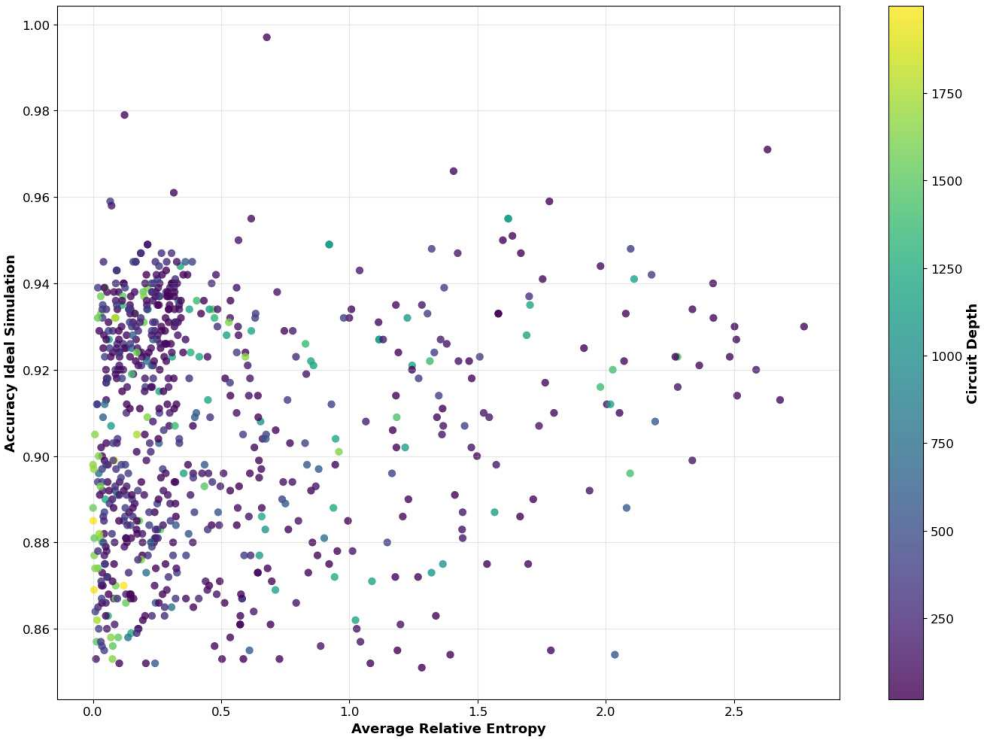

Fig. 3: Ideal-simulation accuracy versus average quantum relative entropy. The marginal plot shows the associated transpilation-depth distribution. The figure illustrates that average quantum relative entropy does not directly contribute to ideal-simulation accuracy.

should generally increase with transpilation depth, decrease with average cross-class quantum relative entropy, and be better explained when both quantities are considered together.

The statistical comparison also serves as a robustness check against over-interpreting representative examples. Because the experiments include multiple architectures, encodings, measurement strategies, datasets, and noisy backends, the regression analysis helps assess whether the proposed interpretation persists at the model-device observation level. Additional backend-wise results are reported in Appendix B.

One important feature of the experimentally observed data is that transpilation-depth values span a much wider numerical range, with substantially larger values compared to the average quantum relative entropy values. As a result, transpilation depth is expected to dominate global linear correlations. Consequently, average quantum relative entropy should not be interpreted as a replacement for transpilation depth, but rather as a complementary distinguishability indicator.

Predictor	Pearson r	Spearman ρ
L_{tr}	0.62199	0.52701
$\log(1 + L_{\mathrm{tr}})$	0.68441	0.52701
$\mathcal{D}_{avg}$	-0.08293	-0.36927
$\log(1 + \mathcal{D}_{avg})$	-0.11451	-0.36927

Table 2: Pairwise correlation of transpilation-depth(L_{tr}) and Average Quantum Relative Entropy($\mathcal{D}_{avg}$) variables with accuracy difference $\Delta\mathrm{Acc} = \mathrm{Acc}_{\mathrm{ideal}} - \mathrm{Acc}_{\mathrm{noisy}}$.

The pairwise correlations in Table 2 show that transpilation depth is the strongest global predictor of simulation-to-noisy degradation. The positive correlation for L_{tr} indicates that deeper transpiled circuits generally experience larger degradation. Average quantum relative entropy shows a negative association with degradation, especially under Spearman correlation, indicating that models with larger class-state distinguishability tend to exhibit lower degradation in rank-order terms. The weaker Pearson correlation for $\mathcal{D}_{avg}$, however, shows that average quantum relative entropy is not a strong standalone global linear predictor.

Linear Regression model	CV R^2	CV R^2 SD	CV MAE	CV MAE SD
L_{tr}	0.38629	0.01983	0.07858	0.00134
$\log(1 + L_{\mathrm{tr}})$	0.47398	0.01858	0.07446	0.00141
$\mathcal{D}_{avg}$	0.00592	0.00358	0.10908	0.00143
$\log(1 + \mathcal{D}_{avg})$	0.01214	0.00508	0.10811	0.00141
$L_{\mathrm{tr}} + \mathcal{D}_{avg}$	0.41738	0.01634	0.07591	0.00126
$\log(1 + L_{\mathrm{tr}}) + \log(1 + \mathcal{D}_{avg})$	0.49008	0.01658	0.07545	0.00126

Table 3: Cross-validated (CV) linear regression comparison for accuracy degradation across all evaluated QPBC model-device observations. Here, transpilation-depth is L_{tr}, Average Quantum Relative Entropy is $\mathcal{D}_{avg}$, R^2 is the coefficient of determination regression score function, MAE is Mean absolute error regression loss, and SD refers to Standard deviation.

The cross-validated regression results in Table 3 provide modest support for the interpretation of average quantum relative entropy as an additional robustness indicator. The transpilation-depth-only model obtains a cross-validated R^2 of 0.38629, whereas adding $\mathcal{D}_{avg}$ increases the value to 0.41738. A similar pattern is observed for the logarithmic variables, where R^2 increases from 0.47398 for $\log(1 + L_{\mathrm{tr}})$ alone to 0.49008 when $\log(1 + \mathcal{D}_{avg})$ is added.

At the same time, the Mean Absolute Error (MAE) regression loss values show that the improvement is not uniform across all metrics. In particular, the logarithmic combined model improves R^2 but does not improve MAE relative to the logarithmic

Idx.	Depth	Avg. Relative Entropy	Ansatz	Model Type	Encode	Acc. Sim.	Acc. Device	Δ Acc. Diff.
1	67	0.0931	RXRZ_CNOT	VQC_QNN_EXP	AngleY	0.943	0.929	0.014
2	67	1.1136	RXRZ_CNOT	VQC_QNN_PROB	AngleX	0.931	0.926	0.005
3	152	0.0540	INV_QFT_U3_MODEL	DATA_REUP_PROB	AngleY	0.860	0.810	0.050
4	152	1.0121	RY_CRX	DATA_REUP_PROB	AngleZ	0.878	0.873	0.005
5	15369	2.3209	RY_CZ	DATA_REUP_PROB	Ampli.	0.937	0.504	0.433
6	1416	0.1506	QNN_RY_CNOT	VQC_QNN_PROB	Ampli.	0.919	0.788	0.131
7	1427	2.0946	RXRZ_CRZPool	DATA_REUP_PROB	IQP	0.896	0.888	0.008
8	245	0.1673	QNN_RY_CNOT	VQC_QNN_PROB	IQP	0.930	0.877	0.053
9	54	0.0244	H_CZ_RX	VQC_QNN_EXP	AngleY	0.929	0.884	0.045
10	67	1.0540	RXRZ_CNOT	VQC_QNN_EXP	AngleX	0.853	0.857	-0.004

Table 4: Representative QPBC model results on the Qiskit_Fake_Fez backend. Depth denotes backend-specific transpilation depth after compilation to Fez. Acc_sim denotes ideal-simulation accuracy, Acc. Device denotes noisy-simulator accuracy, and Δ Acc. Diff. denotes Acc. Sim. - Acc. Device.

depth-only model. Therefore, these results should be interpreted as statistical support for the proposed qualitative indicator, not as evidence of an exact predictive model. Overall, the regression analysis suggests that backend-specific transpilation depth captures the dominant noise-exposure effect, while average quantum relative entropy provides additional information about learned class-state distinguishability that helps explain robustness differences among models with comparable depth.

6.3 Representative Data Values and Data Correlation

In this subsection, both Table 4 and Table 5 provide representative values observed for the Qiskit Fez backend (noise simulator and real Fez hardware). More values are available in the Appendix B for different devices, and further, more data is available on the GitHub code repository provided after Section 8. These tables also help in observing the correlation between the values mentioned in the prior section and how they influence each other.

Table 4 summarizes the performance variations observed across different QPBC models, which differ in architecture and measurement (Model Type), Ansatz, depth (transpilation depth), and encoding methods (Encode). The representative cases do not indicate that the observed degradation trend is tied to a single implementation architecture, ansatz, or measurement aspect when it comes to the difference in accuracy between ideal and noisy systems. However, a consistent qualitative relationship can be seen between the differences in accuracy, average quantum relative entropy, and transpilation depth.

It is noteworthy that models with similar depths exhibit smaller accuracy differences, particularly when they have high average quantum relative entropy. Furthermore, this accuracy difference tends to increase with greater depths. It is important to recognize that the decline in accuracy for a noisy system is not solely determined by the depth of the model or the accuracy achieved in ideal simulation conditions.

Amplitude encoding frequently shows large degradation in the evaluated noisy settings, largely because its state-preparation overhead produces high transpilation depth. In Table 4, we present the highest depth and average quantum relative entropy recorded for Amplitude encoding, alongside data reflecting its lowest depth and corresponding average quantum relative entropy. In both cases, substantial accuracy degradation is evident when tested within a noisy system.

Given the high cost of execution on actual quantum hardware and the time required for such complex experiments, we limit our evaluations to extensive testing on noisy simulations. This involves mimicking the noise characteristics observed in those particular backends and subsequently validating the results against actual hardware using a smaller subset of the dataset to identify any discrepancies. In Table 5, we assess the same models presented in Table 4, utilizing a reduced validation dataset to examine the anticipated differences between the outcomes of the simulated noisy environment and the results obtained from the real Fez hardware. This approach also offers insight into how these findings may translate to actual hardware backends.

To validate the results obtained from the Qiskit Fake Fez backend simulator, we utilized a validation dataset derived from the test dataset through 'K Medoids' clustering, identifying points that best represent the overall dataset. Subsequently, we executed these data points within the same trained model under three different conditions: ideal simulation, noisy simulations, and real hardware. In this evaluation, we focus not on the overall accuracy percentage but on the precise values recorded after measurement for each specific data point. In particular, for Table 5, we present the absolute mean difference in measurement values across all data points. Consequently, the difference between $|\text{Real} - \text{Fake}|$ is minimal, indicating that the results do not significantly diverge from those of the noisy simulations. Moreover, the values of $|\text{Sim} - \text{Fake}|$ and $|\text{Sim} - \text{Real}|$ demonstrate a closer alignment with the accuracy achieved in a noisy setting, as presented in Table 4.

The quantum relative entropy values are dependent on direction, so we evaluated them relative to both classes. In our comparison data, the columns labeled rel_l2-l1 and rel_l1-l2 represent the two cross-class directional averages. The columns rel_l2-l1 and rel_l1-l2 denote the two directional averages of quantum relative entropy. In

Idx.	Depth	Avg. Relative Entropy	Ansatz	Model Type	Encode	|Sim−Fake|	|Sim−Real|	|Real−Fake|
1	67	0.0931	RXRZ_CNOT	VQC_QNN_EXP	AngleY	0.04558	0.03699	0.03989
2	67	1.1136	RXRZ_CNOT	VQC_QNN_PROB	AngleX	0.06477	0.07155	0.02001
3	152	0.0540	INV_QFT_U3_MODEL	DATA_REUP_PROB	AngleY	0.02254	0.02867	0.01845
4	152	1.0121	RY_CRX	DATA_REUP_PROB	AngleZ	0.04165	0.05432	0.02387
5	15369	2.3209	RY_CZ	DATA_REUP_PROB	Ampli.	0.37624	0.38517	0.03295
6	1416	0.1506	QNN_RY_CNOT	VQC_QNN_PROB	Ampli.	0.08360	0.10490	0.03120
7	1427	2.0946	RXRZ_CRZPool	DATA_REUP_PROB	IQP	0.23812	0.28752	0.05888
8	245	0.1673	QNN_RY_CNOT	VQC_QNN_PROB	IQP	0.02909	0.03582	0.01914
9	54	0.0244	H_CZ_RX	VQC_QNN_EXP	AngleY	0.03702	0.02816	0.03873
10	67	1.0540	RXRZ_CNOT	VQC_QNN_EXP	AngleX	0.04332	0.04304	0.02802

Table 5: Mean absolute differences in measured prediction values for the representative models from Table 4, comparing ideal simulation, Qiskit Fake Fez noisy simulation, and real Fez hardware on the validation subset.

probability-based measurement, these correspond to labels 0 and 1; in expectation-value measurement, they correspond to labels -1 and 1. We found that the observed values for rel_l2-l1 and rel_l1-l2 were computationally equivalent with negligible differences, allowing us to use only one of the directional values for comparison. The average quantum relative entropy values presented in both Table 4 and Table 5 correspond to rel_l2-l1. This is also true for the values shown in the tables in Appendix B.

Our analysis of bidirectional similarity of average quantum relative entropy revealed that the mean values for the columns rel_l2-l1 and rel_l1-l2 were 0.5941 and 0.5909, respectively. The mean difference between these values for each model (calculated as rel_l2-l1 - rel_l1-l2) was 0.0031. These calculations were performed for all 1132 unique models, and a detailed analysis table, Table: B3, can be found in Appendix B. Overall, our observations indicate that the bidirectional averages show strong empirical agreement and negligible average directional bias.

Both Table 4 and Table 5 offer insights into the research question previously posed: Is there a factor beyond depth contributing to the observed variance in performance between a noisy system and an ideal simulation for a QPBC model? What can we anticipate regarding hardware results based on the average quantum relative entropy

and depth? However, these values alone do not establish a definitive correlation. The following section will connect the precise qualitative relational theory established in Section 3 with the data gathered from the experiments.

