Amplitude-based Input Attribution in Quantum Learning via Integrated Gradients

Nicholas S. DiBrita
Rice University

Jason Han
Rice University

Younghyun Cho
Santa Clara University

Hengrui Luo
Rice University

Tirthak Patel
Rice University

Abstract

Quantum machine learning (QML) algorithms have demonstrated early promise across hardware platforms, but remain difficult to interpret due to the inherent opacity of quantum state evolution. Widely used classical interpretability methods, such as integrated gradients and surrogate-based sensitivity analysis, are not directly compatible with quantum circuits due to measurement collapse and the exponential complexity of simulating state evolution. In this work, we introduce HATTRIQ, a general-purpose framework to compute amplitude-based input attribution scores in circuit-based QML models. HATTRIQ supports the widely-used input amplitude embedding feature encoding scheme and uses a Hadamard test–based construction to compute input gradients directly on quantum hardware to generate provably faithful attributions. We validate HATTRIQ on classification tasks across several datasets (Bars and Stripes, MNIST, and FashionMNIST).

1 Introduction

Quantum machine learning (QML) uses quantum computing to enhance data analysis and pattern recognition in AI. By using quantum features like superposition and entanglement, QML algorithms have the potential to offer speedups over classical methods (DeRieux & Saad, 2025; De La Vega et al., 2023). Current research emphasizes hybrid models, where quantum circuits work alongside classical optimizers (Bharti et al., 2022; Arrasmith et al., 2021), with applications in classification, clustering, and generative tasks (Preskill, 2018; DiBrita et al., 2024; Zhang et al., 2023; Han et al., 2025). While limited by today's hardware, QML holds promise for solving complex problems in fields such as healthcare, finance, and scientific computing as quantum systems advance (Nicoli et al., 2023; Hothem et al., 2024; Preskill, 2023). Despite growing interest and experimental progress, QML models remain difficult to interpret due to the inherent opacity of quantum state evolution and the absence of intermediate observability mid computation (Herbst et al., 2025; Pira & Ferrie, 2024; Heese et al., 2025).

In classical machine learning, interpretability methods such as feature attribution play a critical role in understanding model predictions, particularly in sensitive and mission-critical domains like healthcare and autonomous systems (Radenovic et al., 2022; Zimmermann et al., 2023; Agarwal et al., 2021; Hooker et al., 2019; Alvarez Melis & Jaakkola, 2018). Attribution methods (Rudin, 2019; Krishna et al., 2022) – such as integrated gradients (Sundararajan et al., 2017) – assign importance scores to input features, revealing which aspects of the input most influence the model's output. These methods enhance transparency, support debugging, and build trust in model behavior. In contrast, existing QML pipelines provide little insight into how input features affect final measurement outcomes, especially when data is encoded and compressed into high-dimensional quantum state amplitudes (Jerbi et al., 2021; Bausch, 2020; Preskill, 2018; 2023).

In this work, we propose HATTRIQ[1], a methodology for computing the input attribution scores for quantum circuits. HATTRIQ adapts integrated gradients (Sundararajan et al., 2017) to the quantum circuit setting, enabling attribution for amplitude embedding. Leveraging integrated gradients for

[1] HATTRIQ stands for Hadamard test-based input attribution score scheme for quantum models.

quantum models is challenging, as larger models require working in exponentially large Hilbert spaces and manipulating complex amplitude vectors, making both analysis and simulation resource-intensive (Xiong et al., 2024; Lei et al., 2024). Another challenge is that quantum states are hidden from the user during computation. For large programs, we cannot simply record or log the hidden state after each circuit layer, as any attempt to measure the hidden state collapses the quantum state of the circuit entirely (Gong & Aaronson, 2023; Abbas et al., 2023); traditional (surrogate-based) sensitivity and Sobol/Shapley score methods (Owen, 2014; Cho et al., 2025) cannot preserve unitarity in quantum circuits, making it difficult to understand how different signals are propagated through the computation circuit.

To address this, HATTRIQ implements a Hadamard test–based construction that computes exact gradients directly on quantum hardware, without requiring access to internal quantum states. For fault-tolerant quantum devices, where the impact of hardware noise (Akhalwaya et al., 2024; Wu et al., 2025; Patel et al., 2024) is negligible, we propose a parallelization mechanism to evaluate multiple gradient components concurrently.

Our contributions are as follows.

- We introduce a formalism for computing integrated gradients in QML models that utilize amplitude embedding for input encoding; the formalism also works with other encodings such as angle embedding.
- We present a quantum-native circuit construction based on the Hadamard test to compute exact feature gradients for amplitude-embedded input attribution.
- We provide a multi-ancilla-based parallelization technique to enable gradient computation concurrently on larger quantum devices with sufficient capacity.
- We evaluate HATTRIQ on multiple classification tasks across Bars and Stripes (Bowles et al., 2024), MNIST (LeCun, 1998), and FashionMNIST (Xiao et al., 2017) datasets, demonstrating high-fidelity attribution.
- The code and dataset of HATTRIQ are open-sourced at: https://github.com/positivetechnologylab/HattriQ.

2 RELEVANT CONCEPTS

2.1 QUANTUM STATES AND GATES

Quantum computations are performed by quantum circuits, which manipulate qubits with logic gates. The *state* of a qubit is represented as a vector: $|\psi\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle$, where β_i is a complex coefficient for basis state $|i\rangle$. The probability of measuring the qubit to be in state $|i\rangle$ is $|\beta_i|^2$, which means we must have $|\beta_0|^2 + |\beta_1|^2 = 1$ (Schuld & Killoran, 2019; Silver et al., 2023). For an n qubit system, the statevector is a complex vector $|\psi\rangle \in \mathbb{C}^{2^n}$ that is normalized $\langle\psi|\psi\rangle = 1$. States are then written in terms of an orthonormal basis set; the conventional choice is referred to as the computational basis set. If we define b_k as the bitstring corresponding to integer k, we can define the computational basis as the set $\{|b_k\rangle \; \forall\, k \in \mathbb{Z}, 0 \leq k \leq 2^n - 1\}$. Our state can then be expressed as $|\psi\rangle = \sum_{k=0}^{2^n-1} \beta_k |b_k\rangle$ (Schuld & Killoran, 2019; Silver et al., 2023). Logic gates are represented by unitary matrices (U) acting on states: $U|\psi_1\rangle = |\psi_2\rangle$. Circuits are constructed by composing sequences of gates together (White et al., 2001; Srinivasan et al., 2018).

2.2 PARAMETERIZED QUANTUM CIRCUITS

We study quantum models that feature circuits with trainable gate parameters. These parameterized quantum circuits (PQCs) are also referred to as variational quantum circuits and have found extensive applications in quantum machine learning, quantum chemistry, and other areas of quantum optimization (Bharti et al., 2022; Arrasmith et al., 2021). Often, the trainable gates in PQCs are rotation gates, which rotate the quantum state according to some angle parameter. There are many possible ways to arrange a PQC; the fixed structure of a PQC is referred to as an ansatz, and is analogous to fixing a neural network architecture.

Let $\mathbf{x} \in \mathbb{R}^D$ be a data point, and $V(\mathbf{x})\,|0\rangle = |x\rangle \in \mathbb{C}^{2^n}$ be a the quantum state that encodes it, with $V(\mathbf{x})$ being the circuit that performs the encoding. Let $U(\boldsymbol{\theta})$ be a PQC with trainable parameters $\boldsymbol{\theta}$ (Schleich et al., 2024), and O be a Hermitian operator that represents the observable measured for the model output. We consider quantum models which apply some circuit operations to the input state $|x\rangle$ and then compute an expectation value, written as

$$F(\mathbf{x}\,;\boldsymbol{\theta}) = \langle x|\, U^\dagger(\boldsymbol{\theta})\, O\, U(\boldsymbol{\theta})\, |x\rangle\,. \tag{1}$$

In the more general case, we might compose $F(\mathbf{x}\,;\boldsymbol{\theta})$ with some other (likely nonlinear) function to add complexity to our model: our discussion generalizes simply by applying the chain rule (Arrasmith et al., 2021) in the gradient computation as introduced next.

Remark 2.1. For our discussion, we do not place any specific requirements on U, except that it must be a valid unitary operator. In most applications, however, U will have some fixed structure of gates (ansatz). Some subset of these gates will depend on variational parameters $\boldsymbol{\theta}$, which are then optimized to minimize the loss.

2.3 Integrated Gradients

We base our technique on the integrated gradients method proposed in (Sundararajan et al., 2017). This work studies the problem of attributing the prediction of deep learning networks to input features in a sample. Integrate gradients benefit from an axiomatic formulation, with guarantees about their sensitivity and implementation invariance (Sundararajan et al., 2017; Mudrakarta et al., 2018).

In addition to its superior theoretical properties, this method for attribution also only relies on a small number of model evaluations and gradient computations, without the need for additional knowledge of the hidden state (Sundararajan et al., 2017). This is highly desirable for the quantum setting, where measuring and storing the internal state at multiple points during the computation would incur a high overhead.

