QuantumScouter: Reinforcement Learning-Based Optimization of Variational Quantum Circuits for Differential Cryptanalysis

Gilsang Ahn
Korea University
Seoul, Republic of Korea
gilsang90@korea.ac.kr

Jiwoo Baek
Korea University
Seoul, Republic of Korea
baekjiwoo@korea.ac.kr

Donggun Lee
Korea University
Seoul, Republic of Korea
c15336@korea.ac.kr

Insung Kim
Korea University
Seoul, Republic of Korea
cmcom35@korea.ac.kr

Changmin Lee
Korea University
Seoul, Republic of Korea
changminlee@korea.ac.kr

Seokhie Hong
SmartM2M
Seoul, Republic of Korea
shhong@smartm2m.co.kr

Dongjae Lee
Kangwon National University
Chuncheon, Republic of Korea
dongjae.lee@kangwon.ac.kr

Abstract

Classical deep learning for differential cryptanalysis requires millions of ciphertext pairs, rendering attacks infeasible or easily detectable. This work overcomes this data limitation by introducing quantum differential distinguishers, enabling a practical attacker model where executing few queries is feasible. We design these distinguishers via quantum machine learning based on variational quantum circuits. To address circuit design challenges, we propose QuantumScouter, a reinforcement learning method that discovers compact quantum circuits. Unlike prior work, QuantumScouter explicitly targets metrics like gate count and circuit depth, producing circuits suitable for noisy intermediate-scale quantum hardware. We apply QuantumScouter to the SPECK32/64 and SIMON32/64 ciphers. The models are trained to distinguish ciphertext pairs possessing a meaningful differential from completely random pairs. For SPECK, our approach achieves 0.80 accuracy, outperforming the 0.53 of prior work while reducing required qubits from 16 to 6. For SIMON, we establish a novel quantum baseline with 0.75 accuracy. Notably, under a restricted regime of merely 400 training samples, classical deep learning models struggle. A simple MLP with 361 parameters fails to capture differential characteristics, resulting in an accuracy of 0.51, while a deep ResNet with 64,737 parameters overfits to yield an accuracy of 0.55. In stark contrast, our data-efficient quantum distinguisher extracts meaningful features using only 49 gates, maintaining 0.80 accuracy.

Keywords

Quantum computing, Quantum machine learning, Differential cryptanalysis, Reinforcement learning, SPECK, SIMON

1 Introduction

What can we *actually* do with quantum computers? We are currently living in an era where quantum computers are already available through cloud services, and many companies are competitively developing systems with increasingly large numbers of qubits. Recognizing the disruptive potential of these technologies, the U.S. administration issued an executive order in June 2026 to accelerate quantum computing capabilities, categorizing it as a pivotal strategic asset and a critical national security priority [36].

However, despite the promise of quantum advantage, the technical limitations of these near-term devices remain clear. For example, while Shor's algorithm could theoretically crack the RSA 2048 cipher in about eight hours, a physical realization would require approximately twenty million qubits when accounting for computational error rates and error correction overheads [15]. In reality, the largest quantum computers currently available possess only a few hundred qubits, indicating that we are in the noisy intermediate-scale quantum (NISQ) era rather than a mature fault-tolerant era [27]. Therefore, rather than focusing solely on distant theoretical threats, it is critical to understand what adversaries can practically achieve within limited NISQ resources from a security perspective. To this end, this work explores quantum adaptations of neural differential distinguishers, a cryptanalytic technique leveraging neural networks to detect differential characteristics.

To implement such cryptanalytic algorithms on small NISQ devices, the algorithms must be constrained to low-depth, error-robust, and qubit-efficient architectures. One class of algorithms that satisfies these requirements and can achieve quantum advantage is the variational quantum algorithm (VQA) [10]. VQAs optimize variational quantum circuit (VQC) parameters via a quantum-classical hybrid loop, enabling quantum computers to solve problems more effectively. In the NISQ regime, quantum machine learning (QML) is also implemented as a VQA [32]. By replacing classical neural networks with parameterized quantum circuits, QML leverages the quantum parallelism inherent in superposition and entanglement to provide expected computational advantages [22].

Because VQAs were created to secure quantum advantages on NISQ devices, the optimization of VQCs is crucial for practical implementation. Given the diverse and unstandardized NISQ platforms such as superconductors, photons, and ions [11], adaptive optimization tailored to each system's characteristics is essential. In existing research, an ansatz approach to the design of hardware-efficient (HE) ansatz-based VQC can be efficient to implement [37],

Table 1: Comprehensive comparison of QuantumScouter against classical deep learning and prior quantum baseline models for differential distinguishing tasks

Target Cipher	Model / Architecture	Methodology	Model Size	Depth	Qubits	Accuracy
SPECK32/64	Simple MLP	Classical DL	361 (Params)	–	–	0.51
	Deep ResNet	Classical DL	64,737 (Params)	–	–	0.55
	Prior Classifier [20]	Quantum (Random)	111 (Gates)	22	16	0.53
	Baseline (Model a)	Quantum (HE Ansatz)	243 (Gates)	71	6	0.52
	QuantumScouter (Model b)	**Quantum (RL)**	**49 (Gates)**	**19**	**6**	**0.80**
	QuantumScouter (Model c)	**Quantum (RL)**	**8 (Gates)**	**4**	**6**	**0.73**
SIMON32/64	Baseline (Model a)	Quantum (HE Ansatz)	243 (Gates)	71	6	0.74
	QuantumScouter (Model b)	**Quantum (RL)**	**56 (Gates)**	**21**	**6**	**0.75**
	QuantumScouter (Model c)	**Quantum (RL)**	**4 (Gates)**	**2**	**6**	**0.71**

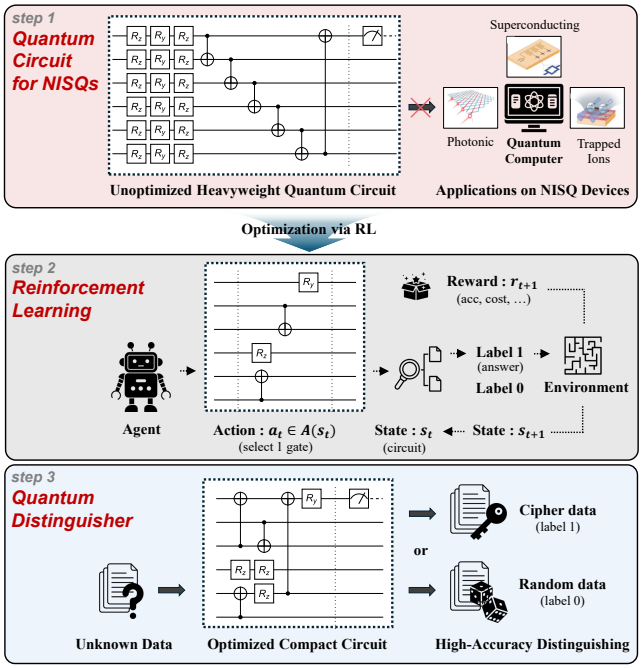


Figure 1: QuantumScouter: RL-based optimization of heavyweight circuits into compact quantum distinguishers for NISQ devices.

but it limits optimization. As NISQ devices develop, the ansatz tends to grow, making it unfavorable for training.

To make these QML models and quantum classifiers practical on NISQ hardware, their circuits must be exceptionally resource-efficient. This necessitates a minimal gate count and a shallow circuit depth to mitigate computational errors and decoherence. To address this challenge, we propose **QuantumScouter**, an RL-based quantum circuit optimization method. The name evokes a 'scout/-explorer' to emphasize that it searches the design space to discover **highly resource-efficient circuit** forms rather than fixing a template a priori. Accordingly, in this work, the term 'optimization' refers to the heuristic search process of discovering practical circuits using RL, rather than a mathematically proven global optimum.

To execute this search, we use reinforcement learning (RL) to optimize the VQC in QML [39]. As illustrated in the three-step framework of Fig. 1, our approach systematically bridges the gap between theoretical quantum algorithms and practical hardware limits. In Step 1, it highlights that the direct implementation of unoptimized, heavyweight quantum circuits on resource-constrained NISQ devices is severely limited due to the difficulty of preserving quantum states, such as superposition and entanglement, within their short coherence times. In Step 2, an RL agent learns to construct optimized circuits through an iterative process of selecting and adding one gate at a time. By utilizing an **adaptive reward model** that reflects the characteristics and specific costs of different NISQ resources, the agent effectively compresses the circuit. Finally, in Step 3, the resulting optimized and compact circuit is deployed as a quantum differential distinguisher. This resource-efficient circuit processes unknown input data to distinguish with high accuracy between cipher data containing meaningful differentials and purely random data.

To demonstrate that QuantumScouter is not limited to this specific cryptanalytic application, we then validate its effectiveness on general classification tasks such as parity and iris [31]. This successful optimization suggests that broader applications of QML are ready for practical implementation. Classifiers required in defense [29] and other critical industries need to process complex and diverse features quickly, and quantum advantages can help them achieve these goals more effectively than classical computers.

Even in computationally demanding simulations, our approach achieved the desired optimization with moderate overhead, proving that these optimized circuits are ready for practical deployment on various NISQ devices.