6.4 Entropy-Depth Trend Curves

The figures presented in this subsection offer experimental evidence for the proportionality outlined in Eq. (21) and (22). These plots are based on data collected from experiments conducted with models trained in simulation and tested on the 'Qiskit Fake Fez' backend, a noise simulator for the IBM Fez Device. For clarity in the graphs, we have focused on points below a transpilation depth of 500; additional points exhibiting similar trends with higher values can be found in Section B.

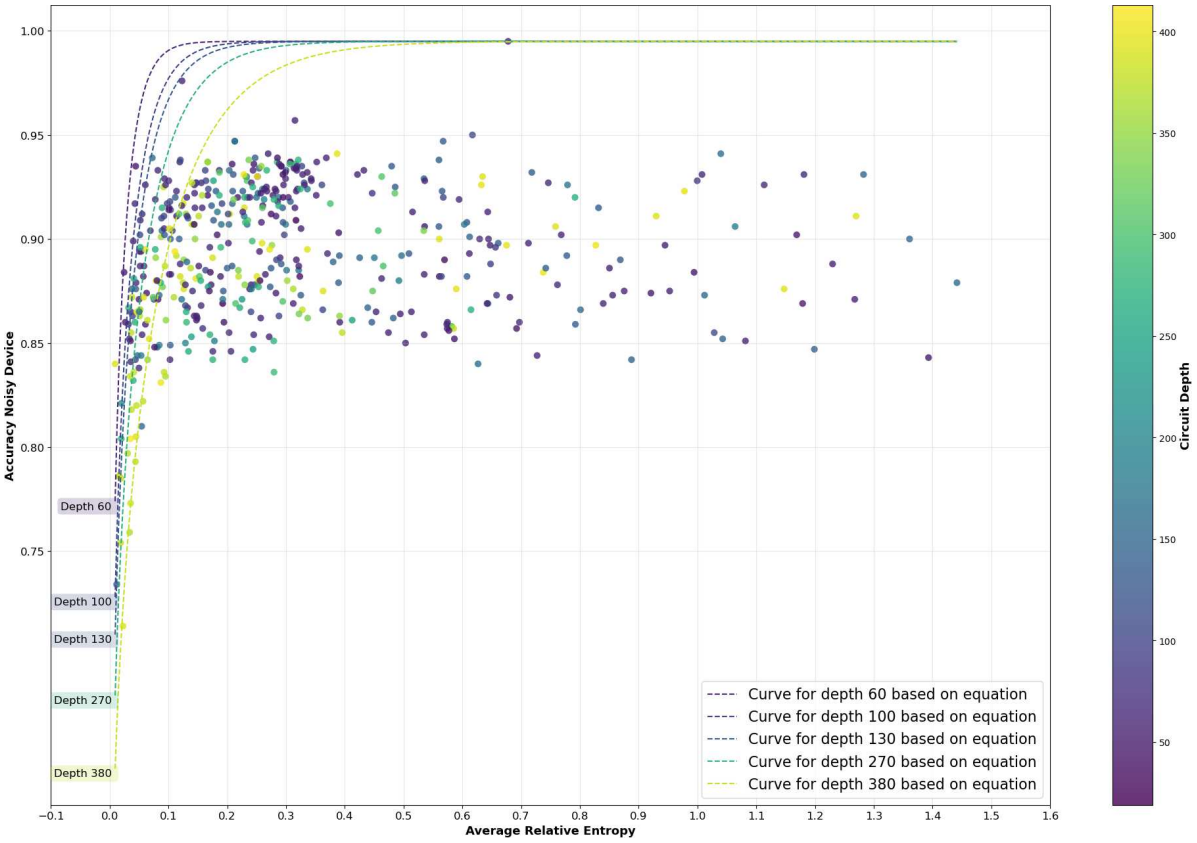

Fig. 4: Noisy accuracy on the Qiskit Fake Fez backend as a function of average cross-class quantum relative entropy for models with transpilation depth ≤ 500. Curves show representative entropy-depth trends based on Eq. (21) using illustrative coefficient choices; they are not unique fitted parameter estimates.

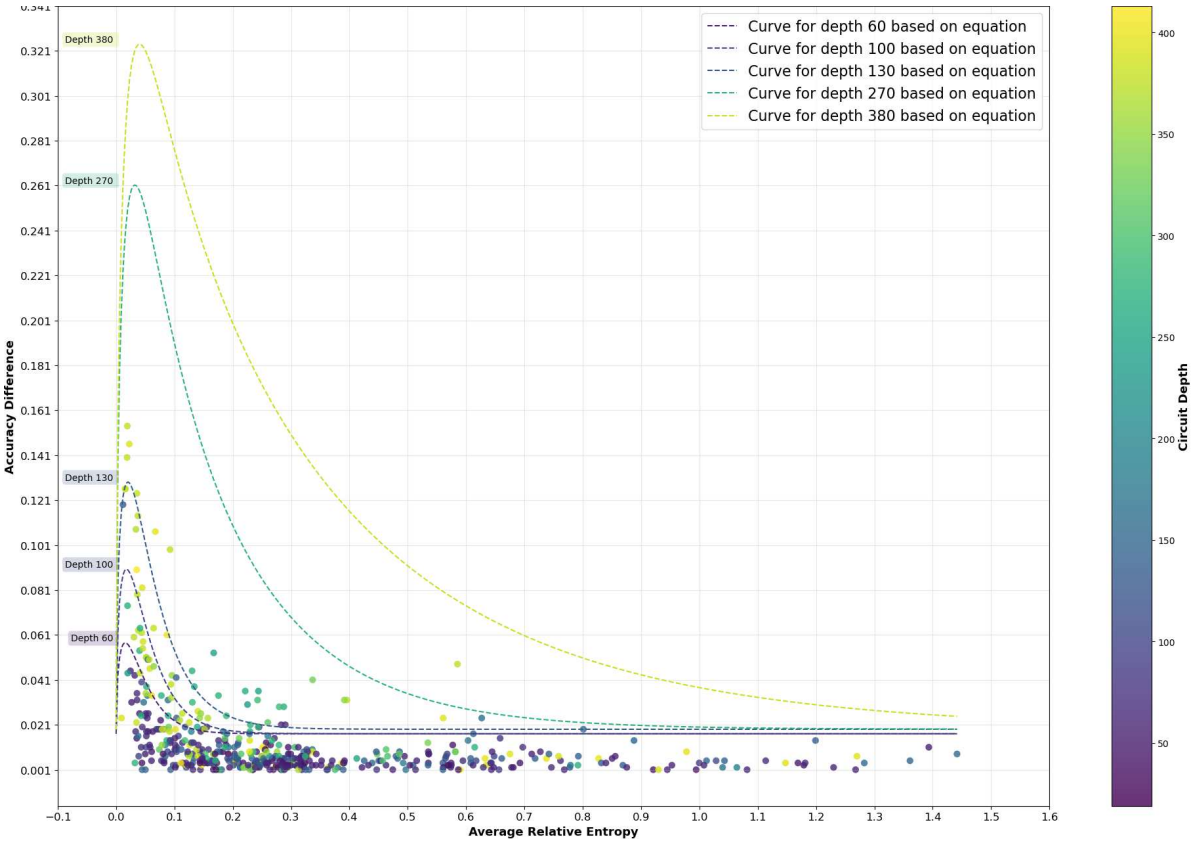

Fig. 5: Accuracy Difference (Ideal-Noisy) on the Qiskit Fake Fez backend as a function of average quantum relative entropy for models with transpilation depth ≤ 500. Curves show representative trends implied by Eq. (22) for illustrative coefficient choices and should be interpreted as qualitative indicators rather than exact predictions.

Each figure includes curves that correspond to specific depths, based on the approximations from Eq. (28) and Eq. (30). This serves to demonstrate that the observed values align with the proposed theory, illustrating consistency with the proposed proportional trend. The transpilation depth values selected for plotting the equation curves are determined by the maximum number of points observed around those specific depths. These points are not limited to any particular model type, encoding, or measurement type of QPBC models.

The coefficients $\prod_{i=1}^{L} \eta_i$, c, c_n, and δ_u are not treated as universal constants. They are backend-, model-, and measurement-dependent calibration quantities. In the curve plots, representative values are chosen to illustrate the theoretical trend implied by Eq. (28) and (30). The coefficients are selected to produce representative trend curves

consistent with the observed value ranges; they are not claimed to be unique fitted parameters. Their purpose is to show consistency with the proposed monotonic structure, not to claim exact point-wise prediction for every model.

Fig. 4 illustrates the noisy-accuracy trend suggested by Eq. (28) under representative coefficient choices. The trend curves suggest that systematic improvement over ideal simulation is not expected, although finite-shot fluctuations and threshold effects can occasionally yield slightly higher noisy accuracy for individual cases.

Fig. 5 showcases the maximum observed difference in performance based on Eq. (30). It also shows that models with lower Average Quantum relative entropy would expect to have higher accuracy difference, and the bound for said relative entropy would grow with the transpilation depth value.

These figures help us visualize the relation between the noisy device performance, average quantum relative entropy, and transpilation depth from both the theoretical curves plotted based on a representative coefficient choices perspective and the experimental points observed. From these plots, one can estimate how a model might perform using approximated parameter values in Eq. (28) and Eq. (30). Further work is in progress to identify effective ways to estimate these values without the requirement of extensive evaluations.

7 Discussion

The theoretical analysis and experimental evaluations provide several insights into the noisy-execution robustness of QPBC models. The results indicate that neither ideal-simulation accuracy nor circuit depth should be used as a standalone robustness indicator. To comprehend the performance disparity between ideal and noisy systems, it is useful to consider both transpilation depth and average quantum relative entropy. A higher average cross-class quantum relative entropy, combined with lower backend-specific transpilation depth, is associated with better retention of ideal-simulation performance under noisy execution.

This article aims to address three research questions, as stated previously, regarding the observed discrepancy in accuracy between similar systems, the observable degradation trends on noisy quantum systems, and the possibility of assessment on classical systems. To tackle these questions, we provide a theoretical framework along with an extensive empirical analysis conducted across multiple datasets and various implementations of general QPBC models, supporting the observed theoretical associations.

'RQ1' explores why, when exposed to a similar noise channel of comparable size, some circuits experience substantially higher degradation in accuracy even though they share many similarities in structure and performance. Our research addresses this question through the reduction in distinguishability between states in a quantum classification model. We use the average quantum relative entropy between classes as a distinguishability measure. This perspective helps explain differences in classification deterioration under a noise channel, since relative entropy is also reduced by noisy evolution. Hence, we see some quantum classification models being more noise-robust than others under similar circumstances.

'RQ2' aims to understand the extent of performance difference when experimenting on a noisy system, based on the correlation derived from values calculated using average quantum relative entropy and transpilation depth. Equations (28) and (30) provide distinguishability-based indicator curves for interpreting noisy accuracy and simulation-to-noisy degradation. The curve for Eq. (28) indicates that the area under the curve tends to decrease at greater depths, suggesting that accuracy may be lower for the same level of relative entropy. This provides a range of expected accuracy in relation to the average quantum relative entropy for a particular transpilation depth.

The Eq. (30) indicates that the variance in observed accuracy is ideally high at lower levels of average quantum relative entropy in a noisy system, and tends to decrease as average quantum relative entropy increases. The rate at which the expected accuracy difference decreases with increasing average quantum relative entropy is also influenced by depth, with deeper circuits generally requiring larger relative entropy values to maintain a similar degradation level. While estimating the exact values for the degradation indicator is challenging due to the presence of various complicating factors, these values can still be approximated.

Taken together, these equations provide intuition regarding the range of average quantum relative entropy required, relative to the transpilation depth of a given device, in order to expect a certain level of performance. In this sense, the relative-entropy range associated with robust behavior depends on the desired precision of the results, the transpilation depth of the circuit, and reasonable assumptions regarding the difficult-to-estimate variables appearing in the formulation.

'RQ3' examines the feasibility of predicting outcomes using these methods without the need for quantum devices while maintaining classical computation. In principle, such pre-assessment is feasible for classically simulable system sizes, because both transpilation-depth calculation and average quantum relative entropy computation can be performed classically. Moreover, the processes of transpilation and the computation of average quantum relative entropy are entirely classical. However, this leads to questions about the scalability of these methods, as well as concerns regarding the precision of addressing variable factors in relation to direct assessments of quantum hardware. Nevertheless, these classical evaluation methodologies do provide an estimate of the expected degradation for QPBC models, particularly when approximate knowledge of the remaining uncertain factors is available.

Empirical observations provided additional insights beyond the general discussion of observed performance reductions. Notably, the complex transpilation process for the Amplitude Encoding methods led to circuits with a high transpilation depth. In the evaluated noisy settings, amplitude encoding frequently led to large degradation, primarily because state-preparation overhead produced high transpilation depths. A similar, albeit less pronounced, issue was encountered with the 'IQP' encodings. This highlights the importance of considering the costs associated with implementing these methods in practical applications.

The differences in each device and the characteristics across various vendors imply that values obtained from one device cannot be readily applied to another. Additionally, considering variations in error rates, native gate sets, and architecture, each device exhibits a unique transpilation depth. Consequently, a transpilation depth deemed

low for one device may yield different outcomes for another. This leads us back to the discussion in the previous Section 2: what should be regarded as 'shallow enough' for robust execution? The answer depends strongly on the target device and implementation setting. This dependency is influenced by several factors, most notably the average quantum relative entropy and the error SDPI coefficient observed for the said transpilation depth. Although the results observed from the devices may appear distinct when examined individually, they exhibit similar structures, as illustrated in Section 6 and Appendix B.

Alternatively, results obtained from noise simulators appear to translate qualitatively to corresponding hardware backends, provided that device-specific calibration effects are taken into account. Hence, the noise simulators appear to provide a useful proxy for the qualitative trends expected on real hardware, although extensive hardware validation is still necessary for precise deployment assessment.

Even though these conclusions are all derived from the experiments conducted on fixed 8-qubit systems for binary classification and the majority of values have been acquired from noise simulations, they do provide key insights into the correlations between the average quantum relative entropy, transpilation depth, and possible noise robustness. Hence, exploring these ideas could enhance this research further for implementing noise-resilient systems on NISQ devices.