Definition 2.2 (Attribution Score)**.** *The integrated gradients attribution of a sample* $\mathbf{x}$ *relative to baseline* $\mathbf{x}'$ *is given as the following integral:*

$$IG_i(x) = (x_i - x_i') \int_0^1 \frac{\partial F(x' + \alpha \cdot (x - x'))}{\partial x_i} d\alpha. \tag{2}$$

The calculated value IG_i *is the integrated gradients attribution for the* i^{th} *feature, and it represents the contribution that it makes to the final model prediction.*

3 Feature Gradients

In this section, we introduce the most popular schemes for encoding data features into a quantum circuit calculation: (1) angle embedding and (2) amplitude embedding (Havlíček et al., 2019; Schuld & Petruccione, 2018; Lloyd et al., 2020; Iten et al., 2016; Schuld & Killoran, 2019). For each of these encoding methods, we introduce our methodology for computing the gradients with respect to those encoded features, attributing the circuit output to features.

3.1 Angle Embedding (or Encoding)

For angle-embedded data, the preparation circuit $V(\mathbf{x})$ consists of rotation gates, $\{R(x_i)\}$ each of which depends on an angle parameter. The angle parameters used are the features x_i. In such cases, the gradient with respect to the features can be natively calculated using the well-known parameter shift rule (Mitarai et al., 2018; Schuld et al., 2019), which allows for computing the gradient of quantum circuits by re-executing those circuits with shifted parameter values. For a quantum gate parametrized by θ_i and with only two distinct eigenvalues $\pm r$, it has been shown (Schuld et al., 2019):

$$\tfrac{\partial F}{\partial \theta_i} = r\left[F(\theta_i + s) - F(\theta_i - s)\right] \tag{3}$$

where $s = \frac{\pi}{4r}$ is the required shift. While at first glance this formula is reminiscent of a standard finite difference, it differs in that the shift s is not taken to be infinitesimal, and the result of this calculation is exact. This requires two additional circuit evaluations per parameter, making the gradient calculation linear with respect to the number of parameters. While Eq. 3 is not generally applicable to all gates, many parameterized gates, like single qubit rotations, do satisfy the eigenvalue requirements, and parameter shift has been utilized in a variety of quantum optimization settings (Schuld et al., 2020; Arrasmith et al., 2021). Additional rules have been formulated that generalize this result to additional kinds of parameterized gates (Wierichs et al., 2022).

While its simplicity makes angle embedding an attractive choice for near-term applications, the number of encoded features typically grows only linearly with the number of qubits (Schuld et al., 2020), meaning the angle encoding does not make full use of the exponentially large Hilbert space, and does not reach the information upper bound on a sphere (Luo et al., 2024).

3.2 Amplitude Embedding (or Encoding)

In the amplitude embedding case (Khan et al., 2024), data features are encoded as amplitudes of the input state: $|x\rangle = \sum_i x_i |b_i\rangle$. Unlike this angle embedding case, this allows for encoding exponentially many input features relative to the number of qubits, expanding the information capacity in the circuit. While it is generally true that the preparation circuit $V(\mathbf{x})$ is unitary, utilizing the parameter shift rule for this purpose is not possible due to the complexity of the circuit's dependence on the input features. In particular, most state preparation circuits will have structures that change based on particular $|x\rangle$ (Buhrman et al., 2024), meaning any differentiation routine will necessarily depend on a complex and changing parameterization. Furthermore, there may be state preparation routines not satisfying the two-eigenvalue criteria mentioned above. In such cases, one would need to use the linear combination of unitaries approach (Schuld et al., 2019), which requires additional matrix decompositions and circuit evaluations. To address this challenge, we provide a novel circuit-based method of calculating the input gradients, which is independent of the routine used for $V(\mathbf{x})$.

Lemma 3.1 (Input Gradient)**.** *For the general case, assume the amplitudes of an amplitude-encoded input are complex valued, so that each $x_k = c_k + \mathbf{i}\, d_k$. Then, the input gradients with respect to the function given in Eq. 1 are given by the following for the real values and complex-valued components.*

$$\tfrac{\partial F}{\partial c_k} = 2\,\mathrm{Re}[\langle b_k|\, U^\dagger(\boldsymbol{\theta})\, O\, U(\boldsymbol{\theta})\, |x\rangle] \qquad \tfrac{\partial F}{\partial d_k} = 2\,\mathrm{Im}[\langle b_k|\, U^\dagger(\boldsymbol{\theta})\, O\, U(\boldsymbol{\theta})\, |x\rangle]$$

Proof. The result is elegant to prove upon judicious use of the product rule for derivatives. A full explicit calculation is deferred to Appendix A. □

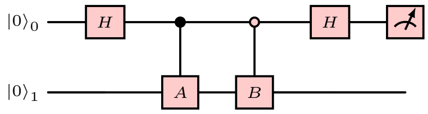
Lemma 3.1 gives a compact expression for the k^{th} component of the gradient in terms of the trained model circuit $U(\boldsymbol{\theta})$, its Hermitian conjugate $U^\dagger(\boldsymbol{\theta})$, Hermitian observable O, and amplitude embedded state $|x\rangle$. In this work, we are primarily concerned with the case where all amplitudes are real, $x_i = c_i$, as this is the most common case encountered when using classical data.

Remark 3.2. If we add the constraint that O be unitary as well as Hermitian, then $U^\dagger(\boldsymbol{\theta}) O U(\boldsymbol{\theta})$ corresponds to a valid quantum circuit. The obvious choices for O that satisfy this are Pauli operators or strings of Pauli operators (Dion et al., 2024), which are available on most devices as both measurement and gate operations.

4 Calculating on Quantum Hardware

4.1 Hadamard Test

Definition 4.1 (Hadamard Test)**.** *Given unitary operators A and B such that $A|0\rangle = |a\rangle$ and $B|0\rangle = |b\rangle$, the Hadamard test (Montanaro & de Wolf, 2013; Audenaert et al., 2008) is a method for encoding the value $\mathrm{Re}[\langle a|b\rangle]$ into the expectation value of a quantum circuit observable. This is achieved by the following circuit:*



The circuit for computing $\mathrm{Im}[\langle a|b\rangle]$ is the same, with the addition of an $S^\dagger$ gate after the first H gate (Aharonov et al., 2006).

After the initial Hadamard gate $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$, we have the state $\frac{1}{\sqrt{2}}[|0\rangle_0 + |1\rangle_0]\,|0\rangle_1$. Applying A conditioned on 1 and B conditioned on 0 gives the entangled state

$$\tfrac{1}{\sqrt{2}}[\,|0\rangle_0\, B\,|0\rangle + |1\rangle_0\, A\,|0\rangle\,] = \tfrac{1}{\sqrt{2}}[\,|0\rangle_0\,|b\rangle + |1\rangle_0\,|a\rangle\,].$$

The final Hadamard gate gives us

$$\tfrac{1}{\sqrt{2}}\,(\tfrac{1}{\sqrt{2}}(|0\rangle_0 + |1\rangle_1)\,|b\rangle_1 + \tfrac{1}{\sqrt{2}}(|0\rangle_0 - |1\rangle_1)\,|a\rangle_1 = \tfrac{1}{2}(|0\rangle\,(|b\rangle + |a\rangle) + |1\rangle\,(|b\rangle - |a\rangle)).$$

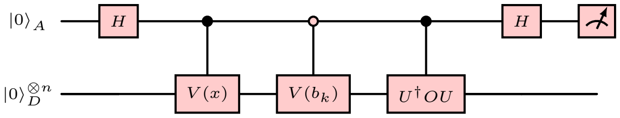
From this we can compute the probability of measuring qubit 0 to be 0 as

$$P(0) = \tfrac{1}{2}[\,\langle b| + \langle a|) \cdot \tfrac{1}{2}(|b\rangle + |a\rangle\,] = \tfrac{1}{4}[\,\langle b|b\rangle + \langle b|a\rangle + \langle a|b\rangle + \langle a|a\rangle\,] = \tfrac{1}{2}[\,1 + \mathrm{Re}[\langle a|b\rangle]\,].$$

This allows us to estimate the desired inner product by sampling from the probability distribution of additional qubits entangled with the system (Schuld et al., 2019). Hadamard tests have been used previously to compute certain kinds of parameter gradients (Bharti et al., 2022; Schuld et al., 2019), but not for feature gradients. We re-frame our formulation in Lemma 3.1 in order to allow for hardware native calculations of the gradients for input amplitudes.

4.2 Gradient Calculation for Input Attribution

Lemma 3.1, in conjunction with Definition 4.1, implies that we can calculate the feature gradient of a quantum model using circuit evaluations. We propose a circuit based on a Hadamard test:



Here, H is the aforementioned Hadamard gate, used to create equal superposition states. $V(x)$ is the preparation circuit that prepares $|x\rangle$. Similarly, $V(b_k)$ prepares the computational basis state $|b_k\rangle$. The wires that extend from one qubit register to another indicate control operations; these are multiqubit operations where the state of one or more qubits in a target register undergoes some transformation, predicated on the state of the control register. In the above circuit, the control is always the ancilla qubit (indexed by A), while the targets are always the data qubits (indexed by D). The target state that triggers the control operation is indicated by the circle: filled circles indicate control gates triggered by the $|1\rangle_A$ state, while empty circles indicate control gates triggered by $|0\rangle_A$. As an example, consider the first controlled gate, controlled $V(\mathbf{x})$. This gate prepares $|x\rangle$ on the data register D when the ancilla $A\ |1\rangle$, and does nothing when A is in the $|0\rangle$ state. Having generic controlled operations can incur additional circuit overhead; however, advances in global pulse operations and reconfigurable connectivity can mitigate this (Delakouras et al., 2025).

