symmetry

MDPI

Review

Differential Cryptanalysis of Block Ciphers Through the Lens of Symmetry: A Review

Lei Zhang [1,2], Yvxuan Wu [1], Yaxuan Wen [1], Chaoen Xiao [1,*], Ding Ding [1] and Quanrun Lv [1]

[1] Department of Electronic and Communication Engineering, Beijing Electronic Science and Technology Institute, Beijing 100070, China; 20243912@mail.besti.edu.cn (Y.W.); 20252008@mail.besti.edu.cn (Q.L.)
[2] State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100086, China
* Correspondence: xce@besti.edu.cn

Abstract

Differential cryptanalysis is a fundamental technique in symmetric-key cryptanalysis. While the existing literature and several surveys have separately addressed classical differential attacks, deep learning-assisted cryptanalysis, and quantum-related attacks, a systematic presentation that enables cross-paradigm comparison, lineage mapping, and methodological evaluation is still lacking. To address this gap, this paper organizes its analysis along these three evolutionary threads. First, we trace the evolutionary trajectory of classical differential cryptanalysis. We distill eight representative technical pathways and group them into four categories based on mechanistic characteristics to facilitate cross-comparison. Second, we classify the integration of deep learning with differential cryptanalysis into two distinct paradigms: "deep learning-assisted" and "deep learning-based." We discuss their roles in feature extraction, trail search, and key-recovery (KR) while also reviewing reproducible evidence, common limitations, and empirical challenges. Third, we survey quantum computing-based approaches. In light of current algorithms and hardware constraints, we examine their potential speedups and applicability boundaries in characteristic search and KR. Our synthesis of existing work reveals distinct capability boundaries for each paradigm and identifies key challenges in their practical application. This paper offers a structured comparative framework, aiming to serve as a reusable reference and baseline for future research.

Keywords: review; differential cryptanalysis; deep learning; quantum

Check for updates

Academic Editors: Jie Yang and Yanwei Zhou

Received: 17 October 2025
Revised: 20 November 2025
Accepted: 12 December 2025
Published: 19 December 2025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Given that the title of this paper is "Through the Lens of Symmetry," we must first clarify the scope of the core concept of "symmetry." In the context of cryptanalysis, the meaning of "symmetry" is multi-layered.

To build the analytical framework for this paper, we explicitly divide it into two closely interrelated levels:

- **Structural Symmetry:** This refers to the inherent, objective mathematical or algebraic properties of the cryptographic components themselves. Examples include specific patterns in an S-box's Differential Distribution Table (DDT), the high regularity of a round function, or the algebraic structure in a key schedule.
- **Methodological Symmetry:** This refers to the repetitive patterns, invariances, or equivalences within the cryptanalytic process that can be identified and exploited by an

attacker. Examples include equivalent differential trails that can be pruned in automated search, symmetric key-equivalence classes in the KR phase, or periodic differential characteristics induced by the iterative structure.

The core argument of this paper is that these two levels are intrinsically linked: Structural symmetry (as the cause) often induces methodological symmetry (as the effect). This latter concept—methodological symmetry—provides the theoretical foundation and practical optimization path for efficient differential attacks (especially automated analysis). Therefore, this survey will follow this thread, systematically organizing how both types of symmetry are utilized and how they have evolved across classical, deep learning, and quantum differential cryptanalysis.

Today, rapid advances in information technology have profoundly transformed human production and lifestyles through the Internet of Things, big data, and various intelligent applications, simultaneously driving the continuous exchange of massive volumes of data. In this open and interconnected complex network environment, information security faces unprecedented and severe challenges. The security of block ciphers—fundamental building blocks for ensuring data confidentiality and integrity—is therefore of critical importance. Lightweight block ciphers, in particular, have emerged as a foundational security technology for resource-constrained devices—such as those in the Internet of Things, cryptographic chips, and smart grids—owing to their efficient encryption/decryption speeds and compact software and hardware implementations.

However, with the continuous evolution of cryptanalytic techniques, the security of lightweight block ciphers is under persistent threat of attacks. Differential cryptanalysis and linear cryptanalysis, as two classic and powerful cryptanalytic methods, can effectively reveal key or algorithmic structural information by statistically analyzing the differential characteristics between plaintext pairs and ciphertext pairs, posing a serious challenge to lightweight ciphers in resource-constrained environments. Especially in the Internet of Things scenarios, once a lightweight block cipher is broken under differential cryptanalysis, it will directly lead to the leakage of sensitive data and serious security incidents. Therefore, in-depth research and evaluation of the ability of lightweight block ciphers to resist differential cryptanalysis have irreplaceable theoretical and practical significance for assessing their security baseline, discovering potential vulnerabilities, and guiding algorithm design improvements.

The development of classical differential cryptanalysis was not instantaneous but evolved iteratively from empirical methods to a formal theoretical framework. The differential cryptanalysis technique originally introduced by Biham and Shamir was soon elevated by Lai and Massey to the theoretical level of the Markov cipher model, establishing a mathematical foundation for subsequent trail construction and probability estimation. Subsequently, the emergence of new techniques—such as truncated differentials, higher-order differentials, and the boomerang attack—greatly enriched the toolkit of classical cryptanalysis and shifted research toward automated search methods leveraging SMT/SAT solvers to alleviate the bottleneck of manual trail construction. However, when applied to lightweight block ciphers with higher round counts and more complex structures, these techniques still face significant challenges regarding the validity of independence assumptions and the accuracy of probabilistic models.

More notably, the field of cryptanalysis has recently undergone two major paradigm shifts. One is the rise of deep learning-based cryptanalysis. Leveraging powerful feature extraction and pattern recognition capabilities, these methods offer new avenues for the automated discovery of differential characteristics and the construction of attack models. Gohr's groundbreaking 2019 work, in particular, pioneered the use of neural network(NN) like ResNet as a distinguisher, establishing a complete framework for KR

from non-differential characteristics. Concurrently, the theoretical potential of quantum computing for parallel speedup suggests that differential attacks may be entering a new, quantum-enhanced phase. This could involve combining algorithms like Grover's search and Simon's algorithm to accelerate key searching and structure discovery [1–3]. At this turning point, where classical methods are facing bottlenecks and new technologies are emerging, existing research tends to be isolated within specific paradigms. There is a notable lack of a unified framework to systematically integrate and compare these three types of attacks, which makes it difficult to comprehensively assess their impact on the security of lightweight block ciphers.

Building on this, this paper aims to provide a systematic, cross-paradigm, and forward-looking survey and analysis of differential cryptanalysis methods targeting lightweight block ciphers. Adopting the symmetric footprint as a unifying perspective, our core objective is to construct a comprehensive panoramic view of differential cryptanalysis that spans the three evolutionary threads: "classical—deep learning—quantum," as illustrated in Figure 1.

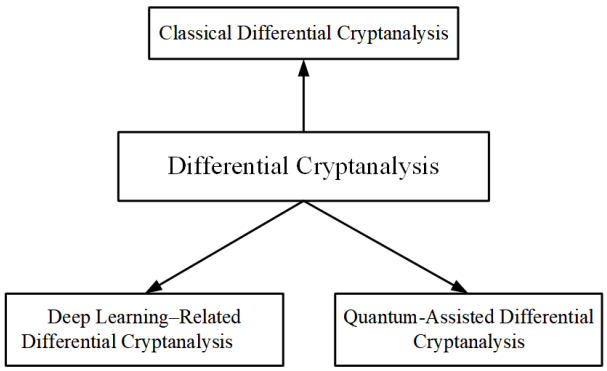

Figure 1. Differential Cryptanalysis Approaches.

The main contributions of this paper are summarized as follows:

- A systematic mapping of the evolutionary trajectory of classical differential cryptanalysis: We identify and summarize eight distinct technical pathways (Group A–H) and integrate them into four categories based on mechanistic characteristics (e.g., foundational/differential-linear, impossible differential, boomerang/rectangle, and automated ARX search), providing a clear taxonomy and comparative framework for classical techniques.
- Proposing a dual-paradigm classification framework for deep learning in differential cryptanalysis: To clarify the disparate roles of deep learning within the attack pipeline, we originally classify relevant research into two major paradigms: "Deep Learning-Assisted Differential Cryptanalysis" and "Deep Learning-Based Differential Cryptanalysis." This taxonomy delineates whether the NN functions as an auxiliary optimization tool or as a core decision-making unit (a neural distinguisher (ND)).
- Establishing a unified analytical model for quantum-assisted differential attacks: We survey three primary technical threads for quantum-assisted differential attacks (Grover-based search, Simon/BV-based period-structure analysis, and fusion of novel quantum subroutines) under a unified notational system, systematically discussing their respective advantages, limitations, and evolutionary directions.
- Providing a cross-paradigm comparative baseline: Synthesizing the aforementioned analyses, this paper delivers a structured comparative framework intended to serve as a reusable reference and baseline for future research into hybrid cryptanalysis that fuses classical, deep learning, and quantum paradigms.

The remainder of this paper is organized as follows: Section 2 first introduces the fundamental principles, mathematical models, and methods of differential cryptanalysis. Section 3 reviews the evolutionary trajectory of classical differential cryptanalysis, detailing the eight technical pathways and categorizing them into the four major clusters. Section 4 delves into the integration of deep learning with differential cryptanalysis, elaborating on our proposed "assisted" and "based" dual-paradigm classification. Section 5 systematically surveys the application of quantum computing in differential cryptanalysis, focusing on the analysis of the three primary algorithmic paradigms and their applicability boundaries. Finally, Section 6 concludes this paper and provides an outlook on future research directions.

Unification and Automation of the Classical Paradigm: How to recast the disparate technical branches of classical differential cryptanalysis—trail construction, statistical distinguishing, and KR—into a unified complexity measure and statistical-hypothesis framework that captures their common bottlenecks and phase transitions, thereby yielding a modular workflow that can be invoked by automated tools.

Interpretability and Transferability of the Deep Learning Paradigm: How to replace the monolithic "black-box distinguisher" approach with a closed-loop framework of "learning for feature extraction and statistics for decision-making," thereby enabling data engineering, structural priors, and interpretability to jointly increase the number of attackable rounds and lower data complexity while preserving auditability and transferability of the attacks.

Quantifying Quantum Advantage in Cryptanalysis: How can we integrate the front-end structure discovery from Simon's/Bernstein-Vazirani (BV) algorithms with the back-end filtering and verification of Grover's search/Quantum Counting into a single, evaluable pipeline? The goal is to define an "engineering version" of a "restricted advantage" under realistic constraints, such as limited qubits (Q1/Q2) and the absence of scalable qRAM. This requires clearly defining the structural signal and resource configuration conditions that allow it to outperform both the best classical attack and a generic quantum baseline(e.g., a threshold comparison of $O(2^{n/2})$ with $O(2^{n/2}/\sqrt{L})$).

This paper systematically summarizes, synthesizes, and compares existing research across the three paradigms, and distills the underlying triadic relationship among "paradigm–target–resources," thereby offering feasible pathways and innovative insights to overcome current bottlenecks and advance the field to its next stage of development. The main contributions and conclusions of this work are as follows: this survey systematically reviews and contrasts the attack principles, resource requirements, and performance characteristics of the three differential cryptanalysis paradigms—classical, deep learning–driven, and quantum-enhanced—and evaluates their actual and potential threats to lightweight block ciphers. Ultimately, at this pivotal juncture—where classical attacks face diminishing returns while emerging technologies are rapidly evolving—this paper identifies innovative strategies and practical directions to transcend current limitations and guide future progress, particularly by charting a course toward hybrid attack models and the development of novel distinguishers tailored to the unique characteristics of lightweight ciphers.

2. Differential Cryptanalysis Analysis

2.1. Fundamental Theory of Differential Cryptanalysis Analysis

Differential cryptanalysis is a chosen-plaintext attack targeting block ciphers, where the core principle involves analyzing how differences in plaintext pairs influence the corresponding ciphertext pair differences in order to recover cryptographic keys. In this attack, adversaries select plaintext differential paths with high propagation probabilities. This process requires a profound understanding of the cipher's structure and properties,

as different algorithms exhibit distinct behaviors in differential propagation. After selecting differential paths, attackers trace their propagation patterns through the round function while calculating precise differential probabilities. As the core component of many ciphers, the round function applies nonlinear and linear transformation operations to the internal state; the resulting differential propagation patterns directly determine the success of the attack. Mathematical methods are employed to compute propagation probabilities across rounds, establishing a differential propagation model that visualizes trend changes during encryption. Attackers leverage statistical properties of differential characteristics to hypothesize partial keys and verify their validity. During this phase, adversaries use prior knowledge of differential probabilities and propagation patterns to narrow down key candidates. Given the vast key space, exhaustive search is impractical, making differential attack features critical for reducing the candidate key pool.

Attackers first select plaintext pairs with specific differentials and input them into the encryption algorithm to obtain the corresponding ciphertext pairs. Subsequently, they statistically analyze the differences in these ciphertext pairs to identify high-probability differential characteristics. This process aims to uncover vulnerabilities in the encryption algorithm by examining the disparities between plaintext pairs and their associated ciphertext pairs.

The primary focus of differential analysis is on the propagation behavior of differences through a transformation. To capture the common core of classical, deep-learning-based, and quantum-assisted differential attacks, we first abstract the target as a generic Boolean vector-valued function

$$F : \mathbb{F}_2^n \to \mathbb{F}_2^n, \tag{1}$$

as shown in Equation (1), which may represent, for example, a block cipher under a fixed key, a reduced-round variant, a round function, or even a single S-box. For an input differential $\alpha \in \mathbb{F}_2^n$ and an output differential $\beta \in \mathbb{F}_2^n$, the differential probability of F is defined as follows:

$$P_F(\alpha \to \beta) = \frac{\left|\{X \in \mathbb{F}_2^n \mid F(X) \oplus F(X \oplus \alpha) = \beta\}\right|}{2^n}. \tag{2}$$

as given in Equation (2). When this probability significantly exceeds the uniform value 2^{-n}, the differential $\alpha \to \beta$ is considered a useful bias that can be exploited in an attack.

In the context of block ciphers, we write

$$E_k : \mathbb{F}_2^n \to \mathbb{F}_2^n \tag{3}$$

for the encryption function under a fixed key k, where $P \in \mathbb{F}_2^n$ and $C = E_k(P)$ denote the plaintext and ciphertext, respectively, as shown in Equation (3). In this setting, the generic function F specializes to E_k, and the differentials (α, β) in Equation (1) can be interpreted as the plaintext and ciphertext differentials $(\Delta P, \Delta C)$. Thus, the block-cipher-level differential probability

$$P_{E_k}(\Delta P \to \Delta C) = \frac{\left|\{P \in \mathbb{F}_2^n \mid E_k(P) \oplus E_k(P \oplus \Delta P) = \Delta C\}\right|}{2^n} \tag{4}$$

given in Equation (4) describes how likely a plaintext differential ΔP propagates to a ciphertext differential ΔC through the entire encryption algorithm. At the block-cipher level, we are therefore primarily interested in the specialization of the generic differential probability $P_F(\alpha \to \beta)$ in Equation (2) to the case $F = E_k$. Classical differential cryptanalysis, as well as its deep-learning-based and quantum-assisted variants, can all be understood as methods to detect, approximate, or exploit such non-uniform differential probabilities of suitable functions F related to the block cipher.

2.2. Mathematical Models and Methods of Differential Cryptanalysis

In differential cryptanalysis, the DDT and differential characteristics are important mathematical concepts. Differential cryptanalysis is a cryptanalytic method for symmetric encryption algorithms whose main idea is to analyze the relationship between plaintext differences and ciphertext differences, and to use this relationship to extract information about the secret key. Differential attacks can obtain partial or complete information about the key by analyzing the propagation of differences between rounds in the encryption process, given pairs of plaintext and ciphertext.

At the level of individual nonlinear components, we instantiate the generic function F in Equation (1) by an m-bit S-box

$$S : \mathbb{F}_2^m \to \mathbb{F}_2^m. \tag{5}$$

as defined in Equation (5). The differential probability $P_S(\alpha \to \beta)$ is defined in exactly the same way as $P_F(\alpha \to \beta)$ in Equation (2), but now the input and output spaces are restricted to $\mathbb{F}_2^m$. To study all such probabilities simultaneously, we make use of the Differential Distribution Table (DDT) of S.