A limitation of the current formulation is that several coefficients, particularly $\prod_{i=1}^{L} \eta_i$, c_n, and δ_u, are difficult to estimate exactly without backend-specific calibration or validation data. Therefore, the proposed framework should currently be viewed as a classical pre-assessment and model-selection tool rather than as an exact hardware-accuracy predictor. Its main value lies in identifying models that are more likely to retain performance under noisy execution by combining learned class-state distinguishability with backend-specific transpilation depth.

Further work is certainly needed and is underway on several fronts. Primarily, there is a need to develop intuitive methods to obtain key variables, such as the collective SDPI coefficient, threshold errors, and drift coefficients, without relying on extensive experiments and data processing. Additionally, it is valuable to explore effective training methods for QPBC models and other quantum classification models, aiming for both high accuracy and high average quantum relative entropy. A scalable approach for evaluating these parameters in large quantum systems would significantly enhance the advancement of this field. More broadly, this study suggests a direction for hardware-aware QML model evaluation that combines quantum-information-theoretic distinguishability with backend-specific compilation properties.

8 Conclusion

In this article, we examined discrepancies between ideal-simulation performance and noisy-execution performance for trained QPBC models. Even when models have similar ideal accuracy and comparable transpilation depth, they can exhibit different levels of noisy-execution degradation. We also investigated the expected degradation trends on a noisy system and whether this degradation can be estimated classically

without extensive execution on quantum hardware. Furthermore, we explored the idea that accuracy derived from ideal simulations or circuit depth cannot be treated as a reliable standalone indicator of a QPBC model's robustness against noise on Noisy Intermediate-Scale Quantum (NISQ) devices.

To address this gap, we propose utilizing average quantum relative entropy between the classes as a supplementary measure, alongside transpilation depth, to better understand the potential decline in accuracy observed when a model trained in an ideal simulation is tested within a noisy system. In this context, Average quantum relative entropy quantifies the distinguishability of learned pre-measurement class-state ensembles, while backend-specific transpilation depth acts as a proxy for cumulative noise exposure after compilation.

In this research, we develop a theoretical relation linking average quantum relative entropy, its noisy-channel contraction, transpilation depth, and the degradation of classification accuracy under noisy execution. We also provide distinguishability-based indicators for noisy-execution accuracy and the possible accuracy degradation between ideal and noisy execution. To support our theoretical claims, we conducted experiments across a wide range of QPBC implementations and noisy quantum systems, showcasing empirical consistency of the proposed relation across architectures, encodings, datasets, and backends.

In response to the questions raised during this evaluation, we conclude that differences in accuracy arise because class-state distinguishability contracts under noisy quantum channels, even when circuits have similar transpilation depths. Generally, deeper compiled circuits require higher class distinguishability to maintain accurate performance in the presence of noise, and this degradation increases as exposure to noise accumulates, while also depending on measurement thresholds, finite-shot effects, backend calibration, and residual device-specific factors. For smaller systems, a classical pre-assessment of expected degradation is feasible, although its precision depends on the estimation of device-dependent coefficients and other backend-specific factors. Overall, our theoretical analysis and empirical findings indicate that QPBC models with higher average quantum relative entropy and lower transpilation depths are more likely to preserve their ideal simulation performance when executed in noisy environments. Taken together, these results provide a practical basis for screening trained QPBC models according to expected degradation trends before deployment, potentially reducing exploratory hardware experimentation for model selection.

These experiments were conducted within a constrained system to ensure comparability and were extensively evaluated on noisy hardware simulators, with subset verification performed on corresponding real hardware for binary classification problems. Ongoing research aims to enhance the estimation of device-dependent coefficients and to scale the methods for larger system sizes. Nonetheless, this work has opened new avenues for developing improved NISQ device-aware Quantum Machine Learning algorithms by integrating quantum information theory with the practical hardware-aware implementation of quantum algorithms.

Statements and Declarations

Data and Code Availability. The datasets generated and/or analyzed during the current study, along with code used for training, noisy-backend testing, transpilation-depth generation, and entropy analysis, are available in the following repositories:

- VQC-QNN Training and Testing Code and Data (Shinde 2026b):https://github.com/AakashShindeHelsinki/VQC_QNN_QML_Model_Training_and_Benchmarking
- Data Reupload Training and Testing Code and Result Data (Shinde 2026a):https://github.com/AakashShindeHelsinki/DataReupload_QML_Training_and_Benchmarking
- QML model testing on noisy devices runs, transpilation depth generation, depth-entropy analysis, real hardware runs, and data processing (Shinde 2026c): https://github.com/AakashShindeHelsinki/NoisyDev_Testing_and_Ent_Data_Processing
- Circuit example for ansatz definition: https://github.com/AakashShindeHelsinki/Quantum_Circuit_Diagrams_for_RelEnt_analysis

Any additional derived data supporting the findings of this study are available from the corresponding author on reasonable request.

Acknowledgments. We would like to express our special thanks to Valter Uotila for his valuable comments on the research and the creation of this manuscript. We also extend our gratitude to Ilmo Salmenperä, Frans Perkkola, Tung Bui Dang, and Leo Becker for their insightful discussions and guidance during the early phases of the research. The authors wish to acknowledge CSC – IT Center for Science, Finland, for computational resources.

AI Acknowledgments. AI-assisted tools were used only for language polishing, grammar correction, and editorial feedback on the presentation. All scientific reasoning, theoretical development, experimental design, validation, and interpretation were performed by the human authors, who take full responsibility for the manuscript.

Author Contribution. Aakash Ravindra Shinde contributed to conceptualization, methodology, software, formal analysis, investigation, and the original draft. Arianne Meijer-van de Griend contributed to conceptualization, theoretical verification, and manuscript review. Jukka K. Nurminen supervised the work and contributed to theoretical verification. All authors read and approved the final manuscript.

Funding. No funds, grants, or other support were received specifically for this study. The corresponding author is supported by the Finnish Doctoral Programme Network for Quantum Technology (QDOC) under the auspices of the Finnish Quantum Flagship.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Competing interests. The authors declare no competing interests.

References

Bowles, J., Ahmed, S., Schuld, M.: Better than classical? the subtle art of benchmarking quantum machine learning models (arXiv:2403.07059) (2024) https://doi.org/10.48550/arXiv.2403.07059 . arXiv:2403.07059 [quant-ph]

Bermejo, P., Braccia, P., Rudolph, M.S., Holmes, Z., Cincio, L., Cerezo, M.: Quantum convolutional neural networks are (effectively) classically simulable. arXiv preprint arXiv:2408.12739 (2024)

Buonaiuto, G., Gargiulo, F., De Pietro, G., Esposito, M., Pota, M.: The effects of quantum hardware properties on the performances of variational quantum learning algorithms. Quantum Machine Intelligence **6**(1), 9 (2024)

Buchanan, S., Gilboa, D., Wright, J.: Deep networks and the multiple manifold problem. arXiv preprint arXiv:2008.11245 (2020)

Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Ahmed, S., Ajith, V., Sohaib Alam, M., Alonso-Linaje, G., AkashNarayanan, B., Asadi, A., Arrazola, J.M., Azad, U., Banning, S., Blank, C., Bromley, T.R., Cordier, B.A., Ceroni, J., Delgado, A., Di Matteo, O., Dusko, A., Garg, T., Guala, D., Hayes, A., Hill, R., Ijaz, A., Isacsson, T., Ittah, D., Jahangiri, S., Jain, P., Jiang, E., Khandelwal, A., Kottmann, K., Lang, R.A., Lee, C., Loke, T., Lowe, A., McKiernan, K., Meyer, J.J., Montañez-Barrera, J.A., Moyard, R., Niu, Z., O'Riordan, L.J., Oud, S., Panigrahi, A., Park, C.-Y., Polatajko, D., Quesada, N., Roberts, C., Sá, N., Schoch, I., Shi, B., Shu, S., Sim, S., Singh, A., Strandberg, I., Soni, J., Száva, A., Thabet, S., Vargas-Hernández, R.A., Vincent, T., Vitucci, N., Weber, M., Wierichs, D., Wiersema, R., Willmann, M., Wong, V., Zhang, S., Killoran, N.: pennylane.math.quantum — PennyLane — docs.pennylane.ai. https://docs.pennylane.ai/en/stable/_modules/pennylane/math/quantum.html#relative_entropy. [Accessed 08-06-2026]

Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Ahmed, S., Ajith, V., Sohaib Alam, M., Alonso-Linaje, G., AkashNarayanan, B., Asadi, A., Arrazola, J.M., Azad, U., Banning, S., Blank, C., Bromley, T.R., Cordier, B.A., Ceroni, J., Delgado, A., Di Matteo, O., Dusko, A., Garg, T., Guala, D., Hayes, A., Hill, R., Ijaz, A., Isacsson, T., Ittah, D., Jahangiri, S., Jain, P., Jiang, E., Khandelwal, A., Kottmann, K., Lang, R.A., Lee, C., Loke, T., Lowe, A., McKiernan, K., Meyer, J.J., Montañez-Barrera, J.A., Moyard, R., Niu, Z., O'Riordan, L.J., Oud, S., Panigrahi, A., Park, C.-Y., Polatajko, D., Quesada, N., Roberts, C., Sá, N., Schoch, I., Shi, B., Shu, S., Sim, S., Singh, A., Strandberg, I., Soni, J., Száva, A., Thabet, S., Vargas-Hernández, R.A., Vincent, T., Vitucci, N., Weber, M., Wierichs, D., Wiersema, R., Willmann, M., Wong, V., Zhang, S., Killoran, N.: pennylane.measurements.state — PennyLane — docs.pennylane.ai. https://docs.pennylane.ai/en/stable/_modules/pennylane/measurements/state.html#density_matrix. [Accessed 08-06-2026]

Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Ahmed, S., Ajith, V., Sohaib Alam, M., Alonso-Linaje, G., AkashNarayanan, B., Asadi, A., Arrazola, J.M., Azad, U.,

Banning, S., Blank, C., Bromley, T.R., Cordier, B.A., Ceroni, J., Delgado, A., Di Matteo, O., Dusko, A., Garg, T., Guala, D., Hayes, A., Hill, R., Ijaz, A., Isacsson, T., Ittah, D., Jahangiri, S., Jain, P., Jiang, E., Khandelwal, A., Kottmann, K., Lang, R.A., Lee, C., Loke, T., Lowe, A., McKiernan, K., Meyer, J.J., Montañez-Barrera, J.A., Moyard, R., Niu, Z., O'Riordan, L.J., Oud, S., Panigrahi, A., Park, C.-Y., Polatajko, D., Quesada, N., Roberts, C., Sá, N., Schoch, I., Shi, B., Shu, S., Sim, S., Singh, A., Strandberg, I., Soni, J., Száva, A., Thabet, S., Vargas-Hernández, R.A., Vincent, T., Vitucci, N., Weber, M., Wierichs, D., Wiersema, R., Willmann, M., Wong, V., Zhang, S., Killoran, N.: PennyLane: Automatic differentiation of hybrid quantum-classical computations. arXiv e-prints, 1811–04968 (2018) https://doi.org/10.48550/arXiv.1811.04968 arXiv:1811.04968 [quant-ph]

Bärligea, A., Poggel, B., Lorenz, J.M.: Scalability challenges in variational quantum optimization under stochastic noise. Physical Review A **112**(3), 032407 (2025)

Brun, T.A.: Quantum error correction. arXiv preprint arXiv:1910.03672 (2019)

Barr, B., Xu, K., Silva, C.T., Bertini, E., Reilly, R., Bruss, C.B., Wittenbach, J.D.: Towards ground truth explainability on tabular data. ArXiv **abs/2007.10532** (2020)

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., *et al.*: Variational quantum algorithms. Nature Reviews Physics **3**(9), 625–644 (2021)

Datta, N.: Min-and max-relative entropies and a new entanglement monotone. IEEE Transactions on Information Theory **55**(6), 2816–2826 (2009)

Du, Y., Hsieh, M.-H., Liu, T., Tao, D.: Expressive power of parametrized quantum circuits. Physical Review Research **2**(3), 033125 (2020)

De Palma, G., Marvian, M., Rouzé, C., França, D.S.: Limitations of variational quantum algorithms: a quantum optimal transport approach. PRX Quantum **4**(1), 010309 (2023)

Fontana, E., Fitzpatrick, N., Ramo, D.M., Duncan, R., Rungger, I.: Evaluating the noise resilience of variational quantum algorithms. Physical Review A **104**(2), 022403 (2021)

Goldt, S., Mézard, M., Krzakala, F., Zdeborová, L.: Modeling the influence of data structure on learning in neural networks: The hidden manifold model. Physical Review X (2019)

Guyon, I.M.: Design of experiments for the nips 2003 variable selection benchmark. (2003). https://api.semanticscholar.org/CorpusID:115452637

Havlíček, V., Córcoles, A.D., Temme, K., Harrow, A.W., Kandala, A., Chow,

J.M., Gambetta, J.M.: Supervised learning with quantum-enhanced feature spaces. Nature **567**(7747), 209–212 (2019)

Hur, T., Kim, L., Park, D.K.: Quantum convolutional neural network for classical data classification. Quantum Machine Intelligence **4**(1), 3 (2022)

Hubregtsen, T., Pichlmeier, J., Stecher, P., Bertels, K.: Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability. Quantum Machine Intelligence **3**(1), 9 (2021)

Huo, Y., Wei, J., Kverne, C., Akewar, M., Bhimani, J., Patel, T.: Revisiting noise-adaptive transpilation in quantum computing: How much impact does it have? In: 2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pp. 1–9 (2025). IEEE

IonQ, Inc.: Simulation with Noise Models. (2026). IonQ, Inc.. Accessed 08-06-2026. https://docs.ionq.com/guides/simulation-with-noise-models

Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C.J., Lishman, J., Gacon, J., Martiel, S., Nation, P.D., Bishop, L.S., Cross, A.W., Johnson, B.R., Gambetta, J.M.: Quantum computing with Qiskit (2024). https://doi.org/10.48550/arXiv.2405.08810

Khanal, B., Rivas, P.: Evaluating the impact of noise on variational quantum circuits in nisq era devices. In: 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), pp. 1658–1664 (2023). IEEE

Lidar, D.A., Brun, T.A. (eds.): Quantum Error Correction. Cambridge University Press, ??? (2013). https://doi.org/10.1017/CBO9781139034807