Theorem 4.2. *The above circuit returns the k^{th} element of the gradient provided in Lemma 3.1, encoded in the probability of the event where qubit A is measured as 0.*

Proof. The result can be seen almost directly from considering definition 4.1. An explicit calculation of the result is provided in Appendix B. The resulting measurement probability on the A register is

$$P(A = 0) = \tfrac{1}{2}(1 + \mathrm{Re}[\,\langle b_k|\,U^\dagger OU\,|x\rangle\,])$$

Comparing with Lemma 3.1, we see that the k^{th} entry of the gradient is contained within the probability of measuring the ancilla to be 0. We can repeat this procedure for each of the components. For a fixed number of measurement shots, this gives a linearly scaling relationship with the number of input features, i.e., one circuit required per input feature. □

4.3 Gradient Calculation Parallelization

We can further parallelize the component operations (the k's in Theorem 4.2) by increasing the number of ancilla qubits. For instance, if three components need to be executed in parallel, the following circuit is capable of calculating the k^{th}, l^{th}, and m^{th} components concurrently. The circuit uses two ancilla qubits instead of one and measures both the ancilla simultaneously.

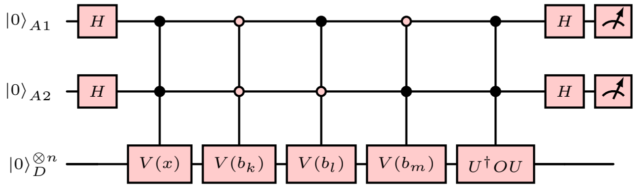


Table 1: Datasets and models used for HATTRIQ's evaluation, including the accuracies achieved.

Dataset	Binary Classes	Encoding	Circuit Structure		Accuracy (%)	
			# Qubits	# Layers	Training	Testing
Bars & Stripes	(Bars, Stripes)	Amplitude	4	8	96	95
		Angle	8	8	95	95
NIST	(0,1), (3,4), (5,6), (6,9), (1,7)	Amplitude	6	6	98, 100, 98, 96, 93	99, 100, 100, 98, 88
MNIST	(0,1), (3,4), (5,6), (6,9), (1,7)	Amplitude	10	10	92, 88, 87, 62, 87	91, 82, 87, 68, 83
Fashion MNIST	(Dress,Shirt), (Boot,Trousers), (Coat,Sandal), (Bag,Sandal), (Boot,Dress)	Amplitude	10	10	74, 100, 96, 74, 90	70, 99, 95, 69, 91

A similar calculation as provided in Theorem 4.2 gives an output probability of

$$P(A_1A_2 = 00) = \tfrac{1}{16}\left[4 + 2\,\mathrm{Re}[\langle b_k|\,\tilde{O}\,|x\rangle + \langle b_l|\,\tilde{O}\,|x\rangle + \langle b_m|\,\tilde{O}\,|x\rangle]\right].$$

Similar expressions exist for $P(A_1A_2 = 01)$, $P(A_1A_2 = 10)$, and $P(A_1A_2 = 11)$, allowing us to generate a simple linear system to determine the unknown gradient components, which can then be solved by a linear solver. Since this constructive proof reduces the problem of feature encoding into solving a linear system, increasing the number of ancilla qubits allows us to push this technique further, with $2^m - 1$ gradient components from m ancilla qubits. This is explored more thoroughly in Appendix C, where more in-depth calculations of the two ancilla case are also provided.

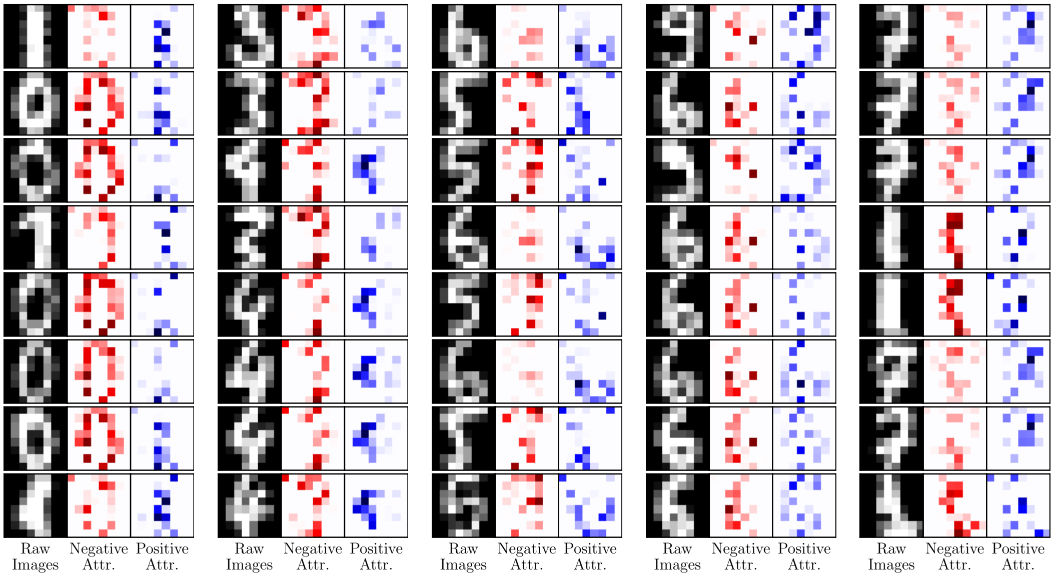
5 Evaluation and Discussion

5.1 Experimental Setup

We test our technique on a variety of image datasets, including Bars and Stripes (Bowles et al., 2024), MNIST (LeCun, 1998), NIST (similar to MNIST, but with reduced resolution: 8×8), and FashionMNIST (Xiao et al., 2017). For each dataset, we construct binary classification tasks by randomly selecting two classes. Training, inference, and gradient calculations are all performed in simulation, assuming ideal error-corrected hardware. As simulation is computationally prohibitive for larger systems, we focus on smaller, but representative, datasets to evaluate HATTRIQ. All simulation code is written in Python 3.12.1, using Qiskit 2.0.0 (Javadi-Abhari et al., 2024) and PennyLane 0.41.1 (Bergholm et al., 2022). Data preprocessing was performed with scikit-learn 1.6.1 (Pedregosa et al., 2011), and the optimization for training circuit parameters was performed using COBYLA (COB, 1994), as implemented in Scipy 1.15.1 (Virtanen et al., 2020). Due to the difficulty of training the angle-embedded model, we used a gradient descent optimizer implemented in PennyLane. Experiments are run on a local cluster, consisting of nodes with the AMD EPYC 7702P 64-core processor. We spawn virtual machines with 8 cores and 32 GB of memory.

5.2 Model Architecture

To focus on the general applicability of our technique, we choose to train relatively simple models (model properties are shown in Table 1) based on the hardware-efficient ansatz, which is composed of alternating layers of single-qubit rotation gates and two-qubit CNOT gates (Arrasmith et al., 2021). An example of this structure is shown in Appendix D for reference. Data is encoded into the system with an amplitude encoding scheme, where the intensity of a pixel corresponds to the amplitude of one of the basis states. For these datasets, it is not generally true that each data point is normalized with $|\mathbf{x}| = 1$, meaning we can not directly encode them as quantum states $|x\rangle = \sum_i x_i\,|b_i\rangle$, but must first apply some transformation. The easiest of these is to simply divide each data point by its



(a) 0 / 1 (b) 3 / 4 (c) 5 / 6 (d) 6 / 9 (e) 1 / 7

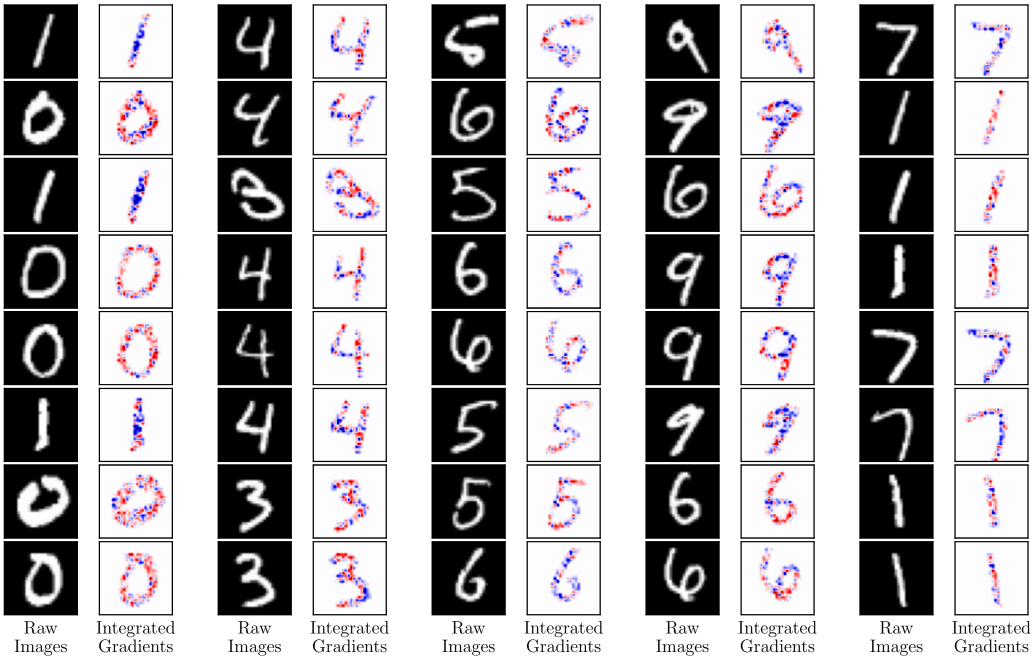
Figure 1: Sample images and the accompanying integrated gradients attribution for various samples from the NIST dataset. Quantum models were trained for various binary classification tasks. Blue indicates positive attribution, red indicates negative attribution, and white indicates neutral attribution. We see patches and patterns of strong attributions for the trained classifier models.