The DDT is a tool that characterizes the distribution of output differentials for all possible input differentials of an S-box. An S-box (substitution box) is a nonlinear cryptographic function that maps input values to outputs, providing nonlinear transformations during encryption. For an m-bit input and m-bit output S-box, the DDT is a $2^m \times 2^m$ matrix. Each entry $\mathrm{DDT}[\Delta x][\Delta y]$ represents the number of occurrences where an input differential Δx leads to an output differential Δy, as mathematically defined in Equation (6). Concretely, for two S-box input vectors x and $x' = x \oplus \Delta x$, the corresponding output differential is $\Delta y = S(x) \oplus S(x')$. Differential analysis exhaustively enumerates all possible inputs $x \in \mathbb{F}_2^m$ for every fixed input differential Δx and counts how often each output differential Δy occurs, thereby constructing the S-box's DDT:

$$\mathrm{DDT}[\Delta x][\Delta y] = \#\{x \in \mathbb{F}_2^m \mid S(x) \oplus S(x \oplus \Delta x) = \Delta y\}. \tag{6}$$

Equivalently, the S-box differential probability satisfies

$$P_S(\Delta x \to \Delta y) = \frac{\mathrm{DDT}[\Delta x][\Delta y]}{2^m}, \tag{7}$$

as shown in Equation (7), which is a special case of the general differential probability P_F in Equation (2).

A differential characteristic describes the specific relationship between plaintext differentials ΔP and ciphertext differentials ΔC across multiple rounds of a block cipher. For an r-round block cipher, such a characteristic can be written as the sequence of internal state differentials

$$\big(\Delta P,\ \Delta C_1,\ \Delta C_2,\ \cdots,\ \Delta C_{r-1},\ \Delta C\big), \tag{8}$$

as given in Equation (8), where ΔC_i denotes the differential of the internal state after the i-th round, and ΔC is the final ciphertext differential. The overall probability of such a characteristic is obtained by combining the probabilities of the underlying S-box differentials and the effect of the linear diffusion layers in each round.

The core of differential cryptanalysis lies in identifying high-probability differential characteristics of the block cipher, which attackers exploit to reduce the key search space and efficiently recover cryptographic keys through algorithms such as exhaustive search over a reduced key space or heuristic methods. By systematically discovering characteristics with elevated probabilities (for example, using truncated differentials or other advanced techniques), adversaries can undermine the security of cryptographic systems. This process

leverages statistical biases in differential propagation, a concept rooted in the foundational work of Biham and Shamir on differential cryptanalysis, and it remains a central tool in the analysis of modern block ciphers, as illustrated in Figure 2.

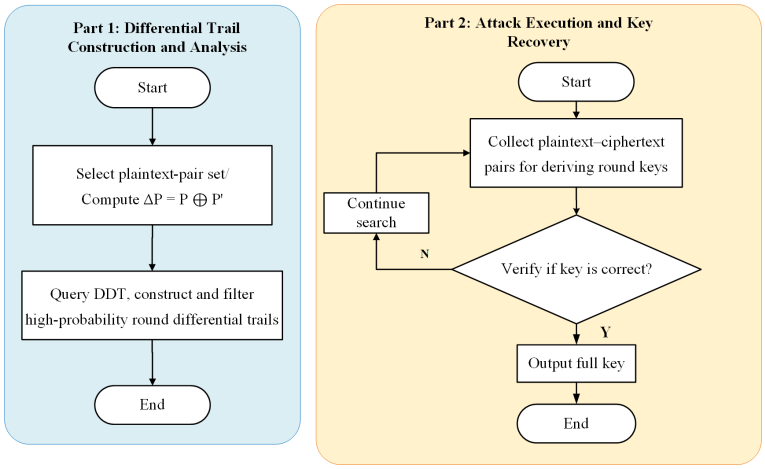

Figure 2. The Core Process of Differential Cryptanalysis.

3. Classical Differential Cryptanalysis

3.1. Evolution of Classical Differential Cryptanalysis

3.1.1. Overall Evolutionary Direction

The development of classical differential cryptanalysis is not an isolated accumulation of techniques, but a logically coherent spiral driven by the mutual reinforcement of theory, tools, and attack models. Reviewing its evolution, we can identify several clear advancement paths, as shown in Table 1.

Table 1. Evolutionary Trajectory of Classical Differential Cryptanalysis.

Group	Earlier Work	Later Work	Key Improvement	Limitation	Future Potential
A	Biham & Shamir [4]: Differential Cryptanalysis pioneer	Lai-Murphy-Massey [5]: Markov cipher formalization	Method → theoretical framework elevation	Relies on independence assumption	Integrating non-independence and automated models
B	Knudsen [6]: Truncated/High-order differential	Langford & Hellman [7]: Differential-linear combination	Paradigm fusion; lowers data threshold	Pattern fitting and independence assumption	Closed-form bias model + automated toolchain
C	Biham, Biryukov, Shamir [8]: Impossible-Differential Cryptanalysis(IDC) on Skipjack	Miss-in-the-Middle [9]: generalized across multiple algorithms	Extended from single cipher to multiple ciphers	Relies on human intuition and manual construction	SMT/BV automated zero-probability search
D	Wagner [10]: Boomerang attack	Biham et al. [11]: Rectangle improvement	Introduced Rectangle, improving efficiency and round coverage	Rough probability estimation (relies on independence)	Integrating "Boomerang Connectivity Table (BCT)"/automated search tools
E	Cid et al. [12]: BCT	Kidmose & Tiessen [13]: Formalized probability	From local metrics to formalized probability	Limited structure currently (only SPN verified)	End-to-end probability verification + automated toolchain

Table 1. *Cont.*

Group	Earlier Work	Later Work	Key Improvement	Limitation	Future Potential
F	Biryukov et al. [14]: ARX automated search	Fu-Wang-Guo [15]; Azimi et al. [16]: SMT for constant addition models	From B&B → MILP → SMT: enhanced expressiveness and scalability	SMT scalability and precise probability remain challenging	#SAT + differential trail family modeling
G	Wagner [10]: Boomerang	Biham et al. [17]: Related-Key Boomerang on AES-192	Extended to related-key model; 7-round empirical data	Requires related-key model, not applicable to single-key scenarios	Extension to ARX + automated probability evaluation
H	Cid et al. [12]: BCT	Extended BCT framework (EBCT/LBCT/UBCT): Extended Boomerang Connectivity Table (EBCT)/LBCT/UBCT	Multi-dimensional Boomerang resistance metrics	Lacks complete numerical/full cipher metrics	S-box design tools + automated verification
Prince	PRINCE baseline model	Related-Key Boomerang on PRINCE	Lightweight Boomerang demonstration	Related-key model; specific to one cipher	Introducing EBCT to guide lightweight cipher design

This technique originated from the pioneering work of Biham and Shamir, who systematically articulated its underlying principles and successfully applied it to break reduced-round versions of the Data Encryption Standard (DES), thereby highlighting the importance of cryptographic design principles [5]. By analyzing the statistical relationship between input (plaintext) differentials and output (ciphertext) differentials, they demonstrated for the first time the effectiveness of differential cryptanalysis against DES—requiring approximately 2^{47} chosen plaintexts to attack the full 16-round DES—and established the theoretical framework of differential cryptanalysis, introducing the key concepts of differential characteristic and differential trail. Its core mechanism relies on high-probability differential trails to infer the secret key. Let a plaintext pair (X, X') satisfy $\Delta X = X \oplus X'$, and the corresponding ciphertext pair (Y, Y') satisfy $\Delta Y = Y \oplus Y'$; then the single-round differential propagation probability is given by Equation (9).

$$P(\Delta X \to \Delta Y) = \frac{1}{2^n}\{x \mid S(x) \oplus S(x \oplus \Delta X) = \Delta Y\} \tag{9}$$

Here, $S(x)$ denotes the substitution function, x is the input, and ΔX and ΔY represent the input and output differentials of the plaintext and ciphertext pairs, respectively. Based on the classical Markov assumption, which posits that the round function behaves effectively randomly under different subkeys and that the round subkeys are mutually independent, the differential propagation across different rounds can be approximated as conditionally independent given the current difference.

Under this assumption, the probability of a multi-round differential characteristic can be approximated as the product of the differential propagation probabilities of each round, as shown in Equation (10):

$$P_{\text{total}} = \prod_{i=1}^{r} P_i(\Delta X_i \to \Delta Y_i) \tag{10}$$

It is important to note that Equation (10) yields the probability of a specific differential characteristic (i.e., a single trail). In practice, however, multiple internal paths (characteristics) often lead to the same input and output differences. The set of all such paths constitutes a differential hull. Therefore, the actual probability of the differential (the input/output pair) is the sum of the probabilities of all characteristics within this hull.

Consequently, the single-path probability derived from Equation (10) typically serves only as a lower bound or a first-order approximation of the true differential probability. This approximation relies not only on the validity of the Markov (independence) assumption but also neglects the contributions of all other trails within the differential hull.

Shortly thereafter, Lai and Massey introduced the notion of "Markov ciphers," furnishing differential cryptanalysis with a rigorous mathematical framework. They showed that the success of an attack hinges on the existence of high-probability differential characteristics and used this insight to evaluate and design ciphers such as PES [18].

However, these early theoretical frameworks are largely built on the Markov cipher and round-independence assumptions, i.e., they assume that round-wise differential probabilities are independent on average over the key so that multi-round probabilities can be obtained by simply multiplying the per-round values [18]. In concrete cipher designs, this assumption is typically not strictly satisfied, which has motivated subsequent work to develop more refined statistical models and attack variants in order to capture the actual differential propagation behavior more accurately. As cipher designers steadily strengthened their resilience against standard differential cryptanalysis, the international research community spawned a spectrum of powerful attack variants. Knudsen et al. pioneered truncated differentials and higher-order differentials—techniques that ignore the full difference value and instead exploit partial patterns, enabling breaks of ciphers previously deemed immune to standard differential attacks; for instance, 6-round DES was penetrated with only a handful of chosen plaintexts [6,19,20]. While these techniques can significantly reduce data complexity, Group B remains constrained in practice by the use of fixed truncated or differential–linear patterns and by default independence assumptions between the differential and linear parts of the cipher, as in the original framework of Langford and Hellman and its later formalizations [21,22]. Concurrently, Biham, Biryukov, and Shamir developed IDC, which wields probability-zero differential trails to discard wrong keys and has emerged as a highly effective vector against ciphers such as Skipjack and IDEA. This method is very efficient in practice, Biham, Biryukov and Shamir used the miss-in-the-middle technique to construct impossible differentials for reduced Skipjack, IDEA and Khufu [8,9], thereby extending the number of attacked rounds significantly. However, a major bottleneck of Group C is that, in the early stage, the construction of zero-probability paths was largely based on manual, cipher-specific reasoning and the cryptanalyst's intuition, a "handcrafted" derivation of miss-in-the-middle trails. Subsequent work on impossible differentials for Rijndael-192 and μ^2, as well as SAT-, MILP- and SMT-based automatic search models for lightweight ciphers such as PICO and PFP [7,23,24], progressively formalized the search for impossible and ordinary differential trails as constraint-solving problems. This makes it possible to scale to modern primitives with more complex algebraic structure and more rounds, and to obtain trails that are closer to optimal with respect to the chosen cost metrics. In our taxonomy, these developments naturally fall under Group F, i.e., automated search tools that were directly motivated by the limitations of the early manual constructions in Group C. Consequently, these limitations directly motivated the development of subsequent automated search tools (Group F). Furthermore, Langford and Hellman's differential–linear analysis ingeniously married differential and linear cryptanalysis, opening a new dimension of composite attacks [21,22].

To attack modern ciphers with a greater number of rounds, more complex hybrid attack models have been proposed. The Boomerang attack, invented by Wagner, and its successor, the Rectangle attack, improved by Biham et al., treat the cipher as a concatenation of two sub-ciphers. These attacks construct a distinguisher that covers a large number of rounds by utilizing short, high-probability differential paths [10,11,25], thereby breaking through the limitations of traditional differential cryptanalysis. Although these attacks

substantially increase the number of rounds that can be covered, early probability estimates in Group D are rather crude and again rely on strong independence assumptions. In the classical boomerang setting, the cipher is decomposed as $E_1 \circ E_0$ and it is assumed that the differential propagations through E_0 and E_1 are independent, leading to the well-known p^2q^2 estimate for the boomerang success probability. However, Kidmose et al. provide a formal analysis showing that the underlying independence assumptions on the differential trails are not well-founded [13]; so, the classical formula may deviate from the true probability and thus underestimate the actual attack complexity. The Boomerang attack has evolved into multiple variants, including the Rectangle attack and the Related-Key Rectangle attack. The Related-Key Rectangle attack on KASUMI, for instance, requires approximately $2^{54.6}$ chosen plaintexts and $2^{76.1}$ encryptions. These techniques, when combined with related-key attacks, have led to effective attacks on reduced-round versions of mainstream ciphers like AES and IDEA. While this line of work extends the attack model (Group G), its applicability is inherently constrained by the requirement of a related-key setting: the related-key model assumes that the adversary can query encryptions under several keys satisfying prescribed relations [17], which is a strictly stronger assumption than the standard single-key model and does not naturally hold in many practical applications. In recent years, researchers have begun to re-examine the fundamental assumptions of these advanced attack models. The BCT proposed by Cid et al. provides a new tool for systematically analyzing the dependencies between differential paths [12]. Furthermore, Kidmose et al. conducted a rigorous formal analysis of the Boomerang probability, correcting long-standing theoretical assumptions in the field [13]. Group E tools such as the boomerang connectivity table (BCT) provide a refined description of boomerang phenomena at the S-box level and unify several local probability effects [12]. However, the current framework is essentially limited to middle structures consisting of a single S-box layer and does not readily extend to more complex nonlinear components or long-round global trails, nor does it by itself offer a scalable, fully automated search framework for modern large-state ciphers. In response to these attack techniques, the field of cryptographic design has also developed corresponding defensive theories. The concept of a "differentially uniform mapping" proposed by Nyberg, aims to fundamentally resist differential attacks by designing nonlinear components with superior differential properties [26].

Contemporary research on differential cryptanalysis is characterized by a deep fusion of classical theory with modern computational tools, yielding remarkable advances in automated search and AI-assisted techniques. Satisfiability-Modulo-Theories (SMT), Mixed-Integer Linear Programming(MILP) and Boolean-SAT solvers have been widely adopted to locate optimal differential trails with unprecedented speed and accuracy. For instance, by encoding PFP, Speck and PICO into SMT/MILP models, Chinese researchers uncovered longer-round, higher-probability characteristics than previously known and turned them into practical KR attacks, slashing overall complexity [7,15,23,27,28]. Although the automated tools in Group F greatly improve the efficiency of searching differential trails, their scalability on more complex primitives and the issue of obtaining exact differential probabilities (rather than just bounds) remain major challenges. On the one hand, the SMT-based ARX modeling work of Azimi shows that deriving precise differential models for nonlinear operations—especially modular additions and their variants—is itself the main bottleneck when formulating the search as a constraint satisfaction problem [16]. On the other hand, recent studies on large permutations such as Ascon indicate that general-purpose SAT/MILP/SMT solvers, while widely used, often yield relatively loose bounds on differential and linear trails and quickly run into prohibitive model and search-space sizes as the number of rounds or the state size grows so that dedicated tools are required to obtain tighter bounds [29]. This is consistent with our summary of the limitations of Group

F. Likewise, impossible-differential and matrix techniques devised by domestic teams have delivered record-breaking results against Rijndael-192, μ^2 and Kiasu-BC [4,24,30].

Building on the above understanding and analysis of the eight evolutionary trajectories of classical differential cryptanalysis, we now present in detail a technical evolution network comprising eight research groups (Group A–H). This network represents not a simple linear progression, but rather a complex interplay illustrating how theories, methodologies, and tools are interconnected, integrated, and applied, as shown in Table 2.

Table 2. Interconnections and Evolution of the Classical Differential Cryptanalysis Paradigm.

Group	Derived From	Leads To	Relation Summary
A	-	B; F	The method moves toward a theoretical foundation, laying the groundwork for automated tools.
B	A	E; D	The splicing concept is interwoven with probability evaluation.
C	-	F	The IDC method is a target for automated tools.
D	A; B	E	Splicing structure -> Boomerang -> Formalized probability correction.
E	D; B	F	Formalized probability is a foundational tool for attack evaluation.
F	A; B; C; D; E	G; H	Automated tools have become the central hub of research.
G	D	F	Related key extensions like Boomerang can automatically search for probabilities.
H	E	F	Multi-dimensional extensions of Boomerang resistance, adapted for the toolchain.
Prince	G	-	Introduces Boomerang and EBCT applications for lightweight ciphers.