Larocca, M., Thanasilp, S., Wang, S., Sharma, K., Biamonte, J., Coles, P.J., Cincio, L., McClean, J.R., Holmes, Z., Cerezo, M.: Barren plateaus in variational quantum computing. Nature Reviews Physics **7**(4), 174–189 (2025)

Monnet, M., Chaabani, N., Drăgan, T.-A., Schachtner, B., Lorenz, J.M.: Understanding the effects of data encoding on quantum-classical convolutional neural networks. In: 2024 IEEE International Conference on Quantum Computing and Engineering (QCE), vol. 1, pp. 1436–1446 (2024). IEEE

Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, UK (2000)

Nielsen, F.: Generalized bhattacharyya and chernoff upper bounds on bayes error using quasi-arithmetic means. Pattern Recognition Letters **42**, 25–34 (2014) https://doi.org/10.1016/j.patrec.2014.01.002 . arXiv:1401.4788 [cs.CV]

Nuradha, T., Wilde, M.M.: Fidelity-based smooth min-relative entropy: Properties and

applications. IEEE Transactions on Information Theory **70**(6), 4170–4196 (2024)

Phalak, K., Ghosh, S.: Shot optimization in quantum machine learning architectures to accelerate training. IEEE Access **11**, 41514–41523 (2023)

Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., Latorre, J.I.: Data re-uploading for a universal quantum classifier. Quantum **4**, 226 (2020)

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research **12**, 2825–2830 (2011). Accessed 08-06-2026

Recio-Armengol, E., Ahmed, S., Bowles, J.: Train on classical, deploy on quantum: scaling generative quantum machine learning to a thousand qubits (2026). https://arxiv.org/abs/2503.02934

Schetakis, N., Bonfini, P., Alisoltani, N., Blazakis, K., Tsintzos, S.I., Askitopoulos, A., Aghamalyan, D., Fafoutellis, P., Vlahogianni, E.I.: Data re-uploading in quantum machine learning for time series: application to traffic forecasting. arXiv preprint arXiv:2501.12776 (2025)

Seo, J.: Multivariate unbounded quantum regression via log-ratio probabilities mitigating barren plateaus. IEEE Access **14**, 4877–4885 (2026)

Shinde, A.R.: Data Reupload (Probabilistic and Expectation Value Measure) Models for Quantum Machine Learning Testing and Data Processing. https://github.com/AakashShindeHelsinki/DataReupload_QML_Training_and_Benchmarking

Shinde, A.R.: PQCs and QNN (Probabilistic and Expectation Value Measures) Models for Quantum Machine Learning Testing and Data Processing. https://github.com/AakashShindeHelsinki/VQC_QNN_QML_Model_Training_and_Benchmarking

Shinde, A.R.: QML Model Testing on Noisy Devices, Depth-entropy Analysis and Data Processing . https://github.com/AakashShindeHelsinki/NoisyDev_Testing_and_Ent_Data_Processing

Sim, S., Johnson, P.D., Aspuru-Guzik, A.: Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Advanced Quantum Technologies **2**(12), 1900070 (2019)

Sharma, K., Khatri, S., Cerezo, M., Coles, P.J.: Noise resilience of variational quantum compiling. New Journal of Physics **22**(4), 043006 (2020)

Singkanipa, P., Lidar, D.A.: Beyond unital noise in variational quantum algorithms: noise-induced barren plateaus and limit sets. Quantum **9**, 1617 (2025)

Uotila, V., Salmenperä, I., Becker, L., Griend, A., Shinde, A.R., Nurminen, J.K.: Perspectives on utilization of measurements in quantum algorithms. In: 2025 IEEE International Conference on Quantum Software (QSW) (2025). https://doi.org/10.1109/QSW67625.2025.00016

Wang, S., Fontana, E., Cerezo, M., Sharma, K., Sone, A., Cincio, L., Coles, P.J.: Noise-induced barren plateaus in variational quantum algorithms. Nature communications **12**(1), 6961 (2021)

Wilde, M.M.: Mark m. wilde: Quantum information theory. Quantum Information Processing **13**(2), 587–590 (2014) https://doi.org/10.1007/s11128-013-0690-x

Zang, O., Barrué, G., Quertier, T.: Benchmarking data encoding methods in quantum machine learning. arXiv preprint arXiv:2505.14295 (2025)

Appendix A Component Definition and Descriptions

A.1 QPBC model architecture

Here, we define several popular architectures used to implement the QPBC models and our naming conventions for these methods. Graphical representation of these architectures is presented in Fig. 1

A.1.1 Parameterized or Variational Quantum Circuits (PQC/VQC)

Here, we define Parameterized Quantum Circuit (PQC)-based models as multi-layered ansatzes composed of both single and two-qubit quantum gates, without subsequent reduction of qubit operations. The depth of these circuits can be enhanced through the repetition of gate structures, tailored to meet the complexity requirements of the model, thus yielding improved performance outcomes. Data encoding structure is appended once at the beginning of the structure.

While increasing the repetition of the ansatzes enhances the expressibility of PQC models, it also leads to an increase in circuit depth and associated implementational constraints. This trade-off between expressibility and practicality is a critical aspect to consider in designing PQC models.

A.1.2 Quantum Neural Networks (QNN)

QNNs are inherently similar to PQCs, with one key structural difference between PQCs and QNNs being that in QNNs, as depicted in Hur et al. (2022), the ansatzes acting on all qubits are halved with each layer, ultimately leading to the measurement of the remaining qubit.

QNNs have a reduced number of operational ansatzes compared to PQCs/VQCs, which helps decrease the overhead associated with circuit implementation. They have

also been shown to significantly reduce the issues related to the Barren Plateau problem during the training phase. However, there are ongoing questions about the ease of simulating such structures and the expressibility of these models (Bermejo et al. 2024).

A.1.3 Data Reupload

Data reuploading involves repeatedly encoding classical data into quantum circuits, which can significantly improve the expressiveness of quantum models without a substantial increase in computational requirements.

The process of data reuploading is accomplished by linking together repeated units of tunable PQCs and Static Data encoding circuits in a sequence. Consequently, a quantum circuit can be structured as a combination of data reuploading and single-qubit processing units. Several recent studies suggest that data reuploading can enhance the performance and trainability of quantum models (Schetakis et al. 2025).

A.2 Data Encoding methods

Data encoding is a key element of Quantum Machine Learning (QML). Unlike classical methods that use vectors, QML requires transforming classical data into quantum states through encoding circuits. These circuits are referred to as quantum feature maps or quantum encodings, also interchangeably known as embeddings. The encoding process significantly affects QML model performance, so choosing the right encoding method for the dataset is essential (Zang et al. 2025). We have utilized the most commonly used encoding methods for Variational Quantum Classification models in our work.

A.2.1 Angle Encoding

One of the most common encoding methods involves representing data values for each feature as an angle of rotation along different axes: X, Y, or Z. This is expressed mathematically as $R_j(\theta) = e^{-i\frac{\theta}{2}\sigma_j}$, where $j \in \{x, y, z\}$ and σ_j represents the Pauli matrices $\sigma_x, \sigma_y, \sigma_z$. In this encoding method, N feature values are embedded into N qubits (Monnet et al. 2024). For our experimentation, we have tested for all three rotational values across all variational circuits in this manuscript.

A.2.2 Amplitude Encoding

In the case of Amplitude encoding, classical feature vectors are represented as amplitudes. Here, classical feature vector $x = \{x_1, .., x_n\}$ can be written as $|\psi\rangle = \Sigma_{i=1}^{n} x_i |i\rangle$, where $|\psi\rangle$ is the resulting quantum state and $|i\rangle$ are the quantum state for each vector components for i being the binary index of values. An N-dimensional amplitude-encoded classical vector requires $\log_2 N$ qubits. Considering the expensive circuit implementation of state initialization, evaluation of amplitude encoding was essential in understanding the bottleneck of QML implementation (Monnet et al. 2024).

A.2.3 Instantaneous Quantum Polynomial (IQP) Encoding

The instantaneous Quantum Polynomial-time (IQP) encoding maps features into qubits via diagonal gates in an IQP circuit. An IQP circuit is a type of quantum circuit that consists of a sequence of Hadamard gates followed by a sequence of diagonal gates in the computational basis. In this context, the diagonal gates are single-qubit Rz rotations applied to each qubit, encoding the n features. This is followed by two-qubit ZZ entanglers, represented as $e^{-ix_i x_j \sigma_z \otimes \sigma_z}$. This method is capable of encoding N data points into N qubits (Havlíček et al. 2019).

A.3 Measurements

This section provides details regarding the Measurement Schemes used during the experimentation.

A.3.1 Probabilistic measurement

A "shot" in a quantum circuit is a single execution that produces probabilistic measurement results of 0 or 1 for each qubit. In Variational Circuits, quantum information is projected into the computational basis, and measurement probabilities are estimated over multiple shots, as several executions are needed for accurate probability estimation (Uotila et al. 2025).

Although increasing the number of shots improves precision, it also raises resource costs. The statistical error in probability estimation decreases with the square root of the number of shots used. Typically, a default value of 1,024 shots is used, but this can be adjusted according to the desired level of precision.

A.3.2 Expectation Values

The expectation value is the probabilistic average of the results (or measurements) from an experiment. It represents an average of all possible outcomes, weighted by their likelihood. However, it is important to note that the expectation value is not necessarily the most probable value of a measurement; in fact, it can even have zero probability.

Given an observable $A = \Sigma_k \lambda_k |k\rangle \langle k|$, we obtain eigenvalue λ_k with probability p_k, we can naturally define an expectation value of this observable in the state $|k\rangle$ as $\langle A_\psi \rangle \equiv \Sigma_k \lambda_k \langle \psi | P_k | \psi \rangle = \langle \psi | (\Sigma_k \lambda_k P_k) | \psi \rangle = \langle \psi | A | \psi \rangle = Tr(A |\psi\rangle \langle \psi|)$ Brun (2019). In our case, as measured in the Z basis, each time we will get $+1$ or -1 (Uotila et al. 2025).

A.4 Datasets Generation method

For extensive evaluation, datasets with varying complexities and intricacies are essential. Hence, we employed five different methods for dataset generation, each with its own prediction difficulties. For each data generation, we kept the modifiable parameters constant, maintaining significant complexities. The first two of the dataset

generators are chosen for their ability to generate Gaussian and non-Gaussian distributions, respectively. While the final three are inspired by the benchmarking initiative in Bowles et al. (2024).

A.4.1 Make Classification by Scikit-Learn (Linear Dataset Creator)

Make Classification is a dataset generation method based on a Gaussian distribution for classifying clusters. Contrary to Make Classification, we utilize a Synthetic Dataset by Capital One for a non-Gaussian distribution for dataset creation. For the latter three datasets, we draw inspiration from Bowles et al. (2024) for selecting our datasets used for benchmarking. Details of the dataset generator methods and associated classification complexities are as follows.

The algorithm for the Make Classification function is based on the concept in Guyon (2003) and was developed to create the "Madelon" dataset. This function is particularly effective at introducing noise through attributes like correlated, redundant, and uninformative features. Additionally, it generates noise by forming multiple Gaussian clusters within each class and applying linear transformations to the feature space (Pedregosa et al. 2011).

A.4.2 Synthetic Data by Capital One (Non-linear Dataset)

The Synthetic Data generator is a tool for creating tabular datasets that incorporate non-linearity using a Non-Gaussian distribution. The feature set generated is tunable over three parameters: informative, redundant, and nuisance. Informative parameters generate features crucial for determining binary labels, which are defined using Copula theory, a mathematical approach that separates the correlation structure from the marginal distributions of the feature vectors. The Redundant parameter helps create a linear combination of the informative features, while nuisance features help generate features consisting of uncorrelated vectors drawn from the range [-1,1]. For more details about the dataset, please refer to Barr et al. (2020).

A.4.3 Hidden Manifold

Goldt et al. (2019) introduced a data generation procedure designed to explore the impact of data structure, such as the size of a hidden manifold believed to influence problem difficulty, on learning outcomes. The process involves generating inputs on a low-dimensional manifold, which are then labeled by a simple, randomly initialized neural network. Finally, these inputs are projected into a d-dimensional space (Bowles et al. 2024).

A.4.4 Hyperplanes

This data generation method creates several hyperplanes in a k-dimensional space and assigns labels to randomly sampled points based on the parity of perceptron classifiers using these hyperplanes as decision boundaries. A label indicates whether a point is on the positive side of an even number of hyperplanes, aiming to reflect the "global"

structure of the problem related to the arrangement of hyperplanes (Bowles et al. 2024).

A.4.5 Two Curves

This data generation procedure is inspired by a theoretical study, as discussed in Buchanan et al. (2020), which demonstrates how the performance of neural networks is affected by the curvature and the distance between two one-dimensional curves embedded in a higher-dimensional space. Following the implementation in (Bowles et al. 2024), we also used low-degree Fourier series to embed two sets of data sampled from a one-dimensional interval, with one set for each class, representing these data as curves in d-dimensional space, while also adding some Gaussian noise.

Appendix B Device Results

In this section, we provide the tabular data across 51 distinct QPBC models for 8 distinct fake(simulated) noisy backends across different models, encodings, and ansatzes. The following tables contain accuracy under ideal conditions for the model, the calculated average quantum relative entropy using the test dataset, and the accuracy observed for the noisy simulators along with their respective transpilation depth. Values shown in green indicate that the accuracy of the noisy device differs from the accuracy of the simulation by less than 0.03. Values in red represent an accuracy difference greater than 0.08, while values with accuracy differences between 0.03 and 0.08 are shown in blue. The value difference over multiple runs for the same test data and the same device does not differ more than ± 0.001 in Tables B1 and B2.

The IonQ Aria1 simulator results are based on a substantially smaller sample size and should therefore be interpreted separately from the larger backend-wise regression summaries.