As summarized in Table 1, we evaluated QuantumScouter against both classical deep learning and prior quantum baseline models for differential distinguishing tasks. In the case study of SPECK32/64, a prior quantum classifier [20] required 16 qubits and 111 gates to achieve an accuracy of only 0.53. Remarkably, our RL-optimized circuit (**Model b**) distinguishes the differential ciphertext pairs with 0.80 accuracy using only 6 qubits and 49 gates, drastically improving both accuracy and resource efficiency. Furthermore, the highly compact **Model c** reduces the gate count further to just 8 gates while maintaining a practical accuracy of 0.73.

Beyond hardware efficiency, **data efficiency** is critical for establishing realistic threat models. In the field of cryptanalysis, extracting features for key recovery using limited chosen-plaintext queries is paramount. While classical deep learning models demand massive datasets to recognize these patterns, our quantum approach operates with minimal data. Executing massive chosen-plaintext queries against a target device is impractical and easily flagged by security systems. In contrast, performing only 400 queries constitutes a highly stealthy attack vector in operational environments.

To highlight this advantage, Table 1 also presents a performance comparison between the QuantumScouter distinguisher and data-restricted classical models. Under identical data conditions, classical deep learning baselines completely collapse due to either a lack of representational capacity in shallow networks or severe overfitting in heavily parameterized architectures. Conversely, our RL-optimized quantum circuit successfully learns the underlying differential characteristics. This demonstrates that the learned circuit architectures effectively capture algebraic signals within cryptographic permutations, providing a practical, NISQ-ready path toward quantum differential distinguishers. Ultimately, we expect our work to facilitate the implementation of various VQCs, ranging from QML models to complex computational problems, by optimizing their circuit structures to realize quantum advantages under strict hardware constraints.

In summary, the main contributions of this work are as follows:

- **Proposing QuantumScouter:** An RL-based optimization framework for Variational Quantum Circuits in QML.
- **Designing an adaptive reward model:** A flexible approach to tailor quantum circuits to the specific hardware characteristics and constraints of various NISQ devices.
- **Advancing quantum cryptanalysis:** Applying RL-optimized quantum distinguishers to achieve higher accuracy using fewer quantum resources compared to prior baselines.
- **Demonstrating extreme data efficiency:** Validating quantum cryptanalysis with minimal chosen-plaintext queries to establish a practical and realistic threat model.
- **Achieving resource-efficient quantum classification:** Implementing functional quantum classifiers on NISQ devices with reduced qubit counts and circuit depths.

This paper is organized as follows. Section 2 reviews background on QML and RL. Section 3 surveys related work, including studies on quantum differential distinguishers, and situates our approach. Section 4 presents QuantumScouter, a method for quantum circuit optimization that is based on RL. Section 5 and Section 6 present the experimental results, Section 7 provides a discussion of the findings, and Section 8 concludes the paper.

2 Background

2.1 Quantum Machine Learning

QML in the NISQ era relies on quantum data embedding [38] and quantum neural networks (QNNs) [5, 33]. A critical challenge in this environment is efficiently encoding high-dimensional classical data into a limited number of qubits. By leveraging superposition, quantum data embedding represents classical information within a quantum state's Hilbert space. For instance, angle encoding directly

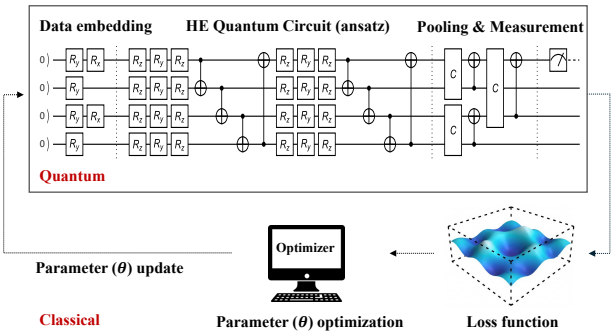


Figure 2: 4-qubit quantum classifier where 6 data points are trained with a 2-layer HE ansatz-based QNN.

maps a scalar data point x_i to a single qubit's continuous state via $x_i \rightarrow \cos(x_i/2)|0\rangle + \sin(x_i/2)|1\rangle$.

In other words, regardless of the size of the data-point x_i, it can be encoded into a single qubit. Consequently, classical data can be processed with fewer qubits, and information can be encoded continuously on one qubit, which provides advantages for data embedding. Furthermore, to achieve exponential data compression, amplitude embedding is widely used. This technique encodes a normalized 2^n-dimensional classical vector into the probability amplitudes of an n-qubit quantum state. Given the stringent resource limits of NISQ devices, this extreme compression is highly advantageous. Therefore, we utilize amplitude embedding in our study to efficiently map 64-bit classical ciphertext data into a compact 6-qubit quantum state. QNNs are implemented as VQCs, and they are trained by updating circuit parameters within the VQA loop. Specifically, parameter optimization in this context refers to updating the rotation angles ($\boldsymbol{\theta}$) of the parameterized quantum gates to minimize a given loss function. The superposition and entanglement created by the circuit operations play a role analogous to layers in a neural network, and quantum parallelism can provide performance benefits [1].

QML is divided into generative models and classifiers according to their roles. A generative model uses the amplitudes of the basis states of a quantum superposition to synthesize image pixels. With n qubits, it can generate 2^n pixels in a single shot. This capability can be used to produce chaotic cipher key images for encryption [2].

A quantum classifier learns data features through superposition and entanglement and then measures a single qubit with the Pauli Z observable. Based on the resulting expectation value, outputs close to -1 are assigned label 0, whereas outputs close to $+1$ are assigned label 1. Quantum classification [14] is the simplest and most powerful QML that can be implemented with quantum data embedding and QNN. Depending on the size of the dataset to be trained, quantum data embedding is used to prepare the state, and the appropriate QNN is configured and trained. Since it is intuitively difficult to construct an appropriate QNN, a structured VQC called an ansatz is used to fit the system qubit size, and it is increased layer by layer until classification is successful.

While many different ansatzes have been studied for different problems in VQA, we typically use the HE ansatz [19, 34] to make

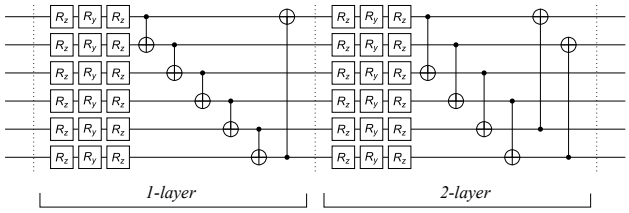


Figure 3: Strongly entangling layers for a six-qubit, two-layer architecture as used in QuantumScouter.

the best use of resources in the NISQ environment. Fig. 2 is an example of a quantum classifier circuit using 6 data point angle encodings and 2-layer HE on a 4-qubit system. However, naively increasing the number of layers to improve classification accuracy inevitably accumulates hardware noise and exacerbates barren plateaus, hindering the training process. This limitation underscores the critical need for advanced circuit optimization techniques that can minimize gate counts while preserving quantum expressivity. Therefore, automating this architecture search to find an optimal balance between expressivity and resource efficiency serves as the primary motivation for our reinforcement learning-based approach.

2.2 Reinforcement Learning

RL is a type of machine learning in which an agent learns how to perform a task through repeated trial-and-error interactions in a dynamic environment. The difference between RL and other machine learning is that it does not rely on a static dataset, but learns from the experience it gathers by operating in a dynamic environment [18].

RL can be represented as shown in Fig. 1, where at each time step t, the agent receives a new state s_{t+1} and a reward r_{t+1} from the environment when it takes an action $a_t \in A(s_t)$ given its state $s_t \in S$. Based on this interaction, the agent develops a policy $\pi : S \rightarrow A$ that maximizes the accumulated reward $R = \sum_{t=1}^{l} r_t$. By quantifying complex, non-formulaic problems into reward signals, RL efficiently optimizes objectives without requiring explicit datasets. Consequently, it is widely applied across diverse fields such as gaming, robotics, and autonomous vehicles [24].

In our work, we use the Deep Q-Network (DQN), a value-based method [3]. The Q network approximates the action value function Q over state s and action a, and the agent updates the network toward the optimal Q using the reward signal.

At each step the agent selects one operation gate as the action. The environment, the VQC, appends the gate to the circuit to produce the next state s_{t+1} and the reward r_{t+1}. The agent updates the function Q using this signal. After learning, under policy π, the state s_t that achieves the maximum reward r_t is taken as the optimized VQC.

2.3 Strongly Entangling Layers

Strongly entangling layers (SEL) are a specific type of variational circuit architecture designed to secure higher quantum expressivity [40]. This dense entanglement explores a broader Hilbert space, maximizing the capacity to capture complex algebraic correlations. As illustrated in Fig. 3, each qubit in a six-qubit SEL undergoes a sequence of parameterized single-qubit rotations (R_z, R_y, and R_z) to generate superposition, followed by CNOT-based entangling blocks. In an n-qubit system, the entanglers are applied sequentially to wires $i = 1, \ldots, n$, where the target wire for a given control i is defined as $j = (i + r) \bmod n$ (with 0 interpreted as n). Here, r serves as the range hyperparameter satisfying $0 < r < n$. To evaluate the versatility of the RL-based QuantumScouter, we apply our method to two distinct architectural variants. The first model applies RL optimization after incorporating the SEL, and the second model applies RL optimization to the entire circuit without using SEL.