norm; however, we find empirically that this can cause issues during training, as the absolute value of a pixel's amplitude can change from image to image, even when the intensity is the same, due to images having differing levels of overall brightness. We instead utilize an encoding scheme that has an overflow state. This overflow state allows us to encode the value of each pixel in a way that is consistent image to image, while maintaining the normalization condition of quantum states. In an n qubit model, we scale $2^n - 1$ pixel values x_i to be within $[0, (\frac{1}{2^n-1})^{\frac{1}{2}}]$. The remaining state, the overflow state, is then assigned the value $(1 - \sum_i^{2^n-1} |x_i|^2)^{\frac{1}{2}}$ so that the final norm of the state is 1. Measurement is performed on a single qubit in the Z basis, i.e., O in Eq. 1 is the single qubit Z operator. While testing, we found improved performance when using a nonlinear tanh activation applied to the output of the circuit. All of our discussion from before still applies upon simple modification using the chain rule.

5.3 HattriQ's Attribution Results

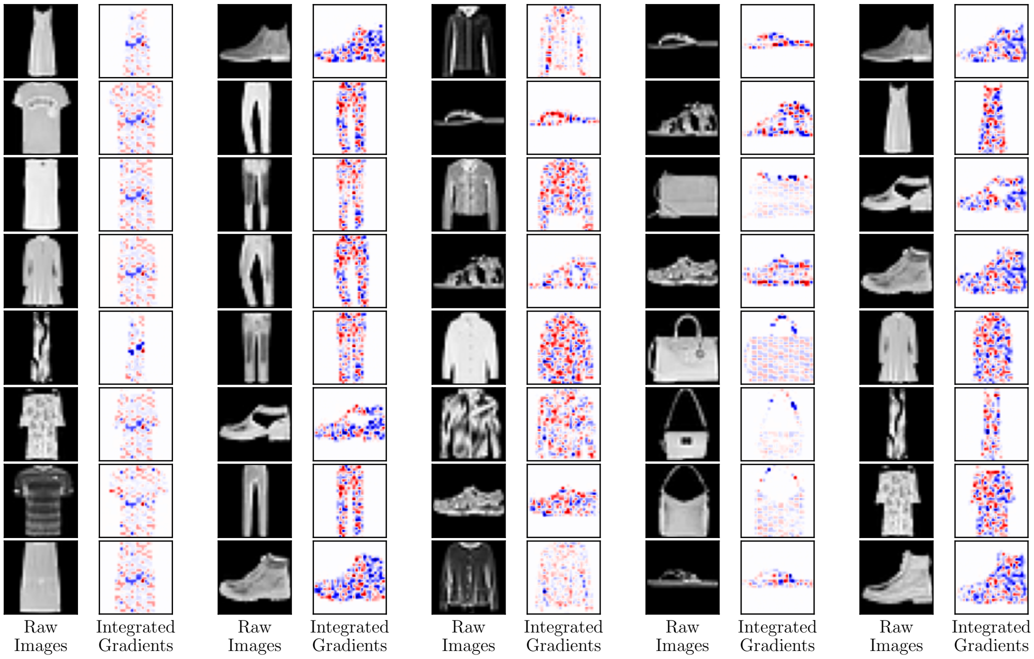
We show attribution scores for a variety of samples from each of the datasets. These samples are chosen randomly for analysis. We use a blank image (0 for all pixel values) for the baseline in all tests. In all plots, negative attributions are plotted in red, while positive attributions are plotted in blue. For visual clarity, attributions are normalized within each sample. Fig. 1 shows the integrated gradient outputs for a variety of samples from the NIST dataset. We see that background pixels have very little importance, as we might expect. We also see that the model has identified features that correspond to the target classes; an example being Fig. 1(b), where we see negative attributions corresponding to the circular shape of the digit 3 toward the upper right, and positive attributions corresponding to pixels near the center left, corresponding to the angled shape of the digit 4.

We see similar trends with MNIST (Fig. 2) and FashionMNIST (Fig. 3), with the model attributing regions of each image to each class. In these larger examples, the attributions appear more mixed spatially. This is especially noticeable in the Dress/Shirt task, which shows a banding pattern forming, in addition to clusters of strong attributions at the center. We see that in some cases, the model picks up distinctive features. One example is the Bag/Sandal task, where we see high attribution along the straps, which are only ever present on bags, but never on sandals. We also see this in the Coat/Sandal task, with the upper area getting consistent negative attributions; this area is unlikely to have any part of a sandal present, due to the low profile near that end of the shoe.



(a) 0 / 1 (b) 3 / 4 (c) 5 / 6 (d) 6 / 9 (e) 1 / 7

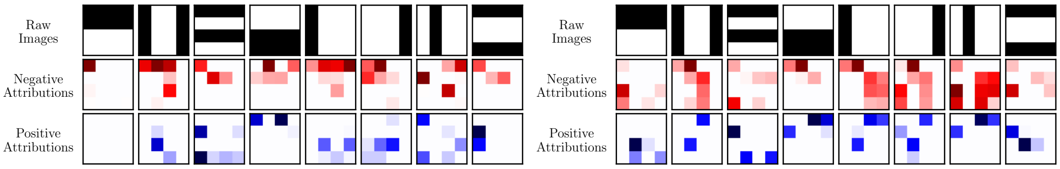
Figure 2: Sample images and gradient attribution for the amplitude-embedded MNIST dataset. We have merged the attributions to show positive and negative attributions in the same image.



(a) Dress / Shirt (b) Boot / Trousers (c) Coat / Sandal (d) Bag / Sandal (e) Boots / Dress

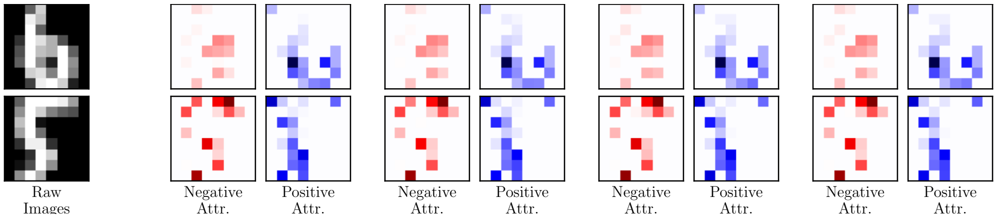
Figure 3: Sample images and attributions for the FashionMNIST dataset using amplitude encoding. We have merged the attributions to show positive and negative attributions in the same image.

We compare the attribution scores of a model using angle embedding and a model using amplitude embedding to see if there are differences in attributions created by different encoding schemes. Despite achieving very similar final accuracy scores (Table 1), we see markedly different attributions for the Bars and Stripes dataset in Fig. 4.



(a) Angle Encoding (b) Amplitude Encoding

Figure 4: Sample images and the accompanying integrated gradients attribution for the Bars and Stripes dataset. Quantum models using (a) angle encoding and (b) amplitude encoding were trained.



(a) Raw Images (b) 10 shots (c) 100 shots (d) 500 shots (e) Exact Sim.

Figure 5: Integrated gradients computed using various amounts of measurement shots (samples). In (b), (c), and (d), gradient components are computed using our circuit-based approach, using 10, 100, and 1000 samples to estimate each component. Overall, we see almost no degradation in the attribution scores, as compared to the results given by exact simulation (e).

5.4 The Effect of Shot Noise

To quantify the resource usage of HattriQ, we study the impact of using reduced measurement shots on the ancilla qubit. We compute attribution scores using 10, 100, and 500 shots to estimate each component $\mathrm{Re}[\langle b_k | U^\dagger O U | x \rangle]$. Just as before, we repeat this for each component k of the gradient, and use numerical integration to compute the IG attribution. We compare against exact simulation, which numerically computes all inner products directly from the state vector. The results of this are shown in Fig. 5. We see that even with an extremely low shot count, the attribution scores computed are largely faithful to the numerically exact ones, with only small deviations appearing in some of the weaker (lower absolute value) attributions.

To further validate HattriQ's attribution scores, we also compute attributions for a null model having randomly generated parameters. These results are provided in Appendix E.

6 Related Work

Several recent works have explored interpretability in QML, though none target input attribution directly, and none provide a general, hardware-compatible gradient-based solution as in HattriQ. Recent efforts in QML interpretability span model-agnostic techniques, gradient-based methods, and visualization tools. Pira et al. (Pira & Ferrie, 2024) and Jahin et al. (Jahin et al., 2023) apply classical attribution methods like LIME and SHAP to QML models, while Heese et al. (Heese et al., 2025) use Shapley values to explain circuit components. These approaches rely on perturbation-based estimates and rely on surrogate-based analysis, hence are not designed for execution on quantum hardware.