Idx	Ansatz	Model	Encoding	Avg. Rel. Ent.	Sim. Acc.	Qiskit Fez		Qiskit Marrakash		Qiskit Torino		Qiskit Prague	
						Acc.	Depth	Acc.	Depth	Acc.	Depth	Acc.	Depth
1	QNN_model_170_pool	VQC_QNN_EXP	AngleY	0.5352	0.892	0.883	62	0.899	62	0.901	62	0.896	62
2	QNN_H_CZ_RX_pool	VQC_QNN_PROB	Amplitude	0.0881	0.932	0.583	1472	0.692	1604	0.515	1606	0.883	1612
3	QNN_H_CZ_RX_pool	VQC_QNN_PROB	AngleX	1.5808	0.933	0.932	46	0.931	46	0.935	46	0.932	46
4	QNN_model_170_pool	VQC_QNN_EXP	IQP	0.2337	0.931	0.928	308	0.918	320	0.923	275	0.932	310
5	H_CZ_RX	VQC_QNN_EXP	AngleY	0.0245	0.929	0.884	54	0.908	55	0.890	56	0.890	54
6	QNN_model_170_pool	VQC_QNN_PROB	IQP	0.6240	0.888	0.890	292	0.889	286	0.893	293	0.887	277
7	RY_RZ_CNOT	VQC_QNN_EXP	IQP	0.0355	0.897	0.773	382	0.767	389	0.729	368	0.817	393
8	RY_CNOT	VQC_QNN_EXP	IQP	0.0191	0.939	0.785	382	0.855	381	0.742	376	0.831	382
9	RXRZ_CNOT	VQC_QNN_PROB	AngleX	0.1864	0.947	0.933	67	0.936	67	0.934	67	0.931	67
10	INV_QFT_U3_MODEL	VQC_QNN_PROB	AngleY	0.2127	0.949	0.947	147	0.945	132	0.943	151	0.940	132
11	RY_CZ	VQC_QNN_EXP	Amplitude	0.1710	0.905	0.714	1745	0.744	1974	0.593	1981	0.799	1835
12	INV_QFT_U3_MODEL	VQC_QNN_EXP	IQP	0.0956	0.904	0.861	350	0.871	340	0.862	384	0.862	385
13	QNN_RY_CRX_pool	VQC_QNN_EXP	AngleX	1.0540	0.853	0.857	58	0.852	58	0.860	58	0.854	58
14	QNN_model_150params	VQC_QNN_PROB	AngleX	0.6778	0.997	0.995	62	0.997	62	0.997	62	0.999	62
15	QNN_H_CZ_RX_pool	VQC_QNN_EXP	AngleY	0.2083	0.942	0.936	47	0.938	47	0.931	47	0.943	47
16	RXRZ_CNOT	DATA_REUP_PROB	AngleX	0.1032	0.925	0.927	67	0.919	67	0.915	67	0.919	67
17	RX_RZ_Multi_CRX	DATA_REUP_PROB	AngleZ	1.6190	0.955	0.917	1193	0.924	1192	0.907	1150	0.932	1100
18	RY_RZ_CNOT	DATA_REUP_PROB	AngleX	1.4200	0.947	0.941	162	0.944	171	0.941	134	0.944	173
19	RXRZ_CRZPool	DATA_REUP_PROB	AngleY	2.5523	0.944	0.942	162	0.944	171	0.940	134	0.943	173
20	H_CZ_RX	DATA_REUP_PROB	Amplitude	1.5761	0.913	0.487	15354	0.456	15606	0.475	15414	0.519	15483
21	RX_RZ_Multi_CRZ	DATA_REUP_PROB	AngleY	1.1158	0.927	0.926	1124	0.929	2083	0.897	2063	0.918	2172
22	QNN_RY_CNOT_pool	VQC_QNN_EXP	AngleY	1.6222	0.966	0.965	62	0.963	62	0.960	62	0.962	62
23	RY_CNOT	DATA_REUP_PROB	IQP	0.0475	0.939	0.504	2519	0.529	2519	0.519	2557	0.531	2580
24	RX_RZ_Multi_CRZ	DATA_REUP_PROB	AngleY	0.9210	0.949	0.907	1100	0.915	1102	0.841	1110	0.937	1064
25	RXRZ_CRZPool	DATA_REUP_PROB	IQP	0.8288	0.926	0.877	1359	0.900	1445	0.861	1365	0.907	1374
26	RX_RZ_Multi_CRZ	DATA_REUP_PROB	AngleY	1.1158	0.927	0.920	1167	0.927	1063	0.897	1113	0.932	1122
27	RXRZ_CRZPool	DATA_REUP_EXP	AngleY	2.6991	0.920	0.920	78	0.922	74	0.919	74	0.922	86
28	RXRZ_CNOT	DATA_REUP_EXP	AngleY	0.0551	0.938	0.912	67	0.923	67	0.908	67	0.920	67
29	RX_RZ_Multi_CRX	DATA_REUP_EXP	AngleY	0.0700	0.907	0.771	1115	0.774	1200	0.694	1180	0.819	1165
30	QNN_RY_CRX_pool	VQC_QNN_EXP	AngleX	1.0540	0.853	0.857	58	0.852	58	0.860	58	0.854	58
31	RXRZ_CNOT	VQC_QNN_EXP	AngleY	0.0930	0.913	0.929	67	0.923	67	0.916	67	0.920	67
32	RY_CZ	VQC_QNN_EXP	Amplitude	0.0830	0.899	0.644	1983	0.648	1979	0.538	1978	0.706	1964
33	QNN_RY_CNOT_pool	VQC_QNN_PROB	Amplitude	0.2120	0.909	0.749	1614	0.818	1656	0.621	1521	0.756	1406
34	INV_QFT_U3_MODEL	VQC_QNN_PROB	IQP	0.1667	0.945	0.937	343	0.940	342	0.936	350	0.940	399
35	RY_CNOT	VQC_QNN_PROB	IQP	0.0380	0.933	0.872	378	0.914	383	0.846	377	0.904	382
36	RY_CNOT	VQC_QNN_PROB	AngleY	0.2970	0.907	0.902	105	0.902	106	0.904	105	0.904	105
37	RY_CRX	VQC_QNN_PROB	IQP	0.2750	0.906	0.897	355	0.911	365	0.891	355	0.902	359
38	RXRZ_CNOT	VQC_QNN_PROB	AngleX	0.3167	0.888	0.896	67	0.890	67	0.893	67	0.891	67
39	H_CZ_RX	VQC_QNN_PROB	AngleY	0.2828	0.875	0.874	54	0.873	54	0.869	54	0.872	55
40	RY_RZ_CNOT	VQC_QNN_PROB	IQP	0.0671	0.959	0.852	399	0.881	398	0.841	379	0.879	386
41	RY_RZ_CNOT	DATA_REUP_EXP	IQP	0.1131	0.935	0.734	1279	0.795	1300	0.674	1281	0.838	1368
42	RXRZ_CRZPool	DATA_REUP_PROB	AngleZ	2.7730	0.930	0.927	88	0.928	73	0.925	69	0.928	73
43	RXRZ_CRZPool	DATA_REUP_PROB	IQP	2.2790	0.923	0.902	1413	0.910	1415	0.908	1382	0.914	1361
44	RXRZ_CRZPool	DATA_REUP_PROB	IQP	1.9780	0.892	0.902	1381	0.905	1438	0.899	1435	0.905	1361
45	RY_RZ_CNOT	DATA_REUP_EXP	AngleX	0.2600	0.940	0.924	180	0.928	133	0.915	177	0.925	192
46	RXRZ_CRZPool	DATA_REUP_EXP	AngleZ	0.5350	0.892	0.937	76	0.941	69	0.940	74	0.941	78
47	QNN_RY_CRZ_pool	VQC_QNN_EXP	AngleY	1.5652	0.939	0.940	47	0.941	47	0.943	47	0.943	47
48	RY_CNOT	VQC_QNN_EXP	IQP	0.0131	0.912	0.786	377	0.829	380	0.727	375	0.817	381
49	RXRZ_CNOT	DATA_REUP_PROB	AngleZ	0.0318	0.902	0.859	67	0.869	67	0.830	67	0.857	67
50	RY_CZ	DATA_REUP_PROB	AngleX	0.0869	0.902	0.859	404	0.869	403	0.830	403	0.857	398
51	RX_RZ_Multi_CRZ	DATA_REUP_PROB	AngleZ	0.5350	0.892	0.831	1192	0.874	1125	0.771	1120	0.877	1170

Table B1: Analysis of variational quantum classification models on Qiskit backends. Backend abbreviations: Marr. = Fake Marrakesh, Tor. = Fake Torino, Prag. = Fake Prague.

Idx	Ansatz	Model	Encoding	Avg. Rel. Ent.	Sim. Acc.	Qiskit Sherbrooke		IQM Aphrodite		IQM Apollo		IonQ Aria1	
						Acc.	Depth	Acc.	Depth	Acc.	Depth	Acc.	Depth
1	QNN_model_170_pool	VQC_QNN_EXP	AngleY	0.5352	0.892	0.900	66	0.905	99	0.895	99	0.898	131
2	QNN_H_CZ_RX_pool	VQC_QNN_PROB	Amplitude	0.0881	0.932	0.883	1804	0.883	616	0.883	578	0.883	527
3	QNN_H_CZ_RX_pool	VQC_QNN_PROB	AngleX	1.5808	0.933	0.938	50	0.934	18	0.930	18	0.934	25
4	QNN_model_170_pool	VQC_QNN_EXP	IQP	0.2337	0.931	0.913	402	0.838	145	0.825	140	0.924	78
5	H_CZ_RX	VQC_QNN_EXP	AngleY	0.0245	0.929	0.843	144	0.674	37	0.656	37	0.882	37
6	QNN_model_170_pool	VQC_QNN_PROB	IQP	0.6240	0.888	0.888	399	0.844	127	0.835	140	0.885	78
7	RY_RZ_CNOT	VQC_QNN_EXP	IQP	0.0355	0.897	0.701	580	0.555	141	0.551	147	0.816	58
8	RY_CNOT	VQC_QNN_EXP	IQP	0.0191	0.939	0.598	505	0.522	159	0.514	160	0.736	72
9	RXRZ_CNOT	VQC_QNN_PROB	AngleX	0.1864	0.947	0.930	90	0.877	36	0.872	36	0.942	24
10	INV_QFT_U3_MODEL	VQC_QNN_PROB	AngleY	0.2127	0.949	0.937	181	0.939	96	0.922	94	0.944	22
11	RY_CZ	VQC_QNN_EXP	Amplitude	0.1710	0.905	0.545	2406	0.531	642	0.499	714	0.787	594
12	INV_QFT_U3_MODEL	VQC_QNN_EXP	IQP	0.0956	0.904	0.787	476	0.587	197	0.567	199	0.853	58
13	QNN_RY_CRX_pool	VQC_QNN_EXP	AngleX	1.0540	0.853	0.856	62	0.850	24	0.854	24	0.851	29
14	QNN_model_150params	VQC_QNN_PROB	AngleX	0.6778	0.997	0.997	66	0.991	26	0.995	26	0.995	26
15	QNN_H_CZ_RX_pool	VQC_QNN_EXP	AngleY	0.2083	0.942	0.927	50	0.925	18	0.916	18	0.940	25
16	RXRZ_CNOT	DATA_REUP_PROB	AngleX	0.1032	0.925	0.907	88	0.859	36	0.850	36	0.908	24
17	RX_RZ_Multi_CRX	DATA_REUP_PROB	AngleZ	1.6190	0.955	0.889	1413	0.654	555	0.653	559	0.930	93
18	RY_RZ_CNOT	DATA_REUP_PROB	AngleX	1.4200	0.947	0.937	239	0.940	22	0.940	22	0.946	21
19	RXRZ_CRZPool	DATA_REUP_PROB	AngleY	2.5523	0.944	0.940	239	0.942	22	0.934	22	0.941	21
20	H_CZ_RX	DATA_REUP_PROB	Amplitude	1.5761	0.913	0.508	17811	0.482	6367	0.482	6365	0.500	5102
21	RX_RZ_Multi_CRZ	DATA_REUP_PROB	AngleY	1.1158	0.927	0.821	2970	0.503	914	0.506	910	0.923	137
22	QNN_RY_CNOT_pool	VQC_QNN_EXP	AngleY	1.6222	0.966	0.961	50	0.952	26	0.957	26	0.960	33
23	RY_CNOT	DATA_REUP_PROB	IQP	0.0475	0.939	0.491	3173	0.508	1005	0.500	989	0.524	405
24	RX_RZ_Multi_CRZ	DATA_REUP_PROB	AngleY	0.9210	0.949	0.796	1671	0.495	556	0.487	544	0.951	93
25	RXRZ_CRZPool	DATA_REUP_PROB	IQP	0.8288	0.926	0.796	1790	0.501	821	0.501	790	0.906	227
26	RX_RZ_Multi_CRZ	DATA_REUP_PROB	AngleY	1.1158	0.927	0.811	1598	0.503	568	0.506	550	0.923	93
27	RXRZ_CRZPool	DATA_REUP_EXP	AngleY	2.6991	0.920	0.918	86	0.915	47	0.920	42	0.918	13
28	RXRZ_CNOT	DATA_REUP_EXP	AngleY	0.0551	0.938	0.892	99	0.798	36	0.792	36	0.904	21
29	RX_RZ_Multi_CRX	DATA_REUP_EXP	AngleY	0.0700	0.907	0.694	1517	0.491	524	0.474	547	0.838	93
30	QNN_RY_CRX_pool	VQC_QNN_EXP	AngleX	1.0540	0.853	0.856	62	0.850	24	0.854	24	0.851	29
31	RXRZ_CNOT	VQC_QNN_EXP	AngleY	0.0930	0.913	0.893	98	0.827	36	0.797	36	0.922	24
32	RY_CZ	VQC_QNN_EXP	Amplitude	0.0830	0.899	0.526	2509	0.496	661	0.522	674	0.652	594
33	QNN_RY_CNOT_pool	VQC_QNN_PROB	Amplitude	0.2120	0.909	0.545	1817	0.470	661	0.482	632	0.790	535
34	INV_QFT_U3_MODEL	VQC_QNN_PROB	IQP	0.1667	0.945	0.932	461	0.708	189	0.693	206	0.937	50
35	RY_CNOT	VQC_QNN_PROB	IQP	0.0380	0.933	0.717	506	0.500	176	0.517	142	0.850	72
36	RY_CNOT	VQC_QNN_PROB	AngleY	0.2970	0.907	0.889	142	0.804	52	0.814	52	0.899	36
37	RY_CRX	VQC_QNN_PROB	IQP	0.2750	0.906	0.897	442	0.728	174	0.736	175	0.898	64
38	RXRZ_CNOT	VQC_QNN_PROB	AngleX	0.3167	0.888	0.893	88	0.875	36	0.880	36	0.887	24
39	H_CZ_RX	VQC_QNN_PROB	AngleY	0.2828	0.875	0.873	144	0.852	37	0.843	37	0.869	37
40	RY_RZ_CNOT	VQC_QNN_PROB	IQP	0.0671	0.959	0.741	505	0.578	123	0.534	146	0.877	58
41	RY_RZ_CNOT	DATA_REUP_EXP	IQP	0.1131	0.935	0.577	1873	0.490	721	0.490	718	0.858	197
42	RXRZ_CRZPool	DATA_REUP_PROB	AngleZ	2.7730	0.930	0.921	97	0.927	24	0.925	20	0.928	13
43	RXRZ_CRZPool	DATA_REUP_PROB	IQP	2.2790	0.923	0.903	2057	0.552	845	0.560	776	0.915	227
44	RXRZ_CRZPool	DATA_REUP_PROB	IQP	1.9780	0.892	0.892	1812	0.595	785	0.583	811	0.906	227
45	RY_RZ_CNOT	DATA_REUP_EXP	AngleX	0.2600	0.940	0.899	182	0.913	22	0.915	22	0.919	21
46	RXRZ_CRZPool	DATA_REUP_EXP	AngleZ	0.5350	0.892	0.936	110	0.934	28	0.936	28	0.934	13
47	QNN_RY_CRZ_pool	VQC_QNN_EXP	AngleY	1.5652	0.939	0.937	50	0.933	18	0.935	18	0.936	29
48	RY_CNOT	VQC_QNN_EXP	IQP	0.0131	0.912	0.540	503	0.501	155	0.501	168	0.736	72
49	RXRZ_CNOT	DATA_REUP_PROB	AngleZ	0.0318	0.902	0.834	98	0.708	36	0.668	36	0.832	24
50	RY_CZ	DATA_REUP_PROB	AngleX	0.0869	0.902	0.834	823	0.708	92	0.668	92	0.819	92
51	RX_RZ_Multi_CRZ	DATA_REUP_PROB	AngleZ	0.5350	0.892	0.629	1613	0.500	571	0.433	541	0.912	93