2.4 Differential Cryptanalysis

Differential cryptanalysis is a technique that exploits deviations in the statistical relationship between input and output differences of a cipher from those expected in a random permutation [8]. Specifically, for a block cipher $E_K(x)$, let two plaintexts P_0 and P_0' produce ciphertexts $C_0 = E_K(P_0)$ and $C_0' = E_K(P_0')$. The input and output differences are defined respectively as:

$$\Delta I := P_0 \oplus P_0', \qquad \Delta O := C_0 \oplus C_0'. \tag{1}$$

When the probability $P(\Delta O \mid \Delta I)$ that a specific output difference ΔO occurs for a given input difference ΔI deviates significantly from the uniform probability, the cipher exhibits biased differential behavior that can be exploited to distinguish it from a random permutation and to reduce the effective key space during an attack.

The propagation characteristics of these differences are quantitatively described by a difference distribution table (DDT), which expresses the probability of each possible output difference for a given input difference. The DDT serves as a fundamental analytical tool for assessing a cipher's resistance to differential attacks, providing a statistical view of how differences evolve through the encryption layers.

Recently, deep-learning-based approaches have been explored to automate and enhance differential cryptanalysis. In particular, neural-network-based distinguishers have been employed to detect non-random patterns in ciphertexts, effectively distinguishing structured ciphertexts from random data and extending the capabilities of traditional differential methods [28, 42].

In this study, we evaluate our methodology on two representative lightweight ciphers: **SPECK32/64**, an ARX-based cipher, and **SIMON32/64**, which utilizes a Feistel network structure. The selection of these two distinct architectures allows for a comprehensive evaluation of our methodology across different cryptographic design philosophies, ensuring that the RL-based optimization can adapt to diverse algebraic structures. For these ciphers, the relationship between input and output differences can be expressed through the following system of equations:

$$\begin{aligned} \Delta I &= P_0 \oplus P_0', \\ C_0 &= \mathrm{E}(P_0), \\ C_0' &= \mathrm{E}(P_0'), \\ \Delta O &= C_0 \oplus C_0' = \mathrm{E}(P_0) \oplus \mathrm{E}(P_0'), \end{aligned} \tag{2}$$

where $E_k(\cdot)$ denotes the encryption function for either SPECK or SIMON. By training quantum classifiers to detect these differential biases (ΔO) stemming from a specific ΔI, we can construct a

quantum-based distinguisher that evaluates the security of these cryptographic primitives.

2.5 SPECK cipher

SPECK32/64 is a lightweight block cipher with a block size of 32 bits and a key size of 64 bits, consisting of a total of 22 rounds of encryption [4]. It was developed by the U.S. National Security Agency (NSA) and is specifically designed for deployment in resource-constrained environments.

With its simple structure, SPECK32/64 enables fast and efficient encryption and decryption, offering high performance in both software and hardware implementations. It has been widely adopted for securing data in various IoT devices and communication systems, and is officially standardized under ISO/IEC 29167-22.

Unlike traditional block ciphers such as AES, SPECK32/64 does not employ S-boxes. Instead, it utilizes an ARX (Addition-Rotation-XOR) structure. Specifically, as an ARX-based cipher utilizing modular addition for non-linearity, it combines this operation with bitwise rotation and XOR to achieve cryptographic diffusion. This design contributes to its high processing speed and low power consumption. Furthermore, SPECK32/64 is flexibly designed to meet a broad range of security requirements, making it suitable for applications ranging from low-security embedded devices to high-security systems. Although the full cipher consists of 22 rounds, in this study we focus on a reduced 5-round version of SPECK32/64 to facilitate comparative analysis with prior research on quantum distinguishers.

We use the 5-round SPECK32/64 to enable like-for-like comparison with prior studies that employ deep learning and QML to distinguish ciphertext pairs exhibiting the target non-zero differential from uniformly random pairs [16].

2.6 SIMON cipher

SIMON32/64 is a lightweight block cipher with a block size of 32 bits and a key size of 64 bits, consisting of a total of 32 rounds of encryption. Like its sister cipher SPECK, it was developed by the NSA and is specifically designed for deployment in highly resource-constrained environments [4].

While SPECK is optimized for software performance, SIMON is uniquely tailored for optimal performance in hardware implementations. Its exceptionally compact design makes it highly suitable for securing data in embedded systems, RFID tags, and other IoT devices where logic gate constraints are severe, which led to its standardization under ISO/IEC 29167-21.

Unlike SPECK32/64, which relies on an ARX structure, SIMON is a Feistel cipher relying on bitwise AND for non-linearity. It completely avoids modular addition and traditional S-boxes. Instead, its round function is constructed using only this basic non-linear AND operation, alongside bitwise rotation and XOR for diffusion. This specific structural choice is incredibly efficient in hardware circuitry, drastically reducing the required gate count and power consumption while maintaining a robust security margin against standard cryptanalytic attacks.

Although the full cipher consists of 32 rounds, in this study we focus on a reduced 5-round version of SIMON32/64 to evaluate our quantum distinguisher.

While our SPECK experiments provide a like-for-like comparison with prior deep learning studies, we utilize the 5-round SIMON to establish a novel quantum baseline. By successfully distinguishing ciphertext pairs with targeted differentials in a Feistel-based cipher, we demonstrate that the optimization capabilities of QuantumScouter are not limited to ARX structures but generalize effectively to fundamentally different cryptographic architectures.

3 Related Work

In this section, we introduce prior works related to the HE ansatz-based classifier and the QCNN-based classifier, which are used as direct comparison targets in our study. We also introduce a preceding study on the quantum distinguisher, which classifies SPECK32/64 ciphertext pairs with meaningful differentials and random ciphertext pairs using a quantum classifier, and which serves as the target of our research.

3.1 Ansatz-based Variational Quantum Circuits

Optimizing the VQC in the NISQ era is a significant challenge. General optimization approaches are limited because VQA configurations vary widely depending on the specific algorithm and the problem being solved. The ansatz, a structured template for such circuits, offers an efficient method for their construction in this context.

Various ansatz designs have been studied for specific applications. These include the resource-efficient HE ansatz—which has forms like linear or full entanglement—and the UCCSD ansatz for molecular chemistry [41].

Although an ansatz is helpful for efficiently constructing a VQC, its layer-by-layer optimization presents a clear limitation; as the system's qubit size increases, the resources per layer grow, making separate optimization efforts essential. While much research has focused on finding good ansatz-based VQCs, there have been some attempts to use machine learning for the optimization of VQC [25].

For instance, RL has been successfully used to optimize the Variational Quantum Eigensolver with a UCCSD ansatz. However, its direct contribution to circuit-level optimization is limited because the agent's reward is based on the final cost function value (i.e., expected eigenvalue) rather than the structure of the circuit itself.

This highlights a critical distinction for practical NISQ applications. While optimizing for the final cost function value is the ultimate goal, this approach does not guarantee that the resulting VQC is resource-efficient or even feasible to run on real hardware. Therefore, for the practical design of VQCs in the NISQ era, optimization must target not only the final result but also the circuit's structural cost.

This is precisely where QuantumScouter excels. By integrating resource-centric metrics—such as gate count, circuit depth, and gate variance—directly into its adaptive reward model, our method is fundamentally geared towards discovering resource-efficient architectures. This makes our approach a more practical and superior strategy for developing VQCs that can actually run and provide an advantage on today's constrained NISQ hardware.

3.2 Quantum Convolutional Neural Network

The advent of convolutional neural networks in 2012 sparked the deep learning boom, driving a wave of research and rapid technological progress. In parallel, the concept was carried over to quantum circuits, giving rise to Quantum CNNs (QCNN), which learn features and then perform pooling to compress information into selected qubits, much like classical CNNs. The model we use as a comparison target in this study is likewise a quantum classifier based on a QCNN [17].

This QCNN model achieves accuracies above 0.9 on MNIST and Fashion-MNIST. However, similar to the quantum classifier discussed next in Section 3.3, our experiments show that on SPECK32/64 ciphertext pairs its accuracy is close to random guessing, indicating that its effectiveness as a ciphertext distinguisher is limited.

3.3 Quantum Differential Distinguisher

After the success of a study that used deep learning to distinguish SPECK ciphertext pairs exhibiting a meaningful differential from random ciphertext pairs [16], follow-up work has applied the idea to quantum classifiers as well [20].

In that paper, the 64-bit data formed by a pair of 32-bit ciphertexts is basis-encoded into four 16-qubit quantum circuits. After passing through a quantum circuit built with a random ansatz, each qubit is measured and the resulting expectation values are used to classify the data. When the 64 expectation values are fed into a classical layer and then an output layer, the model attains an accuracy of 0.76. In contrast, using only the quantum circuit yields an accuracy of 0.53. Because the four 16-qubit circuits do not need to run simultaneously, a single 16-qubit circuit can be executed four times in sequence to obtain the full set of outputs. In this sense, the study effectively demonstrates a quantum classifier that uses 16 qubits and achieves 0.53 accuracy.