Gradient-based methods, such as QGrad-CAM (Lin et al., 2024), demonstrate attribution in hybrid models using class activation maps, but are limited to specific architectures and do not generalize to amplitude encoding schemes. Visualization-driven efforts like QuantumEyes (Ruan et al., 2023) and interpretable model designs (Flamini et al., 2024; Duneau et al., 2024; Flam-Shepherd et al., 2022; Ran & Su, 2023) focus on circuit behavior or latent representations rather than input-level attribution and are limited in hardware compatibility (e.g., photonics or trapped ions). *In contrast,* HattriQ *provides the first gradient-based input attribution method for QML models.* It supports amplitude encoding and enables scalable attribution via Hadamard test circuits and parallel gradient evaluation, making it broadly applicable across quantum models and devices.

7 Conclusion

We presented HATTRIQ, a unified framework for gradient-based feature attribution in quantum machine learning models. As the first-of-its-kind quantum interpretability method, HATTRIQ operates on exponentially scaling amplitude encoding schemes and is designed for execution on quantum hardware, offering circuit-based gradient computations. We plan to extend HATTRIQ to generate parameter/layer attributions for QML models to determine their importance on the QML task with potential disagreements between attributions from multiple runs (Krishna et al., 2022). We also plan to extend HATTRIQ to support QML models with mid-circuit measurements and conditional gate operations, which are starting to become available on quantum hardware. Due to the effectiveness and unitary nature of QML, it is also of interest to explore unitary feature learning, or equivalently, learning from spherical features (Luo et al., 2024). By leveraging a Hadamard test–based construction and a multi-ancilla parallelization strategy, HATTRIQ enables scalable, implementation-agnostic input attribution with fidelity guarantees.

8 Acknowledgement

This work was supported by Rice University, Santa Clara University, the Rice University George R. Brown School of Engineering and Computing, and the Rice University Department of Computer Science. This work was supported by the DOE Quantum Testbed Finder Award DE-SC0024301, the Ken Kennedy Institute, and Rice Quantum Initiative, which is part of the Smalley-Curl Institute. Hengrui Luo was supported by the U.S. Department of Energy under Contract DE-AC02-05CH11231 and the U.S. National Science Foundation NSF-DMS 2412403.

References

A direct search optimization method that models the objective and constraint functions by linear interpolation. Springer, 1994.

Amira Abbas, Robbie King, Hsin-Yuan Huang, William J Huggins, Ramis Movassagh, Dar Gilboa, and Jarrod McClean. On quantum backpropagation, information reuse, and cheating measurement collapse. *Advances in Neural Information Processing Systems*, 36:44792–44819, 2023.

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

Dorit Aharonov, Vaughan Jones, and Zeph Landau. A polynomial quantum algorithm for approximating the jones polynomial. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pp. 427–436, 2006.

Ismail Yunus Akhalwaya, Shashanka Ubaru, Kenneth L Clarkson, Mark S Squillante, Vishnu Jejjala, Yang-Hui He, Kugendran Naidoo, Vasileios Kalantzis, and Lior Horesh. Topological data analysis on noisy quantum computers. In *The Twelfth International Conference on Learning Representations*, 2024.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

Koenraad MR Audenaert, Michael Nussbaum, Arleta Szkoła, and Frank Verstraete. Asymptotic error rates in quantum hypothesis testing. *Communications in Mathematical Physics*, 279:251–283, 2008.

Johannes Bausch. Recurrent quantum neural networks. *Advances in neural information processing systems*, 33:1368–1379, 2020.

Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M. Sohaib Alam, Guillermo Alonso-Linaje, B. AkashNarayanan, Ali Asadi, Juan Miguel Arrazola, Utkarsh Azad, Sam Banning, Carsten Blank, Thomas R Bromley, Benjamin A. Cordier, Jack Ceroni, Alain Delgado, Olivia Di Matteo, Amintor Dusko, Tanya Garg, Diego Guala, Anthony Hayes, Ryan Hill, Aroosa Ijaz, Theodor Isacsson, David Ittah, Soran Jahangiri, Prateek Jain, Edward Jiang, Ankit Khandelwal, Korbinian Kottmann, Robert A. Lang, Christina Lee, Thomas Loke, Angus Lowe, Keri McKiernan, Johannes Jakob Meyer, J. A. Montañez-Barrera, Romain Moyard, Zeyue Niu, Lee James O'Riordan, Steven Oud, Ashish Panigrahi, Chae-Yeun Park, Daniel Polatajko, Nicolás Quesada, Chase Roberts, Nahum Sá, Isidor Schoch, Borun Shi, Shuli Shu, Sukin Sim, Arshpreet Singh, Ingrid Strandberg, Jay Soni, Antal Száva, Slimane Thabet, Rodrigo A. Vargas-Hernández, Trevor Vincent, Nicola Vitucci, Maurice Weber, David Wierichs, Roeland Wiersema, Moritz Willmann, Vincent Wong, Shaoming Zhang, and Nathan Killoran. Pennylane: Automatic differentiation of hybrid quantum-classical computations, 2022. URL https://arxiv.org/abs/1811.04968.

Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.*, 94:015004, Feb 2022. doi: 10.1103/RevModPhys.94.015004. URL https://link.aps.org/doi/10.1103/RevModPhys.94.015004.

Joseph Bowles, Shahnawaz Ahmed, and Maria Schuld. Better than classical? the subtle art of benchmarking quantum machine learning models. *arXiv preprint arXiv:2403.07059*, 2024.

Harry Buhrman, Marten Folkertsma, Bruno Loff, and Niels MP Neumann. State preparation by shallow circuits using feed forward. *Quantum*, 8:1552, 2024.

Younghyun Cho, James Demmel, Michał Dereziński, Haoyun Li, Hengrui Luo, Michael Mahoney, and Riley Murray. Surrogate-based autotuning for randomized sketching algorithms in regression problems. *SIAM Journal on Matrix Analysis and Applications*, 46(2):1247–1279, 2025.

Nimrod De La Vega, Noam Razin, Nadav Cohen, et al. What makes data suitable for a locally connected neural network? a necessary and sufficient condition based on quantum entanglement. *Advances in Neural Information Processing Systems*, 36:40994–41033, 2023.

Antonis Delakouras, Georgios Doultsinos, and David Petrosyan. Multi-qubit rydberg gates between distant atoms. *arXiv preprint arXiv:2507.16602*, 2025.

Alexander DeRieux and Walid Saad. eqmarl: Entangled quantum multi-agent reinforcement learning for distributed cooperation over quantum channels. 2025.

Nicholas S DiBrita, Daniel Leeds, Yuqian Huo, Jason Ludmir, and Tirthak Patel. Recon: Reconfiguring analog rydberg atom quantum computers for quantum generative adversarial networks. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–9, 2024.

Maxime Dion, Tania Belabbas, and Nolan Bastien. Efficiently manipulating pauli strings with pauliarray. *arXiv preprint arXiv:2405.19287*, 2024.

Tiffany Duneau, Saskia Bruhn, Gabriel Matos, Tuomas Laakkonen, Katerina Saiti, Anna Pearson, Konstantinos Meichanetzidis, and Bob Coecke. Scalable and interpretable quantum natural language processing: an implementation on trapped ions. *arXiv preprint arXiv:2409.08777*, 2024.

Daniel Flam-Shepherd, Tony C Wu, Xuemei Gu, Alba Cervera-Lierta, Mario Krenn, and Alan Aspuru-Guzik. Learning interpretable representations of entanglement in quantum optics experiments using deep generative models. *Nature Machine Intelligence*, 4(6):544–554, 2022.

Fulvio Flamini, Marius Krumm, Lukas J Fiderer, Thomas Müller, and Hans J Briegel. Towards interpretable quantum machine learning via single-photon quantum walks. *Quantum Science and Technology*, 9(4):045011, 2024.

Weiyuan Gong and Scott Aaronson. Learning distributions over quantum measurement outcomes. In *International Conference on Machine Learning*, pp. 11598–11613. PMLR, 2023.

Jason Han, Nicholas S DiBrita, Younghyun Cho, Hengrui Luo, and Tirthak Patel. Enqode: Fast amplitude embedding for quantum machine learning using classical data. *arXiv preprint arXiv:2503.14473*, 2025.

Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.

Raoul Heese, Thore Gerlach, Sascha Mücke, Sabine Müller, Matthias Jakobs, and Nico Piatkowski. Explaining quantum circuits with shapley values: Towards explainable quantum machine learning. *Quantum Machine Intelligence*, 7(1):1–33, 2025.

Sabrina Herbst, Sandeep Suresh Cranganore, Vincenzo De Maio, and Ivona Brandic. Exploring channel distinguishability in local neighborhoods of the model space in quantum neural networks. 2025.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Daniel Hothem, Ashe Miller, and Timothy Proctor. What is my quantum computer good for? quantum capability learning with physics-aware neural networks. *arXiv preprint arXiv:2406.05636*, 2024.

Raban Iten, Roger Colbeck, Ivan Kukuljan, Jonathan Home, and Matthias Christandl. Quantum circuits for isometries. *Physical Review A*, 93(3):032318, 2016.

Md Abrar Jahin, Md Sakib Hossain Shovon, Md Saiful Islam, Jungpil Shin, Muhammad Firoz Mridha, and Yuichi Okuyama. Qamplifynet: pushing the boundaries of supply chain backorder prediction using interpretable hybrid quantum-classical neural network. *Scientific Reports*, 13(1): 18246, 2023.

Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit, 2024.