Table B2: Analysis of variational quantum classification models on Sherbrooke, IQM, and IonQ backends. Backend abbreviations: Sherb. = Fake Sherbrooke, Aph. = Fake Aphrodite, Apo. = Fake Apollo.

Statistic	Value
Number of unique non-null model entries	1132
Mean of avg. rel. ent. : mean(rel_l2-l1)	0.59414
Mean of avg. rel. ent. : rel_l1-l2	0.59097
Average Mean avg. rel. ent. : [mean(rel_l2-l1) + mean(rel_l1-l2)]/2	0.59256
Median of avg. rel. ent. : median(rel_l2-l1)	0.31849
Median of avg. rel. ent. : median(rel_l1-l2)	0.31914
Average Median of avg. rel. ent. : [median(rel_l2-l1) + median(rel_l1-l2)]/2	0.31882
Standard deviation of avg. rel. ent. : std(rel_l2-l1)	0.65212
Standard deviation of avg. rel. ent. : std(rel_l1-l2)	0.64347
Average Standard deviation of avg. rel. ent. : [std(rel_l2-l1) + std(rel_l1-l2)]/2	0.64779
Mean of difference in avg. rel. ent. : mean(rel_l2-l1 - rel_l1-l2)	0.00316
Median of difference in avg. rel. ent.: median(rel_l2-l1 - rel_l1-l2)	-0.00017
Standard deviation of difference in avg. rel. ent. : std(rel_l2-l1 - rel_l1-l2)	0.10418
Mean average of avg. rel. ent. : mean(rel_l2-l1 + rel_l1-l2)	0.59256
Median average of avg. rel. ent. : median(rel_l2-l1 + rel_l1-l2)	0.31968
Standard deviation of avg. rel. ent. : std(rel_l2-l1 + rel_l1-l2)	0.64600

Table B3: Statistical Representation for the Average Quantum Relative Entropy (avg. rel. ent.) values as computed from Eq. (19). In this context, rel_l2-l1 represents the average quantum relative entropy between label 0 and label 1. When measuring expectation values, these labels correspond to -1 and 1, respectively. Similarly, rel_l1-l2 represents the average quantum relative entropy between label 1 and label 0, and in terms of expectation value measurement, these labels correspond to 1 and -1.

Regression model	vals.	L_{tr}	$\log(1 + L_{\text{tr}})$	$\mathcal{D}_{\text{avg}}$	$\log(1 + \mathcal{D}_{\text{avg}})$	$L_{\text{tr}} + \mathcal{D}_{\text{avg}}$	$\log(1 + L_{\text{tr}}) + \log(1 + \mathcal{D}_{\text{avg}})$
Qiskit_Fake_Fez	1017	0.70547	0.67172	-0.00713	-0.00743	0.71921	0.67163
Qiskit_Fake_Marrakesh	960	0.76435	0.66061	-0.00910	-0.01007	0.77085	0.65967
IQM_Fake_Apollo	850	0.21497	0.67342	0.08475	0.11655	0.33852	0.74186
Qiskit_Fake_Torino	1022	0.64760	0.69145	-0.01143	-0.01065	0.66513	0.69424
Qiskit_Fake_Sherbrooke	908	0.61206	0.70048	-0.00750	-0.00593	0.63606	0.70995
Qiskit_Fake_Prague	953	0.76586	0.67804	-0.00650	-0.00728	0.77443	0.67753
IQM_Fake_Aphrodite	866	0.41661	0.69266	0.10549	0.14218	0.52215	0.74968
IONQ_Fake_Aria1	51	-0.12498	-0.37135	-0.18832	-0.10699	0.04738	-0.16572

Table B4: Linear cross-validated regression comparison R^2 for accuracy degradation across all evaluated QPBC model-device observations.

Appendix C Device Graphs

We present a series of graphs illustrating various devices and transpilation depth values below 2000. The plotted points represent actual observed values for different QPBC models, including transpilation depth, accuracy, and average quantum relative entropy. The line graphs are generated from Equations Eq. (28) and Eq. (30), reflecting observations across a range of representative coefficient choices. Each model features a distinct set of coefficient values used to construct the line graph, but these can be accurately approximated over specific subsets. It is important to highlight that average

quantum relative entropy values are exceptionally small for some observations, which can pose challenges for precision calculations.

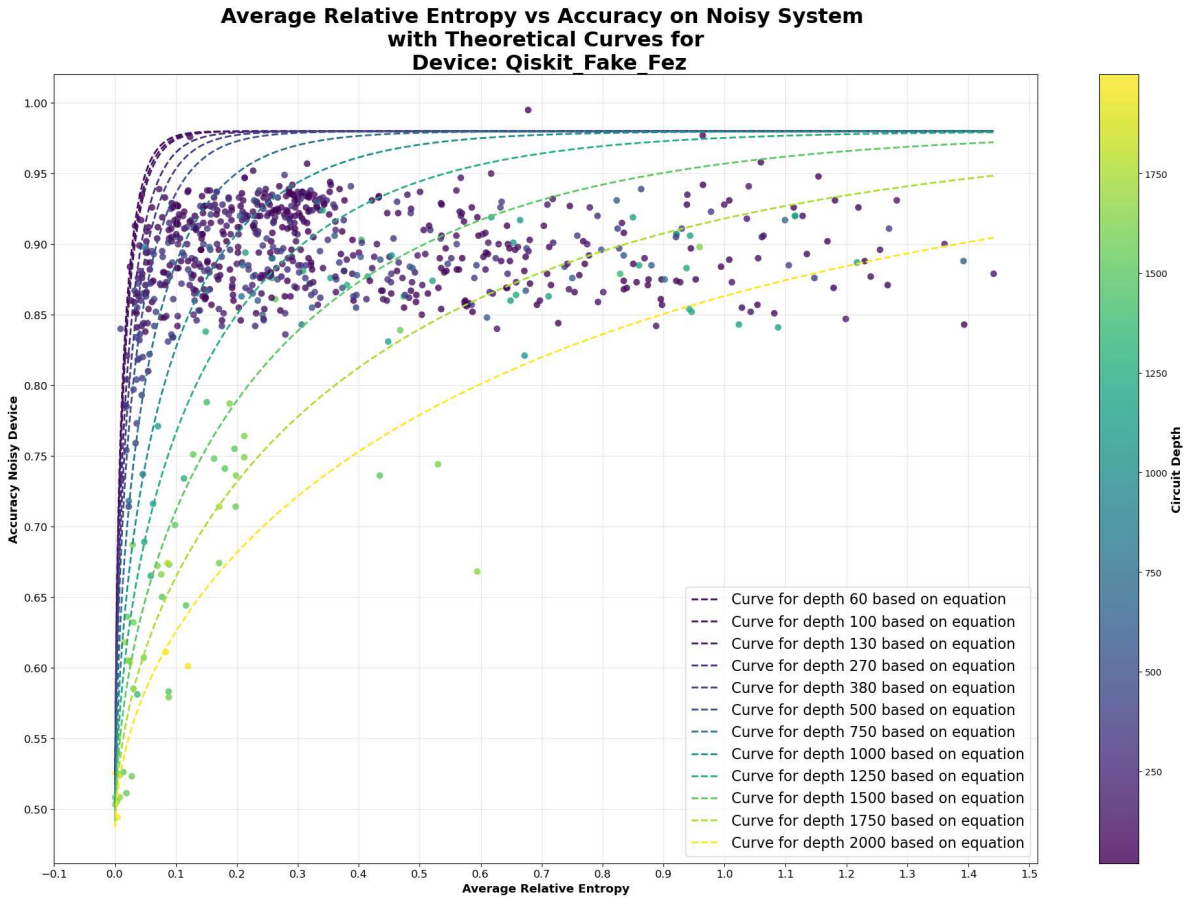

Fig. C1: Average quantum relative entropy vs Accuracy (Ideal-Noisy) with Theoretical Curves for Device: Qiskit Fake Fez, Depth $\leq$ 2000

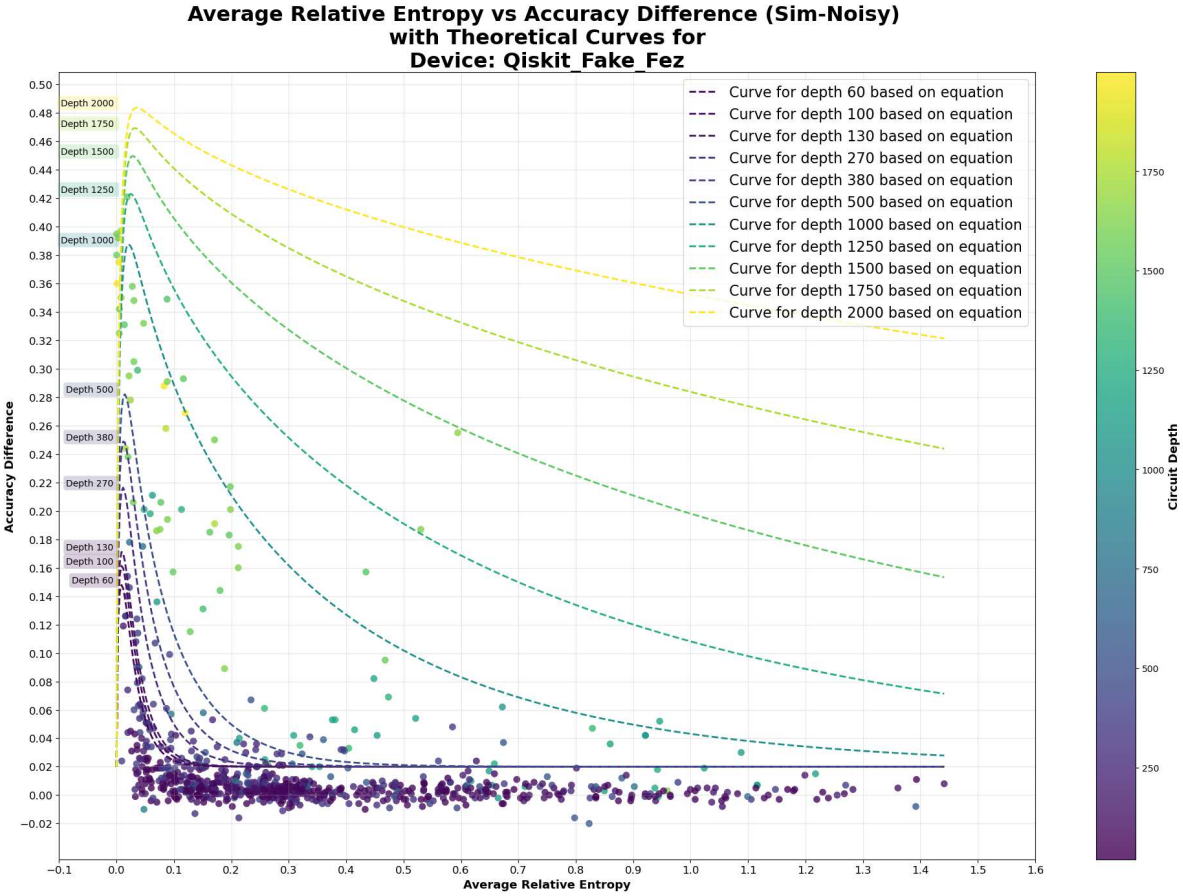

Fig. C2: Average quantum relative entropy vs Accuracy Difference (Ideal-Noisy) with Theoretical Curves for Device: Qiskit Fake Fez, Depth ≤ 2000