A critical distinction between this prior study and our work lies in the data complexity. Their approach relied on massive datasets to train the classifiers. In contrast, our QuantumScouter operates under extreme data starvation, utilizing only a few hundred samples. We deliberately restrict the dataset for two practical reasons. First, the iterative nature of our RL agent requires retraining the VQC every time a gate is added. As discussed later in Section 5.1, this incurs significant computational costs, making the use of massive datasets impractical during the heuristic search. Second, and more importantly, minimizing chosen-plaintext queries establishes a highly realistic and stealthy threat model for cryptanalysis. Despite operating with only a fraction of the data used in the prior study, QuantumScouter significantly outperforms their quantum-only baseline while using drastically fewer qubits and gates.

Although the accuracy of the prior model is modest, it is higher than random guessing at 0.5. The result suggests that, as quantum computing technology advances and environments capable of reliably simulating or executing 64-qubit circuits become available, QML-based classifiers could play a role in cryptography, for example in distinguishing ciphertext pairs with meaningful differentials.

4 QuantumScouter

QuantumScouter is a method that uses RL with an adaptive reward model to optimize quantum circuits. We focus on optimizing circuits for quantum classifiers, a key application in QML. Unlike traditional approaches that rely on predefined HE ansatz templates, our framework dynamically constructs resource-efficient architectures tailored to the strict constraints of the NISQ era.

In this study, we demonstrate the effectiveness of our method on two distinct tasks. We first apply our method to create quantum distinguishers for the SPECK32/64 and SIMON32/64 ciphers in the field of cryptanalysis. The primary goal of these distinguishers is to determine whether a given ciphertext pair exhibits a meaningful differential. To demonstrate broader applicability beyond cryptanalysis, we subsequently validate our approach on general classification tasks such as parity and iris.

Our results show that this approach yields highly efficient distinguisher circuits, requiring fewer gates than conventional HE ansatz-based quantum circuits.

4.1 Cryptanalytic Attack Model

To formalize our security evaluation, we define the threat model based on four standard dimensions. This setting represents a practical regime where classical deep learning models overfit or fail.

Attack Scenario. We consider an attacker evaluating lightweight block ciphers (SPECK and SIMON). The attacker utilizes a NISQ-era quantum computer to perform differential cryptanalysis on 5-round reduced versions of these ciphers.

Attacker Knowledge. The attacker possesses knowledge of the target cryptographic algorithms and their structures. Additionally, they know the optimal input differences (ΔI) required to construct chosen-plaintext pairs.

Attacker Capabilities. The attacker operates under a Chosen Plaintext Attack (CPA) model. To evade detection, they are constrained to a data-starved regime limited to 400 queries, while executing quantum circuits restricted to 6 qubits.

Attack Goal. The objective is to distinguish the target ciphertext pairs from random permutations using minimal data. Isolating these differential biases serves as the first step for a key recovery attack, reducing the cipher's security margin.

4.2 Agent and Environment

The agent adds one operation gate to the quantum circuit as an action. Since the quantum circuit we need to optimize should be a universal qubit operation supported by various NISQs, the types of operation gates with parameters that the agent can choose are rotation gates and interacting CNOT gates.

We do not include an *Undo* action to remove a gate. Instead, our framework achieves this effect through phase cancellation. Within an episode, a suboptimal rotation gate can be neutralized if the agent subsequently places a same-type gate such that the sum of their continuous parameters completes a full cycle (e.g., 2π), forming an Identity operation. Furthermore, across episodes, the RL agent learns to avoid adding inefficient gates to maximize rewards.

Therefore, in an n-qubit quantum system, the size of the agent's action space is $3n + n(n-1) = n(n+2)$, and the number of states that the agent learns if it performs l actions is $\sum_{t=1}^{l}(n(n+2))^t$. Even if we restrict the actions to universal qubit operations, the number of states, i.e., the number of cases of quantum circuits,

grows exponentially with the system size and the length of the operation, so approaches such as brute force are limited. In the environment, the state is represented as a gate-stream. This gate-stream consists of a $4 \times l$ list, where the first two values serve as identifiers for the gate type.

$[1, 0]$ represents a rotation gate, and $[0, 1]$ represents a CNOT gate. The last two values store the specific parameters of the operation. In the case of a rotation gate, the third index indicates the rotation type (encoded as 1 for Pauli X, 2 for Y, and 3 for Z), and the fourth index represents the target qubit number. For a CNOT gate, the third and fourth indices correspond to the control and target qubit numbers, respectively. For example, $[1, 0, 3, 2]$ indicates a rotation gate, applying a Pauli Z operation on qubit 2.

Although the representation could be reduced to three indices by encoding the gate type as a single scalar (e.g., 1 for rotation, 0 for CNOT), we employ a four-index structure. Since the $4 \times l$ gate-stream is flattened into a one-dimensional vector to serve as the input for the DNN in DQN, the use of $[1, 0]$ and $[0, 1]$ functions as a one-hot encoding. This separation enables the neural network to distinguish between gate types and interpret the subsequent parameters, facilitating efficient Q-function updates and reward calculations compared to a scalar representation.

4.3 Adaptive Reward Model

The optimization of VQCs must be achieved in a variety of NISQ system environments, and the reward of RL must be specified to set the best advantage in each NISQ system [23]. For quantum classification of QML, we designed a reward model with the following seven factors to account for the varying characteristics and resource tradeoffs of different NISQ hardware.

- Classification accuracy (per VQA subroutines)
- Classification cost
- Gate variance
- Gate duplication
- Gate type (rotation or interaction)
- CNOT operation (distance between qubits)
- Action step (total gates)

Classification accuracy and **cost** are necessary to achieve the goal of solving the problem, and the VQA subroutines are not directly related to the optimization of the classification quantum circuit, but are related to the total amount of computation.

Gate variance is directly related to the quantum system depth, which is the largest value of the number of gates operating on each qubit, and the depth can be different even if the total gates are the same. Since different NISQ resources have different coherence times, a long depth may not be a disadvantage.

Gate duplication prevents repetition of the same gate, for example, consecutive CNOT – CNOT gates are equivalent to no operation and should be excluded.

The type of gate depends on the cost of the gate physically implemented by NISQ, with interaction gates typically being the most expensive, but the trade-off can be made based on resources.

The distance between two qubits in a CNOT operation needs to be considered as interactions between distant qubits may be either physically impossible or prohibitively expensive depending on the specific NISQ topology.

The action step is related to the total number of gates, and while fewer is generally better, there may be exceptions in certain environments.

Since VQA is inherently an algorithm designed to obtain quantum benefits from NISQ, its implementation and application must be optimized for each NISQ environment, which requires adaptive reward settings. The combination and weight of the previous seven reward factors can be used to optimize the quantum classifier for a specific target NISQ architecture. Formally, the total reward can be formulated as a weighted sum of these factors, allowing for scalable application tailored to the specific constraints of the target hardware.

Note that our RL framework optimizes the core feature extraction and pooling processes. While amplitude embedding is a forward-looking challenge, our compact post-embedding architecture scales with state-preparation advancements, ensuring minimal hardware noise to maximize quantum advantage in the NISQ era.

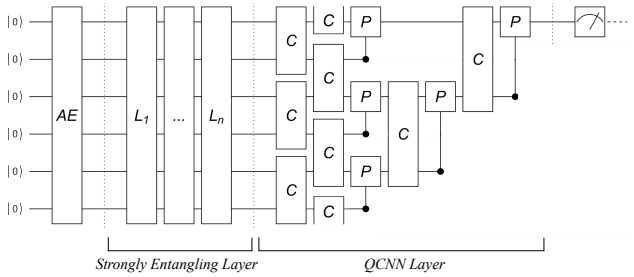
5 Experiments

We conducted the experiments in two stages. First, in Section 5.1, we show that QuantumScouter can be applied as a QML-based cryptanalysis method to distinguish ciphertext pairs with meaningful differentials for both SPECK32/64 (ARX) and SIMON32/64 (Feistel) ciphers. Next, in Section 5.2, to evaluate the general performance and broader applicability of our RL-based approach, we performed gate optimization on general classification tasks, specifically the parity and iris quantum classifiers studied in prior work [12, 30].

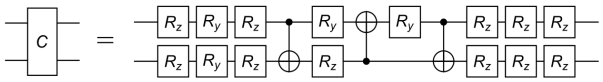
To evaluate these four models, we use classification accuracy and the number of QNN gradient subroutines [35] as performance metrics, alongside circuit depth and total gate count to measure quantum circuit efficiency. Since the focus of our work is to optimize the VQC architecture, the underlying QNN gradient subroutine algorithm is strictly controlled across all experiments.

Since the experiments were conducted via simulation, the adaptive reward model was configured assuming a general NISQ system characterized by short coherence times, low-cost rotation gates, and short-range qubit interactions. Therefore, all seven factors of the reward model presented in Section 4.3 were utilized, calculating the final reward as $r = r_{\text{acc}} \times 15 + r_{\text{cost}} \times 2 + r_{\text{var}} \times 3 + r_{\text{dup}} + r_{\text{type}} \times 3 + r_{\text{cnot}} + r_{\text{step}} \times 5$. These coefficients were determined heuristically through empirical experiments to balance the different scales of each metric. Specifically, a dominant weight of 15 is assigned to r_{acc} to scale its narrow raw range $[0.5, 1.0]$ and ensure accuracy remains the primary objective, while smaller auxiliary weights (e.g., 5 for r_{step}) enforce NISQ circuit compactness without overriding the core accuracy signal.