Sofiene Jerbi, Casper Gyurik, Simon Marshall, Hans Briegel, and Vedran Dunjko. Parametrized quantum policies for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28362–28375, 2021.

Mansoor A Khan, Muhammad N Aman, and Biplab Sikdar. Beyond bits: A review of quantum embedding techniques for efficient information processing. *IEEE access*, 2024.

Satyapriya Krishna, Tessa Han, Alex Gu, Steven Wu, Shahin Jabbari, and Himabindu Lakkaraju. The disagreement problem in explainable machine learning: A practitioner's perspective. *arXiv preprint arXiv:2202.01602*, 2022.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Cong Lei, Yuxuan Du, Peng Mi, Jun Yu, and Tongliang Liu. Neural auto-designer for enhanced quantum kernels. In *The Twelfth International Conference on Learning Representations*, 2024.

Hsin-Yi Lin, Huan-Hsin Tseng, Samuel Yen-Chi Chen, and Shinjae Yoo. Quantum gradient class activation map for model interpretability. In *2024 IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 165–170. IEEE, 2024.

Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*, 2020.

Hengrui Luo, Jeremy E Purvis, and Didong Li. Spherical rotation dimension reduction with geometric loss functions. *Journal of Machine Learning Research*, 25(175):1–55, 2024.

K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Phys. Rev. A*, 98: 032309, Sep 2018. doi: 10.1103/PhysRevA.98.032309. URL https://link.aps.org/doi/10.1103/PhysRevA.98.032309.

Ashley Montanaro and Ronald de Wolf. A survey of quantum property testing. *arXiv preprint arXiv:1310.2035*, 2013.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. Did the model understand the question? In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1896–1906, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1176. URL https://aclanthology.org/P18-1176/.

Kim Nicoli, Christopher J Anders, Lena Funcke, Tobias Hartung, Karl Jansen, Stefan Kühn, Klaus-Robert Müller, Paolo Stornati, Pan Kessel, and Shinichi Nakajima. Physics-informed bayesian optimization of variational quantum circuits. *Advances in Neural Information Processing Systems*, 36:18341–18376, 2023.

Art B Owen. Sobol'indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2 (1):245–251, 2014.

Yash J Patel, Akash Kundu, Mateusz Ostaszewski, Xavier Bonet-Monroig, Vedran Dunjko, and Onur Danaci. Curriculum reinforcement learning for quantum architecture search under hardware errors. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Lirandë Pira and Chris Ferrie. On the interpretability of quantum neural networks. *Quantum Machine Intelligence*, 6(2):52, 2024.

John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

John Preskill. Quantum computing 40 years later. In *Feynman lectures on computation*, pp. 193–244. CRC Press, 2023.

Filip Radenovic, Abhimanyu Dubey, and Dhruv Mahajan. Neural basis models for interpretability. *Advances in Neural Information Processing Systems*, 35:8414–8426, 2022.

Shi-Ju Ran and Gang Su. Tensor networks for interpretable and efficient quantum-inspired machine learning. *Intelligent Computing*, 2:0061, 2023.

Shaolun Ruan, Qiang Guan, Paul Griffin, Ying Mao, and Yong Wang. Quantumeyes: Towards better interpretability of quantum circuits. *IEEE Transactions on Visualization and Computer Graphics*, 30(9):6321–6333, 2023.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1:206–215, 2019.

Philipp Schleich, Marta Skreta, Lasse B. Kristensen, Rodrigo A. Vargas-Hernández, and Alán Aspuru-Guzik. Quantum deep equilibrium models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 31940–31967. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/386432c7534eec9a1cd7cbeea90d7e9f-Paper-Conference.pdf.

Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.

Maria Schuld and Francesco Petruccione. Supervised learning with quantum computers. *Quantum science and technology (Springer, 2018)*, 2018.

Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A*, 99:032331, Mar 2019. doi: 10.1103/PhysRevA.99.032331. URL https://link.aps.org/doi/10.1103/PhysRevA.99.032331.

Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Phys. Rev. A*, 101:032308, Mar 2020. doi: 10.1103/PhysRevA.101.032308. URL https://link.aps.org/doi/10.1103/PhysRevA.101.032308.

Daniel Silver, Tirthak Patel, William Cutler, Aditya Ranjan, Harshitta Gandhi, and Devesh Tiwari. Mosaiq: Quantum generative adversarial networks for image generation on nisq computers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7030–7039, 2023.

Siddarth Srinivasan, Carlton Downey, and Byron Boots. Learning and inference in hilbert space with quantum graphical models. *Advances in Neural Information Processing Systems*, 31, 2018.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3319–3328. JMLR.org, 2017.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Andrew G White, DFV James, William J Munro, and PG Kwiat. Exploring hilbert space: Accurate characterization of quantum information. *Physical Review A*, 65(1):012301, 2001.

David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. *Quantum*, 6:677, March 2022. doi: 10.22331/q-2022-03-30-677.

Yusen Wu, Bujiao Wu, Yanqi Song, Xiao Yuan, and Jingbo Wang. Learning the complexity of weakly noisy quantum states. In *The Thirteenth International Conference on Learning Representations*, 2025.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Hao Xiong, Yehui Tang, Yunlin He, Wei Tan, and Junchi Yan. Node2ket: Efficient high-dimensional network embedding in quantum hilbert space. In *The Twelfth International Conference on Learning Representations*, 2024.

Hao-kai Zhang, Chenghong Zhu, Mingrui Jing, and Xin Wang. Statistical analysis of quantum state learning process in quantum neural networks. *Advances in Neural Information Processing Systems*, 36:33133–33160, 2023.

Roland S Zimmermann, Thomas Klein, and Wieland Brendel. Scale alone does not improve mechanistic interpretability in vision models. *Advances in Neural Information Processing Systems*, 36:57876–57907, 2023.

A Proof of Lemma 3.1: Input Gradients of Quantum Models

For compactness, define $\tilde{O} = U^\dagger(\boldsymbol{\theta})\, O\, U(\boldsymbol{\theta})$ Then, after rewriting Eq. 1, we have:

$$
\begin{aligned}
F(\mathbf{x}\,;\boldsymbol{\theta}) &= \Big(\sum_{i=0}^{2^n-1} \langle b_i|\, x_i^* \Big)\tilde{O}\Big(\sum_{j=0}^{2^n-1} x_j\, |b_j\rangle\Big) = \Big(\sum_{i=0}^{2^n-1} \langle b_i|\, (c_i - \mathbf{i}\, d_i)\Big)\tilde{O}\Big(\sum_{j=0}^{2^n-1} (c_j + \mathbf{i}\, d_j)\, |b_j\rangle\Big) \\
&= \textstyle\sum_{i,j} (c_i - \mathbf{i}\, d_i)(c_j + \mathbf{i}\, d_j)\, \langle b_i|\, \tilde{O}\, |b_j\rangle
\end{aligned}
$$

Taking the derivative with respect to c_k:

$$
\begin{aligned}
\tfrac{\partial F}{\partial c_k} &= \textstyle\sum_{ij} \frac{\partial c_i}{\partial c_k}\, (c_j + \mathbf{i}\, d_j)\, \langle b_i|\, \tilde{O}\, |b_j\rangle + (c_i - \mathbf{i}\, d_i) \frac{\partial c_j}{\partial c_k}\, \langle b_i|\, \tilde{O}\, |b_j\rangle \\
&= \textstyle\sum_{ij} \delta_{ik}\, (c_j + \mathbf{i}\, d_j)\, \langle b_i|\, \tilde{O}\, |b_j\rangle + \delta_{jk} (c_i - \mathbf{i}\, d_i)\, \langle b_i|\, \tilde{O}\, |b_j\rangle \\
&= \textstyle\sum_{j}\, (c_j + \mathbf{i}\, d_j)\, \langle b_k|\, \tilde{O}\, |b_j\rangle + \sum_{i} (c_i - \mathbf{i}\, d_i)\ \langle b_i|\, \tilde{O}\, |b_k\rangle \\
&= \textstyle\sum_{j}\, (c_j + \mathbf{i}\, d_j)\, \langle b_k|\, \tilde{O}\, |b_j\rangle + (c_j - \mathbf{i}\, d_j)\ \langle b_j|\, \tilde{O}\, |b_k\rangle \\
&= \textstyle\sum_{j} 2\ \mathrm{Re}[(c_j + \mathbf{i}\, d_j)\, \langle b_k|\, \tilde{O}\, |b_j\rangle] = 2\, \mathrm{Re}[\langle b_k|\, \tilde{O} \sum_{j}\, (c_j + \mathbf{i}\, d_j)\, |b_j\rangle] \\
&= 2\, \mathrm{Re}[\langle b_k|\, \tilde{O}\, |x\rangle] = 2\, \mathrm{Re}[\, \langle b_k|\, U^\dagger(\boldsymbol{\theta})\, O\, U(\boldsymbol{\theta})\, |x\rangle\,]
\end{aligned}
$$

Here, δ_{ik} is the Kronecker delta, and we have made use of the fact that $\tilde{O}^\dagger = \tilde{O}$. A similar derivation exists for $\frac{\partial F}{\partial d_k}$. We exclude it here for brevity.