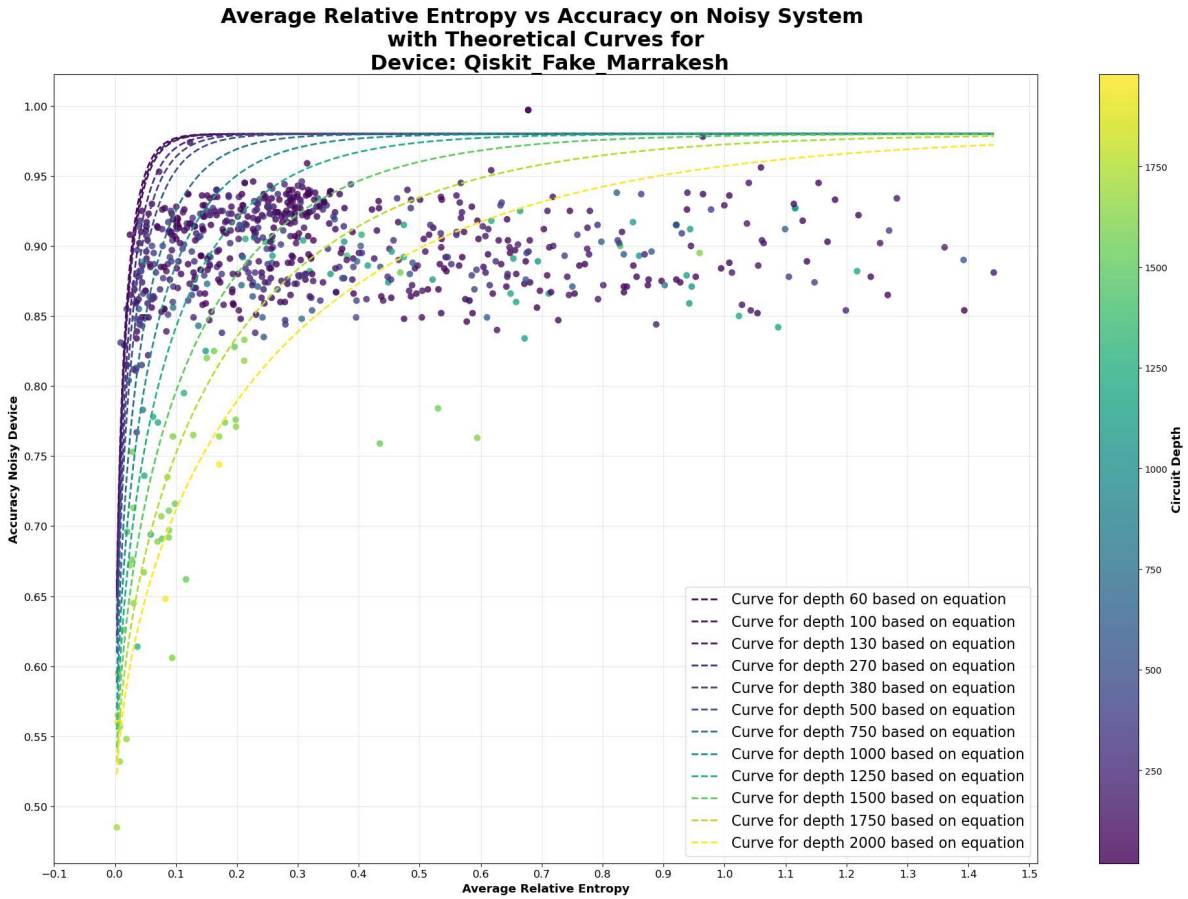

Fig. C3: Average quantum relative entropy vs Accuracy (Ideal-Noisy) with Theoretical Curves for Device: Qiskit Fake Marrakesh, Depth ≤ 2000

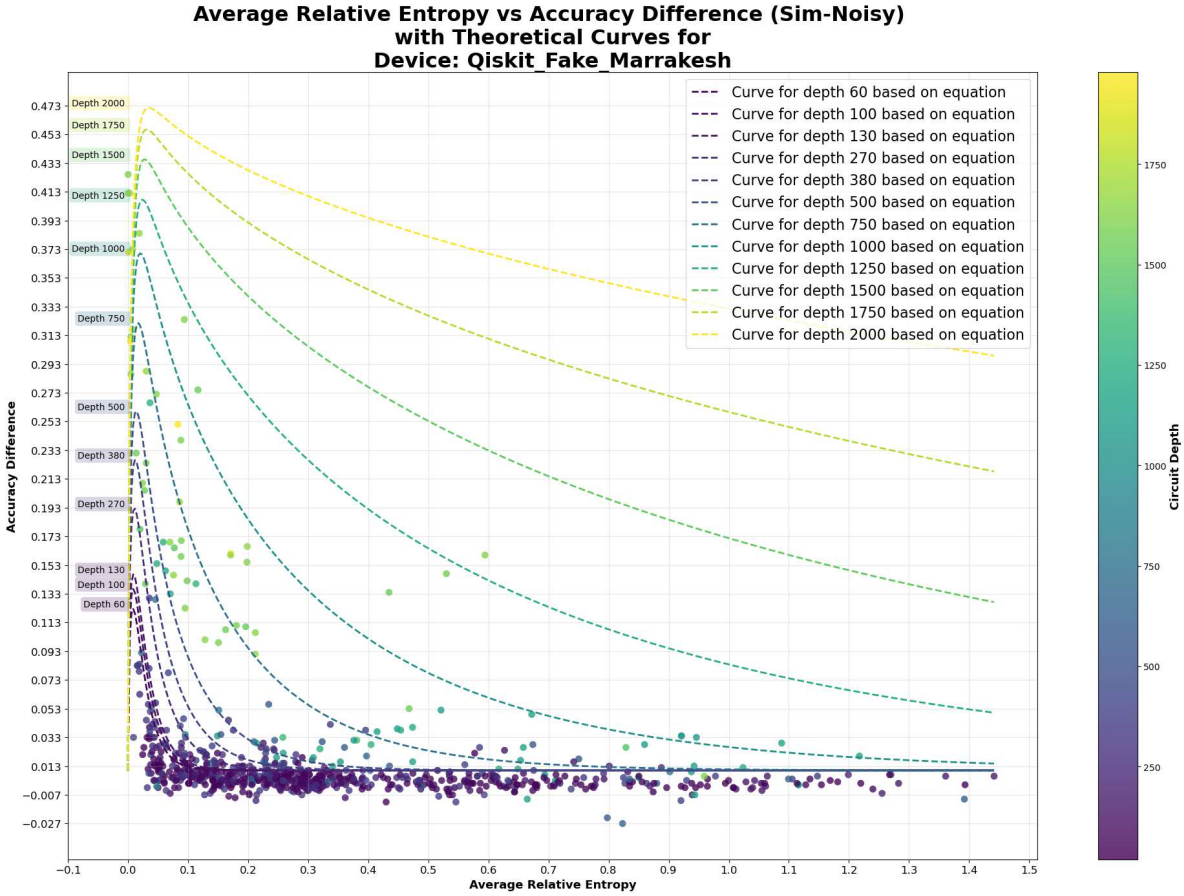

Fig. C4: Average quantum relative entropy vs Accuracy Difference (Ideal-Noisy) with Theoretical Curves for Device: Qiskit Fake Marrakesh, Depth ≤ 2000

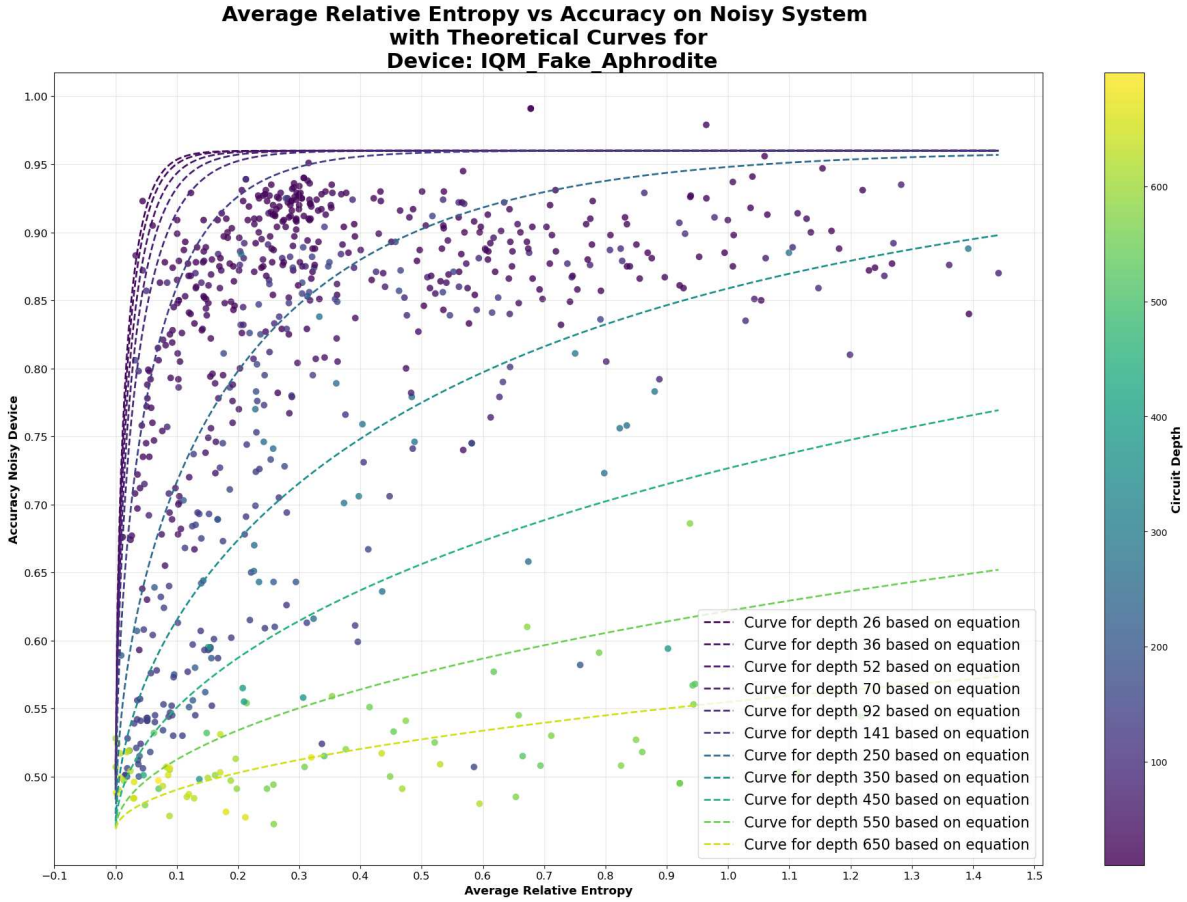

Fig. C5: Average quantum relative entropy vs Accuracy (Ideal-Noisy) with Theoretical Curves for Device: IQM Fake Aphrodite, Depth ≤ 2000

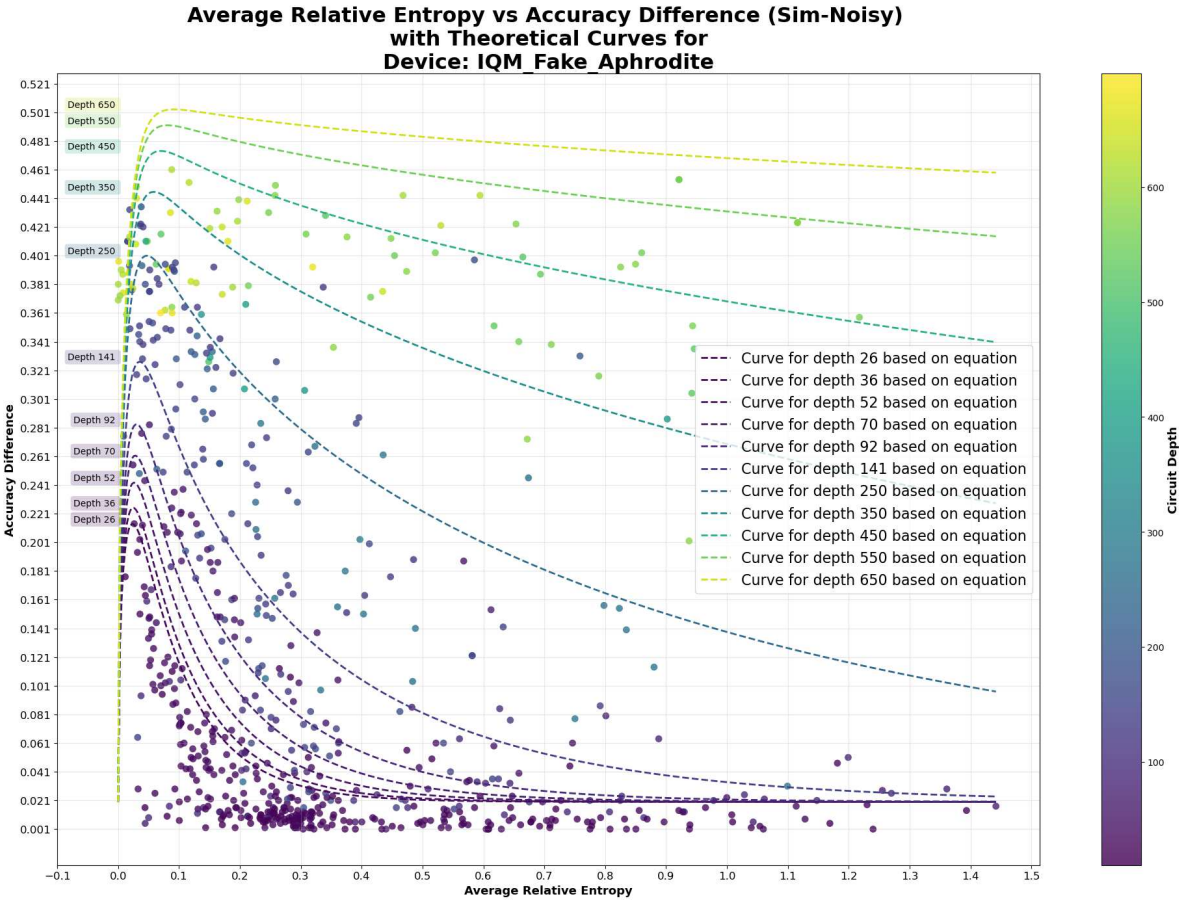

Fig. C6: Average quantum relative entropy vs Accuracy Difference (Ideal-Noisy) with Theoretical Curves for Device: IQM Fake Aphrodite, Depth ≤ 2000