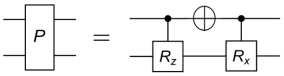
For the RL agent, we use the DQN algorithm with a discount factor of $\gamma = 0.8$. The probability of selecting a random action is determined by an ϵ-greedy policy, where ϵ decays over 500 steps from an initial value of 0.9 down to a minimum of 0.5. The replay memory buffer capacity is set to $10,000$. During training, the target network is softly updated at every step using Polyak averaging ($\tau = 0.05$). The agent and environment are implemented in Python using the PyTorch [26] and PennyLane [6] libraries, respectively. All experiments were conducted on a workstation equipped with an Intel Core i7-8700K CPU (3.70 GHz) and an NVIDIA GeForce



Quantum distinguisher circuit



Circuit for quantum convolution layer



Circuit for quantum pooling layer

Figure 4: The architecture of the distinguisher used in this experiment. (Model a) implements the full circuit as depicted. In (Model b), the QCNN layer is omitted and replaced by a circuit constructed via RL. (Model c) applies the RL method directly after the data embedding.

RTX 4060 Ti GPU (16 GB). Additionally, for intuitive representation, the 'Q3' symbolic quantum simulator package tool, written in the Wolfram language, was utilized [13].

5.1 Distinguisher Experiments

5.1.1 Models. To evaluate the effectiveness of QuantumScouter as a differential distinguisher, we design and compare three distinct quantum classifier models. The goal is to determine whether a given 64-bit input, representing a pair of 32-bit ciphertexts, exhibits a specific differential characteristic from either a 5-round SPECK32/64 or a 5-round SIMON32/64 cipher. All models process a 64-bit input data point, which is encoded into a 6-qubit quantum state using amplitude embedding. Following the data embedding stage, each model employs a distinct architecture, as illustrated in Fig. 4. As previously introduced, we utilize the SEL from Section 2.3, which acts as a feature extractor, and the QCNN structure from Section 3.2, which serves a pooling role to concentrate data.

Model a applies the SEL (2-layer), which is made of HE-ansatz, followed by the QCNN structure, concentrating the processed data into a single qubit for measurement.

Model b employs a hybrid approach. It first applies the SEL, just like Model a, but then utilizes QuantumScouter to construct a circuit in place of the QCNN layer. As described in Section 4.2, an action for the agent is to place a gate, and the maximum action step length is 30.

Model c begins with a bare circuit, without any predefined SEL or QCNN layers. Our RL-based method, QuantumScouter, is then applied from scratch to design the efficient circuit. The maximum action step length is 60.

The purpose of this model design is to experimentally verify whether applying RL to a known powerful architecture—the combination of feature extraction and pooling in Model a—can create a circuit that better analyzes the data's characteristics while using fewer resources than the conventional approach.

5.1.2 Dataset. The primary goal of the quantum distinguisher is to distinguish ciphertext data that is susceptible to differential cryptanalysis from that which is not. As explained in Section 2.4, this requires constructing plaintext and ciphertext pairs that share a specific input difference, denoted as delta (δ).

To create such data for **SPECK32/64**, we first generate a random 32-bit plaintext P_0 (composed of two 16-bit words) and apply the difference $\delta_{SPECK} = (0x0040, 0x0000)$ to generate its counterpart P_0'. This input difference is the same as that used in prior work [16], and the encryption is performed for **5 rounds**. Similarly, for **SIMON32/64**, we apply the input difference $\delta_{SIMON} = (0x0000, 0x0040)$ and encrypt the plaintexts for **5 rounds**. Additionally, we generate another random plaintext P_1, independently of P_0, for both ciphers. By encrypting these three plaintexts P_0, P_0', and P_1, we obtain the corresponding ciphertexts C_0, C_0', and C_1.

Based on this data generation process, the labeling is defined as follows. The ciphertext pair $C_0 \parallel C_1$, derived from unrelated plaintexts with no structured input difference, is labeled as **0 (random data)**. In contrast, the ciphertext pair $C_0 \parallel C_0'$, generated using the fixed input difference δ, is labeled as **1 (cipher data)**. Using this approach, we construct a dataset consisting of 64-bit ciphertext pairs, each labeled with either 0 or 1 depending on whether it is expected to exhibit differential characteristics.

One of the most significant challenges in our approach is the **significant computational cost** inherent in the RL process. For each action taken by the agent, a three-step cycle must be completed: (1) **Action**: a new gate is added to the current circuit; (2) **Training**: the parameters of the newly formed quantum classifier are optimized; and (3) **Evaluation**: the accuracy of the trained classifier is measured to provide a reward signal. This cycle, particularly the training phase, is extremely resource-intensive as it requires repeated quantum circuit simulations to optimize the classifier's parameters. The overall training time scales with the number of episodes, the number of actions per episode, and the cost of each individual classifier training run.

In our experimental setup, Model b explores up to 30 gate additions and Model c up to 60 per episode. Attempting to run a full training course of 500 episodes for each model would therefore demand a prohibitive amount of time and computational resources.

Consequently, due to these severe time and resource constraints, we made the practical decision to limit our dataset to 400 samples for training and 100 samples for testing. While this trade-off was necessary to keep the experimental workload manageable, **it also serves to demonstrate the extreme data efficiency of our proposed method.** Unlike classical deep learning models that often require massive datasets to generalize, QuantumScouter aims to prove that

Table 2: Quantum distinguisher optimization results for SPECK32/64 and SIMON32/64.

Model		a	b	c
		SEL + QCNN	SEL + **RL**	**RL**
SPECK (ARX)	Accuracy	0.52	**0.80**	**0.73**
	Circuit depth	18 + 53 = 71	18 + **1** = **19**	0 + **4** = **4**
	Gates	48 + 195 = 243	48 + **1** = **49**	0 + **8** = **8**
SIMON (Feistel)	Accuracy	0.74	**0.75**	**0.71**
	Circuit depth	18 + 53 = 71	18 + **3** = **21**	0 + **2** = **2**
	Gates	48 + 195 = 243	48 + **8** = **56**	0 + **4** = **4**

RL-optimized quantum circuits can effectively capture differential characteristics even with a very limited number of samples.

5.2 General Classification Experiments

The parity classifier can reproduce the parity function $f : \{0,1\}^2 \to \{1,-1\}$, where $f(x) = 1$ if there are an even number of ones in x and -1 otherwise. It is known that classification is possible with a 2-layer HE ansatz in a 4-qubit system, which requires a total of 32 quantum gates. Of course, classification is not possible with only one layer. To guarantee that our RL agent discovers a resource-efficient architecture, we strictly bound the maximum RL steps (i.e., the maximum number of added gates) to 32, matching this baseline gate count.

The iris classifier classifies the iris dataset, which is widely used in machine learning and statistics. The quantum circuit encodes real vectors as amplitude vectors (amplitude encoding) and trains the model to recognize the first two classes of flowers in the iris dataset. It is known that classification is possible with a 6-layer HE ansatz on a 2-qubit system [31], requiring 42 gates in total. Consequently, the maximum RL steps for this task are deliberately restricted to 42 to ensure the optimization prioritizes resource reduction.

6 Results

6.1 Distinguisher Experiments

As shown in Table 2, the distinguisher optimized via RL is superior to the HE ansatz model across all selected metrics.

In terms of performance, Model b, which retained the SEL and applied RL only to the QCNN layer, exhibited higher performance than Model c, in which all layers after data embedding were optimized via RL. Through this experiment, we have demonstrated that our approach is applicable to the field of cryptography, for tasks such as distinguishing ciphertext pairs that possess meaningful differentials.

Meanwhile, the distinguisher experiments were limited to a highly restricted dataset and only 500 episodes. While necessitated by the **significant computational cost** described in Section 5.1.2, this strict constraint intentionally serves to demonstrate the **extreme data efficiency** of our quantum approach under a highly practical threat model. This contrasts with Section 6.2, where models were trained for significantly longer—5000 episodes for parity and 1000 episodes for iris.

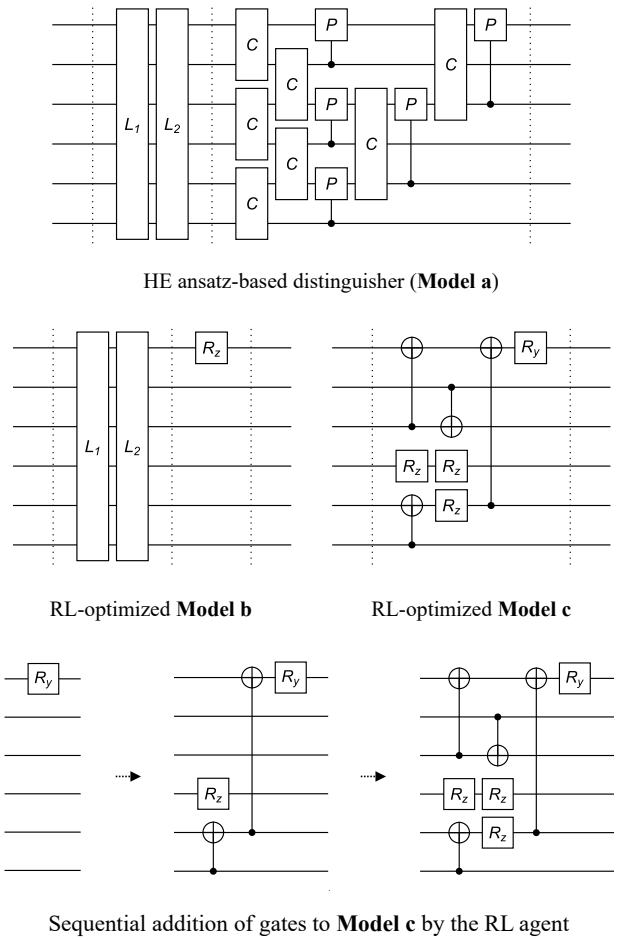


Figure 5: Hardware-efficient common distinguisher circuit (Model a) and RL-optimized circuits for SPECK32/64 (Model b: QCNN layer, Model c: entire circuit).