B Proof of Theorem 4.2: Hadamard Test Computation

We use the subscript A for the state of ancilla qubit(s), and the subscript D for the state of data qubit(s). The final state of the circuit from section 4.2 before measurement is given by:

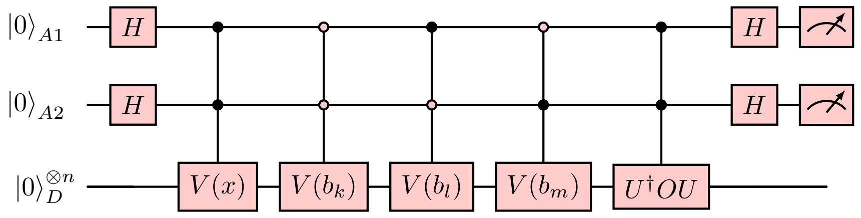
$$
\begin{aligned}
|\psi\rangle &= (I \otimes H) \cdot C\tilde{O} \cdot \bar{C}V(b_k) \cdot CV(x) \cdot (I \otimes H) \cdot (|0\rangle_D^{\otimes n} \otimes |0\rangle_A) \\
&= (I \otimes H) \cdot C\tilde{O} \cdot \bar{C}V(b_k) \cdot CV(x) \cdot \tfrac{1}{\sqrt{2}} \left(|0\rangle_D^{\otimes n} \otimes |0\rangle_A + |0\rangle_D^{\otimes n} \otimes |1\rangle_A\right) \\
&= (I \otimes H) \cdot \tfrac{1}{\sqrt{2}} \left(|b_k\rangle_D \otimes |0\rangle_A + \tilde{O}\, |x\rangle_D \otimes |1\rangle_A\right) \\
&= \tfrac{1}{2} \left(|b_k\rangle_D \otimes (|0\rangle_A + |1\rangle_A) + \tilde{O}\, |x\rangle_D \otimes (|0\rangle_A - |1\rangle_A)\right) \\
&= \tfrac{1}{2} \left((|b_k\rangle + \tilde{O}\, |x\rangle)_D \otimes |0\rangle_A + (|b_k\rangle - \tilde{O}\, |x\rangle)_D \otimes |1\rangle_A\right)
\end{aligned}
$$

Here, C denotes the control operations that trigger on $|1\rangle_A$ and $\bar{C}$ denotes the control operations that trigger on $|0\rangle_A$. From here, using the standard probability rule, we see that

$$
\begin{aligned}
P(A = 0) &= |\tfrac{1}{2}(|b_k\rangle + U^\dagger O U\, |x\rangle)|^2 = \tfrac{1}{4}|\, \langle b_k|b_k\rangle + \langle x|x\rangle + \langle b_k|\, U^\dagger O U\, |x\rangle + \langle x|\, U^\dagger O U\, |b_k\rangle\, | \\
&= \tfrac{1}{2}(1 + \mathrm{Re}[\, \langle b_k|\, U^\dagger O U\, |x\rangle\,])
\end{aligned}
$$

C Full Derivation of Gradient Parallelization

Parallel computation of the gradient entries is made possible by increasing the number of ancilla qubits. For the 2 ancilla case, the circuit looks like:



From this, we can compute the k^{th}, l^{th}, and m^{th} components of the gradient vector. The final state of the circuit is given by the following derivation.

$$
\begin{aligned}
|\psi\rangle &= (\mathbb{1}\otimes H\otimes H)\cdot C\tilde{O}\cdot CV(b_m)\cdot CV(b_l)\cdot CV(b_k)\cdot CV(\mathbf{x})\cdot(\mathbb{1}\otimes H\otimes H)\,|0\rangle_D^{\otimes n}\,|00\rangle_A\\
&= \frac{1}{2}(\mathbb{1}\otimes H\otimes H)\cdot C\tilde{O}\cdot CV(b_m)\cdot CV(b_l)\cdot CV(b_k)\cdot CV(\mathbf{x})\cdot|0\rangle_D^{\otimes n}\big[\,|00\rangle_A+|01\rangle_A+|10\rangle_A+|11\rangle_A\big]\\
&= \frac{1}{2}(\mathbb{1}\otimes H\otimes H)\cdot C\tilde{O}\big[\,|b_k\rangle\,|00\rangle_A+|b_l\rangle\,|01\rangle_A+|b_m\rangle\,|10\rangle_A+|x\rangle\,|11\rangle_A\big]\\
&= \frac{1}{2}(\mathbb{1}\otimes H\otimes H)\big[\,|b_k\rangle\,|00\rangle_A+|b_l\rangle\,|01\rangle_A+|b_m\rangle\,|10\rangle_A+\tilde{O}\,|x\rangle\,|11\rangle_A\big]\\
&= \frac{1}{2}\big[\,|b_k\rangle\,\frac{1}{2}\big(\,|0\rangle+|1\rangle\,\big)\big(\,|0\rangle+|1\rangle\,\big)+|b_l\rangle\,\frac{1}{2}\big(\,|0\rangle+|1\rangle\,\big)\big(\,|0\rangle-|1\rangle\,\big)\\
&\qquad+|b_m\rangle\,\frac{1}{2}\big(\,|0\rangle-|1\rangle\,\big)\big(\,|0\rangle+|1\rangle\,\big)+\tilde{O}\,|x\rangle\,\frac{1}{2}\big(\,|0\rangle-|1\rangle\,\big)\big(\,|0\rangle-|1\rangle\,\big)\big]\\
&= \frac{1}{4}\big[\,|b_k\rangle\,\big(\,|00\rangle_A+|01\rangle_A+|10\rangle_A+|11\rangle_A\,\big)+|b_l\rangle\,\big(\,|00\rangle_A-|01\rangle_A+|10\rangle_A-|11\rangle_A\,\big)\\
&\qquad+|b_m\rangle\,\big(\,|00\rangle_A+|01\rangle_A-|10\rangle_A-|11\rangle_A\,\big)+\tilde{O}\,|x\rangle\,\big(\,|00\rangle_A-|01\rangle_A-|10\rangle_A+|11\rangle_A\,\big)\big]\\
&= \frac{1}{4}\big[\big(\,|b_k\rangle+|b_l\rangle+|b_m\rangle+\tilde{O}\,|x\rangle\,\big)\,|00\rangle_A+\big(\,|b_k\rangle-|b_l\rangle+|b_m\rangle-\tilde{O}\,|x\rangle\,\big)\,|01\rangle_A\\
&\qquad+\big(\,|b_k\rangle+|b_l\rangle-|b_m\rangle-\tilde{O}\,|x\rangle\,\big)\,|10\rangle_A+\big(\,|b_k\rangle-|b_l\rangle-|b_m\rangle+\tilde{O}\,|x\rangle\,\big)\,|11\rangle_A\,\big]
\end{aligned}
$$

From here, we compute the probability of each ancilla bit string using the following derivation.

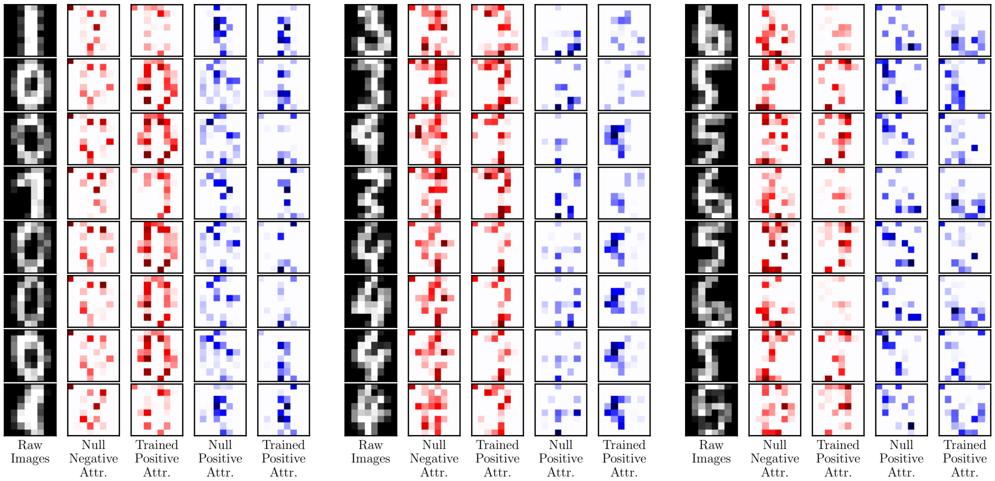
$$
\begin{aligned}
P(A2A1=00) &= \langle\psi|\;\;(\mathbb{1}\otimes|00\rangle_A)(\mathbb{1}\otimes\langle 00|_A)\;\;|\psi\rangle\\
&= \frac{1}{16}\big(\,\langle b_k|+\langle b_l|+\langle b_m|+\langle x|\,\tilde{O}^\dagger\big)\big(\,|b_k\rangle+|b_l\rangle+|b_m\rangle+\tilde{O}\,|x\rangle\,\big)\\
&= \frac{1}{16}\big(\,\langle b_k|b_k\rangle+\langle b_k|\,\tilde{O}\,|x\rangle+\langle b_l|b_l\rangle+\langle b_l|\,\tilde{O}\,|x\rangle+\langle b_m|b_m\rangle+\langle b_m|\,\tilde{O}\,|x\rangle\\
&\qquad+\langle x|\,\tilde{O}^\dagger\,|b_k\rangle+\langle x|\,\tilde{O}^\dagger\,|b_l\rangle+\langle x|\,\tilde{O}^\dagger\,|b_m\rangle+\langle x|\,\tilde{O}^\dagger\tilde{O}\,|x\rangle\,\big)\\
&= \frac{1}{16}\big(4+2\,\mathrm{Re}\,\big[\,\langle b_k|\,\tilde{O}\,|x\rangle+\langle b_l|\,\tilde{O}\,|x\rangle+\langle b_m|\,\tilde{O}\,|x\rangle\,\big]\big)
\end{aligned}
$$