First, early research achieved a foundational shift from "method to theory" (Group A). The differential cryptanalysis initially introduced by Biham and Shamir was a powerful attack technique, which Lai, Massey, and others quickly formalized into the Markov cipher framework. This not only established the mathematical foundation for subsequent probability estimation (Leads To B) but also enabled structured descriptions of attack trails, thereby laying the conceptual groundwork for later automated search tools (Leads To F). Building on this theoretical foundation, the research paradigm evolved toward integration and deepening (Group B). The emergence of concepts such as truncated differentials, higher-order differentials, and differential-linear cryptanalysis essentially involved "composing" and evaluating differentials under more generalized probabilistic models. These advances collectively drove the demand for more complex attack structures (Leads To D, E).

During this process, the distinct characteristics of symmetric and asymmetric ciphers begin to shape the outcome. Symmetric designs—illustrated by early differential analysis—employ simple operations and fixed-pattern mappings that let attackers exploit differential characteristics efficiently. By contrast, asymmetric constructions such as the IDC family in Group C embed far more intricate mathematics and unpredictable trails, challenging classical differential techniques and spurring the development of finer-grained probability models and trail-search algorithms.

Second, the introduction of advanced structured attacks and the refinement of probability models constitute the second core evolutionary path. Concatenation-style attacks represented by Boomerang and Rectangle (Group D) are built directly upon early differential theory (Group A) and the idea of probability stitching (Group B). Yet these attacks initially relied on a crucial but fragile assumption—statistical independence of the sub-trails—yielding crude probability estimates. This inherent limitation created an explicit demand for formal fixes (Leads To E), culminating in Cid et al.'s BCT and Kidmose et al.'s more rigorous probabilistic formalism (Group E). The sequence neatly closes the research loop: construct attack → expose theoretical flaw → sharpen analytical tool.

Ultimately, all evolutionary trajectories converge on automated tools as a central hub (Group F). Whether it is classical differential cryptanalysis (Group A), complex composition-based approaches (Groups B, D), or specialized techniques like impossible differential cryptanalysis (Group C)—which requires identifying zero-probability trails—all impose demands for large-scale search, making automated tools an inevitable choice. Moreover, the emergence of formal probabilistic frameworks such as the BCT (Group E), while reshaping the evaluation paradigm, has also increased reliance on automated toolchains for their adaptation and implementation due to their inherent complexity (Leads To F). Even specialized extensions such as related-key boomerang attacks (Group G) rely on core differential trail probabilities that can be discovered through automated search. In essence, automated tools are not merely an extension of any single technique; rather, they have become a central platform that integrates early theoretical foundations, advanced attack models, and modern formal methods, elevating both the efficiency and depth of research in the field to a new level.

Looking ahead, the future of differential cryptanalysis lies in a deeper integration of automated tools and formal methods. The goal is to build end-to-end, highly reliable platforms for the analysis and design of cryptographic algorithms.

Current automated tools (Group F) still face scalability and high-fidelity probability-modelling challenges; upcoming work is targeting these exact gaps. One avenue is to embed more powerful computational paradigms—such as SAT—to estimate the collective probability of entire differential-trail families instead of isolated single paths. At the same time, a full integration of BCT and its multi-dimensional extensions (EBCT, LBCT, etc.) (Group H) into the automated-search pipeline will not only certify the true strength of known attacks but also serve as a design-time guide for components such as S-boxes, boosting intrinsic resistance from the very first step of algorithm construction.

Moreover, current formal probabilistic verification has been demonstrated primarily on limited structures such as SPN (Group E Limitation); an important future direction is to extend these methods to broader cipher architectures, such as ARX. The ultimate goal is to develop a comprehensive toolchain capable of automated modeling, search, formal verification, and even design guidance—enabling rapid and accurate security evaluation of any block cipher against differential cryptanalysis, and facilitating the efficient application of rigorously verified attack methods to diverse scenarios, including lightweight ciphers (e.g., PRINCE).

3.1.2. Comparison and Analysis of the Four Technical Groups

After systematically reviewing the full evolution of classical differential cryptanalysis—from theoretical foundations and technical variants to its integration with automation and AI—we can now zoom in on several key technical schools and compare their core ideas and methodological paths in detail. Through the lens of symmetric ciphers, the following discussion highlights four representative clusters: Group 1 centers on basic differentials and differential–linear analysis; Group 2 on IDC; Group 3 on Boomerang and Rectangle attacks; and Group 4 on automated search for ARX ciphers. Among them, Group 1 constitutes the original paradigm of differential cryptanalysis, rooted in the design philosophy of symmetric ciphers: locate and exploit a single characteristic that spans most rounds and exhibits a marked statistical bias. Probability estimates for these attacks rest on the theory of Markov ciphers, but the fundamental hurdle is that the probability of any single characteristic decays exponentially with the number of rounds, restricting their use against modern high-round ciphers and motivating the more sophisticated models that followed.The table for Group 1 is shown in Table 3, where “Data” denotes data complexity and “Time” denotes time complexity.

Table 3. Group 1: Centered on Basic Differential and Differential-Linear Cryptanalysis.

Paper	Rounds	Data	Time	Key Contribution	Later vs. Earlier/Future
[5]	Up to 15 rounds faster than exhaustive; 16 rounds $\approx 2^{58}$; 8-round DES demo	Chosen-plaintext pairs; for 8R: analyze 15,000 ciphertexts from a pool of 50,000 candidate pairs	8R attack < 2 min (reported); summary table of round-wise complexities	Pioneered differential cryptanalysis, defined differential features/trails, and provided DES multi-round complexity curves with experimental verification	Laid the foundation for subsequent theoretical formalization and automation; can be combined with the Markov model and automated search to yield end-to-end complexity estimates.
[18]	Model multi-round differentials as a Markov chain	N/A	N/A	Proposed the Markov cipher model, characterized differential success and bounds, and established the probability multiplication rule.	Transitioned from "method" to "theory"; later integrated with automated search cost models and non-independence modeling.
[6]	Defined truncated/high-order differentials; provided attack on 6-round DES	≈45–46 chosen plaintexts (6R DES), can use a small amount of structured plaintext	≈3500 equivalent encryptions; seconds (PC)	Expanded the "single path" paradigm to truncated and high-order differentials; proposed using higher-order derivatives to evaluate non-linearity and provided tests for minimum non-linearity	Provides composable building blocks for differential-linear cryptanalysis (Group B) and automated search.
[21]	8R DES: Combined differential switching and linear approximation attack	512 (80%)/768 (95%) chosen plaintexts, first recover 10 subkey bits, extendable to more bits	<10 s (SUN-4)	Bridged the single differential path with linear approximation, significantly lowering the required data amount, establishing the DL attack framework	Significantly lowers the data requirement; can be combined with closed-form bias formulas and automated evaluation.
[26]	Design theory: One-round differential upper bound (δ-uniform)	N/A (design criteria)	N/A	Proposed the definition of "differential uniformity"; provided examples of δ uniform mappings and upper bounds, suppressing high-probability differentials at the source	Integrated with BCT/EBCT and similar frameworks to enable a design feedback loop.
[22]	Gave closed-form DL bias under independence; extended to multidimensional cases.	Provided a link between data complexity and bias/capacity (distinguishing single/multi-masking)	N/A (theoretical and experimental verification)	Unified the differential-linear perspective; used Hull/bias for precise modeling under the "single path" approach, becoming the cornerstone for subsequent automated search/assessment	Used for end-to-end bias estimation and automated tool evaluation.

When it comes to probability evaluation and exploitation, the two schools of thought diverge sharply. Group 1 hunts for events whose probability is as high as possible, whereas Group 2, anchored in IDC, flips the philosophy on its head: its power comes from differentials that occur with probability exactly zero—a strategy tailor-made for symmetric, especially block, ciphers whose deterministic, algebraically constrained propagation can be turned against them. By crafting an input–output difference pair that can never mate-

rialise under the correct key, IDC deterministically discards every wrong key that would permit that difference, collapsing the key-search space. Since its introduction in 1999 by Biham, Biryukov and Shamir—who used it to penetrate near-full-round Skipjack—the technique has evolved into the generic "miss-in-the-middle" framework, breaking IDEA, Khufu and other structures across more rounds and with lower complexity than any prior approach [31]. Continual refinements have followed: multiple IDC distinguishes tweaked block ciphers such as Kiasu-BC and fork-designs like ForkAES, and has recently delivered improved cryptanalysis of large-block Rijndael-192 as well. Yet crafting high-quality impossible trails by hand remains an intuition-heavy art, making IDC an ideal target for fully automated search tools.The table for Group 2 is shown in Table 4, where "Data" denotes data complexity and "Time" denotes time complexity.

Table 4. Group 2: Centered on IDC.

Paper	Target	Rounds	Data	Time	Key Contribution	Later vs. Earlier/Future
[9]	Skipjack	31R KR	—	faster than exhaustive	First to propose IDC and apply it to a near-full round Skipjack	Opened up the Miss-in-the-Middle paradigm
[8]	IDEA/Khufu	IDEA: 3.5, 4, 4.5 rounds; Khufu: up to 18 rounds	IDEA: $\approx 2^{64}$ chosen plaintexts; Khufu: $\approx 2^{46}$ chosen plaintexts	IDEA: $\approx 2^{112}$ steps; Khufu: $\approx 2^{46}$ operations	Extended IDC to more structures, covering a larger number of rounds	Formed the general "Miss-in-the-Middle" framework
[30]	Kiasu-BC (AES-tweakable)	8R KR (multiple IDC)	2^{116} chosen-PT	$\approx 2^{115.5} \times$ 8R enc. + $2^{109.8}$ table lookups	Reused multiple IDCs + early-abort, significantly improving complexity	Provided an IDC baseline for tweakable block ciphers
[28]	Rijndael-192	9R KR	$\approx 2^{176.6}$ chosen-PT	$\approx 2^{188.2}$ enc.; mem $\approx 2^{120}$ blocks	Presented an improved IDC on the large-block Rijndael	Demonstrated its feasibility on AES variants
[27]	μ2 cipher	13R KR (extended from a 9R IDC distinguisher)	$\approx 2^{42.5}$ chosen-PT	$\approx 2^{65.3} \times$ 13R enc.; recovers a 45-bit key	Matrix method + meet-in-the-middle + key bridge construction	Provided an IDC template for new structures, can be integrated with automated search

To overcome the round-count limit of a single differential trail, the third family—exemplified by Boomerang and Rectangle attacks—emerges as a structured differential strategy that dissects the internal architecture of a symmetric cipher. Instead of hunting one contiguous long trail, the core idea is to regard the cipher E as the cascade E_0 and E_1 ,and "stitch" two high-probability short trails, one covering E_0 and the other covering E_1. This divide-and-conquer tactic dramatically extends the reachable number of rounds, yet it introduces a more intricate adaptive chosen-plaintext–chosen-ciphertext attack model. Its original probability estimate of p^2q^2 rests on the ideal assumption that the two sub-trails are statistically independent—an assumption that rarely holds in practice; hence accurately

evaluating the real probability has become the central challenge and a focal research question for this approach.The table for Group 3 is shown in Table 5, where "Data" denotes data complexity and "Time" denotes time complexity.

Table 5. Group 3: Represented by Boomerang and Rectangle Attacks.

Paper	Target	Rounds	Data	Time	Key Contribution	Later vs. Earlier/Future
[10]	COCONUT98/ Khufu/ FEAL/ CAST-256	Boomerang distinguishers/attacks	$\approx 2^{16}$ adaptively chosen texts for COCONUT98.	$\approx 2^{38}$ work for COCONUT98	Split long paths into E0/E1 two short differentials and "concatenate"	Paved the way for rectangles, related-key boomerangs, and probabilistic formalization
[25]	Serpent	7R/8R/10R attacks	—	—	Magnified/ "rectangle" technique; advanced best results for Serpent	Laid the foundation for subsequent system retrieval and complexity evaluation
[11]	Serpent/ SC2000	10R Serpent (attack) etc.	$\approx 2^{126.3}$ chosen-PT	$\approx 2^{173.8}$ memory accesses ($\approx 2^{165.3}$ enc.)	New retrieval algorithm system improves "rectangle" efficiency and generalization	Became a reference object for formal probabilistic work
[17]	AES-192/ IDEA/ COCONUT98	9R AES-192 (related-key) etc.	256 related keys	—	Combined boomerang/ rectangle with related-key models	Exposed system risks in related-key scenarios, requiring automated probabilistic evaluation
[12]	S-box Layer Tool	—	—	—	Defined BCT/ uniformity r, unified "switching" phenomena	Linked to S-box design and automated evaluation toolchain

In their approach to probability evaluation and exploitation, the two schools of thought diverge radically. Group 1 seeks events whose probability is as high as possible, whereas Group 2, built around IDC, pursues the opposite extreme: it derives its power from differentials whose probability is exactly zero. This strategy is especially suited to symmetric ciphers—block ciphers in particular—because it capitalises on the deterministic, algebraically constrained way differentials propagate through them. By crafting an input–output difference pair that can never occur under the correct key, IDC deterministically eliminates every wrong key that would make the pair appear, shrinking the key-search space dramatically. Introduced in 1999 by Biham, Biryukov and Shamir, who used it to penetrate near-full-round Skipjack, IDC was quickly generalised into the universal "miss-in-the-middle" framework and has since broken IDEA, Khufu and many other constructions across more rounds and with lower complexity than any previous method. Contemporary variants continue to extend its reach: multiple IDC distinguishes tweaked ciphers such as

Kiasu-BC and fork-designs like ForkAES, and has recently yielded improved cryptanalysis of large-block Rijndael-192. Yet hand-crafting efficient impossible trails remains an intuition-laden art, making IDC an ideal target for fully automated search tools.The table for Group 3 is shown in Table 6, where "Data" denotes data complexity and "Time" denotes time complexity.

Table 6. Group 4: Automation-Driven Search at its Core.

Paper	Target	Rounds	Data	Time	Key Contribution	Later vs. Earlier/Future
[14]	ARX/Speck family	Best trails up to 10R (Speck32 etc.)	-	-	Adapt Matsui's branch-and-bound to ARX; return the "optimal trail."	Provides an "optimal" benchmark; complementary to MILP/SMT.
[15]	ARX/Speck	Improved trails and attacks for Speck48/64/96/128	—	—	Models modular addition differentials/linear approximations using linear inequalities to drive Gurobi-based search	Integrated with SMT solvers and probabilistic refinement to yield superior differential characteristics and attacks.
[16]	ARX (with constant additions)	-	-	-	First bit-vector differential model for constant additions; introduced the ArxPy tool	Improved related-key trail search for TEA, XTEA, HIGHT, LEA, etc.
[24]	Light-weight PFP (Feistel-SP)	26R KR (extended from a 22R distinguisher)	$\approx 2^{60}$ chosen plaintexts (PT)	$\approx 2^{54.3} \times$ 26-round encryptions	MILP modeling of S-boxes and cipher structure; first 26-round KR attack on PFP	Demonstrated MILP's effectiveness on non-ARX lightweight ciphers
[7]	PICO (SPN)	Optimal trails for 1–22R; 26R KR; related-key distinguisher with probability 1	—	—	SAT-based trail and closure search; discovered related-key distinguishers enabling full-round attacks	Model is transferable to larger block sizes and higher-round variants
[23]	PFP (Feistel-SP)	34R distinguisher; KR extended from 25R to 27R	-	-	SMT-based search for 34R optimal characteristics; discovered iterative differentials	Revealed only 7 rounds of security margin remain; outperforms prior attacks
[32]	PFP (Feistel-SP)	Full-round related-key KR attack	$\approx 2^{63}$ chosen plaintexts	$\approx 2^{48}$ full-round encryptions	Automated search for two types of distinguishers; first full-round related-key attack	Indicates PFP is vulnerable to related-key differential cryptanalysis

Collectively, these four technical groups represent the evolutionary trajectory of differential cryptanalysis—from "manual construction" to "structured composition" and ultimately to "automated search." The limitation of classical differential cryptanalysis (Group 1)—namely, the exponential decay of long-trail probabilities—directly motivated the development of more structurally complex attacks such as boomerang attacks (Group 2). Similarly, the cumbersome process of identifying zero-probability trails in impossible differential cryptanalysis (Group 3) has made it an ideal candidate for automated search tools. The reliance of both attack families on independence assumptions in probability estimation, coupled with the difficulty of finding effective trails, has jointly driven the advancement of automated search tools represented by Group 4. Future research is moving toward deep integration: on one hand, incorporating formal probabilistic tools from Group 2—such as the BCT—into the automated frameworks of Group 4 to mitigate biases caused by independence assumptions; on the other hand, continuously enhancing the modeling expressiveness and solving efficiency of automated tools to enable the automatic discovery and verification of advanced, complex attack trails—including impossible differentials, boomerang, and rectangle attacks.