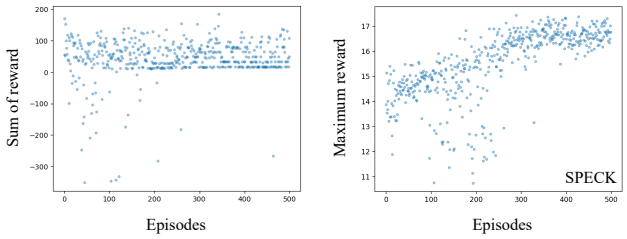


Figure 6: Reward convergence (sum and maximum) for the SPECK32/64 distinguisher (Model b), via QuantumScouter.

6.1.1 SPECK32/64 distinguisher. The results of the distinguisher experiment are shown in Fig. 5, and it is clear at a glance that sufficient optimization was achieved.

In contrast to **Model a**, which applied the well-known SEL and QCNN layers but only achieved an accuracy of **0.52** despite using many gates, Models b and c, which applied RL, show intuitive

Table 3: Comparison of data-restricted classical models and the QuantumScouter distinguisher (Target: SPECK32/64).

Model	Method *Params*	Dataset *Accuracy*
Classical Deep Learning Distinguisher	Simple MLP *361*	400 Train / 100 Test *0.51*
	Deep ResNet *64,737*	400 Train / 100 Test *0.55*
QuantumScouter (Ours)	**RL-based QML** ***49 (Gates)***	400 Train / 100 Test ***0.80***

optimization results. **Model b** optimized the 195-gate, 53-depth QCNN layer down to just a single rotation gate, classifying the dataset with an accuracy of **0.80**. Although this was preceded by a sufficient feature extraction process via the SEL (resulting in a highly compact circuit totaling only 49 gates), this can be interpreted as a remarkable result. **Model c** optimized the entire SEL and QCNN layer structure, consisting of 243 gates and a depth of 71, into just 8 gates with a depth of 4, achieving a classification accuracy of **0.73**.

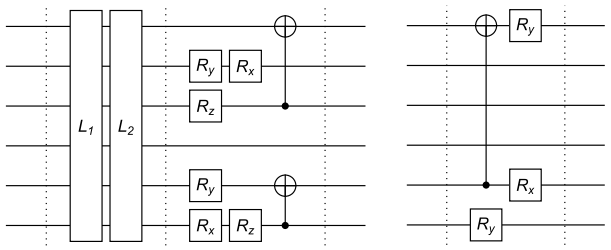
Fig. 6 shows the graphs for the sum of reward and the maximum reward per episode for Model b, which had higher accuracy. After approximately 200 episodes, both graphs exhibit stable convergence, indicating that the agent achieved an effective policy π. Notably, the maximum reward trends upward with continued training, which indicates that the agent is effectively performing quantum circuit optimization through RL.

As summarized in Table 3, we evaluated two classical deep learning baselines against our optimal QuantumScouter configuration (**Model b**) under a restricted condition of 500 total samples.

We evaluated a simple Multi-Layer Perceptron (MLP) and a scaled-down version of Gohr's residual network (ResNet) [16] as classical baselines. We included the **simple MLP**, configured with two hidden layers and 361 **parameters**. This capacity was kept smaller than the training dataset size of 400 samples to prevent memorization while addressing concerns regarding parameter asymmetry. However, as shown in Table 3, the shallow MLP failed to extract differential characteristics, yielding an accuracy of **0.51**. Although the model minimized the training loss, it degenerated to random guessing on the test set, demonstrating that simple classical architectures lack the representational capacity required to unravel cryptographic non-linearity.

To provide sufficient representational capacity, we then evaluated the **scaled-down ResNet architecture** (5 residual blocks and 32 filters). Even with this architectural reduction, the classical model retained 64, 737 **parameters**. Consequently, it overfitted the training data; its training accuracy converged to 1.0, but it failed to generalize, resulting in a test accuracy of **0.55**.

These results demonstrate that classical deep learning models collapse under data starvation regardless of architecture: the shallow model lacked representational capacity, and the deep model overfitted. In contrast, the circuit optimized by **QuantumScouter** captured the dataset features using 49 **gates** and achieved a test



RL-optimized **Model b** RL-optimized **Model c**

Figure 7: RL-optimized distinguisher circuits for SIMON32/64 (Model b: QCNN layer, Model c: entire circuit).

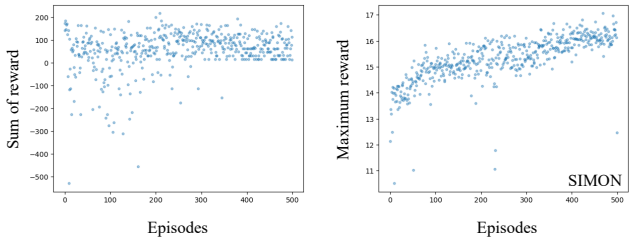


Figure 8: Reward convergence (sum and maximum) for the SIMON32/64 distinguisher (Model b), via QuantumScouter.

accuracy of **0.80**. This highlights an advantage of quantum machine learning in cryptanalysis by proving its ability to analyze cryptographic primitives and detect differential characteristics with data efficiency, making it effective where classical models fail.

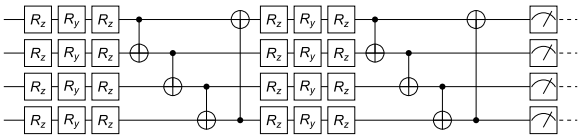
6.1.2 SIMON32/64 distinguisher. Unlike SPECK32/64, there is no prior research on quantum distinguishers for the SIMON32/64, meaning there is no established baseline for comparison. Therefore, our experiments were conducted solely by evaluating and comparing the performance of **Models a**, **b**, and **c**. The resulting optimized circuit architectures for SIMON32/64 are illustrated in Fig. 7.

As shown in Table 2, the baseline HE ansatz (**Model a**) achieved an accuracy of **0.74**, but it requires a massive circuit structure to capture the differential characteristics of the SIMON32/64 cipher. In contrast, the RL-optimized models achieved high accuracy with highly compact structures, even leaving certain qubits completely untouched. This efficiency is a direct result of amplitude embedding. Because the 64-bit input is already distributed across all 2^6 amplitudes of the initial 6-qubit state, the data is globally entangled from the start. This allows the RL agent to naturally perform *quantum feature selection*. By manipulating only a small subset of these pre-mixed qubits, the agent extracts the critical differential features while avoiding redundant gate operations and minimizing hardware noise.

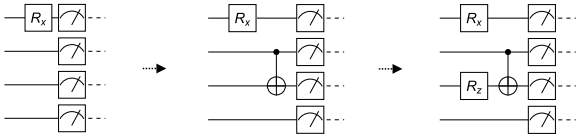
For instance, **Model b** successfully classified the dataset with an accuracy of **0.75**, while drastically compressing the QCNN layer to just 8 gates. Furthermore, **Model c** reduced the entire circuit from the massive baseline down to a mere 4 gates with a depth of 2, while still maintaining a robust accuracy of **0.71**. This highlights

Table 4: Parity and Iris classifier optimization results

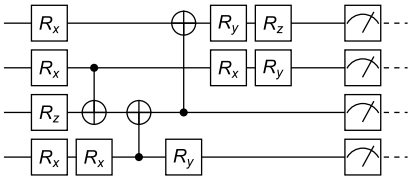
Classifier Metrics	Parity		Iris	
	HE	**RL**	HE	**RL**
Accuracy	1.0	**1.0**	1.0	**1.0**
#Subroutines	9	**4**	60	**22**
Circuit depth	10	**4**	24	**2**
Gates	32	**13**	42	**3**



HE ansatz-based parity classifier



Sequential addition of gates to parity classifier by the RL agent



RL-optimized parity classifier

Figure 9: Optimization of the parity classifier from a complex HE ansatz to a resource-efficient circuit via RL.

the agent's ability to construct a lightweight, highly tailored circuit from scratch.

Finally, the reward convergence for the SIMON32/64 experiment is presented in Fig. 8. Similar to the SPECK32/64 results, the agent discovered a stable policy π, successfully extracting features from the Feistel network despite the lack of a prior structural baseline.

6.2 General Classification Experiments

As shown in Table 4, the RL-optimized circuits outperformed the HE ansatz baselines in all resource metrics. While perfect accuracy (1.0) was maintained for these general datasets, significant improvements were achieved by drastically reducing the number of QNN gradient subroutines, circuit depth, and total gates. Specifically, the optimization reduced the parity circuit's depth from 10 to 4 and gates from 32 to 13, and drastically compressed the iris circuit's depth from 24 to 2 and gates from 42 to 3. This confirms the method's validity for general QML tasks.