$$
\begin{aligned}
P(A2A1=01) &= \langle\psi|\;\;(\mathbb{1}\otimes|01\rangle_A)(\mathbb{1}\otimes\langle 01|_A)\;\;|\psi\rangle\\
&= \frac{1}{16}\big(\,\langle b_k|-\langle b_l|+\langle b_m|-\langle x|\,\tilde{O}^\dagger\big)\big(\,|b_k\rangle-|b_l\rangle+|b_m\rangle-\tilde{O}\,|x\rangle\,\big)\\
&= \frac{1}{16}\big(\,\langle b_k|b_k\rangle-\langle b_k|\,\tilde{O}\,|x\rangle+\langle b_l|b_l\rangle+\langle b_l|\,\tilde{O}\,|x\rangle+\langle b_m|b_m\rangle-\langle b_m|\,\tilde{O}\,|x\rangle\\
&\qquad-\langle x|\,\tilde{O}^\dagger\,|b_k\rangle+\langle x|\,\tilde{O}^\dagger\,|b_l\rangle-\langle x|\,\tilde{O}^\dagger\,|b_m\rangle+\langle x|\,\tilde{O}^\dagger\tilde{O}\,|x\rangle\,\big)\\
&= \frac{1}{16}\big(4+2\,\mathrm{Re}\,\big[-\langle b_k|\,\tilde{O}\,|x\rangle+\langle b_l|\,\tilde{O}\,|x\rangle-\langle b_m|\,\tilde{O}\,|x\rangle\,\big]\big)
\end{aligned}
$$

And similar for $P(A2A1=01)$ and $P(A2A1=0)$. These 4 equations give us a linear system to solve for the 3 unknown gradient components, after measuring the probability distribution for $A2A1$. In general, we need one ancilla bitstring to use as the control for preparation of the state $\tilde{O}\,|x\rangle$; for m ancilla qubits this leaves 2^m-1 ancilla bitstrings that we can use for preparing the $|b_k\rangle$'s, and so we can compute 2^m-1 gradient components with one circuit.

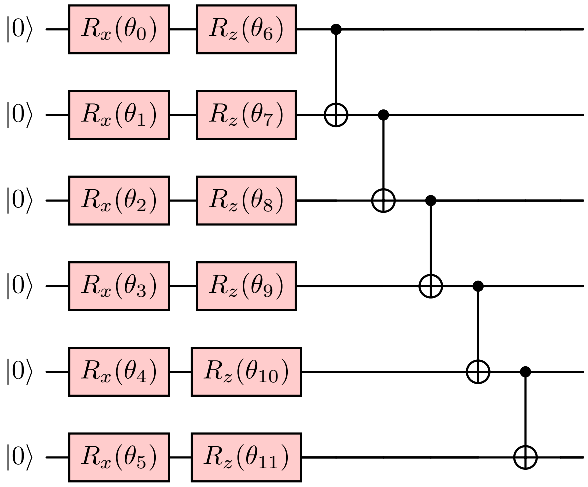
D Circuit Structure Used for QML Models

Our trained circuits are all constructed from a hardware-efficient ansatz, which consists of alternating rows of single-qubit rotations and two-qubit CNOT gates. These layers are repeated multiple times to increase the number of model parameters. An example with 6 qubits is shown below:



(a) 0 / 1 (b) 3 / 4 (c) 5 / 6

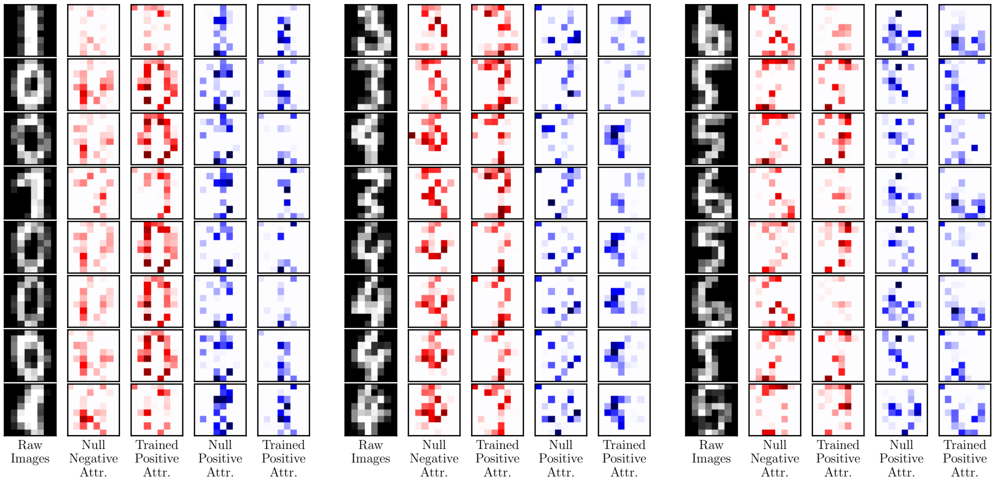
Figure 6: Sample images and the accompanying integrated gradients attribution for various samples from the NIST dataset. Attributions are for untrained null models with parameters sampled from a uniform distribution on the interval $[0, \pi)$. For comparison, we re-plot attributions for trained models from Fig. 1 alongside the null model attributions. Blue indicates positive attribution, red indicates negative attribution, and white indicates neutral attribution. We see from the lack of concentration that the null models fail to identify key features.



In such a circuit, the number of parameters is proportional to the number of qubits × the number of layers. Generally, selecting a layer count between 1× and 2× times the number of qubits provides the best accuracy (as demonstrated by our selection for the number of layers in Table 1).

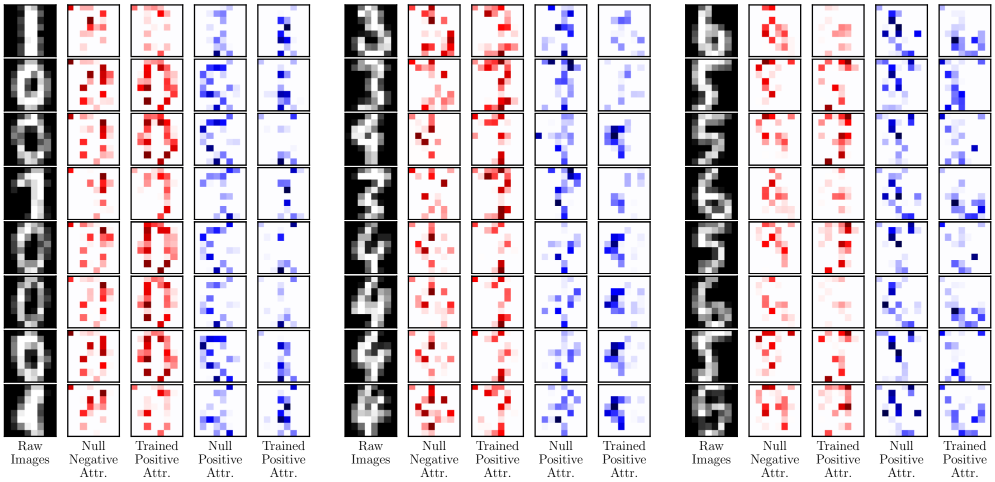
E Validation Against Null Model Attributions

To validate HATTRIQ's attribution scores, we also compute attributions for null models having randomly generated parameters, shown in Fig. 6. Parameters are randomly sampled from either a uniform distribution on the interval $[0, \pi)$, a normal distribution $\mathcal{N}(0, \frac{\pi}{2}^2)$, and a heavy-tailed Student's t-distribution ($\nu = 2$), Attributions are then computed and plotted for the same samples as used in Fig. 1, as shown in Figs. 6, 7, and 8 for the uniform, normal, and t-distributions respectively. Across the various classification tasks, we fail to see notable concentration or clustering of the attribution scores with either of the three distributions, unlike the trained case with HATTRIQ. For



(a) 0 / 1 (b) 3 / 4 (c) 5 / 6

Figure 7: Sample images and the accompanying integrated gradients attribution for various samples from the NIST dataset. Attributions are for untrained null models with parameters sampled from a Gaussian distribution $\mathcal{N}(0, \frac{\pi}{2}^2)$. For comparison, we re-plot attributions for trained models from Fig. 1 alongside the null model attributions. Blue indicates positive attribution, red indicates negative attribution, and white indicates neutral attribution. We see from the lack of concentration that the null models fail to identify key features.



(a) 0 / 1 (b) 3 / 4 (c) 5 / 6

Figure 8: Sample images and the accompanying integrated gradients attribution for various samples from the NIST dataset. Attributions are for untrained null models with parameters sampled from Student's distribution with $\nu = 2$. For comparison, we re-plot attributions for trained models from Fig. 1 alongside the null model attributions. Blue indicates positive attribution, red indicates negative attribution, and white indicates neutral attribution. We see from the lack of concentration that the null models fail to identify key features.

instance, the angular edge on the left side of digit four is only identified and attributed by HattriQs, while the three null attributions provide attribution scores all across the image.