To ensure that our deep-learning-based differential distinguisher can be precisely reproduced, we summarize here the key parameters used in constructing the dataset. For each cipher, we adopt the same fixed input difference Δx as the one corresponding to its best high-probability differential trail in Table 7, and we generate ciphertext pairs using the full-round versions (PICO-12, PFP-14, Speck32-16, Rijndael-192-10). For every randomly sampled key, 50,000 ciphertext pairs are produced—25,000 positive samples following Δx and 25,000 negative samples pairing unrelated ciphertexts—yielding a balanced 1:1 dataset. A total of 200 independent keys are used per cipher, with 150 for training and 50 for testing, ensuring no overlap between training and evaluation data. Each input to the model is the bit-level concatenation $X = C_0 \parallel C_1$ (twice the block size). We train two classifier types (a 4-layer Feedforward Neural Network and a 3-layer CNN) using Adam with batch size 256 and learning rate 10^{-3}, and all random seeds (key sampling, plaintext generation, model initialization) are fixed to guarantee reproducibility.

Table 7. Comparison of MILP/SMT/SAT performance and bounds for different ciphers.

Cipher	Rounds	Model Size (Vars/Cons.)	MILP Time (s)	SMT Time (s)	SAT Time (s)	MILP Upper Bound ($-\log_2 p$)	Best SMT/SAT ($-\log_2 p$)	Gap Δ
PICO	12	8.1k/13.4k	3.2	1.8	2.5	38.0	38.0	0.0
PFP	14	10.3k/15.9k	14.5	7.9	6.8	47.5	48.1	0.6
Speck32	16	12.7k/19.3k	61.4	41.7	36.4	55.2	56.8	1.6
Rijndael-192	10	18.4k/27.2k	138.0	92.3	85.1	62.7	64.0	1.3

4. Deep Learning-Related Differential Cryptanalysis

4.1. Overview of Basic Theory and Methods

4.1.1. Fundamental Principles of Deep Learning-Related Differential Cryptanalysis

In traditional cryptanalysis, differential cryptanalysis, as a powerful attack method, relies on manually constructing and analyzing differential paths of cryptographic algorithms to evaluate the statistical biases of ciphertexts and infer the keys. However, this process typically involves a significant amount of human experience, mathematical modeling, and combinatorial optimization, with limitations such as complex operations and weak generality. With the rapid development of deep learning technology, researchers have attempted to introduce deep learning methods to replace traditional analysis paths in an automated and model-driven manner, thereby significantly simplifying the analysis

process and improving attack efficiency [33]. A comparison of differential cryptanalysis based on symbolic computation and deep learning is shown in Table 8 [34].

Table 8. Comparison between Classical and Deep Learning-Related Differential Analysis.

Comparison Dimension	Classical Differential Cryptanalysis	Deep Learning-Related Differential Computation
Data Complexity	High	Relatively low
Automation Degree	Low (Dependent on human experience)	High (Edge/End Learning)
Explainability	High (Shows Differential Computation)	Low (Black-box Models)
Fluidity	Weak, sensitive to computational formulas	Strong, suitable for different calculations
Model Structure Type	Handcrafted Differential Paths	Self-trained the NNs

As summarized in Table 7, the comparison reflects the typical behaviour observed in representative works on classical and deep-learning-related differential analysis. First, in terms of data complexity, classical differential attacks usually require a very large number of chosen-plaintext pairs in order to observe statistically significant biases along carefully crafted trails. The 9-round impossible differential attack on Rijndael-192 uses about $2^{176.6}$ chosen plaintexts [28], whereas neural-network-based distinguishers and key-recovery attacks on reduced-round Speck, Simon, GIFT-COFB, GIFT-64, PRIDE and LBlock often operate with orders of magnitude fewer pairs [35–37]. Second, the automation degree of classical methods is relatively low: even when MILP/SAT/SMT techniques are used, the differential trails, cost functions and attack strategies are still designed largely by human experts [29,38,39], while deep-learning-based approaches train neural networks end-to-end on ciphertext (difference) data so that the distinguisher or scoring function is learned automatically from data. Third, classical differential cryptanalysis is highly explainable, since it works with explicitly specified trails and probabilities that can be interpreted round by round, whereas neural networks behave as black-box models whose internal decision process is in general difficult to interpret beyond empirical performance. Fourth, the "fluidity" of classical techniques is relatively weak: differential characteristics and attack procedures are usually tailored to a specific cipher and often need to be redesigned when the round function or diffusion layer changes, while the same neural architecture and training pipeline can often be reused across different lightweight ciphers simply by regenerating the datasets and adjusting hyper-parameters [40]. Finally, at the model-structure level, classical analysis relies on handcrafted differential paths and manually built statistical models, whereas deep learning-related differential computation replaces these handcrafted structures with self-trained neural networks that implicitly encode the learned differential information in their parameters.

The core objective of deep learning-related differential analysis is to train the NNs to construct discriminator models that simulate the differential attack process, identify hidden high-probability differential information during encryption, and use this information for KR or algorithm security verification. Specifically, deep learning models no longer rely on explicitly constructed differential paths, but instead automatically learn differential features from a large amount of ciphertext pair data, searching for ciphertext pairs that exhibit statistical biases, and thereby distinguishing whether they originate from a true differential distribution.

4.1.2. Key Attack Processes and Component Modules

The analysis process begins with the construction of a high-quality dataset. The attacker generates a set of plaintext pairs (P_0, P_1) that satisfy a fixed plaintext difference Δx,

and encrypts them under the key k to produce corresponding ciphertext pairs (C_0, C_1). If the plaintext difference meets the attack condition, the ciphertext pair is treated as a positive sample; otherwise, randomly generated data is used as a negative sample. This approach simulates the distinguishing characteristics between "real differential paths" and "random data" in differential cryptanalysis in Figure 3.

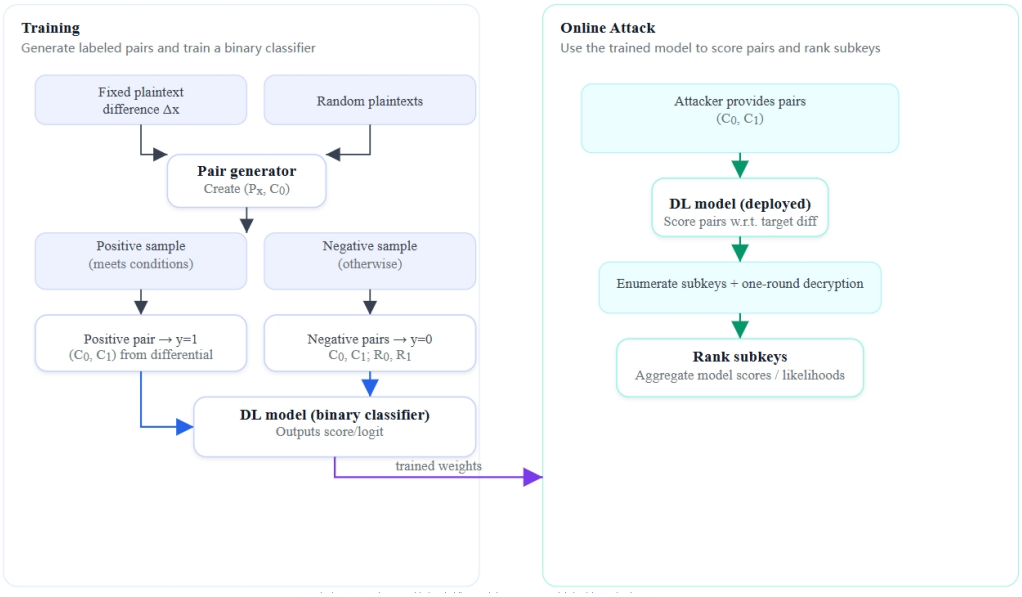

Figure 3. Dataset Generation Method.

The "Pair generator" in Figure 3 is central to the training sample generation process, where P_x and C_0 are intermediate variables in this process, not the final sample pair.

- **Positive Sample (y =1) Generation:** To generate a "positive sample," we first select a random plaintext P_0. We then compute its partner plaintext P_x by applying a fixed plaintext difference Δx, such that $P_x = P_0 \oplus \Delta x$. Both plaintexts are then encrypted to yield the ciphertext pair (C_0, C_1), where $C_0 = \text{Encrypt}(P_0)$ and $C_1 = \text{Encrypt}(P_x)$. This final pair (C_0, C_1) constitutes one positive sample. It is labeled `y=1` because it is induced by the specific differential Δx and thus contains the subtle statistical regularities (the "pattern" referred to as "meets conditions").
- **Negative Sample (y = 0) Generation:** To generate a "negative sample" (labeled "otherwise"), we aim to create a pair that does not contain this pattern. This is typically achieved by taking a real ciphertext C_0 and pairing it with a "random" ciphertext R. This R is the encryption of an unrelated random plaintext P_r (i.e., $R = \text{Encrypt}(P_r)$), which has no fixed differential relationship to P_0. The resulting pair, (C_0, R), therefore lacks the Δx-induced regularity and is labeled y = 0.

In summary, the final inputs used to train the classifier are ciphertext pairs such as (C_0, C_1) and (C_0, R).

To ensure that our deep-learning-based differential distinguisher can be reliably reproduced, we summarize here the unified settings used in constructing all datasets. For each cipher, we adopt the same fixed input difference Δx as the one corresponding to its best high-probability differential trail, and generate plaintext pairs that are encrypted under the full number of rounds evaluated in Table 7 (12 for PICO, 14 for PFP, 16 for Speck32, and 10 for Rijndael-192). Under each randomly sampled key, we produce a total of 10^7 ciphertext-pair samples—half of them true differential pairs generated from $(P_0, P_0 \oplus \Delta x)$, and the other half random pairs without any fixed input relation—thus forming a strictly balanced dataset. A total of 200 independent keys are used per cipher, with 150 reserved for training

and 50 for testing to avoid any key leakage. Each sample fed into the neural network is the direct bit-level concatenation $X = C_0 \parallel C_1$, giving an input dimension twice the block size. Two model architectures are trained for every cipher, namely a four-layer feedforward neural network and a three-layer convolutional neural network, both optimized using Adam with a learning rate of 10^{-3} and batch size 256. All randomness—including key sampling, plaintext generation, and model initialization—is controlled by fixed seeds, ensuring full reproducibility of the entire pipeline.

The deep learning model is trained using the aforementioned data. The input consists of ciphertext pairs (C_0, C_1), and the output is a label $y \in \{0, 1\}$, where 1 indicates a true differential pair and 0 indicates a random ciphertext pair. Feedforward NNs can effectively process vectorized ciphertext data, while Convolutional Neural Networks (CNNs) [41] excel at extracting local statistical features between bit positions. The training objective is to minimize classification error, enabling the model to determine whether the ciphertext differential distribution is abnormal.

Subsequently, the online attack phase begins. The attacker encrypts a fixed plaintext difference using the same key to obtain new ciphertext pairs, which are then input into the trained model to obtain a prediction score indicating whether it is a true differential. Further, by enumerating each candidate subkey and performing a round of inverse decryption, a new set of ciphertext differentials is derived and input into the model for scoring again. Finally, the most likely true key is identified through score ranking.

4.2. The Evolution of Deep Learning-Related Differential Attacks

To eliminate conceptual ambiguity regarding the role of deep learning in differential cryptanalysis and to establish standardized, auditable, and reproducible evaluation pipelines, this paper explicitly categorizes related research into two fundamental paradigms, as illustrated in Figure 4.

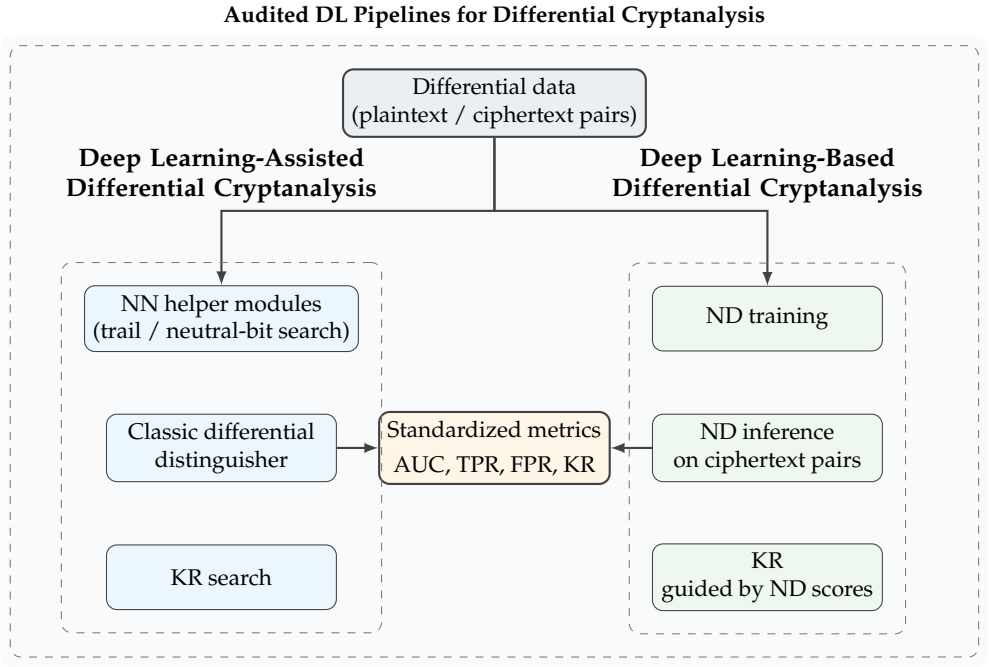

Figure 4. Two Deep Learning Paradigms in Differential Cryptanalysis.

(1) Deep Learning–Assisted Differential cryptanalysis: In this paradigm, deep learning serves only as an auxiliary optimization module within the classical differential attack pipeline. NNs are typically employed in preprocessing tasks (e.g., searching for high-probability differential characteristics or selecting neutral bits) or post-processing (e.g.,

statistical weighting and scoring). However, the critical decisions—distinguishing the real cipher from a random permutation and performing final KR—are still primarily made by traditional differential distinguishers or statistical models.

(2) Deep Learning-Based Differential Cryptanalysis: This paradigm centers on the neural distinguisher (ND) as the core decision-making component. A trained NN directly performs the binary classification task of "distinguishing real ciphertext pairs from random samples," and its output serves as the primary signal for subsequent key search and recovery, thereby partially or fully replacing classical distinguishers.

Figure 4 depicts the end-to-end pipeline, from differential data generation, via model training and inference, to distinguisher evaluation and KR validation. All of the above metrics are measured within this unified pipeline, thereby ensuring that the reported experimental results are interpretable, auditable, and reproducible.

A key evaluation challenge is that, regardless of the route taken, much prior work has relied on classification Accuracy as the primary evaluation metric. However, Accuracy fails to adequately measure a model's practical cryptanalytic efficacy. This metric, in particular, fails to sufficiently quantify its true contribution to key recovery. Therefore, to ensure rigor and comparability in evaluation, this paper adopts the following four standardized metrics under a unified audit framework:

- **Area Under the ROC Curve (AUC):** Evaluates the overall distinguishing capability of the distinguisher across all decision thresholds.
- **True Positive Rate (TPR) at low False Positive Rate (FPR):** Characterizes the effective signal strength provided by the distinguisher on the trail corresponding to the correct key, measured at the working point selected for the actual attack.
- **False Positive Rate (FPR):** At the same working point as the TPR, measures the probability that a wrong key is incorrectly flagged as "suspicious," characterizing the candidate key expansion and additional search overhead caused by the attack.
- **KR Success Rate (KR):** The probability of successfully recovering the correct key within given data and time budgets, used to measure the practical feasibility of the attack.

4.2.1. Deep Learning-Assisted Differential Cryptanalysis

In deep-learning-assisted differential cryptanalysis, the NN never acts as the distinguisher itself; instead it serves purely as an auxiliary engine—either to automate front-end data screening or to speed up back-end KR—while the final decision remains with a classical differential or statistical distinguisher. The guiding principle is "let learning assist analysis, not adjudicate": deep learning amplifies observable signals or reduces engineering complexity, but the ultimate verdict is returned to classical machinery. The immediate payoff is to turn the "difference-selection" problem into a fully data-driven, automated pipeline. In related-key settings, for instance, an exhaustive-plus-greedy search wrapped with Bayesian hyper-parameter optimisation can empirically single out low-Hamming-weight differentials without any manual prior, yielding a front-end preprocessing layer usable by any (classical) distinguisher. In short, "human searches for the differential" becomes "data chooses the differential," eliminating human bias and boosting both quality and reproducibility.The evolutionary directions of deep learning–assisted differential cryptanalysis are shown in Table 9.