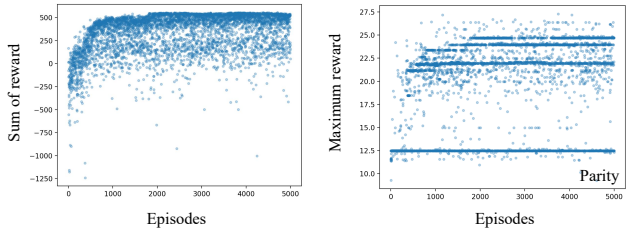
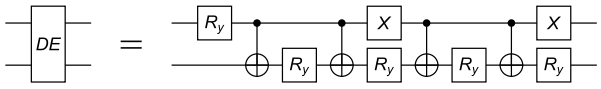
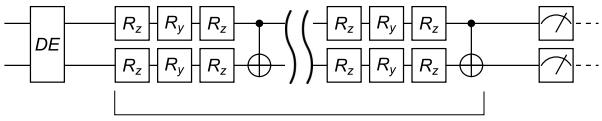


Figure 10: Reward convergence (sum and maximum) for the parity classifier, optimized via QuantumScouter.

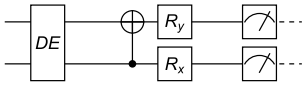


Quantum data embedding



6-layer

HE ansatz-based iris classifier



RL-optimized iris classifier

Figure 11: Optimization of the 2-qubit iris classifier from a 6-layer HE ansatz to a resource-efficient circuit via RL.

6.2.1 Parity classifier. Fig. 9 illustrates the complete optimization process for the parity classifier. The **top** panel shows the baseline: a complex, 2-layer fully entangled HE ansatz parameterized by θ (for R_x, R_y, R_z axes). The **middle** sequence demonstrates the RL agent sequentially constructing the circuit by adding gates.

In sharp contrast, the **bottom** panel presents the final RL-optimized circuit, which is drastically more compact. This highly hardware-efficient architecture demonstrates the effectiveness of our adaptive reward model, achieving a shallow depth, distributed gates, and minimal, short-range interaction gates suitable for typical NISQ constraints.

To verify the learning process, Fig. 10 plots the reward convergence. The left graph shows the accumulated reward ($R = \sum_{t=1}^{l} r_t$), indicating that while training progressed over 5000 episodes, 2000 episodes are sufficient to converge on an effective policy π to obtain the optimized quantum circuit s_t. Given the massive search space of $\sum_{t=1}^{32}(4\times6)^t$ for a 4-qubit system over 32 steps, this demonstrates extremely efficient learning. Furthermore, the consistent upward trend in the maximum reward (right graph) confirms that the agent genuinely learned to construct the optimal classifier rather than succeeding by chance.

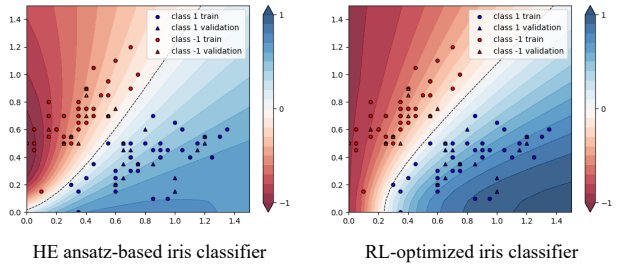


Figure 12: Comparison of decision boundaries between HE ansatz and RL-optimized 2-qubit iris classifiers.

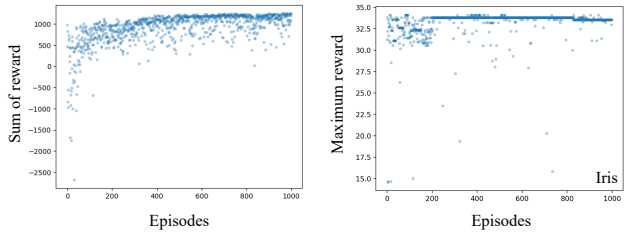


Figure 13: Reward convergence (sum and maximum) for the iris classifier, optimized via QuantumScouter.

6.2.2 Iris classifier. As shown in Fig. 11, QuantumScouter successfully compressed the iris classifier down to just 3 operation gates, excluding the initial quantum data embedding.

Fig. 12 visually confirms that despite this extreme circuit reduction, the RL-optimized model perfectly maintains the decision boundaries of the baseline HE ansatz, accurately separating the two iris classes on the validation set. Finally, Fig. 13 displays the reward convergence for the iris task. Although the agent explored a massive search space of $\sum_{t=1}^{42}(2\times4)^t$ across the bounded 42 steps for the 2-qubit system, it achieved a stable policy even faster, converging within just 500 episodes since the optimal solution required only 3 gates. The steady increase in both total and maximum rewards reaffirms the robustness and adaptability of the RL framework across different classification tasks.

7 Discussion

Through these experiments, we demonstrated that applying QuantumScouter to quantum classifier circuits enables the creation of optimized circuits with fewer gates and shallower depths.

First, Section 6.1 details QuantumScouter's application to quantum differential distinguishers. As shown in Table 2, both Models b and c outperformed the baseline HE ansatz (**Model a**). For SPECK32/64 (ARX), **Model b** replaced the 195-gate QCNN structure with just a single gate, increasing accuracy from **0.52** to **0.80**. Similarly, for SIMON32/64 (Feistel), it compressed the layer to 8 gates, achieving **0.75** accuracy. This proves that expert-designed feature extractors (SEL) remain valid, while RL optimally fine-tunes the complex subsequent structures. Meanwhile, **Model c** built circuits from scratch without SEL or QCNN layers, achieving accuracies of **0.73** (8 gates) for SPECK32/64 and **0.72** (4 gates) for SIMON32/64, serving as a clear testament to QuantumScouter's architecture-agnostic capabilities.

Next, to demonstrate general applicability beyond cryptanalysis, we validated our method in Section 6.2. As detailed in Table 4, for general classification tasks like parity and iris, QuantumScouter compressed large HE ansatz circuits into resource-efficient ones while maintaining an accuracy of **1.0**.

We acknowledge the classical simulation environment and its computational cost, which constrained our distinguisher dataset to 400 training and 100 test samples over 500 episodes, serving as a **Proof of Concept** for extreme data-starvation scenarios. It is important to clarify that we do not claim our quantum neural distinguisher currently surpasses the absolute limits of classical cryptanalysis. Traditional statistical methods based on Difference Distribution Tables (DDT) [7], differential trail searches [9] and Markov cipher models [21] are highly optimized; in fact, a manually crafted DDT-based distinguisher can achieve an accuracy of approximately 0.91 on 5-round SPECK32/64 even with restricted datasets [16]. Instead, the significance of our findings lies in the paradigm of machine learning-based cryptanalysis. Classical deep learning models require hundreds of thousands of chosen-plaintext queries to decipher non-linear characteristics. As demonstrated in Table 3, these models collapse under data starvation due to a lack of representational capacity or severe overfitting. In contrast, our RL-optimized quantum circuit extracts meaningful differential features utilizing only 49 gates. Ultimately, this highlights a promising trajectory for future research where advancements in quantum computing technology and the growing representational capacity of quantum hardware will enable quantum distinguishers to potentially surpass the accuracy of DDT-based classifications while maintaining high data efficiency.

Finally, comparing our best model with the prior quantum classifier [20] developed for SPECK32/64 (Table 1) reinforces this advantage. While the prior 16-qubit ansatz achieved an accuracy of **0.53**, our RL-based distinguisher achieved **0.80** using just 6 qubits, with fewer gates and a shallower depth, demonstrating its viability for physical NISQ hardware. Despite these advantages, we acknowledge the inherent scalability challenges of RL. Although QuantumScouter optimizes circuits for reduced-round ciphers, scaling this approach to full-round ciphers presents a challenge due to the exponential growth of the agent's state space. Therefore, future work should explore state space compression techniques or hierarchical RL to manage this combinatorial complexity, rule out statistical variance through large-scale hardware testing, and discover deeper, more complex quantum distinguishers.

8 Conclusion

In this study, we proposed QuantumScouter, a novel RL-based methodology for optimizing VQCs tailored to NISQ hardware. Applied to differential distinguishers, it generated circuits with significantly higher accuracy, fewer gates, and shallower depths than conventional HE ansatz models, demonstrating that RL-discovered circuits offer superior structural adaptability and efficiency for quantum cryptanalysis.

The potential of QuantumScouter extends to broader data-driven cryptanalysis. Notably, our experiments demonstrated high data

efficiency, detecting differential characteristics using minimal ciphertext pairs. While classical deep learning models require massive datasets, this quantum efficiency presents a stealthy, practical attack vector. Consequently, we recommend that cryptographic security evaluations and standardization processes incorporate QML-based cryptanalysis as a baseline threat model.

Although conducted in a simulation environment, QuantumScouter can be ported to quantum computers via cloud services. By refining the adaptive reward model on actual hardware, this RL-based optimization provides a practical pathway for advancing QML and applied cryptanalysis in the NISQ era.

Acknowledgments

Acknowledgments will be added in the final version of this paper.