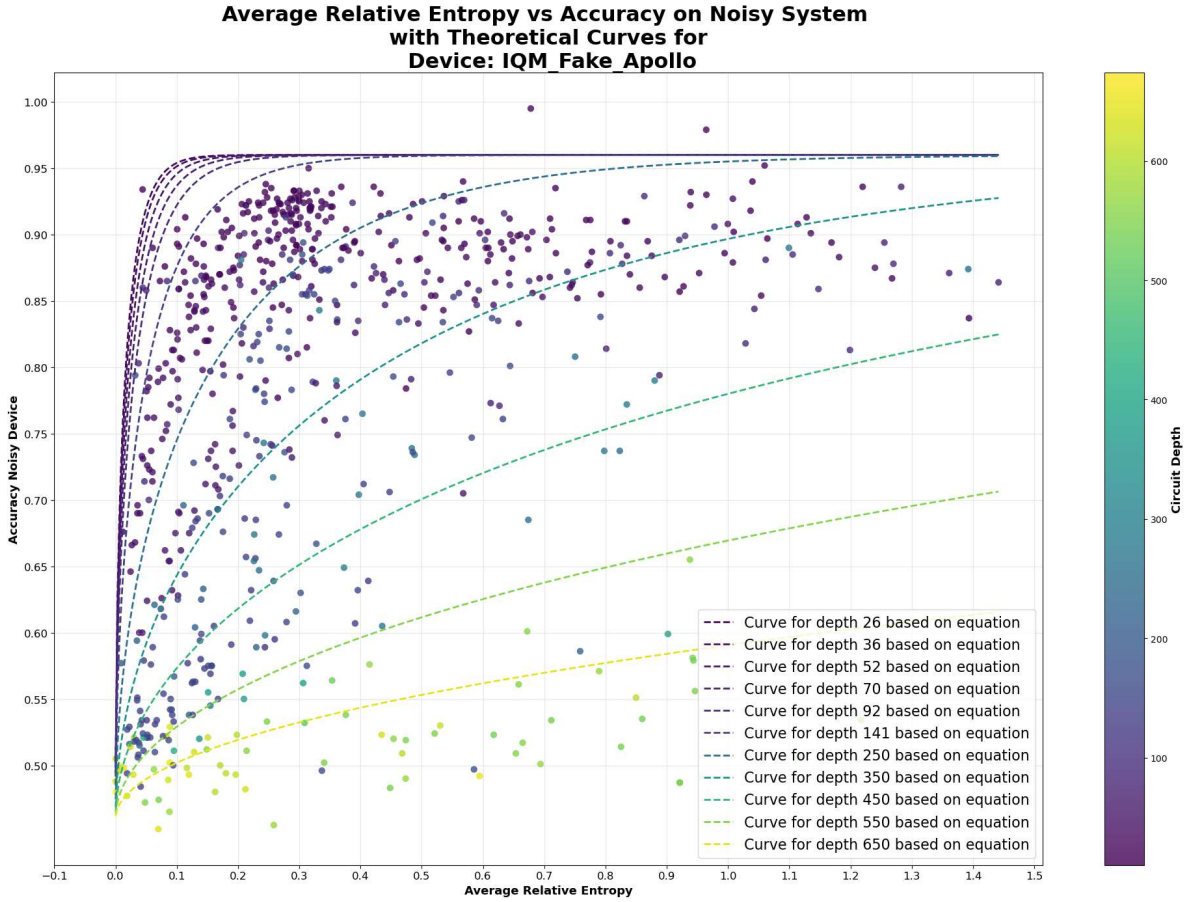

Fig. C7: Average quantum relative entropy vs Accuracy (Ideal-Noisy) with Theoretical Curves for Device: IQM Fake Apollo, Depth ≤ 2000

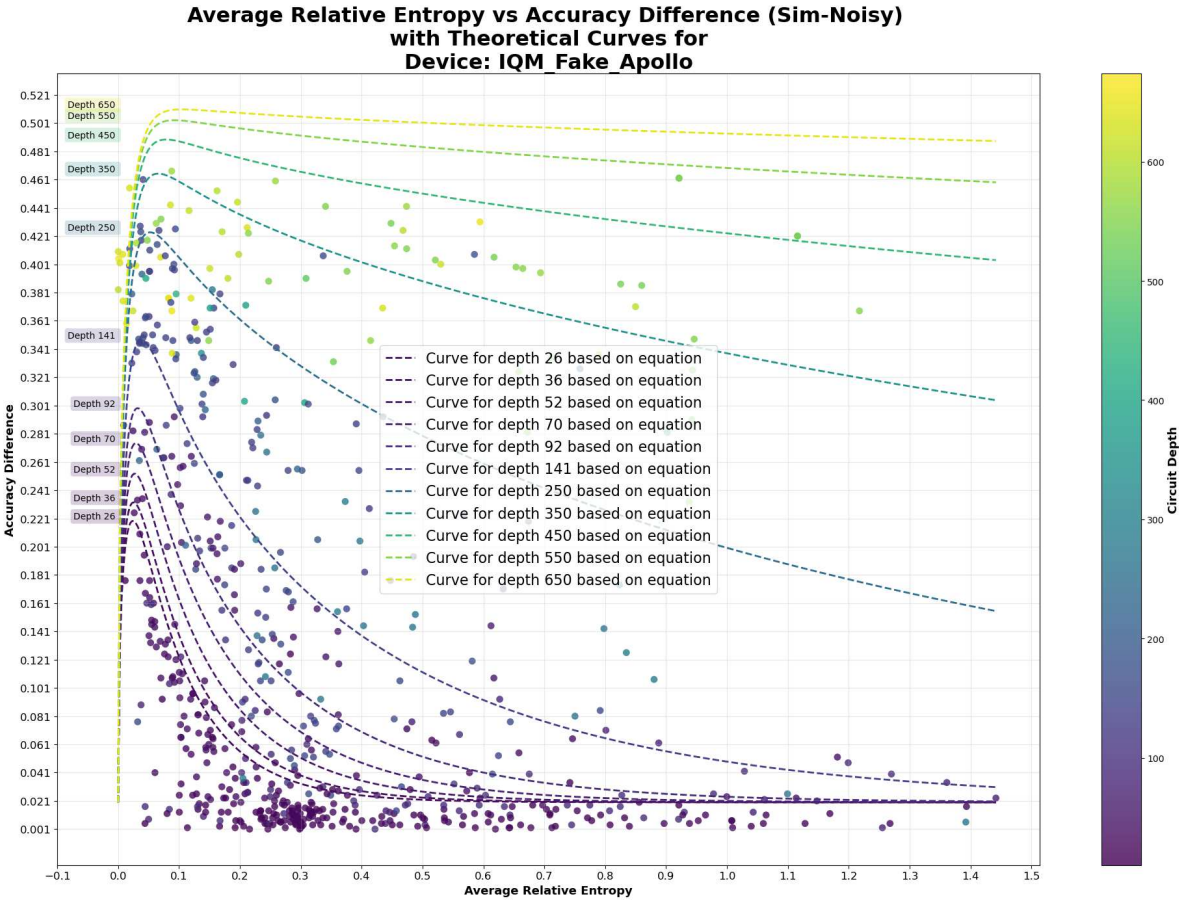

Fig. C8: Average quantum relative entropy vs Accuracy Difference (Ideal-Noisy) with Theoretical Curves for Device: IQM Fake Apollo, Depth ≤ 2000

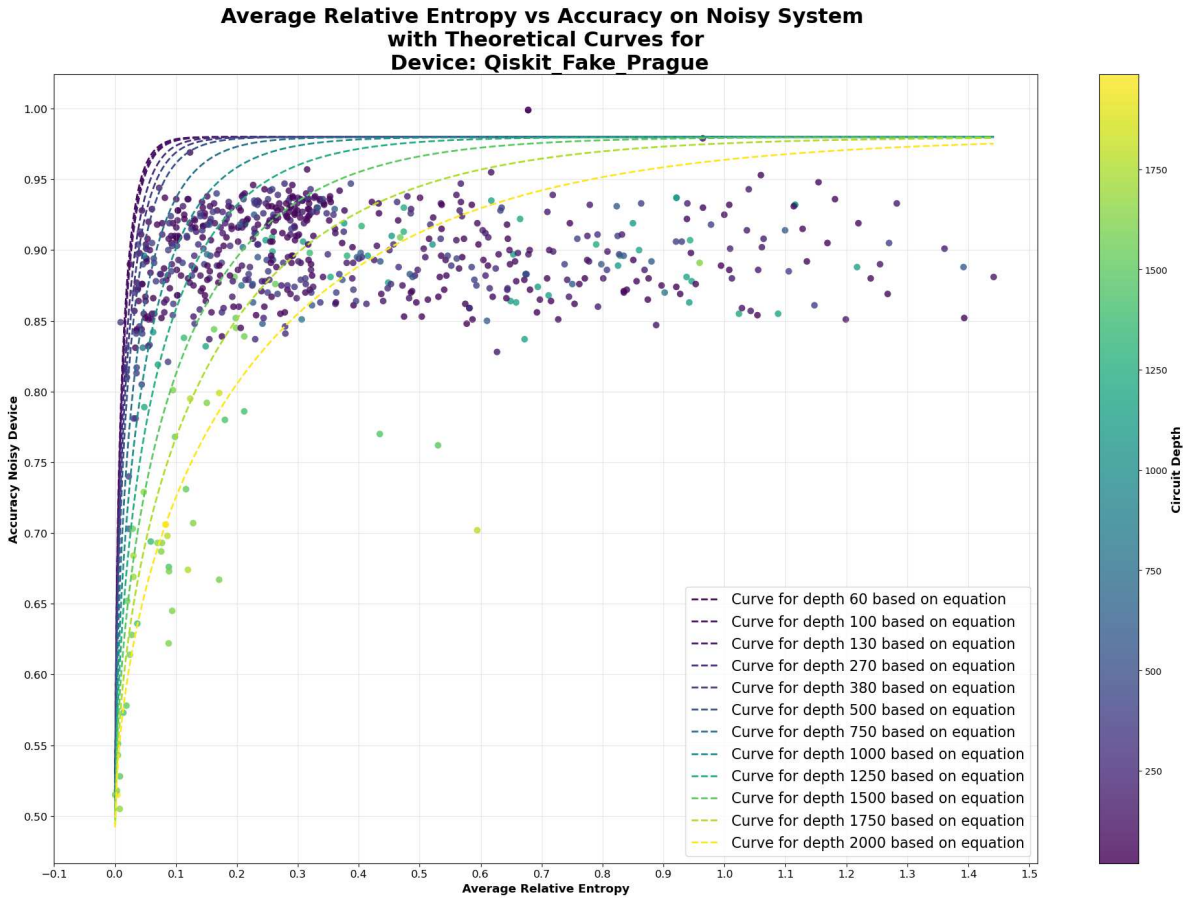

Fig. C9: Average quantum relative entropy vs Accuracy (Ideal-Noisy) with Theoretical Curves for Device: Qiskit Fake Prague, Depth ≤ 2000

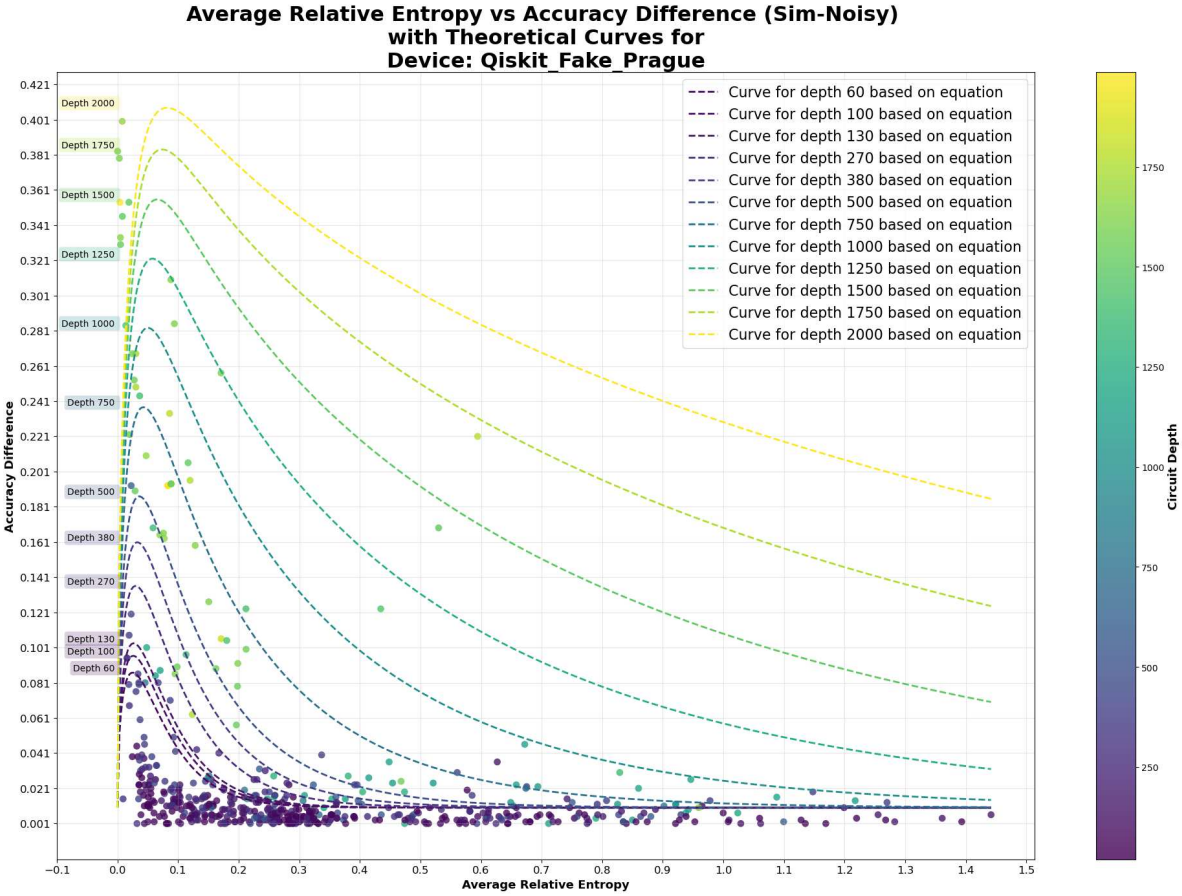

Fig. C10: Average quantum relative entropy vs Accuracy Difference (Ideal-Noisy) with Theoretical Curves for Device: Qiskit Fake Prague, Depth ≤ 2000