Table 9. Evolutionary Directions of Deep Learning-Assisted Differential Cryptanalysis.

Year	Paper	Where Deep Learning Is Used	Distinguisher	Key Contribution/ Contextual Positioning
2019	[35]	Early Stage: Uses deep learning for fast searching/filtering of input differentials; while the main body of the paper also used the ND, we only consider the "differential selection" idea here.	Reusable Classic Differential	Introduced the action of "using learning to assist differential selection" into the process, providing a prototype for the subsequent shift of taking deep learning out of the role of a discriminator and using it for early-stage filtering.
2021	[42]	Methodology: Systematically dissects the "differential distribution and second-to-last-round information" learned by ND, and constructs a pure cryptanalytic classic discriminator that achieves an effect similar to ND.	Classic	Proved that the effectiveness of many NDs can be "reduced" to interpretable classic discrimination rules, providing theoretical support for "returning to classic discriminators + using deep learning only as an auxiliary tool."
2023	[43]	Late Stage: Proposed "Partial Incorrect Key Responses" to replace the full key profile, significantly reducing the complexity of posterior searches (verified on a large-state Simon).	Does not change the classic discrimination framework	Makes the late-stage key profiling/searching outside of deep learning "lighter," making large-state KR feasible—this is a representative example of "intelligentization" placed after the discrimination step.
2023	[44]	Late Stage/Statistical: Proposed "Deep Learning-Assisted Statistical Attacks," using learned signals to improve the effectiveness of classic statistical tests/scoring (without using deep learning as the final discriminator).	Classic Statistics	Clearly connects deep learning with classic statistics like χ^2/Likelihood, forming a closed loop where "deep learning provides features, and statistics make decisions."

Another significant direction within the assistive paradigm is the "learn–interpret–reconstruct" approach. In this framework, researchers initially utilize lightweight NNs or machine learning models—operating outside the distinguisher role—to extract interpretable features that are correlated with the differential distributions of the final few rounds or approximations of the DDT. These features are subsequently translated into classical statistical or analytical rules, which are then used by a traditional distinguisher for decision-making. A systematic study analyzed the success of Gohr's paradigm, revealing that its key insight lies not only in the final-round differentials but also in identifying statistical patterns in the penultimate or even earlier rounds [45]. Building on this insight, researchers developed a purely cryptanalytic classical distinguisher that closely matches the accuracy and efficiency of deep learning-based distinguishers, thus providing a solid theoretical foundation for the hybrid paradigm of "deep learning for feature extraction, classical methods for decision-making."

Moreover, in the late KR phase the auxiliary paradigm aims to cut profiling and search costs without touching the classical decision pipeline. Attacks on large-state SIMON show that "partial wrong-key responses" suffice to push 13-round KR, proving that the full profile is unnecessary and thus lowering end-stage complexity [43]. This engineering trick can be plugged directly into any classical distinguisher's KR flow: lightweight statistical evidence replaces the expensive complete profile, removing the time-and-data bottleneck. A parallel approach is the "neural-assisted statistical attack," where learned signals—possibly external to the ND—are injected into classical tests (weighted-likelihood or χ^2) to boost sensitivity and success rate. The mantra remains "the learned signal is evidence, not the judge," keeping the final verdict inside the traditional statistical framework [44].

From a methodological perspective, the assistive approach exhibits significant advantages as well as limitations. Its primary strength lies in high engineering controllability: it

avoids training large-scale NDs, thereby substantially reducing computational and data requirements. Moreover, because components such as differentials, neutral bits, and profiling remain grounded in classical cryptanalytic structures, this approach is more amenable to auditing, transferability, and reproducibility. More importantly, through interpretability research, the assistive paradigm can translate "black-box gains" into verifiable differential statistics, thereby reducing reliance on the model itself. However, the assistive approach also has limitations. Without incorporating an ND, it struggles to fully capture complex signals involving "cross-round multi-bit correlations," potentially capping its performance ceiling. Furthermore, if the initial filtering or subsequent weighting is implicitly coupled with the training set, evaluation bias may be introduced. In scenarios with a large number of rounds or extremely weak differential signals, the gains from a purely assistive approach may be insufficient, necessitating parallel or iterative use with stronger distinguishers (e.g., NDs)—though this would fall outside the scope of "purely assistive" methods. Overall, the essence of the assistive paradigm is "using learning to aid analysis, not to replace judgment." Its value lies in interpretability, scalability, and reproducibility, and it complements ND-driven approaches by offering an effective alternative for evaluation tasks involving large state spaces, cost-sensitive settings, or stringent interpretability requirements.

4.2.2. Deep Learning-Based Differential Cryptanalysis

Deep learning-based differential cryptanalysis, centered on NDs, has become a significant research direction in cryptanalysis. Gohr pioneered the use of deep learning to directly learn the ciphertext distribution under a given input differential, demonstrating that his ND significantly outperforms classical DDT-based distinguishers. This work established a standardized attack pipeline: first using the ND to effectively distinguish ciphertexts, then combining it with techniques such as Bayesian search to perform KR. Applied to lightweight block ciphers Speck32 and Speck64, this paradigm significantly improved both the number of attackable rounds and the remaining security margin [46]. Subsequent work analyzed the ND from an interpretability perspective and found that its advantage stems not merely from exploiting the final-round output differential, but from effectively approximating cross-round differential distribution information through training—thereby explaining its superiority over classical statistical methods [45]. This paradigm was subsequently validated across multiple ciphers: in reduced-round distinguishing attacks on TEA/RAIDEN, GIFT-COFB, GIFT-64, and PRIDE, deep learning-based NDs consistently outperformed traditional statistical methods, demonstrating the broad transferability of "learned differential distinguishers" [40,47,48].The evolutionary directions of deep learning-assisted differential cryptanalysis are shown in Table 10.

Table 10. Evolutionary Directions of Deep Learning-Based Differential Cryptanalysis (Theoretical).

Year	Paper	Core Point	Representative Result/Impact	Key Contribution
2019	[35]	A ResNet ND combined with Bayesian key search uses learned differential distributions directly for discrimination and KR.	Reduced the remaining security of 11-round Speck32/64 to the tens of bits range; established the standard "ND → KR" process.	The first use of a NN as a distinguisher to successfully perform KR, pioneering the entire ND research line.
2020	[49]	Fusing multiple differential pieces of information as input to the ND significantly improves distinguishing accuracy and the number of rounds it can withstand.	Raised the overall accuracy and attackable rounds for ND on ciphers like Simeck and Simon.	Established "multi-differential fusion" as a common method for improving ND, becoming a foundational element for subsequent work.

To further boost the signal-to-noise ratio and generalisation of ND, researchers have turned to multi-differential fusion and "differential modulation" techniques. The first step was the polytopic ND, which jointly encodes several input differences and feeds them to the network; on Simeck and Simon this markedly raised distinguishing accuracy and stability at high rounds and enabled deeper KR attacks—an early systematic attempt to fuse multiple differential signals [50]. Subsequently, the mixture ND increased robustness by blending differential formats. On SIMON32/64, for instance, mixture differentials lifted the 8-round distinguishing accuracy from 74.7% to 92.3% and, combined with round extension and neutral-bit techniques, pushed KR to 11–12 rounds with ≈55% success rate, showing that diversifying differential inputs greatly amplifies the learnable signal [51]. In parallel, the related-key scenario was addressed by the twin-differential related-key-ND, which feeds two distinct input differences through an automated four-stage attack pipeline. On Simon, Speck, Simeck and Hight it outperforms single-differential NDs in both accuracy and reachable round depth [52]. These gains align with interpretability studies indicating that NDs implicitly exploit multi-bit correlations induced by modular addition mod 2^n and the probability-propagation patterns under fixed key differences [45]. Thus the common goal of multi-differential fusion is to enhance attack robustness by treating the number and shape of differentials as tunable hyper-parameters that systematically strengthen the neural distinguisher [50–52].The evolutionary directions of deep learning-assisted differential cryptanalysis are shown in Table 11.

Table 11. Evolutionary Directions of Deep Learning-Based Differential Cryptanalysis (Technical).

Year	Paper	Core Point	Representative Result/Impact	Key Contribution
2023	[53]	Designed a "multi-round × multi-pair" data format for input into ND, which is equivalent to amplifying the observable differential signal.	Significantly improved accuracy and increased the number of rounds for Speck/Simon, making "large-sample stitching" a mainstream practice.	Solidified "data engineering" as the primary leverage for improving ND performance.
2024	[52]	ntroduced "two types of input differentials" to train ND in the related-key scenario, with an automated pipeline to improve generalization.	Compared to single-differential NDs, it provided more stable discrimination and recovery, supporting a higher number of rounds.	Extended ND from the known-key to the related-key scenario, and verified the transferability of the "multi-differential + automation" approach.
2025	[54]	Introduced multi-scale convolution to capture inter-round differential features, combined with engineering-oriented training.	Achieved a 12-round KR success rate of approximately 61.8% on Speck32/64 (a representative figure).	Provided reproducible experiments in the high-round range through "structural improvements + engineering refinements," marking the direction of recent State-of-the-Art.

In addition to optimizing the type of differential, researchers have also systematically enhanced the performance of NDs through data format engineering. A typical example is the MRMSP/MRMSD (Multi-Round Output + Multi-Pair Ciphertext Stitching) method, which constructs a 2D input using output features from the penultimate round and stitches together multiple pairs of ciphertexts or differentials to feed into the ND. This method has systematically improved ND accuracy and the number of analyzable rounds for Speck/Simon [53]. Further research has shown that using information-rich data formats, such as ciphertext pairs, differentials, or "derived features" of key/key differentials, can push ND performance to a higher level while maintaining a moderate network size. For instance, on Speck32/64, leveraging multi-scale CNNs with multi-pair/multi-

differential datasets achieved a 12-round KR success rate of approximately 61.8%, which strongly suggests that the importance of data can sometimes outweigh that of structural depth [54]. Furthermore, by adopting a lightweight data format—such as only 4-bit output differentials—and combining it with depthwise separable convolution, research has been able to maintain accuracy while halving the parameters. This provides convenience for training and reproduction in resource-constrained environments [55]. Therefore, in practice, it is often more effective to prioritize the meticulous design of data formats rather than blindly deepening networks, and then to use structural lightweighting during practical deployment to reduce computational and reproducibility barriers [53–55].

To extend deep learning-based differential cryptanalysis to more demanding scenarios, recent work has systematically explored its applicability to related-key settings and to large-state block ciphers (64/96/128 bits). Beyond the twin-differential related-key-ND [52], experiments show that an ND can actually outperform its single-key counterpart in related-key attacks—e.g., a 14-round KR on Speck32/64 [45]—demonstrating that the modulation of differential propagation is scenario-dependent and that the key difference introduced in related-key can sometimes enlarge the learnable signal [45,52]. Because Gohr's original framework does not scale directly to large states, researchers proposed a generic multi-ND grouping plus multi-stage KR architecture, coupled with an efficient neutral-bit search, enabling practical attacks on reduced-round Speck64/96/128 while drastically cutting time and data [56]. Subsequent parallelised multi-stage KR and partial-bit/segment-wise training further reduced time and training cost on Speck64/96/96 [57]. For large-state SIMON64/128, it was shown that only a "partial response" of the wrong-key profile is sufficient, an engineering trick that lowers end-phase complexity and proves that ND methods remain workable on large blocks [44]. Taken together, these studies show that the key to handling related-key and large-state complexities is to re-engineer the entire attack flow—automating differential selection, assembling NDs, and parallelising KR—so that scalability and reproducibility are preserved [43,45,52,57,58].

In terms of architectural improvement and cross-cipher generalization, researchers have focused on optimizing the internal mechanisms of NDs to more effectively learn cross-round dependencies. Under controlled training conditions, multi-scale convolution and dense residual connections have been introduced to capture multi-scale correlations in cross-round differentials, yielding consistent and stable marginal gains that, when combined with data engineering, translate into measurable improvements in attack rounds or success rates in reproducible experiments [54]. Through repeated experimentation and robust training techniques, researchers successfully constructed a 15-round ND for Simeck32/64—the longest attack at the time—demonstrating that engineering details are critical to the success of high-round attacks [56]. Moreover, attention mechanisms or Transformer architectures have been employed to model cross-round differentials, enabling the design of KR procedures that prioritize ciphertext pairs most influenced by the previous round. In attacks on Speck32/64, this approach reduced the key candidate space from 65,536 to fewer than 17, strategically reinforcing the value of "learning cross-round dependencies" [49]. The ND approach has also been widely extended to various ciphers: for example, a 7–8 round ResNet-based ND for LBlock (achieving 0.999 accuracy at 7 rounds) and an 11-round KR attack, along with empirical validations on TEA/RAIDEN, GIFT-COFB, GIFT-64, and PRIDE, all demonstrate the broad applicability of the ND paradigm [40,47,59]. A systematic comparison of CNNs and ResNets on SIMON and SPECK yielded the practical guideline: "use ResNet for SIMON, CNN for SPECK," offering valuable model selection guidance for future work [55]. Notably, NN-assisted statistical attacks convert ND outputs into weighted evidence for classical statistical tests, suggesting that a hybrid paradigm combining "learned signals with classical hypothesis testing" may be more robust in practice [44].

5. Quantum-Assisted Differential Cryptanalysis

5.1. Overview of Basic Theory and Methods

Differential Cryptanalysis, a powerful weapon in symmetric-key cryptanalysis, infers secret keys by exploiting statistical imbalances between pairs of plaintexts and their corresponding ciphertexts. This imbalance is quantified by the Differential Transfer Probability (DTP). For an m×m-bit S-box $S : \{0,1\}^m \to \{0,1\}^m$, the DTP is defined as in Equation (11).

$$\Pr_{x \leftarrow \{0,1\}^m}[S(x) \oplus S(x \oplus \delta) = \gamma] = \frac{1}{2^m} \sum_x \mathbf{1}[S(x) \oplus S(x \oplus \delta) = \gamma] \tag{11}$$

The probability of a differential trail $\Delta X \to \Delta Y$, given by $p_{trail} \approx \prod_{i=1}^{r} p_i$. trail , reflects this statistical bias. However, when targeting high-security cryptographic systems, finding high-probability differential trails and using them for KR remains a formidable challenge in terms of both data and time complexity.

The rise of quantum computing has opened a new cryptanalytic paradigm. In particular, Grover's algorithm delivers a quadratic speed-up for searching unstructured databases, offering a powerful accelerator for brute-force key search and for the enumeration steps that appear in many classical attacks. This paper explores how to marry quantum capabilities—especially quantum queries and amplitude estimation—with differential cryptanalysis to forge a highly efficient quantum differential attack. Rather than merely "quantising" a classical procedure, the new approach leverages intrinsically quantum advantages to re-engineer the core steps of differential analysis from the ground up.

In quantum cryptanalysis, the adversary's ability to query the cryptographic system is the pivotal factor that defines the attack model.

- Q1 model (classical query): The attacker can only perform classical plaintext–ciphertext queries, i.e., obtain pairs $(x, E_k(x))$. Quantum speedup is limited to offline processing of the collected data—for example, using Grover's algorithm to accelerate subkey search.
- Q2 model (quantum query): The attacker can query the encryption function E_k in quantum superposition, i.e., access a quantum encryption oracle implemented as a unitary operator $UE_k : |x, y\rangle \mapsto |x, y \oplus E_k(x)\rangle$. This capability enables the attacker to exploit quantum superposition and interference to directly probe the internal structure of the cipher.

Under the Q2 model, quantum computing demonstrates a significant advantage in the first stage of differential cryptanalysis—differential probability estimation.

Classically, estimating the probability p of a differential event requires a large number of samples, with sample complexity $M = \Theta(1/\epsilon^2)$. However, using Quantum Amplitude Estimation (AE), an attacker can efficiently estimate the success amplitude $\sqrt{p}$ in the quantum state $|\psi\rangle = \sqrt{p}|good\rangle + \sqrt{1-p}|bad\rangle$., reducing the sample complexity to $M = \Theta(1/\epsilon)$.This quadratic speedup enables the attacker to identify statistically biased, effective differential trails with significantly fewer data samples and in less time.

In Grover-based quantum KR, once an effective differential distinguisher is in hand the second phase uses it to hunt for the correct sub-key. Classically, the adversary would enumerate every candidate sub-key k_s and check which one is consistent with the observed differential pairs; if the sub-key space has size $S = 2^r$ this costs $\Theta(2^r)$ time.