References

[1] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner. 2021. The power of quantum neural networks. *Nature Computational Science* 1 (2021), 403–409. doi:10.1038/s43588-021-00084-1

[2] G. S. Ahn and S. H. Hong. 2025. QryptGen: a quantum GAN-based image encryption key generator using chaotic data distributions. *Quantum Information Processing* 24, 136 (2025). doi:10.1007/s11128-025-04750-5

[3] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. 2017. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38. doi:10.1109/MSP.2017.2743240

[4] R. Beaulieu, D. Shors, J. Smith, S. T-Clark, B. Weeks, and L. Wingers. 2015. The SIMON and SPECK lightweight block ciphers. In *Proc. 52nd Annual Design Automation Conference (DAC '15)*. ACM, 1–6. doi:10.1145/2744769.2747946

[5] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf. 2020. Training deep quantum neural networks. *Nature Communications* 11, 808 (2020). doi:10.1038/s41467-020-14454-2

[6] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, and N. Killoran. 2018. PennyLane: Automatic differentiation of hybrid quantum-classical computations. arXiv:1811.04968 doi:10.48550/arXiv.1811.04968

[7] E. Biham and A. Shamir. 1991. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology* 4, 1 (1991), 3–72. doi:10.1007/BF00630563

[8] E. Biham and A. Shamir. 1993. *Differential Cryptanalysis of the Data Encryption Standard*. Springer New York. doi:10.1007/978-1-4613-9314-6

[9] A. Biryukov, A. Roy, and V. Velichkov. 2014. Differential Analysis of Block Ciphers SIMON and SPECK. In *Fast Software Encryption (FSE 2014)*. Springer, 546–570. doi:10.1007/978-3-662-46706-0_28

[10] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3 (2021), 625–644. doi:10.1038/s42254-021-00348-9

[11] B. Cheng, X. H. Deng, X. Gu, Y. He, G. Hu, and D. Yu. 2023. Noisy intermediate-scale quantum computers. *Frontiers of Physics* 18, 21308 (2023). doi:10.1007/s11467-022-1249-z

[12] M. S. Choi. 2022. A Quantum Computation Workbook. Springer Nature. Retrieved 2026-03-10 from https://link.springer.com/content/pdf/10.1007/978-3-030-91214-7.pdf

[13] M. S. Choi. 2023. Q3: Symbolic quantum simulation. Github. Retrieved 2026-03-10 from https://github.com/quantum-mob/Q3

[14] E. Farhi and H. Neven. 2018. Classification with Quantum Neural Networks on Near Term Processors. arXiv:1802.06002 doi:10.48550/arXiv.1802.06002

[15] C. Gidney and M. Ekera. 2021. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum* 5 (2021), 433. doi:10.22331/q-2021-04-15-433

[16] A. Gohr. 2019. Improving attacks on round-reduced SPECK32/64 using deep learning. In *Advances in Cryptology – CRYPTO 2019*. Springer, 150–179. doi:10.1007/978-3-030-26951-7_6

[17] L-H. Gong, J-J. Pei, T-F. Zhang, and N-R. Zhou. 2024. Quantum Convolutional Neural Network based on variational quantum circuits. *Optics Communications* 550, 129993 (2024). doi:10.1016/j.optcom.2023.129993

[18] L. P. Kaelbling, M. L. Littman, and A. W. Moore. 1996. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* 4 (1996), 237–285. doi:10.1613/jair.301

[19] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta. 2017. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* 549, 7671 (2017), 242–246. doi:10.1038/nature23879

[20] H. Kim, K. Jang, S. Lim, Y. Kang, W. Kim, and H. Seo. 2023. Quantum Neural Network Based Distinguisher on SPECK32/64. *Sensors* 23, 12 (2023), 5683. doi:10.3390/s23125683

[21] X. Lai, J. L. Massey, and S. Murphy. 1991. Markov ciphers and differential cryptanalysis. In *Advances in Cryptology — EUROCRYPT '91*. Springer, 17–38. doi:10.1007/3-540-46416-6_2

[22] Y. Li, M. Tian, G. Liu, C. Peng, and L. Jiao. 2020. Quantum Optimization and Quantum Learning: A Survey. *IEEE Access* 8 (2020), 23568–23593. doi:10.1109/ACCESS.2020.2970105

[23] B. Nash, V. Gheorghiu, and M. Mosca. 2020. Quantum circuit optimizations for NISQ architectures. *Quantum Science and Technology* 5, 025010 (2020). doi:10.1088/2058-9565/ab79b1

[24] R. Nian, J. Liu, and B. Huang. 2020. A review On reinforcement learning: Introduction and applications in industrial process control. *Computers and Chemical Engineering* 139, 106886 (2020). doi:10.1016/j.compchemeng.2020.106886

[25] M. Ostaszewski, L. M. Trenkwalder, W. Masarczyk, E. Scerri, and V. Dunjko. 2021. Reinforcement learning for optimization of variational quantum circuit architectures. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. Curran Associates, Inc., 18182–18194. https://proceedings.neurips.cc/paper/2021/hash/9724412729185d53a2e3e7f889d9f057-Abstract.html

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, and S. Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

[27] J. Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2, 79 (2018). doi:10.22331/q-2018-08-06-79

[28] R. Rajan, R. K. Roy, D. Sen, and G. Mishra. 2022. Deep Learning-Based Differential Distinguisher for Lightweight Cipher GIFT-COFB. In *Machine Intelligence and Smart Systems*. Springer, 397–406. doi:10.1007/978-981-16-9650-3_31

[29] B. J. Schachter. 2015. Target classification strategies. In *Proc. SPIE*, Vol. 9476. SPIE, 947602. doi:10.1117/12.2176193

[30] M. Schuld. 2019. Variational classifier. Pennylane. Retrieved 2026-03-10 from https://pennylane.ai/qml/demos/tutorial_variational_classifier

[31] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe. 2020. Circuit-centric quantum classifiers. *Physical Review A* 101, 032308 (2020). doi:10.1103/PhysRevA.101.032308

[32] M. Schuld and N. Killoran. 2019. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* 122, 040504 (2019). doi:10.1103/PhysRevLett.122.040504

[33] M. Schuld, I. Sinayskiy, and F. Petruccione. 2014. The quest for a Quantum Neural Network. *Quantum Information Processing* 13 (2014), 2567–2586. doi:10.1007/s11128-014-0809-8

[34] V. V. Shende, I. L. Markov, and S. S. Bullock. 2004. Minimal universal two-qubit controlled-NOT-based circuits. *Physical Review A* 69, 062321 (2004). doi:10.1103/PhysRevA.69.062321

[35] Y. Suzuki, H. Yano, R. Raymond, and N. Yamamoto. 2021. Normalized Gradient Descent for Variational Quantum Algorithms. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE. doi:10.1109/QCE52317.2021.00015

[36] The White House. 2026. Executive Order: Ushering in the Next Frontier of Quantum Innovation. Presidential Actions. https://www.whitehouse.gov/presidential-actions/2026/06/ushering-in-the-next-frontier-of-quantum-innovation/ Accessed: 2026-07-06.

[37] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson. 2022. The Variational Quantum Eigensolver: A review of methods and best practices. *Physics Reports* 986 (2022), 1–128. doi:10.1016/j.physrep.2022.08.003

[38] M. Weigold, J. Barzen, F. Leymann, and M. Salm. 2021. Encoding patterns for quantum algorithms. *IET Quantum Communication* 2, 4 (2021), 141–152. doi:10.1049/qtc2.12032

[39] M. Wiering and M. van Otterlo. 2012. Reinforcement Learning. *Adaptation, learning, and optimization* 12 (2012), 729. doi:10.1007/978-3-642-27645-3

[40] Xanadu Quantum Technologies. 2013. Strongly entangled layers. Pennylane. Retrieved 2026-03-10 from https://docs.pennylane.ai/en/stable/code/api/pennylane.StronglyEntanglingLayers.html

[41] R. Xia and S. Kais. 2020. Qubit coupled cluster singles and doubles variational quantum eigensolver ansatz for electronic structure calculations. *Quantum Science and Technology* 6, 015001 (2020). doi:10.1088/2058-9565/abbc74

[42] T. Yadav and M. Kumar. 2021. Differential-ML distinguisher: Machine learning based generic extension for differential cryptanalysis. In *Proc. LATINCRYPT*. Springer, 191–212. doi:10.1007/978-3-030-88238-9_10

A Open Science

To ensure reproducibility, we provide the source code and the datasets used in this research. The artifacts are available at the following anonymous repository:

https://anonymous.4open.science/r/QuantumScouter

B Ethical Considerations

This research focuses on the theoretical cryptanalysis of existing cryptographic algorithms using quantum machine learning simulations. The proposed framework, QuantumScouter, is designed to rigorously evaluate the potential vulnerabilities of current ciphers under realistic noisy intermediate-scale quantum (NISQ) hardware constraints, rather than to facilitate or deploy malicious real-world attacks. Furthermore, all experiments were conducted using simulated environments and standardized cryptographic benchmarks (e.g., SPECK, SIMON); no human subjects, personally identifiable information (PII), or proprietary sensitive data were involved in any phase of this study. Ultimately, the primary objective of this work is strictly defensive. By proactively identifying and analyzing the capabilities of quantum-assisted cryptanalysis, we aim to contribute to the academic understanding of quantum threats and assist in the ongoing transition toward robust post-quantum security systems.