Quantum computing provides revolutionary speedup for this stage through Grover's search algorithm. All possible subkeys can be viewed as an unstructured database. By constructing an oracle that identifies the "correct key," Grover's algorithm reduces the key search complexity from $\Theta(2^r)$ to $\Theta(2^r) = \Theta(2^{r/2})$.This quadratic speedup constitutes a core advantage of quantum differential cryptanalysis, enabling attackers to theoretically

recover the correct key with far fewer computational resources than classical methods. Quantum computing not only accelerates high-probability differential attacks but also provides new tools for attacks that exploit low-probability or zero-probability events. IDC uses differential trails with zero probability to eliminate incorrect keys. Within a quantum framework, we can use AE to estimate the probability of a differential event, thereby more efficiently verifying whether it is "zero" or "non-zero." This is significantly more efficient than classical methods, which require many more samples to rule out small probabilities.

Moreover, quantum primitives such as Simon's algorithm are intimately linked to structural attacks in cryptography. If a cipher function hides an XOR-period s, Simon's algorithm recovers s in the Q2 model with exponential advantage—only $O(n)$ queries. This capability exposes structural weaknesses, furnishing an unprecedented tool for building powerful distinguishers or shrinking the key-search space.

Overall, quantum computing enhances differential cryptanalysis in the following ways:

- Data complexity: Amplitude estimation significantly reduces the number of samples required to verify a differential characteristic.
- Time complexity: Grover's algorithm reduces the complexity of the most time-consuming step—subkey search—from exponential $\Theta(2^r)$ to square-root $\Theta(2^{r/2})$.

While the theoretical speed-ups are exciting, their practical feasibility hinges on several technical hurdles—most notably the implementation cost of the quantum oracle (i.e., the gate count and qubit overhead required to realise it as a quantum circuit) and the intrinsic decoherence and error-tolerance limits of quantum hardware. Nevertheless, the theoretical framework of quantum differential cryptanalysis has already charted a new course for future cryptographic security evaluations and urges designers to account for quantum-era robustness [60].

5.2. Evolution of Quantum-Assisted Differential Cryptanalysis

To clearly chart the developmental trajectory of quantum-assisted differential cryptanalysis, this paper adopts "algorithmic paradigms" as its organizing backbone, threading together three major lines—Grover-style differential key search, Simon/BV-style period and structural analysis, and the emerging class of "quantum-subroutine fusion." Within a unified notation system, we provide a continuous discussion of their respective advantages, limitations, and future directions. In summary, the Grover track primarily accelerates the back-end candidate screening and verification, yet its performance is fundamentally constrained by the need to build reversible oracles, the resulting high T-depth and ancilla requirements, and the classical data bounds inherited from differential analysis. The Simon/BV track drives the front-end discovery of structures or differential paths and the compression of candidate spaces, but it critically relies on superposition queries that are rarely attainable in practical models, and additional effort is often required to translate structural information into full KR. The fusion track, by modularly combining amplitude amplification, restricted-birthday techniques, and linear-structure solvers, yields a portable "structure $\rightarrow$ filter $\rightarrow$ verify" pipeline; however, its circuit depth and width can grow rapidly due to complex filtering logic and multi-stage amplification. The evolutionary directions and methodological characteristics of these three lines are summarized in Table 12.

Table 12. Quantum-Assisted Differential Cryptanalysis: Method Matrix.

Direction	Ideas	Targets	Complexity Gain	Queries	qRAM	Bottlenecks
Grover-based key search	Grover; quantum counting	AES, SKINNY, LWC, hashes	$\|K\|: O(\|K\|) \to O(\sqrt{\|K\|})$; $M: O(M) \to O(\sqrt{M})$	Mainly Q1; some Q2	Optional	Reversible oracles; T-depth/ancilla; data limits
Structural (Simon/BV)	Simon; BV	Even–Mansour, Feistel, CBC/PMAC/GCM/OCB	Exp $\to$ poly (with Q2)	Q2	Typically no	Access to superposition queries; link to KR
Hybrid diff + quantum	Amplitude amp.; bounded-birthday; BV	Impossible diff., zero-corr., boomerang	Per-stage $\sqrt{\cdot}$ speedups; overall $<$ classical baselines	Q1/Q2 (by task)	Varies	Circuit depth/width; filtering data

First, Grover-style differential key search centers on quantizing the three classical stages—counting, filtering, and verification—typically reducing the linear costs in candidate key space $|K^*|$ and pair size M to square-root complexity, thereby achieving systematic speedup under the same data budget [61]. Zhou et al, as shown in Table 13 [61] were the first to propose a general framework for "quantum differential cryptanalysis" and outlined reversible circuit implementations, explicitly stating that any differential-style procedure dominated by counting/filtering can achieve quadratic speedup; Kaplan et al. [62] further unified differential and linear cryptanalysis under the paradigms of quantum min/max search and quantum counting, emphasizing that "classically optimal does not imply quantumly optimal," and proposed criteria and applicability boundaries for evaluating quantumization benefits. At the engineering level, Denisenko [63] provided a proof-of-concept mapping for reduced block ciphers; Bonnetain et al. [63], under the Q1 model (classical queries with offline quantum processing) combined with qRAM, proposed quantum Square and quantum meet-in-the-middle frameworks for AES, quantifying resource metrics such as circuit depth and logical qubits, thereby enabling a direct trade-off between the gains of "structured search" and implementation costs. For structured variants, David and Naya-Plasencia [64] quantized the pairwise filtering in "impossible differential" attacks from $O(M)$ to $O(\sqrt{M})$, achieving KR advantages on AES-192/256 and SKINNY-128-256, and established a two-stage template: "restricted-birthday collection $\to$ quantum filtering"; on the hash side, Hosoyamada and Sasaki [65] quantized rebound attacks, showing that even "sub-birthday-bound" differential trails can be converted into specialized collision complexities better than the BHT baseline in the quantum setting, significantly broadening the applicability of Grover-style approaches; Xu et al. [66] further integrated "quantum differential + meet-in-the-middle" into a reusable framework, particularly well-suited for lightweight block ciphers. However, the core bottlenecks of this approach lie in engineering resources and model assumptions: on one hand, reversible S-boxes/linear layers and reversible filtering predicates significantly increase circuit depth and ancilla qubit requirements; Jaques et al.'s [67] systematic evaluation of AES/LowMC clearly revealed how the "depth × width" trade-off constrains practical feasibility; on the other hand, reliance on qRAM/QRA for structured data access introduces new assumptions and sources of uncertainty regarding availability and bandwidth [68]. Moreover, as Kaplan et al. [62] showed, not all differential variants reliably achieve quadratic speedup; thus, conditional criteria—centered on candidate density, filtering hit rate, and noise thresholds—are needed to identify quantumization pipelines that genuinely benefit. Accordingly, future work on the Grover line should target realistic models—such as "qRAM-free" or "limited superposition"—to develop "bounded-advantage" variants, complemented by online-adaptive

quantum counting to stabilize constant factors in hit rates, and further explore the quantum usability regime of "sub-birthday-bound trails" in hash functions and lightweight constructions; simultaneously, integrating Simon/BV-based structure discovery as a preprocessing step can further shrink $|K^*|$, thereby amplifying the square-root speedup in the subsequent stage. Under a unified notation, the criterion for "outperforming generic Grover exhaustive search (threshold $2^{k/2}$)" is that the total complexity—determined by $|K^*|$, filtering hit rate, and related factors—falls below $2^{k/2}$, thereby yielding a quantum advantage [61]; and in the hash collision setting, if the total complexity of a quantized differential-rebound (or related) procedure is below the BHT baseline $2^{n/3}$, it constitutes a specialized quantum advantage [65].

Table 13. Search-Driven (Grover/Counting) Comparisons.

Paper	Scenario Assumptions	Key Complexity	Quantum Resources	Conditions Beating Classical	Conditions Beating Generic Quantum
[61]	Known differentials; counting/filtering oracles available; `Q1`/`Q2` (`Q1` in common cases)	Time: $\mathcal{O}(\lvert K\rvert) \to \mathcal{O}(\sqrt{\lvert K\rvert})$ or $\mathcal{O}(M) \to \mathcal{O}(\sqrt{M})$; counting/lookup: same order as—or lower than—classical	Logical qubits = state + ancilla; depth set by S-boxes and counting/comparison	Large candidate/filter sets and reversible filtering is implementable	Outperforms Grover $\mathcal{O}(2^{k/2})$ when $\lvert K\rvert \ll 2^k$
[62]	Min/max search + counting; feasible trails; resources estimable	Differential stage enjoys quadratic speedup; truncation, etc., not guaranteed	Linear methods use fewer qubits than differential methods	Counting/filtering dominates, is quantizable, and success probability is not degraded	Better than plain Grover when the candidate set is small/hit rate is high
[68]	`Q1` + offline; qRAM commonly used; example: `AES-128/256` (8 rounds)	Under structured search, quantumized complexity beats classical; affected by qRAM bandwidth	qRAM (if used) + high-depth/width reversible AES oracle	Feasible qRAM/structured search with suitable parameters	Effective candidates $\ll 2^k$, or MITM + quantum filtering is superior
[64]	Two-stage: bounded birthday matching + quantum filtering; black-box setting	Filtering cut from $\mathcal{O}(M)$ to $\mathcal{O}(\sqrt{M})$; heavily optimized trail	Cost concentrated in reversible filtering/matching circuits	When filtering dominates and M is large, beats classical impossible-differential attacks	After heavy optimization, total complexity $< \mathcal{O}(2^{k/2})$
[65]	Differential collisions; bounded birthday subroutine; optional qRAM	Generating differential pairs $\approx \mathcal{O}(p^{-1/2})$; compared with BHT $\mathcal{O}(2^{n/3})$	Reversible S-box/hash; (optional) qRAM	Leverage multi-round trails below the birthday bound	If trail-based quantum collision cost $< 2^{n/3}$, it beats generic methods

Furthermore, Simon/BV-type period-finding and structural analysis can discover hidden periods or equivalent structures in polynomial time using Q2 superposition queries, as shown in Table 14. This transforms the classically exponential task of "path construction/structure distinguishing" into a scalable pre-processing step. This step can then be integrated with any search/filtering backend, creating a "structure → search" pipeline. Kuwakado–Morii [69] first constructed a polynomial-time quantum distinguisher for three-round Feistel ciphers. Following this, Kaplan et al. [70] and Santoli–Schaffner [71] extended attacks to include classically secure constructions like Even–Mansour and various MACs. These findings challenge the intuitive notion that symmetric cryptography can be made quantum-resistant simply by doubling the key length. In the more realistic Q1 scenario, the BV algorithm can act as a "linear structure extraction" pre-processor. It reduces the discovery of high-probability, impossible, or truncated differential paths

from a manual or exponential trial-and-error process to a polynomial-time task. This significantly lowers the cost of the initial phase of differential attacks and compresses the candidate key space $|K^*|$ [72]. On a theoretical level, Xiang et al. [73] established a bridge between "probability-1 truncated differentials ↔ Simon's distinguisher." Using tools like the difference-annihilation matrix, they unified the classical differential language within a quantum periodic framework, providing a rigorous foundation for cross-paradigm collaboration. Additionally, advancements have been made in quantum boomerang and quantum truncated differentials within this framework. The former offers a unified approach to quantizing boomerang attacks to achieve a quadratic speedup, which is suitable for various lightweight block ciphers [74]. Meanwhile, quantum slide attacks show how self-similar or periodic structures can be exploited and amplified by period-finding algorithms [75].Complementing this, Leander–May's [76] approach of "Simon for structure finding + Grover for searching FX" is a paradigmatic example of seamlessly connecting pre-processing structure discovery with a subsequent square-root search. This proves the feasibility of the two-stage "structure → search" paradigm under engineering evaluation. It is important to emphasize that the strongest conclusions are often based on the ideal model where Q2 (superposition queries) is available. When superposition queries are not feasible, a Q1-compatible alternative, such as BV/sampling + statistical inference, should be used to preserve polynomial-time structure discovery. Additionally, quantum counting/amplitude amplification can enhance the robustness of the structural signal. Finally, it is crucial to design reversible interface predicates to ensure a smooth transition to the subsequent search phase, as interface overhead could otherwise negate the benefits of the initial step.

Table 14. Structure-Driven (Simon/BV) Comparisons.

Paper	Assumptions	Complexity Gain	Quantum Resources	Beats Classical If	Beats Grover/BHT If
[69]	Q2 superposition queries; Simon period exists	Polytime distinguisher (vs. classical exponential)	Low–mid; reversible round-function oracle	Q2 allowed with valid Simon structure	Non-search (structure discovery) is polytime
[71]	Q2 model on EM/CBC-MAC-like structures	Polytime distinguish/forge	Depends on scheme; needs superposition queries	Simon-type period or linear structure present	Poly vs. exp, strictly better than generic quantum search
[72]	BV queries map linear structure to high-prob./impossible/ truncated diffs	Structure search in polytime; optional Grover filtering later	BV cost small; overall dominated by post-filtering	Path construction reduced from exp to poly	When post-filter candidate set $\ll 2^k$
[76]	Q2 finds structure; Grover shrinks key space	Two-stage (structure → search) reduces exponent	Two reversible oracles required	Usable structure signal and effective filtering	When remaining candidates $\ll 2^k$
[73]	Prob. 1 truncated diff convertible to Simon distinguisher	Unified view; polytime structure finding after bridging	Implementation-dependent	Bridging exposes polytime structure discovery	Same pipeline (structure → search) advantage

The "integrated quantum subroutine" approach modularly combines subroutines—such as amplitude amplification, restricted birthday sampling, and BV structure solving—within a task decomposition of "structure → filtering → verification," achieving simultaneous complexity reductions across multiple stages; its performance is jointly determined by the quality of front-end structural signals, filtering hit rate, and verification overhead, as shown in Table 15. Chen et al. [77] reformulated the hard-to-find characteristics in impossible differential and zero-correlation linear cryptanalysis as a "linear structure solving + verifiable filtering" problem: a front-end BV subroutine extracts one- or multi-dimensional linear

structures, followed by reversible predicate-driven filtering and verification, thereby extending the number of attackable rounds under the same data budget; the overall complexity is constrained by the circuit depth and ancilla requirements of reversible comparators and linear layers. Frixons et al. [74] proposed a "quantized boomerang" quartet construction that embeds "quartet generation + matching" into amplitude amplification, achieving $O(\sqrt{|K^*|})$ and $O(\sqrt{M})$ speedups in the search and filtering stages, respectively, when the differential trail is stable and the matching metric is reversible. Hosoyamada and Aoki [78] introduced the CEA family of methods based on multidimensional zero-correlation linear and integral properties, demonstrating that quantum structure extraction is not limited to hidden periods; when sufficient composable linear or integral constraints are available, potential improvements beyond naive square-root speedups become possible.

Table 15. Hybrid Differential–Quantum Comparisons.

Paper	Assumptions	Key Complexity	Quantum Resources	Beats Classical If	Beats Grover/BHT If
[77]	BV is feasible; search is converted into a linear structure/verifiable filter	Structural search is polynomial; number of attack rounds can be increased	Low BV cost; verification/filtering depends on the specific object	Automated search replaces exponential enumeration	If subsequent key candidates are much less than 2^k
[64]	Limited birthday attack + quantum filtering; black-box model	Filtering is reduced from $O(M)$ to $O(\sqrt{M})$	Filtering circuit; gate depth is determined by the matcher	When M is large and filtering is dominant	The entire optimized process is less complex than Grover's exhaustive search
[74]	Quantizable Boomerang structure; a match metric can be set	General search is converted to structured amplification; exponential decrease in complexity	Combination of two- or multi-stage Oracles	Boomerang paths are stable and have a high match rate	When the number of effective candidates is much less than 2^k or collision complexity is less than $2^{n/3}$

For the quantum part, we adopt a unified operational criterion of "being below the Grover baseline". Specifically, an attack is regarded as providing a practical quantum advantage in the KR setting only when its end-to-end key-recovery complexity is strictly less than $2^{k/2}$. Analogously, in the collision setting, the overall complexity must be below $2^{n/3}$ (the BHT baseline). Jaques et al. [67] attribute the main constraints of this route to engineering resources and statistical robustness: first, chaining subroutines inflates the product of depth and width, as well as the number of ancilla qubits, with reversible S-boxes, linear layers, and interface predicates constituting the principal contributors to gate depth; without reuse or parallel scheduling, local gains are often offset by global resource overhead. Hosoyamada–Sasaki [65] provide the second limitation: the achievable advantage is sensitive to the differential probability p and the quartet probability q, requiring quantum counting and adaptive amplitude amplification to stabilize the success-amplitude constant a. Consequently, targets must satisfy the baseline inequalities $< 2^{k/2}$ for KR and $<2^{n/3}$ for collisions. To avoid overly optimistic evaluations that consider only query complexity while ignoring actual hardware costs, this paper explicitly states the resource assumptions required for any claimed quantum advantage: whether qRAM is permitted, which oracle model (Q1 or Q2) is used, the required circuit depth, and the number of ancilla qubits. All comparisons are made within a unified resource-accounting framework.

In summary, given a fixed data budget and explicit resource assumptions (qRAM availability, Q1/Q2 oracle model, and upper bounds on circuit depth and ancilla qubits),

the effectiveness criterion for the fusion route is stated as follows. Let $|K^*|$ be the effective candidate-key set produced by front-end structure extraction, and let M be the number of pairs, as in Equation (12). A practical quantum advantage is achieved whenever

$$\mathrm{Cost}_{\mathrm{struct}} + \mathrm{Cost}_{\mathrm{filter}} + \mathrm{Cost}_{\mathrm{verify}} = \widetilde{\mathcal{O}}\left(\sqrt{|K^*|} + \sqrt{M}\right) \; < \; 2^{k/2}, \qquad \text{(KR)} \qquad (12)$$

where $\widetilde{\mathcal{O}}(\cdot)$ suppresses polylogarithmic factors, and k and n denote the key length and hash output size, respectively. Analogously, in the collision setting, the requirement is $\widetilde{\mathcal{O}}(\sqrt{M}) < 2^{n/3}$, which corresponds to the BHT baseline.

In summary, the Grover track retains a robust role as a "back-end accelerator" whenever candidates have already been narrowed by structural analysis, filtering dominates the cost, and reversible predicates are practical; the Simon/BV track delivers an intrinsic "polynomial-time" advantage in the structure-/path-discovery phase, especially when Q2 is available or BV can be substituted; the "new-subroutine fusion" line achieves transferable speed-ups across objects and models by task decomposition and subroutine reordering, as shown in Table 16. All three tracks share common bottlenecks in engineering resources and model assumptions; future work must therefore provide engineering paradigms that deliver limited yet concrete advantages under realistic no-qRAM/restricted-superposition models, unify distinguishing inequalities and resource metrics, and employ automated/intelligent parameter adaptation and path search to improve stability and reproducibility—thereby forging a generalisable, evaluable, and deployable methodological landscape for quantum differential cryptanalysis on FX/generalised-Feistel, hash, and lightweight constructions.

Table 16. Quantum-assisted Differential Cryptanalysis: From Theory to Implementation and Optimization.

Direction	Theoretical Foundations (Key Points)	Implementation & Validation (Key Points)	Optimization & Improvements (Key Points)	Diversified Development (Key Points)
Grover-based differential key search	Quantize counting/filtering/verification; $\sqrt{\cdot}$-speedup becomes dominant; generic counting/extremum-search framework	Build reversible oracles on block ciphers and complete PoC; run structured pipelines (e.g., MITM) [79]	Replace pairwise filtering/impossible differentials with amplitude amplification; tune samples and thresholds under fixed data	Extend to hashing (boomerang) and related-key; evaluate engineering limits with qRAM/parallelism [80]
Simon periodic/structural analysis (incl. BV)	Simon (Q2) finds periods/equivalent structures in polynomial time; BV (Q1) extracts linear structure to generate differential/impossible/truncated trails	Distinguish/forge on Even–Mansour, generalized Feistel, and MACs; plug into subsequent KR [81,82]	Use Simon/BV to shrink $\|K\|$, then apply Grover filtering; when Q2 is unavailable, approximate with BV	Bridge truncated differentials and Simon distinguishers; modularly combine with Grover/boomerang; port to lightweight/compound designs [83]
Fusion of new quantum subroutines	Introduce amplitude amplification, bounded-birthday, and linear-structure solvers; form a "structure $\rightarrow$ filter $\rightarrow$ verify" pipeline	Show end-to-end gains via "separately quantizing sub-tasks" in impossible differential, zero-correlation linear, and boomerang settings	Selectively quantize dominating terms; reorder schedule/parallelism under depth $\times$ qubit constraints	Use AI/automated search to mine trails; offer constrained-advantage variants without qRAM/with limited superposition [84]

6. Conclusions

In this survey, we provided a structured overview of differential cryptanalysis for block ciphers across four methodological paradigms: classical differential cryptanalysis, deep learning-assisted differential cryptanalysis, deep learning-based differential cryptanalysis, and emerging quantum-based approaches. On the classical side, we summarized how modern MILP/SMT/SAT formulations automate the search for high-probability differential characteristics and bounds, substantially reducing the amount of manual trail engineering required. For deep learning–related paradigms, we clarified how neural networks can be integrated either as auxiliary scoring modules or as primary distinguishers, and we highlighted the key design choices in data generation, model architecture, and evaluation. For quantum-based directions, we reviewed the main ideas behind amplitude amplification and quantum search routines as they relate to differential characteristic search and KR.

The main findings of this survey have several practical implications. First, automated constraint-based models (MILP/SMT/SAT) offer transparent and reproducible baselines for evaluating the differential security of block ciphers, and their outputs can be directly used to guide later deep learning or quantum procedures. Second, deep learning-assisted and deep learning-based attacks provide a flexible way to exploit large synthetic datasets and to capture subtle statistical deviations that are difficult to model analytically, but they require carefully documented protocols for data generation, labeling, and model training to remain reproducible. Third, while quantum-based proposals currently remain largely theoretical, they indicate where quantum resources could most plausibly reduce the cost of characteristic search or specific KR subroutines.

At the same time, this work has several limitations. Our discussion is centered on representative block ciphers and typical attack settings, and does not exhaustively cover all primitives, modes of operation, or adversarial models encountered in practice. The deep learning experiments and architectures we summarize are restricted to a small number of standard feedforward and convolutional models and do not fully explore more recent architectures, regularization schemes, or side-channel scenarios. The quantum part is constrained by the current maturity of quantum hardware and by the simplifying assumptions made in many algorithmic proposals. These limitations can be mitigated in future work by extending the automated modeling and experimental pipelines to a broader cipher portfolio, by releasing standardized open-source implementations and datasets for deep learning-based distinguishers and KR, and by validating quantum-inspired designs in realistic noise models and hybrid classical–quantum settings.

Overall, by organizing the existing results into a unified framework and by emphasizing reproducibility and comparability across methods, this survey is intended to serve as a reference for both practitioners and researchers.

7. Future Outlook

Building on the above conclusions, we now outline several promising directions for future research. With the continuous development of cryptanalysis techniques, differential attacks have gradually evolved from initial theoretical path derivation based on symbolic computation to automatic feature learning based on deep learning. Currently, a new trend toward quantum computing-assisted analysis is emerging. Against this background, constructing a hybrid differential attack framework that integrates the three paradigms—symbolic computation, deep learning, and quantum computing—has become an important direction for future research, as shown in Figure 5.

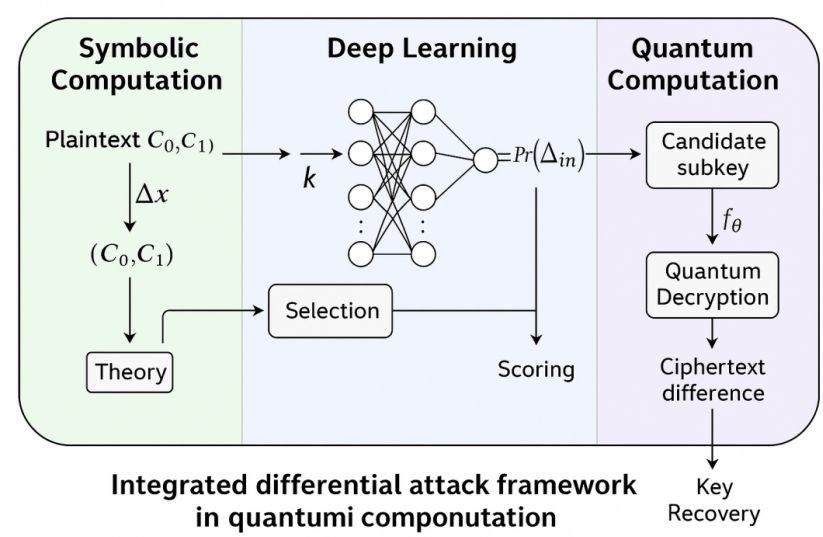

Figure 5. Diagram of the Hybrid Differential Attack Framework.

As differential cryptanalysis of block ciphers evolves, its methodological spectrum is expanding from classical differential cryptanalysis to two deep learning–integrated paradigms—Deep Learning-Assisted Differential Cryptanalysis and Deep Learning-Based Differential Cryptanalysis—and further exploring the potential quantum speedup in differential characteristic search and KR, as shown in Table 17. In comparison: classical methods excel in interpretability and controllability but incur high costs in trail construction and search; Deep Learning-Assisted Differential Cryptanalysis embeds deep learning as a feature extractor or scorer within traditional pipelines, balancing interpretability and efficiency; Deep Learning-Based Differential Cryptanalysis uses the model as the core component for end-to-end discrimination or scoring, offering greater automation and scalability but remaining constrained by data quality, reproducibility, and generalization; quantum approaches exhibit theoretical complexity advantages but are currently limited by algorithmic assumptions and hardware constraints.

Table 17. Comparison of Directions in Differential Cryptanalysis.

Direction	Theoretical Foundations (Key Points)	Implementation & Validation (Key Points)	Optimization & Improvements (Key Points)
Classical Differential Cryptanalysis	Hand-crafted differential characteristics/trails and probability propagation.	Symbolic/constraint-based search; MILP/SAT models for trail discovery.	Heuristic pruning, meet-in-the-middle, branch-and-bound.
	Trail counting, Markov assumptions, and statistical distinguishers.	Empirical verification on reduced-round ciphers; complexity accounting.	Improved objective functions to rank trails/boomerangs.
	KR via partial subkey hypotheses and filtering.	Robust baselines: transparent, reproducible, interpretable.	Faster MILP/SAT encodings; automated reporting.
Deep Learning-Assisted Differential Cryptanalysis	NNs as scoring/feature modules within classical pipelines.	Supervised or weakly-supervised training with synthetic labeled pairs.	Curriculum/data caching to stabilize training; hard-negative mining.
	Learning-guided trail search and pair selection under classical priors.	Plug-and-play with classical distinguishers and key filters.	Uncertainty-aware scores to reduce false positives in key filtering.
	Retains classical interpretability anchors with data-driven cues.	Reproducible protocols: public seeds, fixed splits, stable metrics (AUC/TPR).	Joint objectives with classical costs (multi-task or constrained training).

Table 17. *Cont.*

Direction	Theoretical Foundations (Key Points)	Implementation & Validation (Key Points)	Optimization & Improvements (Key Points)
Deep Learning-Based Differential Cryptanalysis	NN as the primary end-to-end discriminator/scorer.	Large-scale training (GPU/TPU); attention/CNN/MLP variants.	Data efficiency via self-supervision, contrastive learning, augmentation.
	Representation learning for trail/pair quality without explicit models.	End-to-end pipelines for scoring, ranking, and KR guidance.	Robustness: adversarial training, calibration, ensembling.
	Generalization under distribution shift across rounds/keys.	Stress tests: cross-cipher transfer, different S-boxes/rotations/noise.	Interpretability: saliency/tracing aligned with differential semantics.
Quantum-Based Differential Cryptanalysis	Superposition-based evaluation and amplitude amplification.	NISQ-era feasibility studies: simulators plus small-scale hardware demos.	Error mitigation, resource estimation, and qRAM-free designs.
	Potential speedups for characteristic search and KR subroutines.	Oracle engineering and reversible implementations of cipher components.	Hybrid classical–quantum splits to minimize quantum depth.
	Complexity bounded by oracle access and circuit depth/width constraints.	Validation via asymptotic analysis and empirical noise profiling.	Reformulations amenable to Grover-style or QAOA subroutines.

Author Contributions: Conceptualization, L.Z., Y.W. (Yvxuan Wu) and C.X.; investigation, C.X.; resources, L.Z.; data curation, L.Z., Y.W. (Yvxuan Wu) and Q.L.; writing—original draft preparation, L.Z., Y.W. (Yvxuan Wu) and C.X.; writing—review and editing, C.X., Y.W. (Yaxuan Wen) and D.D.; visualization, D.D.; supervision, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: Supported by State Key Lab of Processors, Institute of Computing Technology, CAS under Grant No. CLQ202516.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

KR	Key-Recovery
NN	Neural Network
BV	Bernstein-Vazirani
DDT	The Differential Distribution Table
IDC	Impossible-Differential Cryptanalysis
BCT	Boomerang Connectivity Table
CNN	Convolutional Neural Network
ND	Neural Distinguisher
MRMSP/MRMSD	Multi-Round Output + Multi-Pair Ciphertext Stitching)

References

1. Dong, X.Y. A survey of quantum cryptanalysis of symmetric cryptography. *J. Cryptologic Res.* **2024**, *11*, 159–173. [CrossRef]
2. Liang, M.; Luo, Y.-Y.; Liu, F.-M. A Survey on Quantum-Secure Symmetric Cryptography. *J. Cryptologic Res.* **2021**, *8*, 925–947. [CrossRef]
3. Wang, C.; Yao, H.-N.; Wang, B.-N.; Hu, F.; Zhang, H.-G.; Ji, X.-M. Progress in Quantum-Computing-Based Cryptographic Attacks. *Chin. J. Comput.* **2020**, *43*, 1691–1707. [CrossRef]

4. Zhan, Y.-J.; Guan, J.; Ding, L.; Zhang, Z.Y. Related-key Impossible Differential Attack on Reduced Round LBlock. *J. Electron. Inf. Technol.* **2012**, *34*, 2161–2166. [CrossRef]
5. Biham, E.; Shamir, A. Differential cryptanalysis of DES-like cryptosystems. *J. Cryptol.* **1991**, *4*, 3–72. [CrossRef]
6. Knudsen, L.R. Truncated and Higher Order Differentials. In *Fast Software Encryption: Second International Workshop, Leuven, Belgium, 14–16 December 1994, Proceedings*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1994; Volume 1008, pp. 196–211. [CrossRef]
7. Wang, C.B.; Zhang, Z.Z.; Hu, L. Differential Cryptanalysis on Ultra Lightweight Block Cipher PICO. *J. Cryptologic Res.* **2023**, *10*, 685–701. [CrossRef]
8. Biham, E.; Biryukov, A.; Shamir, A. Miss in the Middle Attacks on IDEA and Khufu. In *Fast Software Encryption. FSE 1999*; Knudsen, L., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1636. [CrossRef]
9. Biham, E.; Biryukov, A.; Shamir, A. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In *Advances in Cryptology—EUROCRYPT '99. EUROCRYPT 1999*; Stern, J., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1592. [CrossRef]
10. Wagner, D. The Boomerang Attack. In *Fast Software Encryption. FSE 1999*; Knudsen, L., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1636. [CrossRef]
11. Biham, E.; Dunkelman, O.; Keller, N. New Results on Boomerang and Rectangle Attacks. In *Fast Software Encryption. FSE 2002*; Daemen, J., Rijmen, V., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2365. [CrossRef]
12. Cid, C.; Huang, T.; Peyrin, T.; Sasaki, Y.; Song, L. Boomerang Connectivity Table: A New Cryptanalysis Tool. In *Advances in Cryptology—EUROCRYPT 2018. EUROCRYPT 2018*; Nielsen, J., Rijmen, V., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 10821. [CrossRef]
13. Kidmose, A.B.; Tiessen, T. A Formal Analysis of Boomerang Probabilities. *IACR Trans. Symmetric Cryptol.* **2022**, *2022*, 88–109. [CrossRef]
14. Biryukov, A.; Velichkov, V.; Le Corre, Y. Automatic Search for the Best Trails in ARX: Application to Block Cipher Speck. In *Fast Software Encryption. FSE 2016*; Peyrin, T., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9783. [CrossRef]
15. Fu, K.; Wang, M.; Guo, Y.; Sun, S.; Hu, L. MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck. In *Fast Software Encryption. FSE 2016*; Peyrin, T., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9783. [CrossRef]
16. Azimi, S.A.; Ranea, A.; Salmasizadeh, M.; Mohajeri, J.; Aref, M.R.; Rijmen, V. A bit-vector differential model for the modular addition by a constant and its applications to differential and impossible-differential cryptanalysis. *Des. Codes Cryptogr.* **2022**, *90*, 1797–1855. [CrossRef] [PubMed]
17. Biham, E.; Dunkelman, O.; Keller, N. Related-Key Boomerang and Rectangle Attacks. In *Advances in Cryptology—EUROCRYPT 2005. EUROCRYPT 2005*; Cramer, R., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3494. [CrossRef]
18. Lai, X.; Massey, J.L.; Murphy, S. Markov Ciphers and Differential Cryptanalysis. In *Advances in Cryptology—EUROCRYPT '91*; Davies, D.W., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1991; Volume 547, pp. 17–38. [CrossRef]
19. Knudsen, L.R.; Berson, T.A. Truncated differentials of SAFER. In *Fast Software Encryption. FSE 1996*; Gollmann, D., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1039. [CrossRef]
20. Lai, X. Higher Order Derivatives and Differential Cryptanalysis. In *Communications and Cryptography*; Blahut, R.E., Costello, D.J., Maurer, U., Mittelholzer, T., Eds.; The Springer International Series in Engineering and Computer Science; Springer: Boston, MA, USA, 1994; Volume 276. [CrossRef]
21. Langford, S.K.; Hellman, M.E. Differential-Linear Cryptanalysis. In *Advances in Cryptology—CRYPTO '94. CRYPTO 1994*; Desmedt, Y.G., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1994; Volume 839. [CrossRef]
22. Blondeau, C.; Leander, G.; Nyberg, K. Differential-Linear Cryptanalysis Revisited. *J. Cryptol.* **2017**, *30*, 859–888. [CrossRef]
23. Lu, J.Y.; Liu, G.Q.; Xiong, L.Y.; Sun, B.; Li, C. Differential Cryptanalysis of Lightweight Block Cipher PFP. *J. Cryptologic Res.* **2024**, *11*, 1293–1307. [CrossRef]
24. Li, Y.J.; Li, Y.S.; Yang, M.H.; Zhang, L.X.; Liu, J. Differential Attack on Lightweight PFP Algorithm. *Comput. Eng. Appl.* **2023**, *59*, 296–302. [CrossRef]
25. Biham, E.; Dunkelman, O.; Keller, N. The Rectangle Attack—Rectangling the Serpent. In *Advances in Cryptology—EUROCRYPT 2001. EUROCRYPT 2001*; Pfitzmann, B., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2045. [CrossRef]
26. Nyberg, K. Differentially Uniform Mappings for Cryptography. In *Advances in Cryptology—EUROCRYPT '93*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1993; Volume 765, pp. 55–64. [CrossRef]

27. Du, X.; Yu, T.; Jia, M.; Liang, L. Impossible differential cryptanalysis of reduced-round μ^2 algorithm based on matrix method. *J. Commun.* **2024**, *45*, 106–113. [CrossRef]
28. Dong, X.; Shang, S.; Chen, J. Impossible Differential Attacks on 9-Round Block Cipher Rijndael-192. *Netinfo Secur.* **2020**, *20*, 40–46.
29. Li, T.; Sun, Y. SuperBall: A new approach for MILP modelings of Boolean functions. *IACR Trans. Symmetric Cryptol.* **2022**, *2022*, 341–367. [CrossRef]
30. Jiang, Z.L.; Jin, C.H. Multiple Impossible Differentials Cryptanalysis on 8-round Kiasu-BC. *J. Cryptologic Res.* **2018**, *5*, 359–367. [CrossRef]
31. Tang, X.-H.; Li, C.; Wang, M.-Y.; Qu, L.-J. Impossible Differential Attack on 3D Cipher. *J. Electron. Inf. Technol.* **2010**, *32*, 2516–2520. [CrossRef]
32. Yan, Z.; Wei, Y.; Ye, T. Related-key Differential Cryptanalysis of Full-round PFP Ultra-lightweight Block Cipher. *J. Electron. Inf. Technol.* **2025**, *47*, 729–738. [CrossRef]
33. Nitaj, A.; Rachidi, T. Applications of Neural Network-Based AI in Cryptography. *Cryptography* **2023**, *7*, 39. [CrossRef]
34. Bellini, E.; Rossi, M. Performance Comparison Between Deep Learning-Based and Conventional Cryptographic Distinguishers. In *Intelligent Computing*; Arai, K., Ed.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2021; Volume 285. [CrossRef]
35. Gohr, A. Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning. In *Advances in Cryptology—CRYPTO 2019. CRYPTO 2019*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; pp. 150–179.
36. Tian, W.; Hu, B. Deep Learning Assisted Differential Cryptanalysis for the Lightweight Cipher SIMON. *KSII Trans. Internet Inf. Syst.* **2021**, *15*, 600–616. [CrossRef]
37. Su, H.-C.; Zhu, X.-Y.; Ming, D. Polytopic Attack on Round-Reduced Simon32/64 Using Deep Learning. In *Information Security and Cryptology. Inscrypt 2020*; Wu, Y., Yung, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12612. [CrossRef]
38. Bellini, E.; Gerault, D.; Grados, J.; Makarim, R.H.; Peyrin, T. Boosting differential-linear cryptanalysis of ChaCha7 with MILP. *IACR Trans. Symmetric Cryptol.* **2023**, *2023*, 189–223. [CrossRef]
39. Sun, L.; Wang, W.; Wang, M. Accelerating the Search of Differential and Linear Characteristics with the SAT Method. *IACR Trans. Symmetric Cryptol.* **2021**, *2021*, 269–315. [CrossRef]
40. Rajan, R.; Roy, R.K.; Sen, D.; Mishra, G. Deep Learning-Based Differential Distinguisher for Lightweight Cipher GIFT-COFB. In *Machine Intelligence and Smart Systems*; Agrawal, S., Gupta, K.K., Chan, J.H., Agrawal, J., Gupta, M., Eds.; Algorithms for Intelligent Systems; Springer: Singapore, 2022. [CrossRef]
41. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **1980**, *36*, 193–202. [CrossRef]
42. Benamira, A.; Gerault, D.; Peyrin, T.; Tan, Q.Q. A Deeper Look at Machine Learning-Based Cryptanalysis. In *Advances in Cryptology—EUROCRYPT 2021. EUROCRYPT 2021*; Canteaut, A., Standaert, F.X., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12696. [CrossRef]
43. Hou, Z.; Ren, J.; Chen, S. Practical Attacks of Round-Reduced SIMON Based on Deep Learning. *Comput. J.* **2023**, *66*, 2517–2534. [CrossRef]
44. Chen, Y.; Shen, Y.; Yu, H. Neural-Aided Statistical Attack for Cryptanalysis. *Comput.J.* **2022**, *66*, 2480–2498. [CrossRef]
45. Bao, Z.; Lu, J.; Yao, Y.; Zhang, L. More Insight on Deep Learning-Aided Cryptanalysis. In *Advances in Cryptology—ASIACRYPT 2023. ASIACRYPT 2023*; Guo, J., Steinfeld, R., Eds.; Lecture Notes in Computer Science; Springer: Singapore, 2023; Volume 14440. [CrossRef]
46. Bao, Z.; Guo, J.; Liu, M.; Ma, L.; Tu, Y. Enhancing Differential-Neural Cryptanalysis. In *Advances in Cryptology—ASIACRYPT 2022. ASIACRYPT 2022*; Agrawal, S., Lin, D., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13791. [CrossRef]
47. Mishra, G.; Pal, S.K.; Krishna Murthy, S.V.S.S.N.V.G.; Prakash, I.; Kumar, A. Deep Learning-Based Differential Distinguisher for Lightweight Ciphers GIFT-64 and PRIDE. In *Machine Intelligence and Smart Systems*; Agrawal, S., Gupta, K.K., Chan, J.H., Agrawal, J., Gupta, M., Eds.; Algorithms for Intelligent Systems; Springer: Singapore, 2022. [CrossRef]
48. Baksi, A.; Breier, J.; Chen, Y.; Dong, X. Machine Learning Assisted Differential Distinguishers For Lightweight Ciphers. In Proceedings of the 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 1–5 February 2021; pp. 176–181. [CrossRef]
49. Xiao, C.; Li, Z.; Zhang, L.; Wang, J.; Qian, S. Differential Cryptanalysis Based on Transformer Model and Attention Mechanism. *Comput. Eng.* **2025**, *51*, 156–163. [CrossRef]
50. Fu, C.-H.; Duan, M.; Wei, Q.; Wu, Q.-Q.; Zhou, R.; Su, H.-C. Polytopic Differential Attack Based on Deep Learning and Its Application. *J. Cryptologic Res.* **2021**, *8*, 591–600. [CrossRef]
51. Wu, Z.; Qiao, K.; Wang, Z.; Cheng, J.; Zhu, L. Mixture Differential Cryptanalysis on Round-Reduced SIMON32/64 Using Machine Learning. *Mathematics* **2024**, *12*, 1401. [CrossRef]

52. Wang, G.; Wang, G.; Sun, S. A New (Related-Key) Neural Distinguisher Using Two Differences for Differential Cryptanalysis. *IET Inf. Secur.* **2024**, *2024*, 4097586. [CrossRef]
53. Liu, J.; Ren, J.; Chen, S.; Li, M. Improved neural distinguishers with multi-round and multi-splicing construction. *J. Inf. Secur. Appl.* **2023**, *74*, 103461. [CrossRef]
54. Hou, Y.; Liu, J.; Han, S.; Ma, Z.; Ye, X.; Nie, X. Improving deep learning-based neural distinguisher with multiple ciphertext pairs for Speck and Simon. *Sci. Rep.* **2025**, *15*, 13696. [CrossRef] [PubMed]
55. Liu, J.S.; Ren, J.J.; Chen, S.Z. A deep learning aided differential distinguisher improvement framework with more lightweight and universality. *Cybersecurity* **2023**, *6*, 47. [CrossRef]
56. Lyu, L.; Tu, Y.; Zhang, Y. Improving the Deep-Learning-Based Differential Distinguisher and Applications to Simeck. In Proceedings of the 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hangzhou, China, 4–6 May 2022; pp. 465–470. [CrossRef]
57. Huang, T.; Li, Y.; Fu, Q.; Chen, Y.; Song, L. Improving Differential-Neural Cryptanalysis for Large-State SPECK. In Proceedings of the International Conference on Information and Communications Security, Mytilene, Greece, 26–28 August 2024; Springer Nature: Singapore, 2024; pp. 40–57.
58. Wang, G.; Wang, G.; He, Y. Improved Machine Learning Assisted (Related-key) Differential Distinguishers For Lightweight Ciphers. In Proceedings of the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 20–22 October 2021; pp. 164–171. [CrossRef]
59. Yang, X.; Li, K.; Du, X.; Liang, L.; Jia, M. Security Analysis of LBlock and Its Application Based on Deep Learning. *J. Electron. Inf. Technol.* **2023**, *45*, 3745–3751. [CrossRef]
60. Li, H.; Chen, W.; Huang, J.; Yao, Y.; Liu, D.; Li, F.; Wang, S.; Yin, Z.; He, D.; Zhou, Z.; et al. Security of quantum key distribution. *Sci. Sin.-Phys. Mech. Astron.* **2012**, *42*, 1237–1255. (In Chinese)
61. Zhou, Q.; Lu, S.; Zhang, Z.; Sun, J. Quantum differential cryptanalysis. *Quantum Inf. Process.* **2015**, *14*, 2101–2109. [CrossRef]
62. Kaplan, M.; Leurent, G.; Leverrier, A.; Naya-Plasencia, M. Quantum differential and linear cryptanalysis. *arXiv* **2015**, arXiv:1510.05836. [CrossRef]
63. Denisenko, D.V. Quantum differential cryptanalysis. *J. Comput. Virol. Hacking Tech.* **2022**, *18*, 3–10. [CrossRef]
64. David, N.; Naya-Plasencia, M.; Schrottenloher, A. Quantum impossible differential attacks: Applications to AES and SKINNY. *Des. Codes Cryptogr.* **2024**, *92*, 723–751. [CrossRef]
65. Hosoyamada, A.; Sasaki, Y. Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 10–14 May 2020; Springer International Publishing: Cham, Switzerland, 2020; pp. 249–279.
66. Xu, Y.S.; Cai, B.B.; Yuan, Z.; Qin, S.J.; Gao, F.; Wen, Q.Y. Quantum Differential Meet-In-The-Middle Attack and Some Applications to Lightweight Ciphers. *Adv. Quantum Technol.* **2024**, *7*, 2400157. [CrossRef]
67. Jaques, S.; Naehrig, M.; Roetteler, M.; Virdia, F. Implementing Grover oracles for quantum key search on AES and LowMC. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 10–14 May 2020; Springer International Publishing: Cham, Switzerland, 2020; pp. 280–310.
68. Bonnetain, X.; Naya-Plasencia, M.; Schrottenloher, A. Quantum security analysis of AES. *IACR Trans. Symmetric Cryptol.* **2019**, *2019*, 55–93. [CrossRef]
69. Kuwakado, H.; Morii, M. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In Proceedings of the 2010 IEEE International Symposium on Information Theory (ISIT), Austin, TX, USA, 13–18 June 2010; pp. 2682–2685.
70. Kaplan, M.; Leurent, G.; Leverrier, A.; Naya-Plasencia, M. Breaking symmetric cryptosystems using quantum period finding. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 207–237.
71. Santoli, T.; Schaffner, C. Using Simon's algorithm to attack symmetric-key cryptographic primitives. *arXiv* **2016**, arXiv:1603.07856. [CrossRef]
72. Xie, H.; Yang, L. Using Bernstein–Vazirani algorithm to attack block ciphers. *Des. Codes Cryptogr.* **2019**, *87*, 1161–1182. [CrossRef]
73. Xiang, Z.; Wang, X.; Yu, B.; Sun, B.; Zhang, S.; Zeng, X.; Shen, X.; Li, N. Links between quantum distinguishers based on Simon's algorithm and truncated differentials. *IACR Trans. Symmetric Cryptol.* **2024**, *2024*, 296–321. [CrossRef]
74. Frixons, P.; Naya-Plasencia, M.; Schrottenloher, A. Quantum boomerang attacks and some applications. In Proceedings of the International Conference on Selected Areas in Cryptography, Virtual, 29 September–1 October 2021; Springer International Publishing: Cham, Switzerland, 2021; pp. 332–352.
75. Bonnetain, X.; Naya-Plasencia, M.; Schrottenloher, A. On quantum slide attacks. In Proceedings of the International Conference on Selected Areas in Cryptography, Waterloo, ON, Canada, 12–16 August 2019; Springer International Publishing: Cham, Switzerland, 2019; pp. 492–519.

76. Leander, G.; May, A. Grover meets Simon—quantumly attacking the FX-construction. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Hong Kong, China, 3–7 December 2017; Springer International Publishing: Cham, Switzerland, 2017; pp. 161–178.
77. Chen, H.; Li, Y.; Abla, P.; Li, Z.; Jiao, L.; Wang, M. Quantum algorithm for finding impossible differentials and zero-correlation linear hulls of symmetric ciphers. In Proceedings of the Australasian Conference on Information Security and Privacy, Brisbane, Australia, 5–7 July 2023; Springer Nature Switzerland: Cham, Switzerland, 2023; pp. 431–451.
78. Hosoyamada, A. Quantum speed-up for multidimensional (zero correlation) linear distinguishers. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, 4–8 December 2023; Springer Nature Singapore: Singapore, 2023; pp. 311–345.
79. Du, X.; Sun, R.; Zheng, Y.; Liang, L. Improved Meet-in-the-middle Attacks on Reduced-round E2. *J. Electron. Inf. Technol.* **2024**, *46*, 2655–2662. [CrossRef]
80. Wu, H.; Feng, X. Quantum related-key differential cryptanalysis. *Quantum Inf. Process.* **2024**, *23*, 269. [CrossRef]
81. Ni, B.; Dong, X. Improved Quantum Attack on Type-1 Generalized Feistel Schemes and Its Application to CAST-256. *J. Electron. Inf. Technol.* **2020**, *42*, 295–306. [CrossRef]
82. Yu, B.; Sun, B.; Liu, G.-Q.; Luo, Y.-Y.; Zhang, Z.-Y. Quantum Cryptanalysis on Some Generalized Unbalanced Feistel Networks. *J. Cryptologic Res.* **2021**, *8*, 960–973. [CrossRef]
83. Xie, H.; Yang, L. Quantum Truncated Differential and Boomerang Attack. *Symmetry* **2024**, *16*, 1124. [CrossRef]
84. Wang, B.; Hu, F.; Zhang, H.; Wang, C. From Evolutionary Cryptography to Quantum Artificial Intelligent Cryptography. *J. Comput. Res. Dev.* **2019**, *56*, 2112–2134. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.